

FAWAPI



FilesAnywhere

Application Programming Interface

Version 2.28

January 2022

TABLE OF CONTENTS

INTRODUCING THE FILESANYWHERE API	28
API SERVICE DESCRIPTION	29
API OPERATIONS	30
STANDARDS COMPLIANCE.....	35
LET’S GET STARTED!.....	36
ENABLE API ACCESS IN YOUR FILESANYWHERE ACCOUNT	36
OBTAIN A DEVELOPER API KEY	36
OBTAIN THE WEB SERVICE WSDL	37
IMPORT THE WSDL FILE INTO YOUR DEVELOPMENT PLATFORM.....	38
<i>To add a web reference:</i>	38
SAMPLE CODE IN THIS DOCUMENT	39
ADDITIONAL SAMPLE WEB APPLICATION CODE	39
IMPORTANT INFORMATION ABOUT CALLING WEBMETHODS	40
INTRODUCING THE FILESANYWHERE DIGITALLY SIGNED API.....	41
DIGITALLY SIGNED API SERVICE DESCRIPTION	41
LET’S GET STARTED!.....	41
<i>Prerequisites</i>	41
<i>Sample VB.Net Code</i>	42
<i>Sample C#.Net Code</i>	46
SOAP API CALLS.....	51
ACCOUNTCREATEFREE	51
<i>Syntax</i>	51
<i>Usage</i>	51
SOAP 1.1	52
<i>Sample Visual Basic.Net Code</i>	54
<i>Request Arguments</i>	54
<i>Response</i>	55
<i>AccountCreateFreeResult Properties</i>	55
ACCOUNTLOGIN	56
<i>Syntax</i>	56
<i>Usage</i>	56

SOAP 1.1	58
SOAP 1.2	59
Sample Visual Basic.Net Code.....	60
Sample Visual Basic.Net Code For User with UNKNOWN OrgID	60
Request Arguments.....	62
Response.....	63
LoginResult Properties.....	63
APPENDCHUNKWITHFOLDERACCESSKEY	64
Syntax.....	64
Usage	64
SOAP 1.1	66
SOAP 1.2.....	67
Sample Visual Basic.Net Code.....	68
Request Arguments.....	68
Response.....	70
ChunkUploadResult Properties	70
APPENDCHUNKGZIP	71
Syntax.....	71
Usage	71
SOAP 1.1	72
SOAP 1.2.....	73
Sample Visual Basic.Net Code.....	74
Request Arguments.....	74
Response.....	75
AppendChunkResult Properties	75
CLIENTGETADDINVERSION	76
Syntax.....	76
Usage	76
SOAP 1.1	77
SOAP 1.2.....	78
Sample Visual Basic.Net Code.....	79
Request Arguments.....	79
Response.....	80
ClientGetAddInVersionResult Properties.....	80
CLIENTGETUPLOADPATH	81
Syntax.....	81
Usage	81
SOAP 1.1	82
SOAP 1.2.....	83
Sample Visual Basic.Net Code.....	84
Request Arguments.....	84
Response.....	85
ClientGetUploadPathResult Properties	85
CONVERTDOCTOPDF	86
Syntax.....	86
Usage	86
SOAP 1.1	87
SOAP 1.2.....	88
Sample Visual Basic.Net Code.....	88

Request Arguments.....	89
Response.....	90
ConvertDocToPDFResult Properties.....	90
CONVERTDOCTOPDFSTATUS	91
Syntax.....	91
Usage.....	91
SOAP 1.1.....	91
SOAP 1.2.....	93
Sample Visual Basic.Net Code.....	93
Request Arguments.....	94
Response.....	95
ConvertDocToPDFStatusResult Properties	95
CONVERTIMAGETOPDF	96
Syntax.....	96
Usage.....	96
SOAP 1.1.....	98
SOAP 1.2.....	99
Sample Visual Basic.Net Code.....	100
Request Arguments.....	100
Response.....	102
ConvertImageToPDFResult Properties	102
COPYITEMS	103
Syntax.....	103
Usage.....	103
SOAP 1.1.....	104
SOAP 1.2.....	105
Sample Visual Basic.Net Code.....	106
Request Arguments.....	106
Item Properties	106
Response.....	107
CopyItemsResult Properties	107
CREATEFOLDER	108
Syntax.....	108
Usage.....	108
SOAP 1.1.....	109
SOAP 1.2.....	110
Sample Visual Basic.Net Code.....	111
Request Arguments.....	111
Response.....	111
CreateFolderResult Properties.....	112
CREATEFOLDERRECURSIVE.....	112
Syntax.....	113
Usage.....	113
SOAP 1.1.....	114
SOAP 1.2.....	115
Sample Visual Basic.Net Code.....	116
Request Arguments.....	116
Response.....	116
CreateFolderRecursiveResult Properties	117

CUTITEMS	118
Syntax.....	118
Usage.....	118
SOAP 1.1.....	119
SOAP 1.2.....	120
Sample Visual Basic.Net Code.....	121
Request Arguments.....	121
An Item has the following properties:	121
Response.....	122
CutItemsResult Properties.....	122
DELETEITEMS	123
Syntax.....	123
Usage.....	123
SOAP 1.1.....	124
SOAP 1.2.....	125
Sample Visual Basic.Net Code.....	126
Request Arguments.....	126
An Item has the following properties:	126
Response.....	127
DeleteItemsResult Properties	127
FILESHAREAPPENDCOMMENT	128
Syntax.....	128
Usage.....	128
SOAP 1.1.....	129
SOAP 1.2.....	130
Sample Visual Basic.Net Code.....	130
Request Arguments.....	131
Response.....	131
FileShareAppendCommentResult Properties	131
FILESHAREGETHEADERINFO.....	132
Syntax.....	132
Usage.....	132
SOAP 1.1.....	133
SOAP 1.2.....	134
Sample Visual Basic.Net Code.....	134
Request Arguments.....	135
Response.....	135
FileShareHeaderResult Properties	136
FILESHAREGETITEMS	136
Syntax.....	137
Usage.....	137
SOAP 1.1.....	138
SOAP 1.2.....	139
Sample Visual Basic.Net Code.....	140
Request Arguments.....	140
Response.....	141
FileShareItemResult Properties.....	141
A FileShareItem has the following properties:	141
FILESHAREINDEXEDITEMPROPERTIES.....	142

Syntax.....	142
FILESHAREITEMIMAGE	142
Syntax.....	142
FILESHAREITEMTHUMBNAIL	142
Syntax.....	142
FILESHAREREQUESTDOCINDEXING	142
Syntax.....	142
GETCLIENTAPPINFO	143
Syntax.....	143
Usage	143
SOAP 1.1	144
SOAP 1.2.....	145
Sample Visual Basic.Net Code.....	146
Request Arguments.....	146
Response.....	147
GetClientAppInfoResult Properties.....	147
GETDROPBOXDEFAULTS	148
Syntax.....	148
Usage	148
SOAP 1.1	149
SOAP 1.2.....	150
Sample Visual Basic.Net Code.....	151
Request Arguments.....	151
Response.....	152
GetDropBoxDefaultsResult Properties	152
GETELINKDEFAULTS	154
Syntax.....	154
Usage	154
SOAP 1.1	155
SOAP 1.2.....	156
Sample Visual Basic.Net Code.....	157
Request Arguments.....	157
Response.....	158
GetELinkDefaultsResult Properties	158
GETUSERPARAM	160
Syntax.....	160
Usage	160
SOAP 1.1	160
SOAP 1.2.....	161
Sample Visual Basic.Net Code.....	162
Request Arguments.....	162
Response.....	163
UserParamResult Properties.....	163
GROUPSHAREADD	164
Syntax.....	164
Usage	164

<i>SOAP 1.1</i>	166
<i>SOAP 1.2</i>	167
<i>Sample Visual Basic.Net Code</i>	168
<i>Request Arguments</i>	169
<i>Response</i>	171
<i>GroupShareResult Properties</i>	171
INDEXEDITEMPROPERTIES.....	172
<i>Syntax</i>	172
<i>Usage</i>	172
<i>SOAP 1.1</i>	173
<i>SOAP 1.2</i>	174
<i>Sample Visual Basic.Net Code</i>	175
<i>Request Arguments</i>	175
<i>Response</i>	176
<i>IndexedItemPropertiesResult Properties</i>	176
ITEMADVANCEDPROPERTIES.....	176
<i>Syntax</i>	177
<i>Usage</i>	177
<i>SOAP 1.1</i>	178
<i>SOAP 1.2</i>	179
<i>Sample Visual Basic.Net Code</i>	180
<i>Request Arguments</i>	180
<i>Response</i>	181
<i>ItemAdvPropertiesResult Properties</i>	181
ITEMCOMMENTADD.....	182
<i>Syntax</i>	182
<i>Usage</i>	182
<i>SOAP 1.1</i>	183
<i>SOAP 1.2</i>	184
<i>Sample Visual Basic.Net Code</i>	185
<i>Request Arguments</i>	185
<i>Response</i>	186
<i>ItemCommentAddResult Properties</i>	186
ITEMCOMMENTREMOVE.....	187
<i>Syntax</i>	187
<i>Usage</i>	187
<i>SOAP 1.1</i>	188
<i>SOAP 1.2</i>	188
<i>Sample Visual Basic.Net Code</i>	190
<i>Request Arguments</i>	190
<i>Response</i>	191
<i>ItemCommentRemoveResult Properties</i>	191
ITEMCOMMENTUPDATE.....	192
<i>Syntax</i>	192
<i>Usage</i>	192
<i>SOAP 1.1</i>	193
<i>SOAP 1.2</i>	193
<i>Sample Visual Basic.Net Code</i>	195
<i>Request Arguments</i>	195

<i>Response</i>	196
<i>ItemCommentUpdateResult Properties</i>	196
ITEMCOMMENTSLIST	197
<i>Syntax</i>	197
<i>Usage</i>	197
SOAP 1.1.....	198
SOAP 1.2.....	198
<i>Sample Visual Basic.Net Code</i>	199
<i>Request Arguments</i>	199
<i>Response</i>	200
<i>ItemCommentsListResultItemCommentsListResult Properties</i>	200
<i>A CommentList has the following properties:</i>	200
ITEMDESCRIPTION	201
<i>Syntax</i>	201
<i>Usage</i>	201
SOAP 1.1.....	202
SOAP 1.2.....	202
<i>Sample Visual Basic.Net Code</i>	203
<i>Request Arguments</i>	203
<i>Response</i>	204
<i>ItemDescriptionResult Properties</i>	204
ITEMDESCRIPTIONADD	205
<i>Syntax</i>	205
<i>Usage</i>	205
SOAP 1.1.....	206
SOAP 1.2.....	206
<i>Sample Visual Basic.Net Code</i>	207
<i>Request Arguments</i>	207
<i>Response</i>	208
<i>ItemDescriptionAddResult Properties</i>	208
ITEMIMAGE.....	209
<i>Syntax</i>	209
<i>Usage</i>	209
SOAP 1.1.....	211
SOAP 1.2.....	212
<i>Sample Visual Basic.Net Code</i>	213
<i>Request Arguments</i>	213
<i>Response</i>	214
<i>ItemImageResult Properties</i>	214
ITEMPROPERTIES	215
<i>Syntax</i>	215
<i>Usage</i>	215
SOAP 1.1.....	216
SOAP 1.2.....	217
<i>Sample Visual Basic.Net Code</i>	218
<i>Request Arguments</i>	218
<i>Response</i>	218
<i>ItemPropertiesResult Properties</i>	219

ITEMTHUMBNAIL	220
Syntax.....	220
Usage.....	220
SOAP 1.1.....	221
SOAP 1.2.....	222
Sample Visual Basic.Net Code.....	223
Request Arguments.....	223
LOGOUT	224
Syntax.....	224
Usage.....	224
SOAP 1.1.....	225
SOAP 1.2.....	226
Sample Visual Basic.Net Code.....	227
Request Arguments.....	227
Response.....	227
LogoutResult Properties.....	228
PASTE.....	229
Syntax.....	229
Usage.....	229
SOAP 1.1.....	230
SOAP 1.2.....	231
Sample Visual Basic.Net Code.....	232
Request Arguments.....	232
Response.....	232
PasteResult Properties.....	233
RENAMEITEM	234
Syntax.....	234
Usage.....	234
SOAP 1.1.....	235
SOAP 1.2.....	236
Sample Visual Basic.Net Code.....	237
Request Arguments.....	237
Response.....	237
RenameItemResult Properties.....	238
REQUESTDOCINDEXING	239
Syntax.....	239
Usage.....	239
SOAP 1.1.....	240
SOAP 1.2.....	241
Sample Visual Basic.Net Code.....	242
Request Arguments.....	242
An Item has the following properties:	242
Response.....	243
DocIndexingResults Properties	243
SAMLACCOUNTLOGIN.....	244
Syntax.....	244
Usage.....	244
SOAP 1.1.....	245

SOAP 1.2.....	246
Sample Visual Basic.Net Code.....	247
Request Arguments.....	247
Response.....	248
LoginResult Properties.....	248
GOOGLEOAUTHACCOUNTLOGIN.....	249
Syntax.....	249
Usage.....	249
SOAP 1.1.....	250
SOAP 1.2.....	251
Sample Visual Basic.Net Code.....	252
Request Arguments.....	252
Response.....	253
LoginResult Properties.....	253
LDAPACCOUNTLOGIN	254
Syntax.....	254
Usage.....	254
SOAP 1.1.....	255
SOAP 1.2.....	256
Sample Visual Basic.Net Code.....	257
Request Arguments.....	257
Response.....	258
LoginResult Properties.....	258
SSOLOGINURL.....	258
Syntax.....	258
Usage.....	258
SOAP 1.1.....	260
SOAP 1.2.....	261
Sample Visual Basic.Net Code.....	262
Request Arguments.....	262
Response.....	263
SSOLoginURLResult Properties.....	263
SSOLOGINWEBURL.....	264
Syntax.....	264
Usage.....	264
SOAP 1.1.....	265
SOAP 1.2.....	266
Sample Visual Basic.Net Code.....	267
Request Arguments.....	267
Response.....	268
SSOLoginURLResult Properties.....	268
SENDDROPBOXLINK	269
Syntax.....	269
Usage.....	269
SOAP 1.1.....	271
SOAP 1.2.....	271
Sample Visual Basic.Net Code.....	273
Request Arguments.....	273

Response.....	275
SendDropboxLink Properties	275
A ElinkURL has the following properties:	275
SENDITEMSELINKWITHDEFAULTS	277
Syntax.....	277
Usage	277
SOAP 1.1	278
SOAP 1.2.....	279
Sample Visual Basic.Net Code.....	280
Request Arguments.....	281
An Item has the following properties:	281
Response.....	282
SendElinkListResult Properties	282
A ElinkDetails has the following properties:	282
SENDFILESHARELINKWITHDEFAULTS.....	283
Syntax.....	283
Usage	283
SOAP 1.1	284
SOAP 1.2.....	285
Sample Visual Basic.Net Code.....	286
Request Arguments.....	287
Parameters has the following properties:	287
An ElinkItems has the following properties:.....	288
Response.....	289
SendElinkListResult Properties	289
A ElinkDetails has the following properties:.....	289
FAXGETDEFAULTS	290
Syntax.....	290
Usage	290
SOAP 1.1	290
SOAP 1.2.....	291
Sample Visual Basic.Net Code.....	292
Request Arguments.....	293
Response.....	293
FaxDefaultsResult Properties.....	293
FAXSEND.....	295
Syntax.....	295
Usage	295
SOAP 1.1	297
SOAP 1.2.....	298
Sample Visual Basic.Net Code.....	299
Request Arguments.....	299
Response.....	301
FaxSendResult Properties	302
GETFAXSTATUS.....	303
Syntax.....	303
Usage	303
SOAP 1.1	303

<i>SOAP 1.2</i>	304
<i>Sample Visual Basic.Net Code</i>	305
<i>Request Arguments</i>	306
<i>Response</i>	306
<i>FaxSentStatus Properties</i>	306
<i>Syntax</i>	307
<i>Usage</i>	307
<i>SOAP 1.1</i>	308
<i>SOAP 1.2</i>	309
<i>Sample Visual Basic.Net Code</i>	310
<i>Request Arguments</i>	310
<i>Response</i>	310
<i>DownloadFileResult Properties</i>	310
FILEPROPERTIES	311
<i>Syntax</i>	311
<i>Usage</i>	311
<i>SOAP 1.1</i>	312
<i>SOAP 1.2</i>	313
<i>Sample Visual Basic.Net Code</i>	314
<i>Arguments</i>	314
<i>Response</i>	314
<i>FilePropertiesResult Properties</i>	315
FILEADVANCEDPROPERTIES	316
<i>Syntax</i>	316
<i>Usage</i>	316
<i>SOAP 1.1</i>	317
<i>SOAP 1.2</i>	318
<i>Sample Visual Basic.Net Code</i>	319
<i>Arguments</i>	319
<i>Response</i>	319
<i>FileAdvancedPropertiesResult Properties</i>	320
CONVERTIMAGE	321
<i>Syntax</i>	321
<i>Usage</i>	321
<i>SOAP 1.1</i>	322
<i>SOAP 1.2</i>	323
<i>Sample Visual Basic.Net Code</i>	324
<i>Arguments</i>	325
<i>Response</i>	326
<i>ConvertImageResult Properties</i>	326
CONVERTIMAGESTATUS.....	327
<i>Syntax</i>	327
<i>Usage</i>	327
<i>SOAP 1.1</i>	328
<i>SOAP 1.2</i>	328
<i>Sample Visual Basic.Net Code</i>	329
<i>Arguments</i>	330
<i>Response</i>	331
<i>ConvertImageStatusResult Properties</i>	331

GETMEDIASTREAM.....	332
Syntax.....	332
Usage.....	332
SOAP 1.1.....	333
SOAP 1.2.....	334
Sample Visual Basic.Net Code.....	335
Arguments.....	335
Response.....	336
GetMediaStreamResult Properties.....	336
Information about GetMediaStream output usage.....	338
GETIMAGESTREAM.....	338
Syntax.....	339
Usage.....	339
SOAP 1.1.....	340
SOAP 1.2.....	340
Sample Visual Basic.Net Code.....	341
Arguments.....	342
GetImageStreamResult Properties.....	342
GETTIFFPAGECOUNT.....	343
Syntax.....	344
Usage.....	344
SOAP 1.1.....	344
SOAP 1.2.....	345
Sample Visual Basic.Net Code.....	345
Arguments.....	346
GetTIFFPageCountResult Properties.....	346
FAWAPIV2 METHODS.....	346
APNECHONOTIFICATION.....	346
Syntax.....	346
Usage.....	347
SOAP 1.2.....	348
Request Arguments.....	349
Response.....	349
APNEchoNotificationResult Properties.....	349
APNGETNOTIFICATIONPROPERTIES.....	350
Syntax.....	350
Usage.....	350
SOAP 1.2.....	351
Request Arguments.....	352
Response.....	352
APNGetNotificationPropertiesResult Properties.....	352
APNREGISTERDEVICE.....	354
Syntax.....	354
Usage.....	354
SOAP 1.2.....	355
Request Arguments.....	356
Response.....	356

<i>APNRegisterDeviceResult Properties</i>	356
APNSSENDNOTIFICATION	357
<i>Syntax</i>	357
<i>Usage</i>	357
SOAP 1.2	358
<i>Request Arguments</i>	359
<i>Response</i>	359
<i>APNSendNotificationResult Properties</i>	359
APNUNREGISTERDEVICE	361
<i>Syntax</i>	361
<i>Usage</i>	361
SOAP 1.2	362
<i>Request Arguments</i>	363
<i>Response</i>	363
<i>APNUnregisterDeviceResult Properties</i>	363
APNUPDATEREGISTRATION	364
<i>Syntax</i>	364
<i>Usage</i>	364
SOAP 1.2	365
<i>Request Arguments</i>	366
<i>Response</i>	366
<i>APNUnregisterDeviceResult Properties</i>	366
DELETEMEETING	367
<i>Syntax</i>	367
<i>Usage</i>	367
SOAP 1.2	368
<i>Request Arguments</i>	369
<i>A Meeting has the following properties:</i>	369
<i>Response</i>	370
<i>DeleteMeetingResult Properties</i>	370
EDITDOCUMENTWITHZOHO	371
<i>Syntax</i>	371
<i>Usage</i>	371
SOAP 1.2	372
<i>Request Arguments</i>	373
<i>Response</i>	373
<i>EditDocumentWithZohoResult Properties</i>	373
ESENDFILES	374
<i>Syntax</i>	374
<i>Usage</i>	374
SOAP 1.2	375
<i>Request Arguments</i>	376
<i>Response</i>	376
<i>ESendFilesResult Properties</i>	376
GETACCOUNTDETAILS	377
<i>Syntax</i>	377
<i>Usage</i>	377

SOAP 1.2.....	378
HTTP POST.....	379
Request Arguments.....	380
Response.....	380
GetAccountDetailsResult Properties.....	380
GETAPNSETTING	382
Syntax.....	382
Usage	382
SOAP 1.2.....	384
Request Arguments.....	385
Response.....	385
GetAPNSettingResult Properties	385
GETAPNSETTINGS.....	387
Syntax.....	387
Usage	387
SOAP 1.2.....	389
Request Arguments.....	391
Response.....	391
GetAPNSettingsResult Properties.....	391
GETDROPBOXDEFAULTS2	392
Syntax.....	392
Usage	392
SOAP 1.2.....	394
Request Arguments.....	395
Response.....	395
GetDropBoxDefaultsResult2v2 Properties.....	395
GETELINKDEFAULTS2	397
Syntax.....	397
Usage	397
SOAP 1.2.....	398
Request Arguments.....	400
Response.....	400
GetELinkDefaultsResult2v2 Properties.....	400
GETELINKDEFAULTS3	402
Syntax.....	402
Usage	402
SOAP 1.2.....	403
Request Arguments.....	405
Response.....	405
GetELinkDefaultsResult3v2 Properties.....	405
GETFOLDERPROPERTIES	408
Syntax.....	408
Usage	408
SOAP 1.2.....	409
Request Arguments.....	411
Response.....	411
GetFolderPropertiesResult Properties.....	411

GETMEETING.....	412
Syntax.....	412
Usage.....	412
SOAP 1.2.....	414
Request Arguments.....	416
Response.....	416
GetMeetingResult Properties.....	416
A Meeting has the following properties:	416
GETNOTIFICATIONS	418
Syntax.....	418
Usage	418
SOAP 1.2.....	419
Request Arguments.....	420
Response.....	420
GetNotificationsResult Properties	420
GETNOTIFICATIONSETTINGFORDEVICE	422
Syntax.....	422
Usage	422
SOAP 1.2.....	423
Request Arguments.....	425
Response.....	425
GetNotificationSettingForDeviceResult Properties	425
GETNOTIFICATIONSETTINGSFORDEVICE	426
Syntax.....	426
Usage	426
SOAP 1.2.....	427
Request Arguments.....	428
Response.....	429
GetNotificationSettingsForDeviceResult Properties.....	429
GROUPSHAREADD2	430
Syntax.....	430
Usage	430
SOAP 1.2.....	432
Request Arguments.....	433
Response.....	433
GroupShareAdd2Result Properties.....	433
GROUPSHAREREMOVE	435
Syntax.....	435
Usage	435
SOAP 1.2.....	437
Request Arguments.....	438
Response.....	438
GroupShareRemoveResult Properties.....	438
ITEMCOMMENTS	438
Syntax.....	439
Usage	439
SOAP 1.2.....	441

Request Arguments.....	442
Response.....	442
ItemCommentsResult Properties.....	442
ITEMCOMMENTSWITHTHUMBNAIL.....	443
Syntax.....	443
Usage.....	444
SOAP 1.1.....	444
SOAP 1.2.....	445
Sample Visual Basic.Net Code.....	446
Request Arguments.....	447
Response.....	447
ItemCommentsResult Properties.....	447
LISTGROUPSHAREUSERS.....	449
Syntax.....	449
Usage.....	449
SOAP 1.2.....	450
Request Arguments.....	451
Response.....	451
ListGroupShareUsersResult Properties.....	451
LISTITEMS2.....	453
Syntax.....	453
Usage.....	453
SOAP 1.1.....	453
SOAP 1.2.....	455
Sample Visual Basic.Net Code.....	456
Request Arguments.....	456
Response.....	456
ListItemsResult Properties.....	457
An Item has the following properties:.....	457
LISTLOGACTIVITIES.....	458
Syntax.....	458
Usage.....	458
SOAP 1.2.....	461
Request Arguments.....	462
Response.....	463
ListLogActivitiesResult Properties.....	463
LISTMEETINGS.....	465
Syntax.....	465
Usage.....	465
SOAP 1.2.....	467
Request Arguments.....	469
Response.....	469
ListMeetingsResult Properties.....	469
A MeetingFromList has the following properties:.....	469
LISTUSERVOLUMES2.....	471
Syntax.....	471
Usage.....	471
SOAP 1.1.....	471

SOAP 1.2.....	472
Sample Visual Basic .Net Code.....	473
Request Arguments.....	474
Response.....	474
ListUserVolumesResult Properties	474
A UserVolumev2 has the following properties:	474
MOVEITEMS	474
Syntax.....	475
Usage	475
SOAP 1.2.....	477
Request Arguments.....	478
Response.....	478
MoveItemsResultv2 Properties	478
REGISTERDEVICE.....	480
Syntax.....	480
Usage	480
SOAP 1.2.....	481
Request Arguments.....	482
Response.....	482
RegisterDeviceResult Properties	482
SCHEDULEMEETING	484
Syntax.....	484
Usage	484
SOAP 1.2.....	486
Request Arguments.....	487
Response.....	488
ScheduleMeetingResult Properties	488
A Meeting has the following properties:	488
SEARCHFILES	490
Syntax.....	490
Usage	490
SOAP 1.2.....	493
Request Arguments.....	494
Response.....	495
SearchFilesResult Properties	495
HOW TO SEARCH FOR FILES IN THE GROUPSHARE FOLDERS?.....	495
SENDDROPBOXLINK2	496
Syntax.....	496
Usage	496
SOAP 1.2.....	499
Request Arguments.....	501
Response.....	502
SendELinkResult Properties	502
SETACCOUNTDETAILS	503
Syntax.....	503
Usage	503
SOAP 1.2.....	504

<i>Request Arguments</i>	506
<i>Response</i>	506
<i>SetAccountDetailsResultv2 Properties</i>	506
SETAPNSSETTING	507
<i>Syntax</i>	507
<i>Usage</i>	507
SOAP 1.2.....	509
<i>Request Arguments</i>	510
<i>Response</i>	510
<i>SetAPNSSettingResult Properties</i>	510
SETAPNSSETTINGS	512
<i>Syntax</i>	512
<i>Usage</i>	512
SOAP 1.2.....	514
<i>Request Arguments</i>	517
<i>Response</i>	517
<i>SetAPNSSettingsResult Properties</i>	517
SETDROPBOXDEFAULTS	518
<i>Syntax</i>	518
<i>Usage</i>	518
<i>Request Arguments</i>	521
<i>Response</i>	522
<i>SetDropBoxDefaultsResultv2 Properties</i>	522
SETDROPBOXDEFAULTS2	523
<i>Syntax</i>	523
<i>Usage</i>	523
SOAP 1.2.....	525
<i>Request Arguments</i>	526
<i>Response</i>	527
<i>SetDropBoxDefaultsResultv2 Properties</i>	527
SETELINKDEFAULTS.....	528
<i>Syntax</i>	528
<i>Usage</i>	528
SOAP 1.2.....	530
<i>Request Arguments</i>	531
<i>Response</i>	533
<i>SetELinkDefaultsResultv2 Properties</i>	533
SETELINKDEFAULTS2.....	534
<i>Syntax</i>	534
<i>Usage</i>	534
SOAP 1.2.....	537
<i>Request Arguments</i>	538
<i>Response</i>	539
<i>SetELinkDefaultsResultv2 Properties</i>	539
SETNOTIFICATIONSETTINGFORDEVICE.....	540
<i>Syntax</i>	540
<i>Usage</i>	540

SOAP 1.2.....	541
Request Arguments.....	543
Response.....	543
SetNotificationSettingsForDeviceResult Properties.....	543
SETNOTIFICATIONSETTINGSFORDEVICE	544
Syntax.....	544
Usage	544
SOAP 1.2.....	545
Request Arguments.....	546
Response.....	546
SetNotificationSettingsForDeviceResult Properties.....	546
STOPSHARINGGROUPSHARE	547
Syntax.....	547
Usage	547
SOAP 1.2.....	549
Request Arguments.....	550
Response.....	550
StopSharingGroupShareResult Properties	550
UNDOCHECKOUT	551
Syntax.....	551
Usage	551
SOAP 1.2.....	553
Request Arguments.....	554
Response.....	554
UndoCheckoutResult Properties.....	554
UNREGISTERDEVICE	555
Syntax.....	555
Usage	555
SOAP 1.2.....	557
Request Arguments.....	558
Response.....	558
UnregisterDeviceResult Properties.....	558
UPDATEMEETING.....	559
Syntax.....	559
Usage	559
SOAP 1.2.....	560
Request Arguments.....	561
A Meeting has the following properties:	561
Response.....	562
UpdateMeetingResult Properties.....	562
UPLOADFILEWITHCHECKIN	562
Syntax.....	562
Usage	563
SOAP 1.2.....	564
Request Arguments.....	565
Response.....	565
UploadFileResult Properties	565

USERLOOKUP	566
Syntax.....	566
Usage.....	566
SOAP 1.2.....	568
Sample VB.Net Code.....	570
Request Arguments.....	570
Response.....	571
UserLookupResult Properties.....	571
GETPROFILEIMAGE	572
Syntax.....	572
Usage.....	572
SOAP 1.1.....	573
SOAP 1.2.....	573
Sample Visual Basic.Net Code.....	574
Request Arguments.....	574
ACCOUNTLOGIN2	575
Syntax.....	575
Usage.....	575
SOAP 1.1.....	577
SOAP 1.2.....	578
Sample Visual Basic.Net Code.....	579
Request Arguments.....	579
GETGROUPSHAREUSERPROFILEIMAGE.....	580
Syntax.....	581
Usage.....	581
SOAP 1.1.....	582
SOAP 1.2.....	583
Sample Visual Basic.Net Code.....	584
Request Arguments.....	584
GETID3TAG	585
Syntax.....	585
Usage.....	585
SOAP 1.1.....	586
SOAP 1.2.....	587
Sample Visual Basic.Net Code.....	588
Request Arguments.....	588
GETAPIURL	589
Syntax.....	589
Usage.....	589
SOAP 1.1.....	590
SOAP 1.2.....	591
Sample Visual Basic.Net Code.....	592
Request Arguments.....	592
Response.....	592
GetAPIURLResult Properties.....	592
GETCLIENTID.....	593
Syntax.....	593

Usage	593
SOAP 1.1	594
SOAP 1.2	595
Sample Visual Basic.Net Code.....	596
Request Arguments.....	596
Parameters Properties	596
Response.....	596
GetClientIDResult Properties.....	596
GETCOMPANYDETAILS	597
Syntax.....	597
Usage	597
SOAP 1.1	598
SOAP 1.2.....	599
Sample Visual Basic.Net Code.....	600
Request Arguments.....	600
Parameters Properties	600
Response.....	600
GETSTORAGELOCATION.....	602
Syntax.....	602
Usage	602
SOAP 1.1	603
SOAP 1.2.....	604
Sample Visual Basic.Net Code.....	605
Request Arguments.....	605
Response.....	605
StorageLocation Properties.....	606
GETFOLDERACCESSKEY	607
Syntax.....	607
Usage	607
SOAP 1.1	608
SOAP 1.2.....	608
Sample Visual Basic.Net Code.....	610
Request Arguments.....	610
Response.....	611
GetFolderAccessKeyResult Properties.....	611
UPLOADFILEWITHFOLDERACCESSKEY	612
Syntax.....	612
Usage	612
SOAP 1.1	613
SOAP 1.2.....	613
Sample Visual Basic.Net Code.....	615
Request Arguments.....	615
Response.....	616
UploadFileResult Properties	616
UPDATEUSERPROFILE	617
Syntax.....	617
Usage	617
SOAP 1.1	618

SOAP 1.2.....	619
Sample Visual Basic.Net Code.....	620
Request Arguments.....	620
Parameters Properties.....	620
Response.....	621
GETCLIENTDETAILS	622
Syntax.....	622
Usage.....	622
SOAP 1.1.....	622
SOAP 1.2.....	623
Sample Visual Basic.Net Code.....	624
Request Arguments.....	624
Response.....	624
SETPROFILEIMAGE.....	624
Syntax.....	624
Usage.....	624
SOAP 1.1.....	625
SOAP 1.2.....	626
Sample Visual Basic.Net Code.....	626
Request Arguments.....	627
Parameters Properties.....	627
Response.....	627
SENDFILESHARELINK	627
Syntax.....	628
Usage.....	628
SOAP 1.1.....	630
SOAP 1.2.....	631
Sample C#.Net Code.....	633
Request Arguments.....	635
Response.....	637
SendFileShareLinkResult Properties.....	637
ADMINCREATEFOLDER	637
Syntax.....	637
Usage.....	638
SOAP 1.1.....	639
SOAP 1.2.....	640
Sample C#.Net Code.....	641
Arguments.....	642
Response.....	642
AdminCreateFolderResult Properties.....	643
ADMINCREATEGROUPSHARE	643
Syntax.....	643
Usage.....	643
SOAP 1.1.....	645
SOAP 1.2.....	646
Sample C#.Net Code.....	647
Arguments.....	648
Response.....	649

<i>AdminCreateGroupShareResult Properties</i>	650
ADMINUSERADD	650
Syntax.....	650
Usage.....	650
SOAP 1.1.....	651
SOAP 1.2.....	653
<i>Sample C#.Net Code</i>	656
Arguments.....	658
Response.....	664
<i>AdminUserAddResult Properties</i>	664
UPDATEFOLDERPROPERTIES	664
Syntax.....	664
Usage.....	664
<i>Sets folder properties such as file version history and check-in/check-out and enable the notifications on various operations on folder.</i>	664
SOAP 1.1.....	665
SOAP 1.2.....	666
<i>Sample C#.Net Code</i>	668
Arguments.....	670
Response.....	671
<i>UpdateFolderPropertiesResult Properties</i>	671
FILEAGINGRULE.....	672
Syntax.....	672
Usage.....	672
<i>Create or Edit FileLifeCycle and FileArchive Rule on a folder</i>	672
SOAP 1.1.....	672
SOAP 1.2.....	674
<i>Sample C#.Net Code</i>	676
Response.....	678
<i>FileAgingRuleResult Properties</i>	678
FILEAGINGRULEREMOVE.....	679
Syntax.....	679
Usage.....	679
<i>Removes File LifeCycle or Archive rules.</i>	679
SOAP 1.1.....	679
SOAP 1.2.....	680
<i>Sample C#.Net Code</i>	682
Response.....	682
<i>FileAgingRuleRemoveResult Properties</i>	682
INTRODUCING THE FILESANYWHERE REST API	683
REST API SERVICE URL	683
REST API OPERATIONS	683
LET'S GET STARTED!.....	684
SIGN UP FOR A FILESANYWHERE ACCOUNT USING FAWAPI	684

REST API CALLS	684
DOWNLOADFILE	684
Syntax.....	684
HTTPs Request	684
POST: /DownloadFile.....	684
HOST: https://restapi.filesanywhere.com/FAWAPI.svc	684
Filesanywhere-API-Arg : {"ItemPath":"","Token":""}.....	684
Usage	684
Sample C#.Net Code	685
Request Arguments.....	686
Response.....	686
ZIPDOWNLOADFILEORFOLDER	687
Syntax.....	687
HTTP Request.....	687
POST: /ZipDownloadFileOrFolder	687
HOST: https://restapi.filesanywhere.com/FAWAPI.svc	687
Filesanywhere-API-Arg : {"ItemPath":"","Token":""}.....	687
Usage	687
Sample C#.Net Code	687
Request Arguments.....	689
Response.....	689
DOWNLOADWITHCHECKOUT	690
Syntax.....	690
HTTPs Request	690
POST: /DownloadWithCheckout	690
HOST: https://restapi.filesanywhere.com/FAWAPI.svc	690
Usage	690
Sample C#.Net Code	690
Response.....	692
DOWNLOADFROMFILESHARE	692
Syntax.....	692
HTTP Request.....	693
POST: /DownloadFromFileShare.....	693
HOST: https://restapi.filesanywhere.com/FAWAPI.svc	693
Filesanywhere-API-Arg : {"FSToken":"","FileIDList":"","PathList":""}	693
Usage	693
Sample C#.Net Code	693
Response.....	695
DOWNLOADFORMDATA	695
Syntax.....	696
HTTP Request.....	696
POST: /DownloadFormData.....	696
HOST: https://restapi.filesanywhere.com/FAWAPI.svc	696
Filesanywhere-API-Arg	:
{"Token":"","FormID":"","FilterParams":{"FromDataID":"","UptoDataID":"","DateType":"","StartDate":"","EndDate":"","ExportFields":"","ExportType":}}	696
Usage	696
Sample C#.Net Code	696

Response.....	700
PREUPLOADVALIDATION	701
Syntax.....	701
HTTPs Request	701
POST: /PreUploadValidation.....	701
HOST: https://restapi.filesanywhere.com/FAWAPI.svc	701
Filesanywhere-API-Arg : {"UploadPath":"","Token":""}	701
Usage	701
Sample C#.Net Code	701
Request Arguments.....	702
Response.....	702
UPLOADITEM	703
Syntax.....	703
HTTPs Request	703
POST: /UploadItem.....	703
HOST: https://restapi.filesanywhere.com/FAWAPI.svc	703
Filesanywhere-API-Arg : {"UploadPath":"","Token":"","UploadKey":"","Compress":"","FileSize":""}	703
Usage	703
Sample C#.Net Code	704
Request Arguments.....	705
Response.....	705
CHUNKUPLOAD	706
Syntax.....	706
HTTPs Request	706
POST: /ChunkUpload	706
HOST: https://restapi.filesanywhere.com/FAWAPI.svc	706
Filesanywhere-API-Arg : {"UploadPath":"","Token":"","UploadKey":"","Compress":"","FileSize":""}	706
Usage	706
Sample C#.Net Code	706
Request Arguments.....	707
Response.....	708
RESUMEUPLOAD.....	708
Syntax.....	708
HTTPs Request	708
POST: /ResumeUpload.....	708
HOST: https://restapi.filesanywhere.com/FAWAPI.svc	708
Filesanywhere-API-Arg : {"UploadKey":"","Token":"","FileName":""}	708
Usage	708
Sample C#.Net Code	709
Request Arguments.....	709
Response.....	710
INTRODUCING THE FILESANYWHERE DIGITALLY SIGNED REST API.....	710
DIGITALLY SIGNED REST API SERVICE URL	710
LET'S GET STARTED!.....	711

SIGN UP FOR A FILESANYWHERE ACCOUNT USING FAWAPI.....711

Prerequisites..... 711
Please refer support article <https://support.filesanywhere.com/hc/en-us/articles/115003773592> to know more about prerequisites required to use the digitally signed REST API in your API consumer application.
..... 711

NOTE: BELOW SAMPLE CODE IS GIVEN IN THE CONTEXT OF USING DOWNLOADFILE API METHOD. YOU MAY MODIFY THE SAMPLE CODE TO USE ANY OTHER SUPPORTED API METHOD 711

Sample VB.Net Code..... 711
Sample C#.Net Code..... 714

API REVISION HISTORY 717

Version 1.3 717
Version 1.4 717
Version 1.5 718
Version 1.6 718
Version 1.7 718
Version 1.8 719
Version 1.9 719
Version 2.0 719
Version 2.1 719
Version 2.2 721
Version 2.3 722
Version 2.4 722
Version 2.5 722
Version 2.6 723
Version 2.7 723
Version 2.8 723
Version 2.9 723
Version 2.10..... 723
Version 2.11..... 724
Version 2.12..... 724
Version 2.13..... 724
Version 2.14..... 724
Version 2.15..... 724
Version 2.16..... 725
Version 2.17..... 725
Version 2.18..... 725
Version 2.19..... 725
Version 2.20..... 726
Version 2.21..... 726
Version 2.22..... 726
Version 2.23..... 726
Version 2.24..... 727
Version 2.25..... 727
Version 2.26..... 727
Version 2.27..... 727
Version 2.28..... 727

Introducing the FilesAnywhere API

The **FilesAnywhere SOAP API (FAWAPI)** provides a simple web services interface to your FilesAnywhere account. The API (Application Programming Interface) enables your FilesAnywhere account to be integrated with applications and web sites using the standard web services protocol.

SOAP is a simple, XML-based protocol for exchanging messages between a client and a web service. SOAP is independent of any particular operating system, hardware platform, programming language or development environment.

The API is freely available for commercial or non-commercial use by anyone having a FilesAnywhere Account

API Functionality

API functionality can be accessed using any web service enabled development environment. Using the standards-based SOAP interface designed to work with any Internet-development toolkit, you can construct web service client applications that can programmatically:

- Login to your FilesAnywhere account and receive authentication information to be used for subsequent calls.
- Get a list of all User Volumes available to your account.
- Get a list of all items (files or folders) in any folder in your account.
- Create a new folder.
- Rename or Delete any item.
- Copy or Cut and Paste items from one folder to another.
- Download a file.
- Upload a file.
- Logout from your FilesAnywhere account.
- Get basic or advance attributes of an item.
- Get a thumbnail image on an item.
- Send a FileShare link.
- Send a FileReceive Link.
- Get a list of the contents of a FileShare link.

API Service Description

For a formal definition, please review the Service Description at:

<https://api.filesanywhere.com/v2/fawapi.asmx>

API Operations

The following operations are supported. For a formal definition, please review the **Service Description**.

- **AccountCreateFree**
Allows you to create new free user accounts. You must obtain special permission from FilesAnywhere to activate this feature. Send an email to support@filesanywhere.com and we will evaluate your request.
- **AccountLogin**
Logs you into your FilesAnywhere account
- **AppendChunk**
Append a chunk of bytes to a file. The client must ensure that all messages are in sequence.
- **AppendChunkGZip**
Similar to the regular AppendChunk method, except in this method all chunks are decompressed on the server upon receipt, using the GZIP compression standard. This speeds up large file transfers significantly by compressing each chunk. The use of this method requires that you first compress all chunks using the GZIP standard. (Note: if you are using .NET, this compression method is available as the 'GZipStream' class library)
- **ClientGetAddInVersion**
Returns the latest version of the Client Office Add-in
- **ClientGetUploadPath**
Returns the upload path to which client applications will upload files.
- **ConvertDocToPDF**
Convert document file into PDF file.
- **ConvertDocToPDFStatus**
Get the status for Document to PDF conversion.
- **ConvertImageToPDF**
Convert Images into a PDF file.
- **CopyItems**
Copies items to the clipboard for pasting
- **CreateFolder**
Creates a new folder

- **CutItems**
Cuts items and moves it to the clipboard for pasting
- **DeleteItems**
Deletes Items
- **DownloadFile**
Provides a URL from which a file can be downloaded
- **FaxGetDefaults**
This function returns the defaults that were saved from your last fax sent.
- **FaxSend**
This function sends a fax using the proprietary FilesAnywhere Web Fax Service.
- **GetFaxStatus**
This function returns the status of sent fax.
- **FileShareAppendComment**
Allows users of the FileShare link to append comments to a file.
- **FileShareDownloadFiles**
This method allows you to download one or more files or folders from a FileShare link.
- **FileShareGetHeaderInfo**
This method retrieves the header information for a FileShare link.
- **FileShareGetItems**
This method retrieves a list of files and folders for a given path in a FileShare link.
- **FileShareIndexedItemProperties**
(Same as IndexedItemProperties except uses FSToken instead of Token)
- **FileShareItemImage**
(Same as ItemImage except uses FSToken instead of Token)
- **FileShareItemThumbnail**
(Same as ItemThumbnail except uses FSToken instead of Token)
- **FileShareRequestDocIndexing**
(Same as RequestDocIndexing except uses FSToken instead of Token)
- **GetClientAppInfo**
Returns the latest info (version, download url etc.) about a client app

- **GetDropBoxDefaults**
Returns the preferred FileReceive Link settings.
- **GetELinkDefaults**
Returns the preferred file share settings.
- **GroupShareAdd**
Create a new GroupShare: shares a folder from the active session user account to a different user.
- **GetUserParam**
This function returns the userparam for the provided token.
- **IndexedItemProperties**
This function indicates if the referenced file is page-indexed, for viewing in the Online Viewer. If the file is indexed, the function returns a value of 1 along with the number of indexed pages. This function can be used to check the status of a file that has been queued for indexing.
- **ItemAdvancedProperties**
Displays advance attributes of a folder or file. This includes comments, keyword metatags, and the Version Control status if Version Control is enabled.
- **ItemCommentAdd**
Add the comment of a file/folder.
- **ItemCommentRemove**
Remove the comment of a file/folder.
- **ItemCommentUpdate**
Update the comment of a file/folder.
- **ItemCommentsList**
Get the Comments list of a file/folder.
- **ItemCommentsWithThumbnail**
Get the Comments list of a file/folder, including the profile image of the commenter.
- **ItemDescription**
Allows users to get the Description of a file/folder.
- **ItemDescriptionAdd**
Add the Description of a file/folder.

- **ItemImage**
Displays an image of the referenced page within the referenced file. This function only returns an image if the file is done indexing.
- **ItemProperties**
Displays basic file properties. This includes the complete folder path, the file size, and last updated date.
- **ItemThumbnail**
Displays a thumbnail for the item. For image files and documents indexed for the Online Viewer, the thumbnail will be a small picture of the first page of the file. For folders, a folder icon is returned.
- **Logout**
Logs you out from your FilesAnywhere account
- **Paste**
Pastes items in the clipboard into the specified path
- **Renameltem**
Renames an Item
- **RequestDocIndexing**
Sends the system a request to index the referenced file. The system processes these requests in a queue. Once the document pages are indexed for viewing, the ItemImage function may be called to return the page image(s) in the file.
- **SendDropboxLink**
This operation is similar to the SendItemsELinkWithDefaults operation. This produces a web URL that anyone, even non-members, can use to upload files to a designated folder in a user account. An email is sent by the system containing an encrypted web link, which when activated by recipients, will prompt them to upload any files to the specified folder, according to the parameters (options) given when calling the function. If the 'RecipientsEmailAddresses' element is equal to the value 'GENERALLINK', the function will return the FileReceive URL Link without sending an email. This link can be used directly from any location, and the link behaves exactly like the way it does when sent in an email.
- **SendItemsELinkWithDefaults**
This operation is similar to the SendDropboxLink operation, except that it uses the defaults stored in the user profile. It shares files/folders securely with anyone, even non-members, using a web URL link. An email is sent by the system containing an encrypted web link, which when activated by recipients, will prompt them to download the shared files or folders or view files online, according to the parameters (options) given when calling the function. If the 'RecipientsEmailAddresses' element is equal to the value 'GENERALLINK', the function will return the FileShare URL Link without sending an email. This link can

be used directly from any location, and the link behaves exactly like the way it does when sent in an email.

- **UploadFile**
This operation lets you upload a file
- **UserAdd**
This operation lets the Private Site Administrator add a new User.
- **ViewFile**
This provides a temporary URL from which a file can be view.

For each operation, client applications must submit a synchronous request to the FAWAPI web service, wait for the web service's response and then process the results. All requests must be secured via SSL. For detailed information about supported web service operations, see the Service Description at:

<https://api.filesanywhere.com/v2/fawapi.asmx>

Standards Compliance

The API is compliant with the following specifications:

Standard	URL
Simple Object Access Protocol (SOAP) 1.1	http://www.w3.org/TR/2000/NOTE-SOAP-20000508/
Web Service Description Language (WSDL) 1.1	http://www.w3.org/TR/2001/NOTE-wsdl-20010315
WS-I Basic Profile 1.1	http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html

Let's Get Started!

This section helps you in getting started with using the API. It tells you what you need in order to start using the API in your development environment. It includes the following steps:

- Sign up for a FilesAnywhere account.
- Enable API access in your FilesAnywhere account.
- Obtain a Developer API Key.
- Obtain web service WSDL.
- Import the WSDL file into your development platform.
- Study the sample web application.
- Important Information

Enable API access in your FilesAnywhere account

Using your web browser, log in to your FilesAnywhere account and enable the API access in your profile.

Obtain a Developer API Key

Using your web browser, log in to your FilesAnywhere account and click on the “Tools” tab in the toolbar on the top. Then click on the “Developer API Key” link.

Obtain the web service WSDL

The WSDL file is available at <https://api.filesanywhere.com/v2/fawapi.asmx?WSDL>

To access the FilesAnywhere API web service, you need a Web Service Description Language (WSDL) file. WSDL files are used to communicate interface information between web service producers and consumers. A WSDL description allows a client to utilize a web service's capabilities without knowledge of the implementation details of the web service.

A WSDL file contains all of the information necessary for a client to invoke the methods of a web service:

- The data types used as method parameters or return values
- The individual methods names and signatures (WSDL refers to methods as operations)
- The protocols and message formats allowed for each method
- The URLs used to access the web service

Your development platform uses this WSDL to generate an API to access the FilesAnywhere API Web service it defines.

For more information about WSDL, see <http://www.w3.org/TR/wsdl>

Import the WSDL file into your development platform

You need to import the WSDL file into your development platform so that your development environment can generate the necessary objects for use in building client web service applications in that environment.

Here we provide you sample instructions for Microsoft Visual Studio. For instructions about other development platforms, see your platform's product documentation.

In order to use the API in Visual Studio, you must first generate a proxy class for the web service from the WSDL file.

To access the FilesAnywhere API web service from Visual Studio managed code you must:

- Add a web reference for the FAWAPI web service. The web reference creates a proxy class with methods that serve as proxies for each exposed method of the FAWAPI web service.
- Add the namespace for the web reference.
- Create an instance of the proxy class and then access the methods of the class.

To add a web reference:

- First open an existing application or create a new application.
- In the Solution Explorer, right-click on your project and select **Add Web Reference**.
- In the URL text box of the Add Web Reference dialog box, type in the URL of the FAWAPI web service or of the saved WSDL file i.e.:
 - <https://api.filesanywhere.com/v2/fawapi.asmx>
 - or
 - <file:///c:/fawapiWSDL/fawapi.wsdl>
- Click the GO button to retrieve information about the FAWAPI web service.
- In the Web Reference name text box, type in the namespace you will be using for this web reference.
- Click **Add Reference** to add a web reference for the FAWAPI web service. For more information, see the topic “Adding and Removing Web References” in the Visual Studio documentation.

- Visual Studio retrieves the service description and generates a proxy class for the FAWAPI web service to be used by your application.

Sample Code in This Document

The sample code for each specific function is found under each function reference in this document, for SOAP 1.1, SOAP 1.2 and Visual Basic .Net code.

Additional Sample Web Application Code

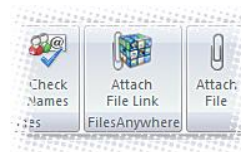
The sample code is available that will demonstrate how the API functions are used in a real-world, working web application.

The API was used exclusively in the development of these entire applications:

<https://m.filesanywhere.com>



FilesAnywhere Outlook Add-in



Apple iPhone Application



Try the Mobile Interface using any browser on any computer. The Outlook Add-in and FA Desktop for Mac runtime applications are available from our downloads page.

The sample code used in the creation of these apps can be sent to you upon request. Please email your request for sample code to support@filesanywhere.com

Important Information About Calling WebMethods

All parameters of methods are required. If you do not have value for some parameters then please pass String.empty ("") for String values and for number value pass 0 (zero)

instead of ignoring them in the API method call. Omitting any parameter will not give a desirable result.

Introducing the FilesAnywhere Digitally Signed API

Digitally signed API Service Description

For a formal definition, please review the Service Description at:

<https://apicert.filesanywhere.com/v2/fawapi.asmx>

Supported Operations

It supports all operations which are already supported by FilesAnywhere API

Let's Get Started!

This section helps you in getting started with using the digitally signed API. It tells you what you need in order to start using the digitally signed API in your development environment. It includes the following steps:

Prerequisites

Please refer support article <https://support.filesanywhere.com/hc/en-us/articles/115003773592> to know more about prerequisites required to use the digitally signed API in your API consumer application.

Note: Below sample code is given in the context of using GetAPIURL API method. You may modify the sample code to use any other supported API method

Important Note: Sample code is using Web Services Enhancements (WSE) 3.0 for Microsoft .NET. Please download and install the runtime from <https://www.microsoft.com/en-us/download/details.aspx?id=14089> on your development environment. Also make sure that your API consumer application is WSE 3.0 enabled; follow steps mentioned in article <https://msdn.microsoft.com/en-us/library/aa528776.aspx>

Sample VB.Net Code

```
Imports System
Imports Microsoft.Web.Services3
Imports Microsoft.Web.Services3.Design
Imports Microsoft.Web.Services3.Security
Imports Microsoft.Web.Services3.Security.Tokens
Imports System.Security.Cryptography.X509Certificates
Imports System.IdentityModel.Selectors

Public Class APICaller
    Private Sub GetAPIURL(ByVal sClientID As String, ByVal objClientCertificate As X509Certificate2, ByVal sAPIKey As String)
        Try
            Dim objProxy As WebReference.FAWAPIv2Wse = New WebReference.FAWAPIv2Wse
            'Create policy
            Dim objClientCertificatePolicyAssertion As ClientCertificatePolicyAssertion = New ClientCertificatePolicyAssertion(sClientID, objClientCertificate)
            Dim objPolicy As Policy = New Policy
            objPolicy.Assertions.Add(objClientCertificatePolicyAssertion)
            ' Add policy to web service proxy
            objProxy.SetPolicy(objPolicy)
            ' Call web method
            Dim sResult = objProxy.GetAPIURL(sAPIKey, sClientID)
            If (sResult.ErrorMessage <> String.Empty) Then
                Console.WriteLine(sResult.ErrorMessage)
            Return
            End If
            Console.WriteLine(sResult.APIURL)
        Catch ex As Exception
            Console.WriteLine(ex.Message)
        End Try
    End Sub
End Class

Public Class ClientCertificatePolicyAssertion
    Inherits SecurityPolicyAssertion
    Public ClientID As String
    Public ClientCertificate As X509Certificate2

    Public Sub New(ByVal sClientID As String, ByVal objClientCertificate As X509Certificate2)
        MyBase.New()
        Me.ClientID = sClientID
    End Sub
End Class
```

```

    Me.ClientCertificate = objClientCertificate
  End Sub

  Public Overrides Function CreateClientInputFilter(ByVal objFilterCreationContext As FilterCreationContext) As SoapFilter
    Return New ClientReceiveSecurityFilter(Me)
  End Function

  Public Overrides Function CreateClientOutputFilter(ByVal objFilterCreationContext As FilterCreationContext) As SoapFilter
    Return New ClientSendSecurityFilter(Me)
  End Function

  Public Overrides Function CreateServiceInputFilter(ByVal objFilterCreationContext As FilterCreationContext) As SoapFilter
    Return Nothing
  End Function

  Public Overrides Function CreateServiceOutputFilter(ByVal objFilterCreationContext As FilterCreationContext) As SoapFilter
    Return Nothing
  End Function
End Class
Public Class ClientSendSecurityFilter
  Inherits SendSecurityFilter
  Private objClientCertificatePolicyAssertion As ClientCertificatePolicyAssertion = Nothing
  Public Sub New(ByVal objClientCertificatePolicyAssertion As ClientCertificatePolicyAssertion)
    MyBase.New(Me.objClientCertificatePolicyAssertion.ServiceActor, True)
    Me.objClientCertificatePolicyAssertion = Me.objClientCertificatePolicyAssertion
  End Sub

  Public Overrides Sub SecureMessage(ByVal objSoapEnvelope As SoapEnvelope, ByVal objSecurity As Security)
    'Add ClientID in SOAP Body as MessageIdentifier
    Dim objMessageIdentifierParam = objSoapEnvelope.CreateElement("MessageIdentifier")
    objMessageIdentifierParam.InnerXml = String.Format("<{0}>{1}</{0}>", "ClientID",
Me.objClientCertificatePolicyAssertion.ClientID)
    objSoapEnvelope.Body.PrependChild(objMessageIdentifierParam)
    If (Not (Me.objClientCertificatePolicyAssertion.ClientCertificate) Is Nothing) Then
      Dim objX509SecurityToken As X509SecurityToken = New
X509SecurityToken(Me.objClientCertificatePolicyAssertion.ClientCertificate)
      If (objX509SecurityToken Is Nothing) Then
        Throw New SecurityFault("Message requirements could not be satisfied. Couldn't generate security token
using given client certificate.")
      End If

      ' Add the X.509 certificate to the header.
      objSecurity.Tokens.Add(objX509SecurityToken)
      ' Specify that the SOAP message is signed using this X.509 certificate.
      Dim objMessageSignature As MessageSignature = New MessageSignature(objX509SecurityToken)
      objSecurity.Elements.Add(objMessageSignature)
    End If
  End Sub
End Class

```

Class ClientReceiveSecurityFilter

Inherits ReceiveSecurityFilter

Private objServerSecurityToken As SecurityToken = Nothing

Private objServerMessageSignature As MessageSignature = Nothing

Public Sub New(ByVal parentAssertion As ClientCertificatePolicyAssertion)

MyBase.New(parentAssertion.ServiceActor, True)

End Sub

Public Overrides Sub ValidateMessageSecurity(ByVal envelope As SoapEnvelope, ByVal security As Security)

If (Not (security) Is Nothing) Then

' Make sure the message was signed with the server's security token.

For Each element As ISecurityElement In security.Elements

If (TypeOf element Is MessageSignature) Then

Me.objServerMessageSignature = CType(element, MessageSignature)

Me.objServerSecurityToken = Me.objServerMessageSignature.SigningToken

End If

Next

If (Not (Me.objServerSecurityToken) Is Nothing) Then

' The given context contains a Signature element.

Me.objServerMessageSignature = CType(Me.objServerMessageSignature, MessageSignature)

If Me.CheckSignature(envelope, security, Me.objServerMessageSignature) Then

' The SOAP message is signed.

If (TypeOf Me.objServerMessageSignature.SigningToken Is X509SecurityToken) Then

' The SOAP message is signed with a X509SecurityToken

Me.AuthenticateToken(CType(Me.objServerMessageSignature.SigningToken, X509SecurityToken))

End If

End If

End If

End If

End Sub

Private Function CheckSignature(ByVal envelope As SoapEnvelope, ByVal security As Security, ByVal signature As MessageSignature) As Boolean

' Now verify which parts of the message were actually signed.

Dim actualOptions As SignatureOptions = signature.SignatureOptions

Dim expectedOptions As SignatureOptions = SignatureOptions.IncludeSoapBody

If ((Not (security) Is Nothing) _

AndAlso (Not (security.Timestamp) Is Nothing)) Then

expectedOptions = (expectedOptions Or SignatureOptions.IncludeTimestamp)

End If

' The <Action> and <To> are required addressing elements.

expectedOptions = (expectedOptions Or SignatureOptions.IncludeAction)

expectedOptions = (expectedOptions Or SignatureOptions.IncludeTo)

If ((Not (envelope.Context.Addressing.FaultTo) Is Nothing) _

AndAlso (Not (envelope.Context.Addressing.FaultTo.TargetElement) Is Nothing)) Then

expectedOptions = (expectedOptions Or SignatureOptions.IncludeFaultTo)

End If

If ((Not (envelope.Context.Addressing.From) Is Nothing) _

AndAlso (Not (envelope.Context.Addressing.From.TargetElement) Is Nothing)) Then

expectedOptions = (expectedOptions Or SignatureOptions.IncludeFrom)

End If

```
If ((Not (envelope.Context.Addressing.MessageID) Is Nothing) _
    AndAlso (Not (envelope.Context.Addressing.MessageID.TargetElement) Is Nothing)) Then
    expectedOptions = (expectedOptions Or SignatureOptions.IncludeMessageId)
End If
```

```
If ((Not (envelope.Context.Addressing.RelatesTo) Is Nothing) _
    AndAlso (Not (envelope.Context.Addressing.RelatesTo.TargetElement) Is Nothing)) Then
    expectedOptions = (expectedOptions Or SignatureOptions.IncludeRelatesTo)
End If
```

```
If ((Not (envelope.Context.Addressing.ReplyTo) Is Nothing) _
    AndAlso (Not (envelope.Context.Addressing.ReplyTo.TargetElement) Is Nothing)) Then
    expectedOptions = (expectedOptions Or SignatureOptions.IncludeReplyTo)
End If
```

' Check if the all the expected options are the present.

```
Return ((expectedOptions And actualOptions) _
    = expectedOptions)
```

End Function

Private Sub AuthenticateToken(ByVal token As X509SecurityToken)

If (token Is Nothing) Then

Throw New SecurityFault("Response message does not meet security requirements. Response message is not signed by server.")

Else

'Add your custom logic to verify the server certificate here and throw errors appropriately

'Following is sample code to verify the server certificate used in signing the response

```
ClientReceiveSecurityFilter.ValidateServerCertificate(token.Certificate)
```

End If

End Sub

Public Shared Function ValidateServerCertificate(ByVal objX509Certificate2 As X509Certificate2) As Boolean

If (objX509Certificate2 Is Nothing) Then

Throw New SecurityFault("Response message does not meet security requirements. Response message is not signed by server.")

End If

```
Dim dtCurrentDate As DateTime = DateTime.Now
```

```
Dim bValid As Boolean = (objX509Certificate2.Verify _
```

```
    AndAlso ((dtCurrentDate <= objX509Certificate2.NotAfter.AddMinutes(5)) _
```

```
    AndAlso (dtCurrentDate >= objX509Certificate2.NotBefore.AddMinutes(-5))))
```

If Not bValid Then

Throw New SecurityFault("Message requirements could not be satisfied. Invalid server certificate.")

End If

```
Dim objX509CertificateValidator As X509CertificateValidator = X509CertificateValidator.ChainTrust
```

Try

```
objX509CertificateValidator.Validate(objX509Certificate2)
```

Catch ex As Exception

```
Throw New SecurityFault(ex.Message)
```

End Try

```
'Throw error if invalid certificate is used while signing or certificate is expired
If objX509Certificate2.Subject <> "<FilesAnywhere Server Certificate Subject Name>" OrElse
    String.IsNullOrEmpty(objX509Certificate2.Thumbprint) OrElse
    (Not String.IsNullOrEmpty(objX509Certificate2.Thumbprint) AndAlso Not
objX509Certificate2.Thumbprint.Equals("<FilesAnywhere Server Certificate Thumbprint >",
StringComparison.OrdinalIgnoreCase)) Then
    Throw New SecurityFault("Message requirements could not be satisfied. Invalid server certificate.")
End If
Return True
End Function
End Class
```

Sample C#.Net Code

```
using System;
using Microsoft.Web.Services3;
using Microsoft.Web.Services3.Design;
using Microsoft.Web.Services3.Security;
using Microsoft.Web.Services3.Security.Tokens;
using System.Security.Cryptography.X509Certificates;
using System.IdentityModel.Selectors;
public class APICaller
{
    private void GetAPIURL(string sClientID, X509Certificate2 objClientCertificate, string sAPIKey)
    {
        try
        {
            WebReference.FAWAPIv2Wse objProxy = new WebReference.FAWAPIv2Wse();
            //Create policy
            ClientCertificatePolicyAssertion objClientCertificatePolicyAssertion = new
ClientCertificatePolicyAssertion(sClientID, objClientCertificate);
            Policy objPolicy = new Policy();
            objPolicy.Assertions.Add(objClientCertificatePolicyAssertion);

            // Add policy to web service proxy
            objProxy.SetPolicy(objPolicy);

            // Call web method
            var sResult = objProxy.GetAPIURL(sAPIKey, sClientID);
            if (sResult.ErrorMessage != string.Empty)
            {
                Console.WriteLine(sResult.ErrorMessage);
                return;
            }
            Console.WriteLine(sResult.APIURL);
        }
        catch (Exception ex)
        {
            Console.WriteLine(ex.Message);
        }
    }
}
```

```
    }
  }
}

public class ClientCertificatePolicyAssertion : SecurityPolicyAssertion
{
    public string ClientID;
    public X509Certificate2 ClientCertificate;
    public ClientCertificatePolicyAssertion(string sClientID, X509Certificate2 objClientCertificate)
    {
        this.ClientID = sClientID;
        this.ClientCertificate = objClientCertificate;
    }

    public override SoapFilter CreateClientInputFilter(FilterCreationContext objFilterCreationContext)
    {
        return new ClientReceiveSecurityFilter(this);
    }

    public override SoapFilter CreateClientOutputFilter(FilterCreationContext objFilterCreationContext)
    {
        return new ClientSendSecurityFilter(this);
    }

    public override SoapFilter CreateServiceInputFilter(FilterCreationContext objFilterCreationContext)
    {
        return null;
    }

    public override SoapFilter CreateServiceOutputFilter(FilterCreationContext objFilterCreationContext)
    {
        return null;
    }
}

public class ClientSendSecurityFilter : SendSecurityFilter
{
    private ClientCertificatePolicyAssertion objClientCertificatePolicyAssertion = null;

    public ClientSendSecurityFilter(ClientCertificatePolicyAssertion objClientCertificatePolicyAssertion)
        : base(objClientCertificatePolicyAssertion.ServiceActor, true)
    {
        this.objClientCertificatePolicyAssertion = objClientCertificatePolicyAssertion;
    }

    public override void SecureMessage(SoapEnvelope objSoapEnvelope, Security objSecurity)
    {
        //Add ClientID in SOAP Body as MessageIdentifier
        var objMessageIdentifierParam = objSoapEnvelope.CreateElement("MessageIdentifier");
        objMessageIdentifierParam.InnerXml = string.Format("<{0}>{1}</{0}>", "ClientID",
objClientCertificatePolicyAssertion.ClientID);
        objSoapEnvelope.Body.PrependChild(objMessageIdentifierParam);
    }
}
```

```
    if (objClientCertificatePolicyAssertion.ClientCertificate != null)
    {
        X509SecurityToken objX509SecurityToken = new
X509SecurityToken(objClientCertificatePolicyAssertion.ClientCertificate);
        if (objX509SecurityToken == null)
        {
            throw new SecurityFault("Message requirements could not be satisfied. Couldn't generate security token
using given client certificate.");
        }

        // Add the X.509 certificate to the header.
objSecurity.Tokens.Add(objX509SecurityToken);

        // Specify that the SOAP message is signed using this X.509
// certificate.
MessageSignature objMessageSignature = new MessageSignature(objX509SecurityToken);
objSecurity.Elements.Add(objMessageSignature);
    }
}
}

class ClientReceiveSecurityFilter : ReceiveSecurityFilter
{
    SecurityToken objServerSecurityToken = null;
    MessageSignature objServerMessageSignature = null;
    public ClientReceiveSecurityFilter(ClientCertificatePolicyAssertion parentAssertion)
        : base(parentAssertion.ServiceActor, true)
    {
    }

    public override void ValidateMessageSecurity(SoapEnvelope envelope, Security security)
    {
        if (security != null)
        {
            // Make sure the message was signed with the server's security token.
            foreach (ISecurityElement element in security.Elements)
            {
                if (element is MessageSignature)
                {
                    objServerMessageSignature = (MessageSignature)element;
                    objServerSecurityToken = objServerMessageSignature.SigningToken;
                }
            }

            if (objServerSecurityToken != null)
            {
                // The given context contains a Signature element.
                objServerMessageSignature = objServerMessageSignature as MessageSignature;

                if (CheckSignature(envelope, security, objServerMessageSignature))
                {
                    // The SOAP message is signed.
                }
            }
        }
    }
}
```



```

        if (objServerMessageSignature.SigningToken is X509SecurityToken)
        {
            // The SOAP message is signed with a X509SecurityToken
            AuthenticateToken(objServerMessageSignature.SigningToken as X509SecurityToken);
        }
    }
}
}

private bool CheckSignature(SoapEnvelope envelope, Security security, MessageSignature signature)
{
    //
    // Now verify which parts of the message were actually signed.
    //
    SignatureOptions actualOptions = signature.SignatureOptions;
    SignatureOptions expectedOptions = SignatureOptions.IncludeSoapBody;

    if (security != null && security.Timestamp != null)
        expectedOptions |= SignatureOptions.IncludeTimestamp;

    //
    // The <Action> and <To> are required addressing elements.
    //
    expectedOptions |= SignatureOptions.IncludeAction;
    expectedOptions |= SignatureOptions.IncludeTo;

    if (envelope.Context.Addressing.FaultTo != null &&
        envelope.Context.Addressing.FaultTo.TargetElement != null)
        expectedOptions |= SignatureOptions.IncludeFaultTo;

    if (envelope.Context.Addressing.From != null &&
        envelope.Context.Addressing.From.TargetElement != null)
        expectedOptions |= SignatureOptions.IncludeFrom;

    if (envelope.Context.Addressing.MessageID != null &&
        envelope.Context.Addressing.MessageID.TargetElement != null)
        expectedOptions |= SignatureOptions.IncludeMessageId;

    if (envelope.Context.Addressing.RelatesTo != null &&
        envelope.Context.Addressing.RelatesTo.TargetElement != null)
        expectedOptions |= SignatureOptions.IncludeRelatesTo;

    if (envelope.Context.Addressing.ReplyTo != null &&
        envelope.Context.Addressing.ReplyTo.TargetElement != null)
        expectedOptions |= SignatureOptions.IncludeReplyTo;
    //
    // Check if the all the expected options are the present.
    //
    return ((expectedOptions & actualOptions) == expectedOptions);
}

private void AuthenticateToken(X509SecurityToken token)
{

```

```
        if (token == null)
        {
            throw new SecurityFault("Response message does not meet security requirements. Response message is
not signed by server.");
        }
        else
        {
            //Add your custom logic to verify the server certificate here and throw errors appropriately
            //Following is sample code to verify the server certificate used in signing the response
            ValidateServerCertificate(token.Certificate);
        }
    }
    public static bool ValidateServerCertificate(X509Certificate2 objX509Certificate2)
    {
        if (objX509Certificate2 == null)
        {
            throw new SecurityFault("Response message does not meet security requirements. Response message is
not signed by server.");
        }
        DateTime dtCurrentDate = DateTime.Now;
        bool bValid = objX509Certificate2.Verify() && dtCurrentDate <= objX509Certificate2.NotAfter.AddMinutes(5)
&& dtCurrentDate >= objX509Certificate2.NotBefore.AddMinutes(-5);
        if (!bValid)
        {
            throw new SecurityFault("Message requirements could not be satisfied. Invalid server certificate.");
        }

        X509CertificateValidator objX509CertificateValidator = X509CertificateValidator.ChainTrust;
        try
        {
            objX509CertificateValidator.Validate(objX509Certificate2);
        }
        catch (Exception ex)
        {
            throw new SecurityFault(ex.Message);
        }

        //Throw error if invalid certificate is used while signing or certificate is expired
        if (objX509Certificate2.Subject != "<FilesAnywhere Server Certificate Subject Name>" ||
String.IsNullOrEmpty(objX509Certificate2.Thumbprint) ||
(!String.IsNullOrEmpty(objX509Certificate2.Thumbprint) &&
!objX509Certificate2.Thumbprint.Equals("<FilesAnywhere Server Certificate Thumbprint >",
StringComparison.OrdinalIgnoreCase)))
        {
            throw new SecurityFault("Message requirements could not be satisfied. Invalid server certificate.");
        }
        return true;
    }
}
```

SOAP API Calls

AccountCreateFree

This method is available to FA partners, used to create new free (web basic) accounts.

Note: Method is not applicable for on- premise clients.

You will need to obtain permission first from FilesAnywhere to activate this feature. Please email support@filesanywhere.com and we will evaluate your request.

Syntax

AccountCreateFreeResult = AccountCreateFree (API_KEY, EmailAddress, UserName, Password)

Usage

Use the AccountCreateFree method to create new free (web basic) accounts. You supply your FilesAnywhere APIKey, the email address of the new account user, optional username (email address will be used if left blank), optional password (if left blank an email with a temporary password will be sent to the new user).

The following parameters are needed to make the AccountCreateFree call:

- APIKey (A Developer API Key. See the “Getting Started” section above, to find out how to obtain a Developer API Key)
- EmailAddress (This is the email address of the new user you are creating).
- Username (This is the username of the new account you are creating). If this is left blank then EmailAddress will be attempted as the Username. Then if the EmailAddress is too long then you will get an error.
- Password (This is the password of the new user you are creating) If this is left blank, then the user will get an email with a temporary password sent to them.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/AccountCreateFree"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <AccountCreateFree xmlns="http://api.filesanywhere.com/">
      <API_KEY>string</API_KEY>
      <EmailAddress>string</EmailAddress>
      <UserName>string</UserName>
      <Password>string</Password>
    </AccountCreateFree>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <AccountCreateFreeResponse xmlns="http://api.filesanywhere.com/">
      <AccountCreateFreeResult>
        <ErrorMessage>string</ErrorMessage>
        <Result>string</Result>
      </AccountCreateFreeResult>
    </AccountCreateFreeResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <AccountCreateFree xmlns="http://api.filesanywhere.com/">
      <API_KEY>string</API_KEY>
      <EmailAddress>string</EmailAddress>
      <UserName>string</UserName>
      <Password>string</Password>
    </AccountCreateFree>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <AccountCreateFreeResponse xmlns="http://api.filesanywhere.com/">
      <AccountCreateFreeResult>
        <ErrorMessage>string</ErrorMessage>
        <Result>string</Result>
      </AccountCreateFreeResult>
    </AccountCreateFreeResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub AccountCreateFree ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim AccountCreateFreeResult As New WebReference.AccountCreateFreeResult
    AccountCreateFreeResult =
        FAWAPI.AccountCreateFree (ApiKey,EmailAddress,Username,Password)
    If AccountCreateFreeResult.ErrorMessage <> "" Then
        MsgBox AccountCreateFreeResult.ErrorMessage
    End If
    AccountCreateFree = Nothing
    FAWAPI = Nothing
End Sub
```

Request Arguments

Name	Data Type	Required	Description
ApiKey	String	Yes	A Developer API Key. The "Getting Started" section describes how to obtain a Developer API Key
EmailAddress	String	Yes	EmailAddress of the user being created.
Username	String	No	Username of the user being created. If this is left blank the email address will be attempted. If this is too long (>20 characters) then an error will occur.
Password	String	No	Password of account being created. If this is left blank an email with a temporary password will be sent to the new user.

Response**AccountCreateFreeResult**

AccountCreateFreeResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the AccountCreateFreeResult method was executed. If the AccountCreateFreeResult call was successful, the ErrorMessage is blank
Result	String	Returns "success" as Result if AccountCreateFree call was a success

AccountLogin

This method logs you into your FilesAnywhere account and starts an API session. It returns back a Token ID which must be used for subsequent API calls.

Syntax

`LoginResult = AccountLogin(APIkey, OrgID, UserName, Password, AllowedIPList, ClientEncryptParam)`

Usage

Use the AccountLogin call to login to your FilesAnywhere account and start a client API session. A client application must first log in and obtain a Token ID before making other API calls.

The following parameters are needed to make the AccountLogin call:

- **APIKey** (A Developer API Key. See the “Getting Started” section above, to find out how to obtain a Developer API Key).
- **OrgID** (Client ID). The OrgID can have the following values:
 - 0 for a FilesAnywhere Web account.
 - 50 for a FilesAnywhere WebAdvanced account.
 - The Private Site Client ID for a FilesAnywhere Private Site.
- **UserName** (FilesAnywhere UserName). This is the FilesAnywhere username using with the user logs in to his FilesAnywhere account.
- **Password** (FilesAnywhere Password). This is the user’s FilesAnywhere account password
- **AllowedIPList** (List of allowed IPs). This parameter is optional. It is a comma-separated list of IPs that may be allowed to make subsequent API calls with the same Token ID which was passed back in this AccountLogin call.
- **ClientEncryptParam** (An optional parameter that can be used to get back an encrypted password which can be used for subsequent logins).
 - If no password encryption is desired, the parameter should be passed as blank.

- If the string “ENCRYPTEDNO” is passed as the value of this parameter, the LoginResult returns back an encrypted password. This encrypted password can be used in subsequent logins using the AccountLogin API method.
- When an encrypted password (returned back as explained above) is used to login using the AccountLogin method, the value of this parameter should be passed as “ENCRYPTEDYES”.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/AccountLogin"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <AccountLogin xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <OrgID>long</OrgID>
      <UserName>string</UserName>
      <Password>string</Password>
      <AllowedIPList>string</AllowedIPList>
      <ClientEncryptParam>string</ClientEncryptParam>
    </AccountLogin>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <AccountLoginResponse xmlns="http://api.filesanywhere.com/">
      <LoginResult>
        <ErrorMessage>string</ErrorMessage>
        <Token>string</Token>
        <ClientEncryptedPassword>string</ClientEncryptedPassword>
      </LoginResult>
    </AccountLoginResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <AccountLogin xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <OrgID>long</OrgID>
      <UserName>string</UserName>
      <Password>string</Password>
      <AllowedIPList>string</AllowedIPList>
      <ClientEncryptParam>string</ClientEncryptParam>
    </AccountLogin>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <AccountLoginResponse xmlns="http://api.filesanywhere.com/">
      <LoginResult>
        <ErrorMessage>string</ErrorMessage>
        <Token>string</Token>
        <ClientEncryptedPassword>string</ClientEncryptedPassword>
      </LoginResult>
    </AccountLoginResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub AccountLogin()  
    Dim FAWAPI As New WebReference.FAWAPI  
    Dim LoginResult As New WebReference.LoginResult  
  
    LoginResult = FAWAPI.AccountLogin(APIKey, OrgID, UserName,  
Password,  
AllowedIPList, ClientEncryptParam)  
    If LoginResult.ErrorMessage <> "" Then  
        MsgBox LoginResult.ErrorMessage  
    Else  
        Dim Token As String = LoginResult.Token  
    End If  
    LoginResult = Nothing  
    FAWAPI = Nothing  
End Sub
```

Sample Visual Basic.Net Code For User with UNKNOWN OrgID

NOTE: Use this sample when you are developing an application that can be used by any type of FilesAnywhere user: Web Basic, WebAdvanced, or Private Site accounts.

Overview of User Types

Web Basic Accounts: These users have basic file storage features. Web Basic accounts are created and login at www.filesanywhere.com.

WebAdvanced Accounts: These users have the basic features plus have advanced features including FTP, SFTP, WebDAV, Backup, and Sync. These WebAdvanced accounts are created and login at <https://backup.filesanywhere.com>.

Private Site Accounts: These users are created by the Administrators of Private Sites (and Dedicated Servers). These sites are private labeled and have their own specific web address for login. Administrators of each site are responsible for creating the users. Private Site users who wish to use an application developed using the API will need to precede their LoginID with the Client ID assigned to their site (ex: "999:username"), and your login process will need to detect this (unless your application is written specifically to make this choice for users of a known Private Site, in the case where the application you are developing is for a specific Private Site).

Recommended Login Method for Unknown User Type (“any” user):

The sample code below shows the logic that is recommended to automatically detect these three possible user types – useful for apps that are available to the public.

First, try to log the user in with OrgID=0 for WebBasic. If that login fails, have a condition to check the login using OrgID=50 for WebAdvanced users.

You may also look for a leading Client ID for Private Site users (branded FA sites).

```
Private Sub AccountLogin()  
    Dim OrgID As Integer  
    Dim StrOrgID As String = ""  
    Dim Position As Integer  
    Dim PrivateSiteLogin As Boolean = False  
    Dim UserName As String = Trim(UCase(txtUserName.Text))  
  
    '-- code added to allow private sites to login  
    '-- private sites can enter orgID:Username (e.g. 105:username)  
    Position = InStr(UserName, ":")  
    If Position > 1 Then '-- must have at least one character before the :  
        StrOrgID = Trim(Left(UserName, Position - 1))  
        If IsNumeric(StrOrgID) Then  
            OrgID = CInt(StrOrgID)  
            UserName = Trim(Mid(UserName, Position + 1))  
            PrivateSiteLogin = True  
            wsLoginResult = wsFAWAPI.Login(OrgID, UserName, txtPassword.Text,  
                AllowedIPList, Nothing)  
        End If  
    End If  
  
    If Not PrivateSiteLogin Then  
        '-- first try to login using OrgID 50, if not found then try using OrgID = 0  
        OrgID = 50  
        wsLoginResult = wsFAWAPI.AccountLogin(OrgID, UserName, txtPassword.Text,  
            AllowedIPList, Nothing)  
  
        If wsLoginResult.ErrorMessage <> "" Then  
            OrgID = 0  
            wsLoginResult = wsFAWAPI.AccountLogin(OrgID, UserName,  
                txtPassword.Text, AllowedIPList, Nothing)  
        End If  
    End If  
End Sub
```

Request Arguments

Name	Data Type	Required	Description
APIKey	String	Yes	A Developer API Key. The “Getting Started” section describes how to obtain a Developer API Key
OrgID	Long	Yes	The Client ID
UserName	String	Yes	FilesAnywhere Username
Password	String	Yes	FilesAnywhere Password
AllowedIPList	String	No	Comma-separated list of IPs to be allowed to make subsequent API calls with the same Token ID passed back by this AccountLogin call
ClientEncryptParam	String	No	<p>A parameter that can be blank or have the following values: “ENCRYPTEDYES”, “ENCRYPTEDNO”.</p> <p>When it has the value “ENCRYPTEDNO”, it means that the password being passed in the AccountLogin method is not encrypted, but, the LoginResult should pass back an encrypted password which will be used in subsequent AccountLogin calls.</p> <p>When an encrypted password is passed to the AccountLogin method, this parameter should be assigned the string value “ENCRYPTEDYES”. This tells the system, that the password being passed is an encrypted password and it will decrypt it before using it.</p>

Response

LoginResult

LoginResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the AccountLogin method was executed. If the AccountLogin call was successful, the ErrorMessage is blank
Token	String	A Token ID to be used in subsequent API calls
ClientEncryptedPassword	String	This will either be blank or will have an encrypted password if the value of the ClientEncryptParam in the AccountLogin call was "ENCRYPTEDNO".

AppendChunkWithFolderAccessKey

This method is used in the process of uploading large files using a `FolderAccessKey`. It appends a chunk of bytes to a file. The method must be called repeatedly until all the chunks have been sent. The client must ensure that all the chunks are sent in the right sequence.

Syntax

```
ChunkUploadResult = AppendChunkWithFolderAccessKey(APIKey, FolderAccessKey,
FileName, ChunkData, Offset, BytesRead, isLastChunk, UploadToken,
LastWriteTimeUTC, CreationTimeUtc)
```

Usage

Use the `AppendChunkWithFolderAccessKey` method to upload very large files using a `FolderAccessKey`. The Client application will have to send the file data in sequential chunks by repeated calls to the `AppendChunkWithFolderAccessKey` method. This method uses the MTOM protocol. MTOM stands for SOAP “Message Transmission Optimization Mechanism”.

The following parameters are needed to make a call to this API:

- `APIKey`. A Developer API Key. See the “Getting Started” section above, to find out how to obtain a Developer API Key.
- `FolderAccessKey`. Unique Access Key of a folder belonging to a specified user that can be generated using the `GetFolderAccessKey` API function.
- `FileName`. Name of the file being uploaded. The upload location will be determined by the `FolderAccessKey`.
- `ChunkData` (A chunk of binary data). Each chunk of data read from the file must be converted to the XML base64Binary format.
- `Offset`. The total number of bytes already sent in previous `AppendChunkWithFolderAccessKey` requests for the current file prior to this request.
- `BytesRead`. The size of the current chunk, in bytes.
- `isLastChunk`. It is a Boolean True or False indicating whether this is the last chunk of the file.

- UploadToken is a unique token required for Uploading a file which you will get as a return value of the first call to the AppendChunkWithFolderAccessKey method.
 - In the first call to the AppendChunkWithFolderAccessKey method, pass the value of UploadToken as blank.
 - In subsequent calls, pass the value of UploadToken which was retrieved as ChunkUploadResult.UploadToken from the first call.
- LastWriteTimeUtc is a UTC date-time stamp to be applied as the Last Modified date on the uploaded file. The value of the parameter can be null or a valid date-time value.
 - When a value is passed to this parameter, the API will update the Last Modified date-time of the file with the parameter value.
 - When a null value is passed to this parameter, the current date time will be applied as the Last Modified date-time of the file.
- CreationTimeUtc is a UTC date-time stamp to be applied as the Creation date on the uploaded file. Value of the parameter can be null or a valid date-time value.
 - When a value is passed to this parameter, the API will update the Creation date-time of the file with the parameter value.
 - When a null value is passed to this parameter, the current date-time will be applied as the Creation date-time of the file.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/AppendChunkWithFolderAccessKey"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <AppendChunkWithFolderAccessKey xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <FolderAccessKey>string</FolderAccessKey>
      <FileName>string</FileName>
      <ChunkData>base64Binary</ChunkData>
      <Offset>long</Offset>
      <BytesRead>int</BytesRead>
      <isLastChunk>boolean</isLastChunk>
      <UploadToken>string</UploadToken>
      <LastWriteTimeUtc>dateTime</LastWriteTimeUtc>
      <CreationTimeUtc>dateTime</CreationTimeUtc>
    </AppendChunkWithFolderAccessKey>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <AppendChunkWithFolderAccessKeyResponse
xmlns="http://api.filesanywhere.com/">
      <ChunkUploadResult>
        <ErrorMessage>string</ErrorMessage>
        <ChunkAppended>boolean</ChunkAppended>
        <UploadToken>string</UploadToken>
        <LastWriteTimeUtc>dateTime</LastWriteTimeUtc>
        <CreationTimeUtc>dateTime</CreationTimeUtc>
      </ChunkUploadResult>
    </AppendChunkWithFolderAccessKeyResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <AppendChunkWithFolderAccessKey xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <FolderAccessKey>string</FolderAccessKey>
      <FileName>string</FileName>
      <ChunkData>base64Binary</ChunkData>
      <Offset>long</Offset>
      <BytesRead>int</BytesRead>
      <isLastChunk>boolean</isLastChunk>
      <UploadToken>string</UploadToken>
      <LastWriteTimeUtc>dateTime</LastWriteTimeUtc>
      <CreationTimeUtc>dateTime</CreationTimeUtc>
    </AppendChunkWithFolderAccessKey>
  </soap12:Body>
</soap12:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <AppendChunkWithFolderAccessKeyResponse
xmlns="http://api.filesanywhere.com/">
      <ChunkUploadResult>
        <ErrorMessage>string</ErrorMessage>
        <ChunkAppended>boolean</ChunkAppended>
        <UploadToken>string</UploadToken>
        <LastWriteTimeUtc>dateTime</LastWriteTimeUtc>
        <CreationTimeUtc>dateTime</CreationTimeUtc>
      </ChunkUploadResult>
    </AppendChunkWithFolderAccessKeyResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```

Private Sub AppendChunkWithFolderAccessKey ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim ChunkUploadResult As New WebReference.ChunkUploadResult
    ChunkUploadResult = FAWAPI.AppendChunkWithFolderAccessKey (APIKey,
FolderAccessKey, FileName, ChunkData, Offset, BytesRead, isLastChunk,
UploadToken, LastWriteTimeUtc, CreationTimeUtc)

    If Not ChunkUploadResult.ChunkAppended Or
        ChunkUploadResult.ErrorMessage <> "" Then
        MsgBox ChunkUploadResult.ErrorMessage
    End If
    ChunkUploadResult = Nothing
    FAWAPI = Nothing
End Sub

```

Request Arguments

Name	Data Type	Required	Description
APIKey	String	Yes	A Developer API Key. See the “Getting Started” section above, to find out how to obtain a Developer API Key
FolderAccessKey	String	Yes	Unique Access Key of a folder belonging to a specified user that can be generated using the GetFolderAccessKey API function.
FileName	String	Yes	Name of the file being uploaded.
ChunkData	Base64Binary	Yes	Chunk of binary data in XML base64Binary format
Offset	Long	Yes	This is the physical location of the current chunk, in the file, in terms of bytes. The current chunk of data will be written starting from the next byte Offset bytes. So, it represents the total no. of bytes already sent in previous AppendChunk requests for the current file
BytesRead	Integer	Yes	Size of the chunk in bytes
islastChunk	Boolean	Yes	True or False indicating whether this is the last chunk for the file
UploadToken	String	Yes	Unique token required to upload a file.

LastWriteTimeUtc	dateTime	Yes	The date-time to be applied to the Last Modified date-time of the uploaded file. If this value is null, then the current date-time will be applied.
CreationTimeUtc	dateTime	Yes	The datetime to be applied to the Creation datetime of the uploaded file. If this value is null, then the current date-time will be applied.

Response**ChunkUploadResult**

ChunkUploadResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the AppendChunkWithFolderAccessKey method was executed. If the AppendChunkWithFolderAccessKey call was successful, the ErrorMessage is blank
ChunkAppended	Boolean	True or False indicating whether the API call was a success or a failure
UploadToken	String	A Token which needs to be used in all subsequent calls of the AppendChunkWithFolderAccessKey method after the first call.
LastWriteTimeUtc	dateTime	With the last call of the AppendChunkWithFolderAccessKey method (when the file has been successfully uploaded), the value of LastModifiedUtc of the uploaded file will be returned, for any other call before the last call, this value will be null.
CreationTimeUtc	dateTime	With the last call of the AppendChunkWithFolderAccessKey method (when the file has been successfully uploaded), the value of CreationTimeUtc of the uploaded file will be returned, for any other call before the last call, this value will be null .

AppendChunkGZip

This method is used in the process of uploading large files. It appends a chunk of bytes to a file. The method must be called repeatedly until all the chunks have been sent. The chunks must be compressed using the GZip standard. The client must ensure that all the chunks are sent in the right sequence.

Syntax

AppendChunkResult = AppendChunkGZip(Token, Path, ChunkData, Offset, BytesRead, isLastChunk)

Usage

Use the AppendChunkGZip method to upload very large files. The Client application will have to send the file data in sequential chunks, which have been compressed using the GZip standard, using repeated calls to the AppendChunkGZip method. This method uses the MTOM protocol. MTOM stands for SOAP “Message Transmission Optimization Mechanism”.

The following parameters are needed to make the AppendChunk call:

- Token (The Token ID returned by the last AccountLogin call).
- Path (Name of the file being uploaded along with the FilesAnywhere Path). The Path should be formatted like `\UserVolume\FolderName\FileName`, where Foldername is the name of the folder in which the file is to be uploaded and UserVolume is the FilesAnywhere UserName in whose account the file has to be uploaded.
- ChunkData (A chunk of binary data). Each chunk of data read from the file must be converted to the XML base64Binary format.
- Offset is the total no. of bytes already sent in previous AppendChunk requests for the current file prior to this request.
- BytesRead is the size of the current chunk, in bytes.
- isLastChunk is a Boolean True or False indicating whether this is the last chunk of the file.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/AppendChunkGZip"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <AppendChunkGZip xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
      <ChunkData>base64Binary</ChunkData>
      <Offset>long</Offset>
      <BytesRead>int</BytesRead>
      <isLastChunk>boolean</isLastChunk>
    </AppendChunkGZip>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    < AppendChunkGZipResponse xmlns="http://api.filesanywhere.com/">
      <AppendChunkResult>
        <ErrorMessage>string</ErrorMessage>
        <ChunkAppended>boolean</ChunkAppended>
      </AppendChunkResult>
    </AppendChunkGZipResponse>
  </soap:Body>
</soap:Envelope>
```


SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <AppendChunkGZip xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
      <ChunkData>base64Binary</ChunkData>
      <Offset>long</Offset>
      <BytesRead>int</BytesRead>
      <isLastChunk>boolean</isLastChunk>
    </AppendChunkGZip>
  </soap12:Body>
</soap12:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <AppendChunkGZipResponse xmlns="http://api.filesanywhere.com/">
      <AppendChunkResult>
        <ErrorMessage>string</ErrorMessage>
        <ChunkAppended>boolean</ChunkAppended>
      </AppendChunkResult>
    </AppendChunkGZipResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```

Private Sub AppendChunkGZip()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim AppendChunkResult As New WebReference.AppendChunkResult
    AppendChunkResult = FAWAPI.AppendChunkGZip(Token, Path, ChunkData,
    Offset, BytesRead, isLastChunk)

    If Not AppendChunkResult.ChunkAppended Or
        AppendChunkResult.ErrorMessage <> "" Then
        MsgBox AppendChunkResult.ErrorMessage
    End If
    AppendChunkResult = Nothing
    FAWAPI = Nothing
End Sub

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
Path	String	Yes	Full FilesAnywhere path of the file in the form \UserVolume\FolderName\FileName
ChunkData	Base64Binary	Yes	Chunk of binary data, compressed in GZip standard format, in XML base64Binary format
Offset	Long	Yes	This is the physical location of the current chunk, in the file, in terms of bytes. The current chunk of data will be written starting from the next byte Offset bytes. So, it represents the total no. of bytes already sent in previous AppendChunkGZip requests for the current file
BytesRead	Integer	Yes	Size of the chunk in bytes
islastChunk	Boolean	Yes	True or False indicating whether this is the last chunk for the file

Response**AppendChunkResult**

AppendChunkResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the AppendChunkGZip method was executed. If the AppendChunkGZip call was successful, the ErrorMessage is blank
ChunkAppended	Boolean	True or False indicating whether the AppendChunkGZip call was a success or a failure

ClientGetAddInVersion

This method returns the current version of the Client Office Add-in.

Syntax

`ClientGetAddInVersionResult = ClientGetAddInVersion(Token, AddInType)`

Usage

Use the `ClientGetAddInVersion` method to get the current version of the Client Office Add-in.

The following parameters are needed to make the `ClientGetAddInVersion` call:

- `Token` (The Token ID returned by the last `AccountLogin` call).
- `AddInType` is the type of the Add-in for which the information is being requested. For example, for the Microsoft Outlook Add-in, this parameter should have the value "OutlookAddin".

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/ClientGetAddInVersion"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ClientGetAddInVersion xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <AddinType>string</AddinType>
    </ClientGetAddInVersion>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ClientGetAddInVersionResponse xmlns="http://api.filesanywhere.com/">
      <ClientGetAddInVersionResult>
        <ErrorMessage>string</ErrorMessage>
        <Version>string</Version>
        <URL>string</URL>
      </ClientGetAddInVersionResult>
    </ClientGetAddInVersionResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ClientGetAddInVersion xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <AddinType>string</AddinType>
    </ClientGetAddInVersion>
  </soap12:Body>
</soap12:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ClientGetAddInVersionResponse xmlns="http://api.filesanywhere.com/">
      <ClientGetAddInVersionResult>
        <ErrorMessage>string</ErrorMessage>
        <Version>string</Version>
        <URL>string</URL>
      </ClientGetAddInVersionResult>
    </ClientGetAddInVersionResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub ClientGetAddInVersion()  
    Dim FAWAPI As New WebReference.FAWAPI  
    Dim ClientGetAddInVersionResult As New WebReference.  
        ClientGetAddInVersionResult  
    ClientGetAddInVersionResult = FAWAPI.ClientGetAddInVersion(Token,  
        AddinType)  
    If ClientGetAddInVersionResult.ErrorMessage <> "" Then  
        MsgBox ClientGetAddInVersionResult.ErrorMessage  
    End If  
    ClientGetAddInVersionResult = Nothing  
    FAWAPI = Nothing  
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
AddinType	String	Yes	The type of Add-in.

Response**ClientGetAddInVersionResult**

ClientGetAddInVersionResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ClientGetAddInVersion method was executed. If the ClientGetAddInVersion call was successful, the ErrorMessage is blank
Version	String	The current Version Number of the Add-in
URL	String	The URL from which the Add-in can be downloaded. The name of the Add-in must be appended to the URL in order to download the latest version of the Add-in.

ClientGetUploadPath

This method returns the path to which client applications will upload files.

Syntax

ClientGetUploadPathResult = ClientGetUploadPath(Token)

Usage

Use the ClientGetUploadPath method to get the path to which client applications will upload files.

The following parameters are needed to make the ClientGetUploadPath call:

- Token (The Token ID returned by the last AccountLogin call).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/ClientGetUploadPath"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ClientGetUploadPath xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </ClientGetUploadPath>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ClientGetUploadPathResponse xmlns="http://api.filesanywhere.com/">
      <ClientGetUploadPathResult>
        <ErrorMessage>string</ErrorMessage>
        <UploadPath>string</UploadPath>
      </ClientGetUploadPathResult>
    </ClientGetUploadPathResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ClientGetUploadPath xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </ClientGetUploadPath>
  </soap12:Body>
</soap12:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ClientGetUploadPathResponse xmlns="http://api.filesanywhere.com/">
      <ClientGetUploadPathResult>
        <ErrorMessage>string</ErrorMessage>
        <Version>string</Version>
        <URL>string</URL>
      </ClientGetUploadPathResult>
    </ClientGetUploadPathResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub ClientGetUploadPath()  
    Dim FAWAPI As New WebReference.FAWAPI  
    Dim ClientGetUploadPathResult As New WebReference.  
        ClientGetUploadPathResult  
    ClientGetUploadPathResult = FAWAPI.ClientGetUploadPath(Token)  
    If ClientGetUploadPathResult.ErrorMessage <> "" Then  
        MsgBox ClientGetUploadPathResult.ErrorMessage  
    End If  
    ClientGetUploadPathResult = Nothing  
    FAWAPI = Nothing  
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call

Response**ClientGetUploadPathResult**

ClientGetUploadPathResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ClientGetAddInVersion method was executed. If the ClientGetAddInVersion call was successful, the ErrorMessage is blank
UploadPath	String	The path to which Client Applications will upload files

ConvertDocToPDF

This method Convert document file into a PDF file.

Syntax

```
ConvertDocToPDFResult=ConvertDocToPDF(Token, SourceFilePath,  
DestinationFilePath)
```

Usage

Use the ConvertDocToPDF method to convert document file to PDF format.

Supported doc files are

"DOC,LWP,WRI,RTF,ANS,TXT,HTM,HTML,XML,WPD,WPS,PSW,PWD,PAB,OLK,MCW,DOT,DOCX,DOTX,DOCM,DOTM,LOG,DIC,ASC,ME,INI,TEXT,NFO,XLS,WK1,WK2,WK3,WK4,WK5,WKI,WKS,WKU,CSV,DIF,PRN,SLK,WQ1,XLT,XLW,XLSX,XLTX,XLSB,XLSM,XLTM,XLAM,PP T,PPTX,POTX,PPSX,PPTM,POTM,PPAM,PPSM,PPS"

The following parameters are needed to make the ConvertDocToPDF call:

- **Token** (The Token ID returned by the last AccountLogin call).
- **SourceFilePath** (The document file which user want to convert into PDF format. The document file path should be formatted like \UserVolume\FolderName\FileName where UserVolume is the FilesAnywhere UserName in whose account the file has to be uploaded.)
- **DestinationFilePath** (The destination path where the converted PDF file will be saved. In case of same PDF file is already exist at destination path then it will be overwritten. The Destination Path should be formatted like \UserVolume\FolderName\, where Foldername is the name of the folder in which the converted PDF file will be saved and UserVolume is the FilesAnywhere UserName.)

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: 64.156.56.142
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/ConvertDocToPDF"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ConvertDocToPDF xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <SourceFilePath>string</SourceFilePath>
      <DestinationFilePath>string</DestinationFilePath>
    </ConvertDocToPDF>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ConvertDocToPDFResponse xmlns="http://api.filesanywhere.com/">
      <ConvertDocToPDFResult>
        <ErrorMessage>string</ErrorMessage>
        <RequestId>int</RequestId>
      </ConvertDocToPDFResult>
    </ConvertDocToPDFResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: 64.156.56.142
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ConvertDocToPDF xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <SourceFilePath>string</SourceFilePath>
      <DestinationFilePath>string</DestinationFilePath>
    </ConvertDocToPDF>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ConvertDocToPDFResponse xmlns="http://api.filesanywhere.com/">
      <ConvertDocToPDFResult>
        <ErrorMessage>string</ErrorMessage>
        <RequestId>int</RequestId>
      </ConvertDocToPDFResult>
    </ConvertDocToPDFResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code


```

Private Sub ConvertDocToPDFTest ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim ConvertDocToPDFResult As New WebReference.
    ConvertDocToPDFResult
    ConvertDocToPDFResult = FAWAPI.
ConvertDocToPDF(Token, SourceFilePath, DestinationFilePath)
    If ConvertDocToPDFResult.ErrorMessage <> "" Then
        MsgBox ConvertDocToPDFResult.ErrorMessage
        Exit Sub
    End If
    Dim iRequestId as Integer = ConvertDocToPDFResult.RequestId
    MsgBox ConvertDocToPDFResult.RequestId

    ConvertDocToPDFResult = Nothing
    FAWAPI = Nothing
End Sub

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
SourceFilePath	String	Yes	The filepath which user wants to convert the document file into PDF. Full FilesAnywhere path of the document file in the form \UserVolume\FolderName\documentFileName.
DestinationFilePath	String	Yes	The Destination path where converted PDF file will be saved. In case of the same name PDF file is already existing at the destination path then it will be overwritten. Full FilesAnywhere path of folder in the form \UserVolume\FolderName\

Response**ConvertDocToPDFResult**

ConvertDocToPDFResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ConvertDocToPDF method was executed. If the ConvertToPDF call was successful, the ErrorMessage is blank
RequestId	Integer	The source filepath and destination file path is valid then it will return the RequestId that means Request is submitted. And by use of this RequestId user can use other API method ConvertDocToPDF status (Token, RequestID) to know the status of Document to PDF conversion.

ConvertDocToPDFStatus

This method allows user to get the status for Document to PDF Conversion.

Syntax

ConvertDocToPDFStatusResult =ConvertDocToPDFStatus(Token,RequestId)

Usage

Use the ConvertDocToPDFStatus method to get the status for document to PDF conversion.

The following parameters are needed to make the ConvertDocToPDFStatus call:

- Token (The Token ID returned by the last AccountLogin call).
- RequestId (The RequestId will be returned from ConvertDocToPDF(Token,SourceFilePath,DestinationFilePath) method),

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: 64.156.56.142
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/ConvertDocToPDFStatus"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ConvertDocToPDFStatus xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <RequestId>int</RequestId>
    </ConvertDocToPDFStatus>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ConvertDocToPDFStatusResponse xmlns="http://api.filesanywhere.com/">
      <ConvertDocToPDFStatusResult>
        <ErrorMessage>string</ErrorMessage>
        <Status>string</Status>
      </ConvertDocToPDFStatusResult>
    </ConvertDocToPDFStatusResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: 64.156.56.142
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ConvertDocToPDFStatus xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <RequestId>int</RequestId>
    </ConvertDocToPDFStatus>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ConvertDocToPDFStatusResponse xmlns="http://api.filesanywhere.com/">
      <ConvertDocToPDFStatusResult>
        <ErrorMessage>string</ErrorMessage>
        <Status>string</Status>
      </ConvertDocToPDFStatusResult>
    </ConvertDocToPDFStatusResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub ConvertDocToPDFStatusTest ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim ConvertDocToPDFStatusResult As New WebReference.
    ConvertDocToPDFStatusResult
    ConvertDocToPDFStatusResult = FAWAPI.
ConvertDocToPDFStatus (Token, RequestId)
    If ConvertDocToPDFStatusResult.ErrorMessage <> "" Then
        MsgBox ConvertDocToPDFStatusResult.ErrorMessage
        Exit Sub
    End If
    Dim sStatus as String = ConvertDocToPDFStatusResult.Status
    MsgBox ConvertDocToPDFStatusResult.Status

    ConvertDocToPDFStatusResult = Nothing
    FAWAPI = Nothing
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
RequestId	Integer	Yes	The Request Id returned back in the last ConverDocToPDF method call.

Response**ConvertDocToPDFStatusResult**

ConvertDocToPDFStatusResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ConvertDocToPDFStatus method was executed. If the ConvertDocToPDFStatus call was successful, the ErrorMessage is blank
Status	String	Status of conversion process of document file to PDF file. The returned status can be 'P', 'Y', 'C' or 'E' 'P' : Pending Request (means request is waiting to submit in process) 'Y' : Request is submitted but conversion is in Process. 'E' : Error occurs while conversion of Doc to PDF 'C' : Conversion of document DF file is completed.

ConvertImageToPDF

This method Convert Images into a PDF file.

Syntax

```
ConvertImageToPDFResult = ConvertImageToPDF(Token,SourceImagePath,
DestinationPath, PDFFileName, PageFormatType, ImageHeight, ImageWidth)
```

Usage

Use the ConvertImageToPDF method to convert images in to PDF format.

Supported image files are "JPG,JPEG,JPE,JFIF,GIF,PNG,BMP,WMF,DIB,TIF,TIFF".

The following parameters are needed to make the ConvertImageToPDF call:

- Token (The Token ID returned by the last AccountLogin call).
- SourceImagePath (The Images path in string array format which user want to convert into a single PDF format. The image path should be formatted like \UserVolume\FolderName\File Name where UserVolume is the FilesAnywhere UserName in whose account the file has to be uploaded.)
- DestinationPath (The destination path where the converted PDF file will be saved. The Destination Path should be formatted like \UserVolume\FolderName\, where Foldername is the name of the folder in which the converted PDF file will be saved and UserVolume is the FilesAnywhere UserName.)
- PDFFileName (The name of converted PDF file which consist of all images. This pdf file will be saved on destination path. In case of PDFfilename passed by user already existing at destination path then it will be overwritten)
- PageFormatType (The images in PDF page displaying in 5 format type. Format types are in between 2 to 6)
 - 2 AutoSize (All images will be display in PDF file at fit to page size)
 - 3 4 X 2 (Images will be display in 4 rows and 2 columns in a page)
 - 4 4 X 1 (Images will be display in 4 rows and 1 column in a page.)
 - 5 2 X 1 (Images will be display in 2 rows and 1 column in a page.)

6 Custom Size (user can change the height and width of image by passing height and width parameter in this method, as per that height and width image will be displayed in pdf format.)

- ImageHeight (User can pass height of image if PageformatType selection is 6 (Custom size)).
- ImageWidth (User can pass width pixels of image if PageFormatType selection is 6(CustomSize))

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/ConvertImageToPDF"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ConvertImageToPDF xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <SourceImagePath>
        <string>string</string>
        <string>string</string>
      </SourceImagePath>
      <DestinationPath>string</DestinationPath>
      <PDFFileName>string</PDFFileName>
      <PageFormatType>string</PageFormatType>
      <ImageHeight>string</ImageHeight>
      <ImageWidth>string</ImageWidth>
    </ConvertImageToPDF>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ConvertImageToPDFResponse xmlns="http://api.filesanywhere.com/">
      <ConvertImageToPDFResult>
        <ErrorMessage>string</ErrorMessage>
        <ImageConverted>boolean</ImageConverted>
      </ConvertImageToPDFResult>
    </ConvertImageToPDFResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ConvertImageToPDF xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <SourceImagePath>
        <string>string</string>
        <string>string</string>
      </SourceImagePath>
      <DestinationPath>string</DestinationPath>
      <PDFFileName>string</PDFFileName>
      <PageFormatType>string</PageFormatType>
      <ImageHeight>string</ImageHeight>
      <ImageWidth>string</ImageWidth>
    </ConvertImageToPDF>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ConvertImageToPDFResponse xmlns="http://api.filesanywhere.com/">
      <ConvertImageToPDFResult>
        <ErrorMessage>string</ErrorMessage>
        <ImageConverted>boolean</ImageConverted>
      </ConvertImageToPDFResult>
    </ConvertImageToPDFResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```

Private Sub ConvertImageToPDFTest ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim ConvertImageToPDFResult As New WebReference.
        ConvertImageToPDFResult
    ConvertImageToPDFResult = FAWAPI.
        ConvertImageToPDF(Token, SourceImagePath, DestinationPath, PDFFileName, PageFo
        rmatType, ImageHeight, ImageWidth)
    If ConvertImageToPDFResult.ErrorMessage <> "" Then
        MsgBox ConvertImageToPDFResult.ErrorMessage
        Exit Sub
    End If
    Dim isConverted as boolean= ConvertImageToPDFResult.ImageConverted
    MsgBox ConvertImageToPDFResult.ImageConverted

    ConvertImageToPDFResult = Nothing
    FAWAPI = Nothing
End Sub

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
SourceImagePath	String()	Yes	The filepath images which user wants to convert into a PDF. Full FilesAnywhere path of the file in the form \UserVolume\FolderName\ImageFileName.
DestinationPath	String	Yes	The Destination path where converted PDF file will be saved. Full FilesAnywhere path of folder in the form \UserVolume\FolderName\
PDFFileName	String	Yes	Name of PDF file which will be saved at destination folder. In case of this PDFFileName already exist at destination path then it will be overwritten.
PageFormatType	String	Yes	There are 5 format types in which images will be displayed in PDF file. 2 = Auto Size

			<p>3 = 4 X 2 (4 Rows and 2 column)</p> <p>4 = 4 X 1 (4 Rows and 1 column)</p> <p>5 = 2 X 1 (2 Rows and 1 column)</p> <p>6 = Custom Size (user has to pass imageheight and imagewidth, as per that height and width the images will be displayed in PDF file.)</p>
--	--	--	---

Response**ConvertImageToPDFResult**

ConvertImageToPDFResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ConvertImageToPDF method was executed. If the ConvertImageToPDF call was successful, the ErrorMessage is blank
ImageConverted	Boolean	If images successfully converted into PDF file then this will be returned true else false.

CopyItems

This method copies items into your FilesAnywhere API clipboard. The items are copied over to the new location in the next Paste call, after which they are removed from the clipboard. If another CopyItems method or CutItems method is called before a Paste call, the items in the clipboard get replaced by the newly selected items of the CopyItems or CutItems call.

Syntax

CopyItemsResult = CopyItems(Token, ItemsToCopy)

Usage

Use the CopyItems method to copy one or more items into your FilesAnywhere API clipboard for the purpose of copying from one location to another by issuing a subsequent Paste call.

The following parameters are needed to make the CopyItems call:

- Token (The Token ID returned by the last AccountLogin call).
- ItemsToCopy is an array of type Item. An item is defined by the following two parameters:
 - Type. This is the Item type and can either be a 'file' or a 'folder'.
 - Path (Name of the item with the FilesAnywhere Path). The Path should be formatted like \UserVolume\FolderName\ItemName, where UserVolume is the FilesAnywhere UserName in whose account the item exists.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/CopyItems"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CopyItems xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ItemsToCopy>
        <Item>
          <Type>string</Type>
          <Path>string</Path>
        </Item>
        <Item>
          <Type>string</Type>
          <Path>string</Path>
        </Item>
      </ItemsToCopy>
    </CopyItems>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CopyItemsResponse xmlns="http://api.filesanywhere.com/">
      <CopyItemsResult>
        <ErrorMessage>string</ErrorMessage>
        <Copied>boolean</Copied>
      </CopyItemsResult>
    </CopyItemsResponse>
  </soap:Body>
</soap:Envelope>
```


SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <CopyItems xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ItemsToCopy>
        <Item>
          <Type>string</Type>
          <Path>string</Path>
        </Item>
        <Item>
          <Type>string</Type>
          <Path>string</Path>
        </Item>
      </ItemsToCopy>
    </CopyItems>
  </soap12:Body>
</soap12:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <CopyItemsResponse xmlns="http://api.filesanywhere.com/">
      <CopyItemsResult>
        <ErrorMessage>string</ErrorMessage>
        <Copied>boolean</Copied>
      </CopyItemsResult>
    </CopyItemsResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```

Private Sub CopyItems ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim CopyItemsResult As New WebReference.CopyItemsResult
    CopyItemsResult = FAWAPI.CopyItems (Token, ItemsToCopy)
    If Not CopyItemsResult.Copied Or CopyItemsResult.ErrorMessage <>
        "" Then
        MsgBox CopyItemsResult.ErrorMessage
    End If
    CopyItemsResult = Nothing
    FAWAPI = Nothing
End Sub

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
ItemsToCopy	Array of type Item	Yes	An array of items.

Item Properties

An Item has the following properties:

Property Name	Data Type	Required	Description
Type	String	Yes	Type can either be a 'file' or a 'folder'
Path	String	Yes	Full FilesAnywhere path of the Item in the form \UserVolume\FolderName\ItemName

Response**CopyItemsResult**

CopyItemsResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the CopyItems method was executed. If the CopyItems call was successful, the ErrorMessage is blank
Copied	Boolean	True or False indicating whether the CopyItems call was a success or a failure

CreateFolder

This method creates a new folder in the specified path.

Syntax

CreateFolderResult = CreateFolder(Token, Path, NewFolderName)

Usage

Use the CreateFolder method to create a new folder.

The following parameters are needed to make the CreateFolder call:

- Token (The Token ID returned by the last AccountLogin call).
- Path (Full FilesAnywhere Path). This is the FilesAnywhere Path of the folder inside which the new folder will be created. The Path should be formatted like \UserVolume\FolderName, where UserVolume is the FilesAnywhere UserName in whose account the folder will be created and FolderName is the folder or subfolder where the new folder will be created.
- NewFolderName is the name of the new folder.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/CreateFolder"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CreateFolder xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
      <NewFolderName>string</NewFolderName>
    </CreateFolder>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CreateFolderResponse xmlns="http://api.filesanywhere.com/">
      <CreateFolderResult>
        <ErrorMessage>string</ErrorMessage>
        <FolderCreated>boolean</FolderCreated>
      </CreateFolderResult>
    </CreateFolderResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <CreateFolder xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
      <NewFolderName>string</NewFolderName>
    </CreateFolder>
  </soap12:Body>
</soap12:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <CreateFolderResponse xmlns="http://api.filesanywhere.com/">
      <CreateFolderResult>
        <ErrorMessage>string</ErrorMessage>
        <FolderCreated>boolean</FolderCreated>
      </CreateFolderResult>
    </CreateFolderResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub CreateFolder()  
    Dim FAWAPI As New WebReference.FAWAPI  
    Dim CreateFolderResult As New WebReference.CreateFolderResult  
    CreateFolderResult = FAWAPI.CreateFolder(Token, Path,  
    NewFolderName)  
    If Not CreateFolderResult.FolderCreated Or  
    CreateFolderResult.ErrorMessage <> "" Then  
        MsgBox CreateFolderResult.ErrorMessage  
    End If  
    CreateFolderResult = Nothing  
    FAWAPI = Nothing  
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
Path	String	Yes	Full FilesAnywhere path of the folder or subfolder in the form \UserVolume\FolderName
NewFolderName	String	Yes	The name of the new folder

Response

CreateFolderResult

CreateFolderResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the CreateFolder method was executed. If the CreateFolder call was successful, the ErrorMessage is blank
FolderCreated	Boolean	True or False indicating whether the CreateFolder call was a success or a failure

CreateFolderRecursive

This method creates a new folder in the specified path. If any of the parent folders in the specified path don't exist then it creates those parent folders also, before creating the specified new folder.

Syntax

CreateFolderRecursiveResult = CreateFolderRecursive(Token, Path, NewFolderName)

Usage

Use the CreateFolderRecursive method to create a new folder. If any of the parent folders in the specified path don't exist then those parent folders are created before creating the specified new folder.

The following parameters are needed to make the CreateFolderRecursive call:

- Token (The Token ID returned by the last AccountLogin call).
- Path (Full FilesAnywhere Path). This is the FilesAnywhere Path of the folder inside which the new folder will be created. The Path should be formatted like `\UserVolume\FolderName1\FolderName2\`, where `UserVolume` is the FilesAnywhere UserName in whose account the folder will be created and `FolderName` is the folder or subfolder where the new folder will be created. It is not necessary that `FolderName1` or `FolderName2` exist. They will be created by this function.
- NewFolderName is the name of the new folder.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.cloudpc.me
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/CreateFolderRecursive"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CreateFolderRecursive xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
      <NewFolderName>string</NewFolderName>
    </CreateFolderRecursive>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CreateFolderRecursiveResponse xmlns="http://api.filesanywhere.com/">
      <CreateFolderRecursiveResult>
        <ErrorMessage>string</ErrorMessage>
        <FolderCreated>boolean</FolderCreated>
        <NewFolderName>string</NewFolderName>
      </CreateFolderRecursiveResult>
    </CreateFolderRecursiveResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.cloudpc.me
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <CreateFolderRecursive xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
      <NewFolderName>string</NewFolderName>
    </CreateFolderRecursive>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <CreateFolderRecursiveResponse xmlns="http://api.filesanywhere.com/">
      <CreateFolderRecursiveResult>
        <ErrorMessage>string</ErrorMessage>
        <FolderCreated>boolean</FolderCreated>
        <NewFolderName>string</NewFolderName>
      </CreateFolderRecursiveResult>
    </CreateFolderRecursiveResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub CreateFolderRecursive()  
  
    Dim FAWAPI As New WebReference.FAWAPIv2SoapClient  
    Dim CreateFolderRecursiveResult As New  
WebReference.CreateFolderRecursiveResult  
    CreateFolderRecursiveResult = FAWAPI.CreateFolderRecursive(Token,  
Path,  
NewFolderName)  
    If Not CreateFolderRecursiveResult.FolderCreated Or  
CreateFolderRecursiveResult.ErrorMessage <> "" Then  
        MsgBox CreateFolderRecursiveResult.ErrorMessage  
    End If  
    CreateFolderRecursiveResult = Nothing  
    FAWAPI = Nothing  
  
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
Path	String	Yes	Full FilesAnywhere path of the folder or subfolder in the form \UserVolume\FolderName1\FolderName2\ the form
NewFolderName	String	Yes	The name of the new folder

Response

CreateFolderRecursiveResult

CreateFolderRecursiveResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the CreateFolder method was executed. If the CreateFolder call was successful, the ErrorMessage is blank
FolderCreated	Boolean	True or False indicating whether the CreateFolder call was a success or a failure
NewFolderName	String	If the new folder name sent in the request contains "." Or ".." or a space in the beginning then it will be removed from the name and the new name will be passed in the response through NewFolderName

CutItems

This method cuts items and puts them into your FilesAnywhere API clipboard. The items are not removed from the original location until a Paste call is made. When a paste call is made, the items are moved from the original location to the new location, after which, they are removed from the clipboard. If another CopyItems method or CutItems method is called before a Paste call, the items in the clipboard get replaced by the newly selected items of the CopyItems or CutItems call.

Syntax

CutItemsResult = CutItems(Token, ItemsToCut)

Usage

Use the CutItems method to keep one or more items in your FilesAnywhere API clipboard for the purpose of moving them from one location to another by issuing a subsequent Paste call.

The following parameters are needed to make the CutItems call:

- Token (The Token ID returned by the last AccountLogin call).
- ItemsToCut is an array of type Item. An item is defined by the following two parameters:
 - Type. This is the Item type and can either be a 'file' or a 'folder'.
 - Path (Name of the item with the FilesAnywhere Path). The Path should be formatted like \UserVolume\FolderName\ItemName, where UserVolume is the FilesAnywhere UserName in whose account the item exists.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/CutItems"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CutItems xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ItemsToCut>
        <Item>
          <Type>string</Type>
          <Path>string</Path>
        </Item>
        <Item>
          <Type>string</Type>
          <Path>string</Path>
        </Item>
      </ItemsToCut>
    </CutItems>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CutItemsResponse xmlns="http://api.filesanywhere.com/">
      <CutItemsResult>
        <ErrorMessage>string</ErrorMessage>
        <Cut>boolean</Cut>
      </CutItemsResult>
    </CutItemsResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <CutItems xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ItemsToCut>
        <Item>
          <Type>string</Type>
          <Path>string</Path>
        </Item>
        <Item>
          <Type>string</Type>
          <Path>string</Path>
        </Item>
      </ItemsToCut>
    </CutItems>
  </soap12:Body>
</soap12:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <CutItemsResponse xmlns="http://api.filesanywhere.com/">
      <CutItemsResult>
        <ErrorMessage>string</ErrorMessage>
        <Cut>boolean</Cut>
      </CutItemsResult>
    </CutItemsResponse>
  </soap12:Body>
</soap12:Envelope>
```


Sample Visual Basic.Net Code

```

Private Sub CutItems ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim CutItemsResult As New WebReference.CutItemsResult
    CutItemsResult = FAWAPI.CutItems (Token, ItemsToCut)
    If Not CutItemsResult.Cut Or CutItemsResult.ErrorMessage <> ""
    Then
        MsgBox CutItemsResult.ErrorMessage
    End If
    CutItemsResult = Nothing
    FAWAPI = Nothing
End Sub

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
ItemsToCut	Array of type Item	Yes	An array of items.

An Item has the following properties:

Property Name	Data Type	Required	Description
Type	String	Yes	Type can either be a 'file' or a 'folder'
Path	String	Yes	Full FilesAnywhere path of the Item in the form \UserVolume\FolderName\ItemName

Response**CutItemsResult**

CutItemsResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the CutItems method was executed. If the CutItems call was successful, the ErrorMessage is blank
Cut	Boolean	True or False indicating whether the CutItems call was a success or a failure

Deleteltems

This method deletes one or more items.

Syntax

DeleteltemsResult = Deleteltems(Token, ItemsToDelete)

Usage

Use the Deleteltems method to delete one or more items.

The following parameters are needed to make the Deleteltems call:

- Token (The Token ID returned by the last AccountLogin call).
- ItemsToDelete is an array of type Item. An item is defined by the following two parameters:
 - Type. This is the Item type and can either be a 'file' or a 'folder'.
 - Path (Name of the item with the FilesAnywhere Path). The Path should be formatted like \UserVolume\FolderName\ItemName, where UserVolume is the FilesAnywhere UserName in whose account the item exists.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/DeleteItems"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <DeleteItems xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ItemsToDelete>
        <Item>
          <Type>string</Type>
          <Path>string</Path>
        </Item>
        <Item>
          <Type>string</Type>
          <Path>string</Path>
        </Item>
      </ItemsToDelete>
    </DeleteItems>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <DeleteItemsResponse xmlns="http://api.filesanywhere.com/">
      <DeleteItemsResult>
        <ErrorMessage>string</ErrorMessage>
        <Deleted>boolean</Deleted>
      </DeleteItemsResult>
    </DeleteItemsResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <DeleteItems xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ItemsToDelete>
        <Item>
          <Type>string</Type>
          <Path>string</Path>
        </Item>
        <Item>
          <Type>string</Type>
          <Path>string</Path>
        </Item>
      </ItemsToDelete>
    </DeleteItems>
  </soap12:Body>
</soap12:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <DeleteItemsResponse xmlns="http://api.filesanywhere.com/">
      <DeleteItemsResult>
        <ErrorMessage>string</ErrorMessage>
        <Deleted>boolean</Deleted>
      </DeleteItemsResult>
    </DeleteItemsResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```

Private Sub DeleteItems ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim DeleteItemsResult As New WebReference.DeleteItemsResult
    DeleteItemsResult = FAWAPI.DeleteItems(Token, ItemsToDelete)
    If Not DeleteItemsResult.Deleted Or DeleteItemsResult.ErrorMessage
    <> "" Then
        MsgBox DeleteItemsResult.ErrorMessage
    End If
    DeleteItemsResult = Nothing
    FAWAPI = Nothing
End Sub

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
ItemsToDelete	Array of type Item	Yes	An array of items.

An Item has the following properties:

Property Name	Data Type	Required	Description
Type	String	Yes	Type can either be a 'file' or a 'folder'
Path	String	Yes	Full FilesAnywhere path of the Item in the form \UserVolume\FolderName\ItemName

Response**DeleteltemsResult**

DeleteltemsResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the Deleteltems method was executed. If the Deleteltems call was successful, the ErrorMessage is blank.
Deleted	Boolean	True or False indicating whether the Deleteltems call was a success or a failure

FileShareAppendComment

This method allows users of the FileShare link to append comments to a file.

Syntax

FileShareAppendCommentResult = FileShareAppendComment(FSToken, FileID, Path, AppendersName, NewComment)

Usage

This method allows users of the FileShare link to append comments to a file. If the user of the FileShare link is viewing the link in the Comments View the user can append their comments to a file. This is a nice feature for photos or documents you want to get feedback on.

The following parameters are needed to make the FileShareAppendComment call:

- FSToken (The FileShare Token ID returned by the last FileShareGetHeaderInfo call).
- FileID – Part of the key to the file or folder item in a FileShare. This is returned from the FileShareGetItems method.
- Path – Second part of the key to the file or folder item in a FileShare. This is returned from the FileShareGetItems method. This the file or folder path to the item you are commenting on.
- AppendersName – Name of the user that is appending the comment. This can be any string 20 characters or less. ex: from first/last name or an email address or an alias. Just something that identifies the user to the owner of the FileShare link.
- NewComment – The comment you are appending. This will NOT replace the current comment it will simply be added on to the existing comment with a header in the following format:

Straight from the database no formatting added

• AppendersName 1/8/2009 4:45PM •
NewComment.

Header data located and formatted (smaller,gray,italics)

• *AppendersName 1/8/2009 4:45PM* •
NewComment.

The header can be identified by
Begin: char(10) + char(149) + space
End: M + space + char(149) + char(10)

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /FAWAPI.asmx HTTP/1.1
Host: 64.156.56.142
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/FileShareAppendComment"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <FileShareAppendComment xmlns="http://api.filesanywhere.com/">
      <FSToken>string</FSToken>
      <FileID>string</FileID>
      <Path>string</Path>
      <AppendersName>string</AppendersName>
      <NewComment>string</NewComment>
    </FileShareAppendComment>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <FileShareAppendCommentResponse xmlns="http://api.filesanywhere.com/">
      <FileShareAppendCommentResult>
        <ErrorMessage>string</ErrorMessage>
        <CommentAppended>boolean</CommentAppended>
      </FileShareAppendCommentResult>
    </FileShareAppendCommentResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /FAWAPI.asmx HTTP/1.1
Host: 64.156.56.142
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <FileShareAppendComment xmlns="http://api.filesanywhere.com/">
      <FSToken>string</FSToken>
      <FileID>string</FileID>
      <Path>string</Path>
      <AppendersName>string</AppendersName>
      <NewComment>string</NewComment>
    </FileShareAppendComment>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <FileShareAppendCommentResponse xmlns="http://api.filesanywhere.com/">
      <FileShareAppendCommentResult>
        <ErrorMessage>string</ErrorMessage>
        <CommentAppended>boolean</CommentAppended>
      </FileShareAppendCommentResult>
    </FileShareAppendCommentResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub FileShareAppendComment ()
Dim FAWAPI As New WebReference.FAWAPI
Dim FileShareAppendCommentResult As New WebReference.FileShareAppendCommentResult
FileShareAppendCommentResult =
FAWAPI.FileShareAppendCommentInfo (FSToken, FileID, Path, AppendersName, NewComment)
If FileShareAppendCommentResult.ErrorMessage <> "" Then
Msgbox FileShareAppendCommentResult.ErrorMessage
End If
FileShareAppendCommentResult = Nothing
FAWAPI = Nothing
End Sub
```

Request Arguments

Name	Data Type	Required	Description
FSToken	String	Yes	ID to be used in subsequent FileShare API calls.
FileID	String	Yes	ID of the FileShare item. This is NOT a unique key. The unique key for a FileShare item is FileID + Path.
Path	String	Yes	Path of the FileShare item.
AppendersName	String	Yes	20 character or less string that identifies the user as the person that made the comment. Ex: first/last name or email.
NewComment	String	Yes	Comment to be appended to a FileShare item.

Response

FileShareAppendCommentResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the FileShareAppendComment method was executed. If the FileShareAppendComment call was successful, the ErrorMessage is blank
CommentAppended	Boolean	If the append procedure was successful this will be true else it will be false.

FileShareGetHeaderInfo

This method retrieves the header information for a FileShare link.

Syntax

FileShareHeaderResult =

FileShareGetHeaderInfo(FileShareKey, ClientID, Password, AllowedIPList)

Usage

This method allows you to get the FileShare header information such as the information about the person sharing the link. It returns a FSToken that is used for validation to make subsequent calls. Think of this method as FileShare Login.

The following parameters are needed to make the FileShareGetHeaderInfo call:

- FileShareKey - is the alphanumeric key that is on the querystring when you click on a FileShare link. The value of “v=” on the querystring is the FileShareKey.
- ClientID (Client ID). The ClientID can have the following values:
 - 0 for a FilesAnywhere Web account.
 - 50 for a FilesAnywhere WebAdvanced account.
 - The Private Site Client ID for a FilesAnywhere Private Site.
- Password - is provided when the creator of the FileShare link password protects their link. If one is required an error letting you know to provide one will be returned.
- AllowedIPList (List of allowed IPs). This parameter is optional. It is a comma-separated list of IPs that may be allowed to make subsequent API calls with the same Token ID which was passed back in this AccountLogin call.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/FileShareGetHeaderInfo"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <FileShareGetHeaderInfo xmlns="http://api.filesanywhere.com/">
      <FileShareKey>string</FileShareKey>
      <ClientID>string</ClientID>
      <Password>string</Password>
      <AllowedIPList>string</AllowedIPList>
    </FileShareGetHeaderInfo>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <FileShareGetHeaderInfoResponse xmlns="http://api.filesanywhere.com/">
      <FileShareHeaderResult>
        <ErrorMessage>string</ErrorMessage>
        <FSToken>string</FSToken>
        <SharedBy>string</SharedBy>
        <FirstName>string</FirstName>
        <LastName>string</LastName>
        <EmailAddress>string</EmailAddress>
        <RequiresPassword>string</RequiresPassword>
        <UserView>string</UserView>
        <PlayOptions>string</PlayOptions>
        <CoverPage>string</CoverPage>
        <WaterMark>string</WaterMark>
        <ViewOnly>string</ViewOnly>
      </FileShareHeaderResult>
    </FileShareGetHeaderInfoResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <FileShareGetHeaderInfo xmlns="http://api.filesanywhere.com/">
      <FileShareKey>string</FileShareKey>
      <ClientID>string</ClientID>
      <Password>string</Password>
      <AllowedIPList>string</AllowedIPList>
    </FileShareGetHeaderInfo>
  </soap12:Body>
</soap12:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <FileShareGetHeaderInfoResponse xmlns="http://api.filesanywhere.com/">
      <FileShareHeaderResult>
        <ErrorMessage>string</ErrorMessage>
        <FSToken>string</FSToken>
        <SharedBy>string</SharedBy>
        <FirstName>string</FirstName>
        <LastName>string</LastName>
        <EmailAddress>string</EmailAddress>
        <RequiresPassword>string</RequiresPassword>
        <UIView>string</UIView>
        <PlayOptions>string</PlayOptions>
        <CoverPage>string</CoverPage>
        <WaterMark>string</WaterMark>
        <ViewOnly>string</ViewOnly>
      </FileShareHeaderResult>
    </FileShareGetHeaderInfoResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub FileShareGetHeaderInfo()  
    Dim FAWAPI As New WebReference.FAWAPI  
    Dim FileShareHeaderResult As New WebReference.FileShareHeaderResult  
        FileShareHeaderResult =  
            FAWAPI.FileShareGetHeaderInfo(FileShareKey, ClientID, Password, AllowedIPList)  
    If FileShareHeaderResult.ErrorMessage <> "" Then  
        MsgBox FileShareHeaderResult.ErrorMessage  
    End If  
    FileShareHeaderResult = Nothing  
    FAWAPI = Nothing  
End Sub
```

Request Arguments

Name	Data Type	Required	Description
FileShareKey	String	Yes	Alphanumeric key that is on the querystring when you click on a FileShare link. The value of "v=" on the querystring is the FileShareKey.
ClientID	String	Yes	The Client ID.
Password	String	No	Password set up by FileShare creator. (optional)
AllowedIPList	String	No	Comma-separated list of IPs to be allowed to make subsequent API calls with the same FSToken ID passed back by this FileShareGetHeaderInfo call

Response

FileShareHeaderResult

FileShareHeaderResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the FileShareGetHeaderInfo method was executed. If the FileShareGetHeaderInfo call was successful, the ErrorMessage is blank
FSToken	String	ID to be used in subsequent FileShare API calls.
SharedBy	String	The person's user name that created the FileShare link.
FirstName	String	The sharer's first name. (may be blank)
LastName	String	The sharer's last name. (may be blank)
EmailAddress	String	The sharer's email address.
RequiresPassword	String	Whether the sharer password protected the FileShare link. "Y" or "N" or blank.
UserView	String	The default view the sharer is requesting to be seen for the FileShare link. "L" is for CommentsView, "F" is for ListView (default), "M" is for IconView, "A" is for AutoPlay (ex: slide show or music player).
PlayOptions	String	The default player the sharer is requesting to be used to play this link. "S" is for slideshow, "D" is for document viewer, "M" is for music player, "V" is for video player. This is only valid for UserView of type "A". (may be blank)
CoverPage	String	Whether there is a cover page. "Y" or "N"
WaterMark	String	Whether there is a WaterMark. "Y" or "N"
ViewOnly	String	Whether or not the FileShare is ViewOnly. "Y" is for NO downloading of files allowed. "N" is to allow downloading.

FileShareGetItems

This method retrieves a list of files and folders for a given path in a FileShare link.

Syntax

FileShareItemResult =

FileShareGetItems(FSToken, FileID, FolderPath, IncludeComments)

Usage

This method retrieves a list of the files and folders that make up a FileShare link.

The following parameters are needed to make the FileShareGetItems call:

- **FSToken** (The FileShare Token ID returned by the last FileShareGetHeaderInfo call).
- **FileID** - Part of the key to the file or folder item in a FileShare. For the first call of this method, the value will be blank.
- **Path** - Second part of the key to the file or folder item in a FileShare. For the first call of this method, the value will be blank.
- **IncludeComments** - Retrieves the comments that may be associated with the FileShare item.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/FileShareGetItems"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <FileShareGetItems xmlns="http://api.filesanywhere.com/">
      <FSToken>string</FSToken>
      <FileID>string</FileID>
      <FolderPath>string</FolderPath>
      <IncludeComments>string</IncludeComments>
    </FileShareGetItems>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <FileShareGetItemsResponse xmlns="http://api.filesanywhere.com/">
      <FileShareItemResult>
        <ErrorMessage>string</ErrorMessage>
        <FileShareItems>
          <FileShareItem>
            <Type>string</Type>
            <Name>string</Name>
            <Size>string</Size>
            <DateLastModified>string</DateLastModified>
            <Path>string</Path>
            <FileID>string</FileID>
            <Comments>string</Comments>
          </FileShareItem>
          <FileShareItem>
            <Type>string</Type>
            <Name>string</Name>
            <Size>string</Size>
            <DateLastModified>string</DateLastModified>
          </FileShareItem>
        </FileShareItems>
      </FileShareItemResult>
    </FileShareGetItemsResponse>
  </soap:Body>
</soap:Envelope>
```

```
        <Path>string</Path>
        <FileID>string</FileID>
        <Comments>string</Comments>
    </FileShareItem>
</FileShareItems>
</FileShareItemResult>
</FileShareGetItemsResponse>
</soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <FileShareGetItems xmlns="http://api.filesanywhere.com/">
      <FSToken>string</FSToken>
      <FileID>string</FileID>
      <FolderPath>string</FolderPath>
      <IncludeComments>string</IncludeComments>
    </FileShareGetItems>
  </soap12:Body>
</soap12:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <FileShareGetItemsResponse xmlns="http://api.filesanywhere.com/">
      <FileShareItemResult>
        <ErrorMessage>string</ErrorMessage>
        <FileShareItems>
          <FileShareItem>
            <Type>string</Type>
            <Name>string</Name>
            <Size>string</Size>
            <DateLastModified>string</DateLastModified>
```

```

        <Path>string</Path>
        <FileID>string</FileID>
        <Comments>string</Comments>
    </FileShareItem>
    <FileShareItem>
        <Type>string</Type>
        <Name>string</Name>
        <Size>string</Size>
        <DateLastModified>string</DateLastModified>
        <Path>string</Path>
        <FileID>string</FileID>
        <Comments>string</Comments>
    </FileShareItem>
    </FileShareItems>
</FileShareItemResult>
</FileShareGetItemsResponse>
</soap12:Body>
</soap12:Envelope>

```

Sample Visual Basic.Net Code

```

Private Sub FileShareGetItems ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim FileShareItemResult As New WebReference.FileShareHeaderResult
        FileShareItemResult =
            FAWAPI.FileShareGetItems (FSToken, FileID, FolderPath, IncludeComments)
    If FileShareItemResult.ErrorMessage <> "" Then
        MsgBox FileShareItemResult.ErrorMessage
    End If
    FileShareItemResult = Nothing
    FAWAPI = Nothing
End Sub

```

Request Arguments

Name	Data Type	Required	Description
FSToken	String	Yes	ID to be used in subsequent FileShare API calls.
FileID	String	No	ID of the FileShare item. On the initial call, this will be blank. This is NOT a unique key. The unique key for a FileShare item is FileID + FolderPath.
FolderPath	String	No	Path of the FileShare item. On the initial call, this will be blank.
IncludeComments	String	No	Retrieve the comments that may be on a FileShare item. "Y" or "N"

Response

FileShareItemResult

FileShareItemResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the FileShareGetItems method was executed. If the FileShareGetItems call was successful, the ErrorMessage is blank
FileShareItems	Array of type FileShareItem	An array of FileShareItems

A FileShareItem has the following properties:

Property Name	Data Type	Required	Description
Type	String	Yes	Type can either be a 'file' or a 'folder'
Name	String	Yes	Name of file or folder.
Size	String	Yes	Size of file in bytes. Folders will always show 0 bytes.
DateLastModified	String	Yes	Date file or folder was modified. This is displayed in EST (eastern standard time)
Path	String	Yes	Full FilesAnywhere path of the Item in the form \UserVolume\FolderName\ItemName
FileID	String	Yes	ID of the FileShare item. This will be used to identify the file or folder for browsing or downloading.
Comments	String	No	Comments that may be associated with the file or folder. If you set IncludeComments to "N" this will be blank. If you set IncludeComments to "Y" this MAY still be blank. The vast majority of files do NOT have comments associated with them.

FileShareIndexedItemProperties

This method is identical to `IndexedItemProperties`. This is used for FileShare links and uses the `FSToken` instead of the `Token` (ctrl+click title for more information)

Syntax

`FileShareIndexedItemPropertiesResult = FileShareIndexedItemProperties (FSToken, Path)`

FileShareItemImage

This method is identical to `ItemImage`. This is used for FileShare links and uses the `FSToken` instead of the `Token`. One slight difference is that this function returns the same `FilePath` that was passed into it to make Adobe Flash development simpler. (ctrl+click title for more information)

Syntax

`FileShareItemImageResult = FileShareItemImage(FSToken, Type, Path, PageNumber, DesiredImageHeight, DesiredImageWidth, ZoomPercentage)`

FileShareItemThumbnail

This method is identical to `ItemThumbnail`. This is used for FileShare links and uses the `FSToken` instead of the `Token`. One slight difference is that this function returns the same `Path` that was passed into it to make Adobe Flash development simpler. (ctrl+click title for more information)

Syntax

`FileShareItemThumbnailResult = FileShareItemThumbnail(FSToken, Type, Path)`

FileShareRequestDocIndexing

This method is identical to `RequestDocIndexing`. This is used for FileShare links and uses the `FSToken` instead of the `Token` (ctrl+click title for more information)

Syntax

`FileShareRequestDocIndexingResult = FileShareRequestDocIndexing(FSToken, Path)`

GetClientAppInfo

This method returns the latest info (version, download url etc.) about a client application.

Syntax

```
GetClientAppInfoResult = GetClientAppInfo(AppName)
```

Usage

Use the GetClientAppInfo method to get the current info (version, download url etc.) about a client application.

The following parameters are needed to make the GetClientAppInfo call:

- **AppName** (The name of the application).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetClientAppInfo"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetClientAppInfo xmlns="http://api.filesanywhere.com/">
      <AppName>string</AppName>
    </GetClientAppInfo>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetClientAppInfoResponse xmlns="http://api.filesanywhere.com/">
      <GetClientAppInfoResult>
        <ErrorMessage>string</ErrorMessage>
        <Version>string</Version>
        <URL>string</URL>
      </GetClientAppInfoResult>
    </GetClientAppInfoResponse>
  </soap:Body>
</soap:Envelope>
```


SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetClientAppInfo xmlns="http://api.filesanywhere.com/">
      <AppName>string</AppName>
    </GetClientAppInfo>
  </soap12:Body>
</soap12:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetClientAppInfoResponse xmlns="http://api.filesanywhere.com/">
      <GetClientAppInfoResult>
        <ErrorMessage>string</ErrorMessage>
        <Version>string</Version>
        <URL>string</URL>
      </GetClientAppInfoResult>
    </GetClientAppInfoResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub GetClientAppInfo()  
    Dim FAWAPI As New WebReference.FAWAPI  
    Dim GetClientAppInfoResult As New WebReference.  
        GetClientAppInfoResult  
    GetClientAppInfoResult = FAWAPI.GetClientAppInfo(AppName)  
    If GetClientAppInfoResult.ErrorMessage <> "" Then  
        MsgBox GetClientAppInfoResult.ErrorMessage  
    End If  
    GetClientAppInfoResult = Nothing  
    FAWAPI = Nothing  
End Sub
```

Request Arguments

Name	Data Type	Required	Description
AppName	String	Yes	The name of the application.

Response**GetClientAppInfoResult**

GetClientAppInfoResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the GetClientAppInfo method was executed. If the GetClientAppInfo call was successful, the ErrorMessage is blank
Version	String	The current Version Number of the Application
URL	String	The URL from which the Application can be downloaded.

GetDropBoxDefaults

This method returns the preferred FileReceive Link settings.

Syntax

```
GetDropBoxDefaultsResult = GetDropBoxDefaults(Token)
```

Usage

Use the `GetDropBoxDefaults` method to get the preferred FileReceive Link settings.

The following parameters are needed to make the `GetDropBoxDefaults` call:

- `Token` (The Token ID returned by the last `AccountLogin` call).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetDropBoxDefaults"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetDropBoxDefaults xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </GetDropBoxDefaults>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetDropBoxDefaultsResponse xmlns="http://api.filesanywhere.com/">
      <GetDropBoxDefaultsResult>
        <ErrorMessage>string</ErrorMessage>
        <ShareDays>int</ShareDays>
        <FlagLog>string</FlagLog>
        <FlagEmail>string</FlagEmail>
        <FlagEmailFromSystem>string</FlagEmailFromSystem>
        <EmailSubject>string</EmailSubject>
        <EmailBody>string</EmailBody>
        <FlagAttachment>string</FlagAttachment>
        <FlagReadReceipt>string</FlagReadReceipt>
        <FlagOverwrite>string</FlagOverwrite>
        <SubFolderNamePrompt>string</SubFolderNamePrompt>
      </GetDropBoxDefaultsResult>
    </GetDropBoxDefaultsResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetDropBoxDefaults xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </GetDropBoxDefaults>
  </soap12:Body>
</soap12:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetDropBoxDefaultsResponse xmlns="http://api.filesanywhere.com/">
      <GetDropBoxDefaultsResult>
        <ErrorMessage>string</ErrorMessage>
        <ShareDays>int</ShareDays>
        <FlagLog>string</FlagLog>
        <FlagEmail>string</FlagEmail>
        <FlagEmailFromSystem>string</FlagEmailFromSystem>
        <EmailSubject>string</EmailSubject>
        <EmailBody>string</EmailBody>
        <FlagAttachment>string</FlagAttachment>
        <FlagReadReceipt>string</FlagReadReceipt>
        <FlagOverwrite>string</FlagOverwrite>
        <SubFolderNamePrompt>string</SubFolderNamePrompt>
      </GetDropBoxDefaultsResult>
    </GetDropBoxDefaultsResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub GetDropBoxDefaults ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim GetDropBoxDefaultsResult As New WebReference.
    GetDropBoxDefaultsResult
    GetDropBoxDefaultsResult = FAWAPI.GetDropBoxDefaults (Token)
    If GetDropBoxDefaultsResult.ErrorMessage <> "" Then
        MsgBox GetDropBoxDefaultsResult.ErrorMessage
    End If
    GetDropBoxDefaultsResult = Nothing
    FAWAPI = Nothing
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call

Response

GetDropBoxDefaultsResult

GetDropBoxDefaultsResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the GetDropBoxDefaults method was executed. If the GetDropBoxDefaults call was successful, the ErrorMessage is blank.
ShareDays	integer	Number of days the recipient can access the FileReceive Link.
FlagLog	String	A "Y" or "N" Flag indicating whether each file drop should be recorded in the File History Log.
FlagEmail	String	A "Y" or "N" Flag indicating whether the FileReceive Link owner should be notified by email about each file drop.
FlagEmailFromSystem	String	A "Y" or "N" Flag indicating whether the email containing the FileReceive Link should have its "Sent From" email address as the FilesAnywhere System email address (indicated by a "Y") or the FileReceive Link owners email address (indicated by a "N").
EmailSubject	String	Text that appears in the email subject.
EmailBody	String	Message that appears in the email body.

FlagAttachment	String	A “Y” or “N” Flag indicating whether the FileReceive Link is included in the text file which is sent as an attachment in the email.
FlagReadReceipt	String	A “Y” or “N” Flag indicating whether the FileReceive Link owner is notified when the recipient clicks on the FileReceive Link.
FlagOverwrite	String	A “Y” or “N” flag indicating whether file drops can overwrite files of the same name.
SubFolderNamePrompt	String	The name of the folder which will be created for the file drops.

GetELinkDefaults

This method returns the preferred file share settings.

Syntax

GetELinkDefaultsResult = GetELinkDefaults(Token)

Usage

Use the GetELinkDefaults method to get the preferred file share settings.

The following parameters are needed to make the GetELinkDefaults call:

- Token (The Token ID returned by the last AccountLogin call).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetELinkDefaults"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetELinkDefaults xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </GetELinkDefaults>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetELinkDefaultsResponse xmlns="http://api.filesanywhere.com/">
      <GetELinkDefaultsResult>
        <ErrorMessage>string</ErrorMessage>
        <ShareDays>int</ShareDays>
        <FlagLog>string</FlagLog>
        <FlagEmail>string</FlagEmail>
        <FlagMediaView>string</FlagMediaView>
        <DownloadLimit>int</DownloadLimit>
        <FlagViewOnly>string</FlagViewOnly>
        <FlagWatermark>string</FlagWatermark>
        <WatermarkCenter>string</WatermarkCenter>
        <WatermarkBottom>string</WatermarkBottom>
        <FlagEmailFromSystem>string</FlagEmailFromSystem>
        <EmailSubject>string</EmailSubject>
        <EmailBody>string</EmailBody>
        <FlagAttachment>string</FlagAttachment>
        <FlagReadReceipt>string</FlagReadReceipt>
      </GetELinkDefaultsResult>
    </GetELinkDefaultsResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetELinkDefaults xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </GetELinkDefaults>
  </soap12:Body>
</soap12:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetELinkDefaultsResponse xmlns="http://api.filesanywhere.com/">
      <GetELinkDefaultsResult>
        <ErrorMessage>string</ErrorMessage>
        <ShareDays>int</ShareDays>
        <FlagLog>string</FlagLog>
        <FlagEmail>string</FlagEmail>
        <FlagMediaView>string</FlagMediaView>
        <DownloadLimit>int</DownloadLimit>
        <FlagViewOnly>string</FlagViewOnly>
        <FlagWatermark>string</FlagWatermark>
        <WatermarkCenter>string</WatermarkCenter>
        <WatermarkBottom>string</WatermarkBottom>
        <FlagEmailFromSystem>string</FlagEmailFromSystem>
        <EmailSubject>string</EmailSubject>
        <EmailBody>string</EmailBody>
        <FlagAttachment>string</FlagAttachment>
        <FlagReadReceipt>string</FlagReadReceipt>
      </GetELinkDefaultsResult>
    </GetELinkDefaultsResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub GetELinkDefaults ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim GetELinkDefaultsResult As New WebReference.
    GetELinkDefaultsResult
    GetELinkDefaultsResult = FAWAPI.GetELinkDefaults (Token)
    If GetELinkDefaultsResult.ErrorMessage <> "" Then
        MsgBox GetELinkDefaultsResult.ErrorMessage
    End If
    GetELinkDefaultsResult = Nothing
    FAWAPI = Nothing
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call

Response

GetELinkDefaultsResult

GetELinkDefaultsResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the GetELinkDefaults method was executed. If the GetELinkDefaults call was successful, the ErrorMessage is blank
ShareDays	integer	Number of days the recipient can access the FileShare link.
FlagLog	String	A "Y" or "N" Flag indicating whether each file download should be recorded in the File History Log.
FlagEmail	String	A "Y" or "N" Flag indicating whether the FileShare owner should be notified by email about each file downloaded.
FlagMediaView	String	<p>A "M" or "L" or "F" Flag indicating the way in which the items will be displayed.</p> <p>M - MediaFolder View: The recipients will be able to see the items as thumbnails, slideshow, streaming media</p> <p>L - Media List View: Same as MediaFolder view, but the items will be displayed in rows with comments and search.</p> <p>F - File Detail View: The item will be displayed in a plain, row by row view.</p>
DownloadLimit	Integer	The limit on the number of downloads for each item by the recipients. A 0 indicates

		unlimited download.
FlagViewOnly	String	A “Y” or “N” Flag. A “Y” indicates that the recipient can only view the items shared by the user. A “N” allows the recipient to download files too.
FlagWatermark	String	A “Y” or “N” flag. A “Y” indicates that the watermark text specified by the user appears on the images and the documents that is viewed in the online viewer.
WatermarkCenter	String	Watermark text that appears at the center of the item viewed in the online viewer.
WatermarkBottom	String	Watermark text that appears at the bottom of the item viewed in the online viewer.
FlagEmailFromSystem	String	A “Y” or “N” Flag indicating whether the email containing the FileShare link should have its “Sent From” email address as the FilesAnywhere System email address (indicated by a “Y”) or the FileShare owners email address (indicated by a “N”).
EmailSubject	String	Text that appears in the email subject.
EmailBody	String	Message that appears in the email body.
FlagAttachment	String	A “Y” or “N” Flag indicating whether the FileShare link is included in the text file which is sent as an attachment in the email.
FlagReadReceipt	String	A “Y” or “N” Flag indicating whether the FileShare owner is notified when the recipient clicks on the FileShare link.

GetUserParam

This method returns the userparam. A string of alphanumeric characters representing permissions for different FilesAnywhere features for a given user.

Syntax

UserParamResult = GetUserParam(Token)

Usage

Use the GetUserParam method to get the userparam for the token provided.

The following parameters are needed to make the GetUserParam call:

- Token (The Token ID returned by the last AccountLogin call).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetUserParam"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetUserParam xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </GetUserParam>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
```


Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetUserParamResponse xmlns="http://api.filesanywhere.com/">
      <UserParamResult>
        <ErrorMessage>string</ErrorMessage>
        <UserParam>string</UserParam>
      </UserParamResult>
    </GetUserParamResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetUserParam xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </GetUserParam>
  </soap12:Body>
</soap12:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetUserParamResponse xmlns="http://api.filesanywhere.com/">
      <UserParamResult>
        <ErrorMessage>string</ErrorMessage>
        <UserParam>string</UserParam>
```

```
</UserParamResult>  
</GetUserParamResponse>  
</soap12:Body>  
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Function GetUserParam() As String  
    Dim FAWAPI As New WebReference.FAWAPI  
    Dim UserParamResult As New WebReference.UserParamResult  
    UserParamResult = FAWAPI.GetUserParam(Token)  
    If UserParamResult.ErrorMessage <> "" Then  
        MsgBox(UserParamResult.ErrorMessage)  
    End If  
    Return UserParamResult.UserParam  
End Function  
  
Private Sub GetValFromUserParam(ByVal UserParamPos As Integer)  
    Dim userparam As String = GetUserParam()  
    Dim sRtn As String  
    If userparam.Length < UserParamPos Then  
        sRtn = " "  
    Else  
        sRtn = userparam(UserParamPos - 1)  
    End If  
    MsgBox(sRtn)  
End Sub
```

UserParamPos = 13 for webdav enabled

UserParamPos = 14 for ftp enabled

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call

Response

UserParamResult

UserParamResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the GetUserParam method was executed. If the GetUserParam call was successful, the ErrorMessage is blank
UserParam	String	The userparam: A string of alphanumeric characters representing permissions for different FilesAnywhere features for a given user.

GroupShareAdd

This function allows the user to share folder under its account or entire account with other existing FilesAnywhere users.

Syntax

`GroupShareAddResult= GroupShareAdd (Token, UserAccountWithWhomToShareFolder, SharedFolderName, SharedFolderAccessLevel, Notes)`

Usage

Use the GroupShareAdd method to share a personal folder with any other FilesAnywhere account.

The following parameters are needed to make the GroupShareAdd call:

- Token (The Token ID returned by the last AccountLogin call).
- UserAccountWithWhomToShareFolder is the FilesAnywhere Account UserName with whom the user wants to share its personal folder.
- SharedFolderName is the user's personal folder path that the user wants to share with the FilesAnywhere existing user specified in the UserAccountWithWhomToShareFolder field. If the SharedFolderName is not specified, then the users root folder is shared with the new user. The SharedFolderName should not start with "\ " and should not end with "\ ". For Example, it should be "abc" or "abc\xyz" or "abc\xyz\1234".
- SharedFolderAccessLevel is the permission flag set for the SharedFolder. Following are the access levels that can be set for the shared folder:
 - F - Full Permission. The user can upload, delete and download files.
 - A - Master access permission. Recipient user can do all as to Full permission, as well as he can share this folder to other users also.
 - R - Read-only Permission. The user can only view and download files from the shared folder.
 - C - Create and Update Permissions.
 - P - Preview only, User can Preview the file from Shared folder, User cannot upload/download the File into Shared folder.

U – Upload only, User can only upload the files into Shared folder.

- Note is the text that the user wants to attach to the new Group Share folder for future reference

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: 64.156.56.142
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GroupShareAdd"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GroupShareAdd xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>

<UserAccountWithWhomToShareFolder>string</UserAccountWithWhomToShareFolder
>
      <SharedFolderName>string</SharedFolderName>
      <SharedFolderAccessLevel>string</SharedFolderAccessLevel>
      <Notes>string</Notes>
    </GroupShareAdd>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GroupShareAddResponse xmlns="http://api.filesanywhere.com/">
      <GroupShareAddResult>
        <ErrorMessage>string</ErrorMessage>
        <GroupShareCreated>boolean</GroupShareCreated>
      </GroupShareAddResult>
    </GroupShareAddResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: 64.156.56.142
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GroupShareAdd xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>

<UserAccountWithWhomToShareFolder>string</UserAccountWithWhomToShareFolder
>
      <SharedFolderName>string</SharedFolderName>
      <SharedFolderAccessLevel>string</SharedFolderAccessLevel>
      <Notes>string</Notes>
    </GroupShareAdd>
  </soap12:Body>
</soap12:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GroupShareAddResponse xmlns="http://api.filesanywhere.com/">
      <GroupShareAddResult>
        <ErrorMessage>string</ErrorMessage>
        <GroupShareCreated>boolean</GroupShareCreated>
      </GroupShareAddResult>
    </GroupShareAddResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub ShareFolder ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim wsGroupShareAddResult As New WebReference.GroupShareAddResult
    wsGroupShareAddResult=wsFAWAPI.GroupShareAdd(Token,
    UserAccountWithWhomToShareFolder, SharedFolderName,
    SharedFolderAccessLevel, Notes)
    If Not wsGroupShareAddResult.GroupShareCreated Or
        wsGroupShareAddResult.ErrorMessage <> "" Then
        MsgBox wsGroupShareAddResult.ErrorMessage
    End If
    wsGroupShareAddResult = Nothing
    FAWAPI = Nothing
End Sub
```


Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
UserAccountWithWhomToShareFolder	string	Yes	FilesAnywhere Account UserName with whom the user wants to share its personal folder
SharedFolderName	String	No	User's personal folder path that the user wants to share with the FilesAnywhere existing user specified in the UserAccountWithWhomToShareFolder field. If the SharedFolderName is not specified then the users root folder is shared with the new user. The SharedFolderName should not start with "\" and should not end with "\". For Example, it should be "abc" or "abc\xyz" or "abc\xyz\1234"
SharedFolderAccessLevel	String	Yes	<p>Shared Folder permission flag. Following are the access levels that can be set for the shared folder:</p> <p>F - Full Permission. The user can upload, delete and download files.</p> <p>A - Master access permission. Recipient user can do all as to Full permission, as well as he can share this folder to other users also.</p> <p>R - Read-only Permission. The user can only view and download files from the shared folder.</p> <p>C - Create and Update Permissions.</p> <p>P - Preview only, User can Preview the file from Shared folder, User cannot upload/download the File into Shared folder.</p> <p>U - Upload only, User can only upload the files into Shared folder.</p>
Notes	string	No	Notes that the user wants to attach to the new Group Share folder for future

			reference
--	--	--	-----------

Response

GroupShareAdd

GroupShareResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while creating the group share. If the GroupShareAdd call was successful, the ErrorMessage is blank or null.
GroupShareCreated	Boolean	Returns true if the Group Share is Created.

IndexedItemProperties

This function indicates if the referenced file is page-indexed, for viewing in the Online Viewer. If the file is indexed, the function returns a value of **1** along with the number of indexed pages. This function can be used to check the status of a file that has been queued for indexing.

Syntax

IndexedItemPropertiesResult= IndexedItemProperties(Token, Path)

Usage

Use the IndexedItemProperties method to get the properties of the page-indexed file.

The following parameters are needed to make the IndexedItemProperties call:

- Token (The Token ID returned by the last AccountLogin call).
- Path (Name of the item with the FilesAnywhere Path). The Path should be formatted like `\UserVolume\FolderName\ItemName`, where UserVolume is the FilesAnywhere UserName in whose account the item exists. The function returns an error like "invalid file extension." for items of type folder and files without proper extensions.

Note: Items with extensions like jpg, jpeg, gif, png, bmp, and wmf are always considered as an indexed file and page count is always **1**.

IndexedItemProperties function should be used only for specific file types like all the Microsoft Office documents, pdf and tif files. The IndexedItemProperties function returns the item indexing request status and the page count for items with extensions like doc, ppt, xls, pdf, tif, vsd, txt, rtf, wri, pub, mpp, mdi, dot, xlt, wq1, wks, wk3, wk4, slk, prn, dif, csv.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /facorpapi/fawapi.asmx HTTP/1.1
Host: 10.10.10.16
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/IndexedItemProperties"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <IndexedItemProperties xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
    </IndexedItemProperties>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <IndexedItemPropertiesResponse xmlns="http://api.filesanywhere.com/">
      <IndexedItemPropertiesResult>
        <ErrorMessage>string</ErrorMessage>
        <Indexed>short</Indexed>
        <PageNumber>long</PageNumber>
      </IndexedItemPropertiesResult>
    </IndexedItemPropertiesResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /facorpapi/fawapi.asmx HTTP/1.1
Host: 10.10.10.16
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <IndexedItemProperties xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
    </IndexedItemProperties>
  </soap12:Body>
</soap12:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <IndexedItemPropertiesResponse xmlns="http://api.filesanywhere.com/">
      <IndexedItemPropertiesResult>
        <ErrorMessage>string</ErrorMessage>
        <Indexed>short</Indexed>
        <PageNumber>long</PageNumber>
      </IndexedItemPropertiesResult>
    </IndexedItemPropertiesResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub IndexedItemProperties ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim IndexedItemPropertiesResult As New WebReference.
    IndexedItemProperties
    IndexedItemPropertiesResult = FAWAPI.IndexedItemPropertiesResult
    (Token, Path)
    If IndexedItemPropertiesResult.ErrorMessage IsNot Nothing Then
        MsgBox IndexedItemPropertiesResult.ErrorMessage
    End If
    IndexedItemPropertiesResult = Nothing
    FAWAPI = Nothing
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
Path	String	No	Full FilesAnywhere path of the Item in the form \UserVolume\FolderName\ItemName. Please refer to the usage section of this function for the list of the items extension that can be passed to this function.

Response

IndexedItemPropertiesResult

IndexedItemPropertiesResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ItemThumbnail method was executed. If the ItemThumbnail call was successful, the ErrorMessage is blank or null.
Indexed	Short	Returns that status of the file that has queued for indexing. Value 1 indicates that the item is indexed, 0 indicates that the item is being indexed.
PageNumber	Long	If the file is indexed then the number of indexed pages is returned else 1 is returned. If the User doesn't have permission to view the Documents image and the Items extension is a valid document extension then the number of pages of the sample document is returned which are 4.

ItemAdvancedProperties

This method returns the advanced item attributes. This includes Item basic properties, comments, and the Version Control status if Version Control is enabled.

Syntax

ItemAdvPropertiesResult = ItemAdvancedProperties(Token, Type, Path)

Usage

Use the ItemAdvancedProperties method to get the advanced attributes of the item.

The following parameters are needed to make the ItemAdvancedProperties call:

- Token (The Token ID returned by the last AccountLogin call).
- Type is the Item type and can either be a 'file' or a 'folder'.
- Path (Name of the item with the FilesAnywhere Path). The Path should be formatted like \UserVolume\FolderName\ItemName, where UserVolume is the FilesAnywhere UserName in whose account the item exists.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/ItemAdvancedProperties"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemAdvancedProperties xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Type>string</Type>
      <Path>string</Path>
    </ItemAdvancedProperties>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemAdvancedPropertiesResponse xmlns="http://api.filesanywhere.com/">
      <ItemsPropertiesResult>
        <VC_Comments>string</VC_Comments>
        <VC_CheckoutComments>string</VC_CheckoutComments>
        <VC_CheckoutStatusDate>dateTime</VC_CheckoutStatusDate>
        <VC_CheckoutStatusByName>string</VC_CheckoutStatusByName>
      </ItemsPropertiesResult>
    </ItemAdvancedPropertiesResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ItemAdvancedProperties xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Type>string</Type>
      <Path>string</Path>
    </ItemAdvancedProperties>
  </soap12:Body>
</soap12:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ItemAdvancedPropertiesResponse xmlns="http://api.filesanywhere.com/">
      <ItemsPropertiesResult>
        <VC_Comments>string</VC_Comments>
        <VC_CheckoutComments>string</VC_CheckoutComments>
        <VC_CheckoutStatusDate>dateTime</VC_CheckoutStatusDate>
        <VC_CheckoutStatusByName>string</VC_CheckoutStatusByName>
      </ItemsPropertiesResult>
    </ItemAdvancedPropertiesResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub getItemAdvancedProperties ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim ItemAdvancedProperties As New WebReference.
    ItemAdvancedProperties
    ItemAdvProperties Result = FAWAPI.ItemAdvancedProperties (Token,
    ItemType, Path)
    If ItemAdvProperties.ErrorMessage <> "" Then
        MsgBox ItemAdvProperties.ErrorMessage
    End If
    ItemAdvProperties = Nothing
    FAWAPI = Nothing
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
Type	String	Yes	Type can either be a 'file' or a 'folder'
Path	String	Yes	Full FilesAnywhere path of the Item in the form \UserVolume\FolderName\ItemName

Response

ItemAdvPropertiesResult

ItemAdvPropertiesResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ItemAdvancedProperties method was executed. If the ItemAdvancedProperties call was successful, the ErrorMessage is blank or null.
Size	String	If the item type is file, size will contain the file size in MB and bytes. If the type is folder the size is null
Count	string	File and folder count (subfolders are not included in the count), if the type is folder otherwise null.
DateCreated	string	Date and time when the item was created.
DateLastModified	string	Date and time when the item was last modified.
VC_Comments	String	Folder version control comments, if version control is enabled otherwise blank
VC_CheckoutComments	String	File checkout comments, if the type is file and it is checked out. If the type is folder then blank.
VC_CheckoutStatusDate	string	File checkout status date, if the type is file and it is checked out. If the type is folder then blank.
VC_CheckoutStatusByName	String	User Name who checked the file, if the type is file and it is checked out. If the type is folder then blank.

ItemCommentAdd

This method performs two operations based on ParentID parameter.

- a) Add comment of a file/folder.
- b) Add Reply of an existing comment for a file/folder.

Syntax

```
ItemCommentAddResult =  
ItemCommentAdd(Token,Path,Comment,FullName,ParentId)
```

Usage

Use the ItemCommentAdd method to add the comment of an item.

The following parameters are needed to make the ItemCommentAdd call:

- Token (The Token ID returned by the last AccountLogin call).
- Path (Name of the item with the FilesAnywhere Path). The Path should be formatted, for file `\UserVolume\FolderName\FileName` and for folder path should be `\UserVolume\FolderName\`, where UserVolume is the FilesAnywhere UserName in whose account the item exists.
- Comment (Comment of a file/folder.)
- FullName (Full name of user who is adding comment of an item, If user has not passed fullname then system will automatically pick up the full name of user from database.)
- ParentId (If user is adding a new comment of an Item then this ParentId should be 0 else if user is replying of any existing comment then this ParentId should be the id of an existing comment.)

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: 64.156.56.142
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/ItemCommentAdd"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemCommentAdd xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
      <Comment>string</Comment>
      <FullName>string</FullName>
      <ParentId>int</ParentId>
    </ItemCommentAdd>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemCommentAddResponse xmlns="http://api.filesanywhere.com/">
      <ItemCommentAddResult>
        <ErrorMessage>string</ErrorMessage>
        <CommentAdded>boolean</CommentAdded>
      </ItemCommentAddResult>
    </ItemCommentAddResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: 64.156.56.142
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ItemCommentAdd xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
      <Comment>string</Comment>
      <FullName>string</FullName>
      <ParentId>int</ParentId>
    </ItemCommentAdd>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ItemCommentAddResponse xmlns="http://api.filesanywhere.com/">
      <ItemCommentAddResult>
        <ErrorMessage>string</ErrorMessage>
        <CommentAdded>boolean</CommentAdded>
      </ItemCommentAddResult>
    </ItemCommentAddResponse>
  </soap12:Body>
</soap12:Envelope>
```


Sample Visual Basic.Net Code

```

Private Sub ItemCommentAdd()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim ItemCommentAddResult As New WebReference.ItemCommentAddResult
    ItemCommentAddResult= FAWAPI.ItemCommentAdd(Token,
Path,Comment,fullName,ParentId)
    If ItemCommentAddResult.ErrorMessage IsNot Nothing Then
        MsgBox ItemCommentAddResult.ErrorMessage
        Exit Sub
    End If
    If ItemCommentAddResult.CommentAdded IsNot Nothing Then
        MsgBox ItemCommentAddResult.CommentAdded
    End If
    ItemCommentAddResult = Nothing
    FAWAPI = Nothing
End Sub

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
Path	String	Yes	Full FilesAnywhere path of the file/folder in the form \UserVolume\FolderName\FileName or \UserVolume\FolderName\
Comment	String	Yes	Comment of an Item
FullName	String	No	Full name of user who is adding the comment of an Item. If user has passed null then system will pick up the full name of User from database.
ParentId	String	No	If user is adding a new comment of an Item then this ParentId should be 0 else if user is replying of any existing comment then this ParentId should be the id of existing comment.)

Response**ItemCommentAddResult**

ItemCommentAddResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ItemCommentAdd method was executed. If the ItemCommentAdd call was successful, the ErrorMessage is blank or null.
CommentAdded	Boolean	If comment is added successfully then this will return True else false.

ItemCommentRemove

This method removes comment of a file/folder.

Syntax

ItemCommentRemoveResult = ItemCommentRemove(Token, CommentId)

Usage

Use the ItemCommentRemove method to remove the comment of a file/folder.

The following parameters are needed to make the ItemCommentRemove call:

- Token (The Token ID returned by the last AccountLogin call).
- CommentId (CommentId of a comment added on a file/folder)

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: 64.156.56.142
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/ItemCommentRemove"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemCommentRemove xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <CommentId>int</CommentId>
    </ItemCommentRemove>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemCommentRemoveResponse xmlns="http://api.filesanywhere.com/">
      <ItemCommentRemoveResult>
        <ErrorMessage>string</ErrorMessage>
        <CommentRemoved>boolean</CommentRemoved>
      </ItemCommentRemoveResult>
    </ItemCommentRemoveResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: 64.156.56.142
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ItemCommentRemove xmlns="http://api.filesanywhere.com/">
```

```
<Token>string</Token>
  <CommentId>int</CommentId>
</ItemCommentRemove>
</soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ItemCommentRemoveResponse xmlns="http://api.filesanywhere.com/">
      <ItemCommentRemoveResult>
        <ErrorMessage>string</ErrorMessage>
        <CommentRemoved>boolean</CommentRemoved>
      </ItemCommentRemoveResult>
    </ItemCommentRemoveResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub ItemCommentRemove()  
    Dim FAWAPI As New WebReference.FAWAPI  
    Dim ItemCommentRemoveResult As New  
WebReference.ItemCommentRemoveResult  
    ItemCommentRemoveResult= FAWAPI.ItemCommentRemove(Token, CommentId)  
    If ItemCommentRemoveResult.ErrorMessage IsNot Nothing Then  
        MsgBox ItemCommentRemoveResult.ErrorMessage  
        Exit Sub  
    End If  
    If ItemCommentRemoveResult.CommentRemoved IsNot Nothing Then  
        MsgBox ItemCommentRemoveResult.CommentRemoved  
    End If  
    ItemCommentRemoveResult = Nothing  
    FAWAPI = Nothing  
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
CommentId	Integer	Yes	CommentId of a comment added on a file/folder.

Response**ItemCommentRemoveResult**

ItemCommentRemoveResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ItemCommentRemove method was executed. If the ItemCommentRemove call was successful, the ErrorMessage is blank or null.
CommentRemoved	Boolean	If comment is Removed successfully then this will return True else false.

ItemCommentUpdate

This method Update the comment of a file/folder.

Syntax

```
ItemCommentUpdateResult = ItemCommentUpdate(Token,ItemComment,CommentId)
```

Usage

Use the ItemCommentUpdate method to update the comment of a file/folder.

The following parameters are needed to make the ItemCommentUpdate call:

- Token (The Token ID returned by the last AccountLogin call).
- ItemComment (The updated comment of a file/folder)
- CommentId (CommentId of a comment added on file/folder)

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: 64.156.56.142
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/ItemCommentUpdate"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemCommentUpdate xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ItemComment>string</ItemComment>
      <CommentId>int</CommentId>
    </ItemCommentUpdate>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemCommentUpdateResponse xmlns="http://api.filesanywhere.com/">
      <ItemCommentUpdateResult>
        <ErrorMessage>string</ErrorMessage>
        <CommentUpdated>boolean</CommentUpdated>
      </ItemCommentUpdateResult>
    </ItemCommentUpdateResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: 64.156.56.142
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ItemCommentUpdate xmlns="http://api.filesanywhere.com/">
```

```
<Token>string</Token>
<ItemComment>string</ItemComment>
<CommentId>int</CommentId>
</ItemCommentUpdate>
</soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ItemCommentUpdateResponse xmlns="http://api.filesanywhere.com/">
      <ItemCommentUpdateResult>
        <ErrorMessage>string</ErrorMessage>
        <CommentUpdated>boolean</CommentUpdated>
      </ItemCommentUpdateResult>
    </ItemCommentUpdateResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub ItemCommentUpdate()  
    Dim FAWAPI As New WebReference.FAWAPI  
    Dim ItemCommentUpdateResult As New  
        WebReference.ItemCommentUpdateResult  
    ItemCommentUpdateResult=  
        FAWAPI.ItemCommentUpdate(Token, ItemComment, CommentId)  
    If ItemCommentUpdateResult.ErrorMessage IsNot Nothing Then  
        MsgBox ItemCommentUpdateResult.ErrorMessage  
        Exit Sub  
    End If  
    If ItemCommentUpdateResult.CommentUpdated IsNot Nothing Then  
        MsgBox ItemCommentUpdateResult.CommentUpdated  
    End If  
    ItemCommentUpdateResult = Nothing  
    FAWAPI = Nothing  
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
ItemComment	String	Yes	Updated comment of an Item
CommentId	Integer	Yes	CommentId of a comment added on a file/folder.

Response**ItemCommentUpdateResult**

ItemCommentUpdateResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ItemCommentupdate method was executed. If the ItemCommentUpdate call was successful, the ErrorMessage is blank or null.
CommentUpdated	Boolean	If comment is Updated successfully then this will return True else false.

ItemCommentsList

This method returns the comments list of a file/folder.

Syntax

```
ItemCommentsListResult = ItemCommentsList(Token,ItemPath)
```

Usage

Use the ItemCommentsList method to get the comments list of an item.

The following parameters are needed to make the ItemCommentsList call:

- Token (The Token ID returned by the last AccountLogin call).
- ItemPath (Name of the file/folder with the FilesAnywhere Path). The Path should be formatted, for file `\UserVolume\FolderName\FileName` and for folder path should be `\UserVolume\FolderName\`, where UserVolume is the FilesAnywhere Username in whose account the item exists.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/ItemCommentsList"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemCommentsList xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ItemPath>string</ItemPath>
    </ItemCommentsList>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemCommentsListResponse xmlns="http://api.filesanywhere.com/">
      <ItemCommentsListResult>
        <ErrorMessage>string</ErrorMessage>
        <CommentList>xmlxml</CommentList>
      </ItemCommentsListResult>
    </ItemCommentsListResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
```

```

    <ItemCommentsList xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ItemPath>string</ItemPath>
    </ItemCommentsList>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ItemCommentsListResponse xmlns="http://api.filesanywhere.com/">
      <ItemCommentsListResult>
        <ErrorMessage>string</ErrorMessage>
        <CommentList>xmlxml</CommentList>
      </ItemCommentsListResult>
    </ItemCommentsListResponse>
  </soap12:Body>
</soap12:Envelope>

```

Sample Visual Basic.Net Code

```

Private Sub ItemCommentsList ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim ItemCommentsListResult As New
WebReference.ItemCommentsListResult
ItemCommentsListResult= FAWAPI.ItemCommentsList(Token,ItemPath)
    If ItemCommentsListResult.ErrorMessage IsNot Nothing Then
        MsgBox ItemCommentsListResult.ErrorMessage
        Exit Sub
    End If
    If ItemCommentsListResult.CommentList IsNot Nothing Then
        Dim dtTemp As New System.Data.DataTable ()
        dtTemp = ItemCommentsListResult.CommentList
    End If
    ItemCommentsListResult = Nothing
    FAWAPI = Nothing
End Sub

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last

			AccountLogin call
ItemPath	String	Yes	Full FilesAnywhere path of the file/folder in the form <code>\UserVolume\FolderName\FileName</code> or <code>\UserVolume\FolderName\</code>

Response

ItemCommentsListResultItemCommentsListResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ItemCommentsList method was executed. If the ItemCommentsList call was successful, the ErrorMessage is blank or null.
CommentList	Return the XML string.	List of comments whose structure is shown as below.

A CommentList has the following properties:

Property Name	Data Type	Description
ID	Int	Id of Comment
Comment	String	Comment added by user
CreatedBy	String	User name who has created this comment
CreatedOn	DateTime	Date on which comment created
ParentId	Int	Value in the parent id represent comment id of existing comment (reply of an existing comment) else it will be 0 means it is comment of a document

ItemDescription

This method returns comment of input file/folder.

Syntax

ItemImageResult = ItemDescription(Token,Path)

Usage

Use the ItemDescription method to get the description of item.

The following parameters are needed to make the ItemDescription call:

- Token (The Token ID returned by the last AccountLogin call).
- Path (Name of the item with the FilesAnywhere Path). The Path should be formatted like \UserVolume\FolderName\ItemName, where UserVolume is the FilesAnywhere UserName in whose account the item exists.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/ItemDescription"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemDescription xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
    </ItemDescription>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemDescriptionResponse xmlns="http://api.filesanywhere.com/">
      <ItemDescriptionResult>
        <ErrorMessage>string</ErrorMessage>
        <Comment>string</Comment>
      </ItemDescriptionResult>
    </ItemDescriptionResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ItemDescription xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
```

```

        <Path>string</Path>
    </ItemDescription>
</soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ItemDescriptionResponse xmlns="http://api.filesanywhere.com/">
      <ItemDescriptionResult>
        <ErrorMessage>string</ErrorMessage>
        <Comment>string</Comment>
      </ItemDescriptionResult>
    </ItemDescriptionResponse>
  </soap12:Body>
</soap12:Envelope>

```

Sample Visual Basic.Net Code

```

Private Sub getItemDescription ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim ItemDescriptionResult As New
WebReference.ItemDescriptionResult
    ItemDescriptionResult= FAWAPI.ItemDescription(Token, Path)
    If ItemDescriptionResult.ErrorMessage IsNot Nothing Then
        MsgBox ItemDescriptionResult.ErrorMessage
        Exit Sub
    End If
    If ItemDescriptionResult.Comment IsNot Nothing Then
        MsgBox ItemDescriptionResult.Comment
    End If
    ItemDescriptionResult = Nothing
    FAWAPI = Nothing
End Sub

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
Path	String	No	Full FilesAnywhere path of the Item in the form \UserVolume\FolderName\ItemName or

			\UserVolume\FolderName\
--	--	--	-------------------------

Response

ItemDescriptionResult

ItemDescriptionResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ItemComment method was executed. If the ItemDescription call was successful, the ErrorMessage is blank or null.
Comment	String	Comment of the item as string.

ItemDescriptionAdd

This method adds description of a file/folder.

Syntax

```
ItemDescriptionAddResult = ItemDescriptionAdd(Token,Path,Description)
```

Usage

Use the ItemDescriptionAdd method to add the description of an item.

The following parameters are needed to make the ItemDescriptionAdd call:

- Token (The Token ID returned by the last AccountLogin call).
- Path (Name of the item with the FilesAnywhere Path). The Path should be formatted like, for file: \UserVolume\FolderName\FileName and for folder: \UserVolume\FolderName\, where UserVolume is the FilesAnywhere UserName in whose account the file/folder exists.
- Description (Description of a file/folder)

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/ItemDescriptionAdd"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemDescriptionAdd xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
      <Description>string</Description>
    </ItemDescriptionAdd>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemDescriptionAddResponse xmlns="http://api.filesanywhere.com/">
      <ItemDescriptionAddResult>
        <ErrorMessage>string</ErrorMessage>
        <DescriptionAdded>boolean</DescriptionAdded>
      </ItemDescriptionAddResult>
    </ItemDescriptionAddResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
```

```

<soap12:Body>
  <ItemDescriptionAdd xmlns="http://api.filesanywhere.com/">
    <Token>string</Token>
    <Path>string</Path>
    <Description>string</Description>
  </ItemDescriptionAdd>
</soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ItemDescriptionAddResponse xmlns="http://api.filesanywhere.com/">
      <ItemDescriptionAddResult>
        <ErrorMessage>string</ErrorMessage>
        <DescriptionAdded>boolean</DescriptionAdded>
      </ItemDescriptionAddResult>
    </ItemDescriptionAddResponse>
  </soap12:Body>
</soap12:Envelope>

```

Sample Visual Basic.Net Code

```

Private Sub ItemDescriptionAdd ()
  Dim FAWAPI As New WebReference.FAWAPI
  Dim ItemDescriptionAddResult As New
  WebReference.ItemDescriptionAddResult
  ItemDescriptionAddResult=
  FAWAPI.ItemDescriptionAdd(Token, Path, Description)
  If ItemDescriptionAddResult.ErrorMessage IsNot Nothing Then
    MsgBox ItemDescriptionAddResult.ErrorMessage
    Exit Sub
  End If
  If ItemDescriptionAddResult.Comment IsNot Nothing Then
    MsgBox ItemDescriptionAddResult.DescriptionAdded
  End If
  ItemDescriptionAddResult = Nothing
  FAWAPI = Nothing
End Sub

```

Request Arguments

Name	Data Type	Required	Description
------	-----------	----------	-------------

Token	String	Yes	The Token ID returned back in the last AccountLogin call
Path	String	Yes	Full FilesAnywhere path of the Item in the form \UserVolume\FolderName\FileName or \UserVolume\FolderName\
Description	String	Yes	Description of a file/folder.

Response

ItemDescriptionAddResult

ItemDescriptionAddResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ItemDescriptionAdd method was executed. If the ItemDescriptionAdd call was successful, the ErrorMessage is blank or null.
DescriptionAdded	Boolean	If Description of a file/folder is added successfully then this parameter will return True else False.

ItemImage

This method returns an image of the referenced page within the referenced file. It only returns an image if the file is done indexing.

Syntax

`ItemImageResult = ItemImage(Token, Type, Path, PageNumber, DesiredImageHeight, DesiredImageWidth, ZoomPercentage)`

Usage

Use the `ItemImage` method to get the image of the item.

The following parameters are needed to make the `ItemImage` call:

- `Token` (The Token ID returned by the last `AccountLogin` call).
- `Type` is the Item type and can either be a 'file' or a 'folder'.
- `Path` (Name of the item with the FilesAnywhere Path). The Path should be formatted like `\UserVolume\FolderName\ItemName`, where `UserVolume` is the FilesAnywhere Username in whose account the item exists.
- `PageNumber`. The Item Page whose image needs to be returned. The Page number passed should always be equal to or less than the item's total page count. If Page number passed is greater than the actual page count then the image of the last page is returned and if the page number passed is 0 or less than 0 then the first page of the indexed item is returned.
- `DesiredImageHeight` is the height of the items image that the user wants to set.
- `DesiredImageWidth` is the width of the items image that the user wants to set.
- `ZoomPercent` is the percentage of the original image size that is returned. `DesiredImageHeight` and `DesiredImageWidth` are ignored if the `ZoomPercentage` passed is not equal to zero. If the zoom percentage is set to 100 or if it is set to 0 and the `DesiredImageHeight` and `DesiredImageWidth` are set to a value greater than the actual Item height and width then the original image is returned. To zoom in or zoom out of an item image, pass the `DesiredImageHeight`, `DesiredImageWidth` and `zoomPercentage` as -1 or pass `DesiredImageHeight` and `DesiredImageWidth` as 0 and `ZoomPercentage` greater than 0. The image size returned for some item types like Microsoft Office files is very large. To scale only the width of such item images so that it fits the screen

width, pass the screen width as the `DesiredImageWidth` and `ZoomPercentage` as `-2`.

Note: Paging works only for specific file types like all the Microsoft Office documents, pdf, tif files. The paging works for items with extensions like doc, ppt, xls, pdf, tif, vsd, txt, rtf, wri, pub, mpp, mdi, dot, xlt, wq1, wks, wk3, wk4, slk, prn, dif, csv. For items with extensions like jpg, jpeg, gif, png, bmp and wmf a valid image of the first page is returned.

If the item's extension is a valid document extension as listed above and if the user doesn't have permission to view the document image the images of the sample documents are returned.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

POST /FACORPAPI/FAWAPI.asmx HTTP/1.1

Host: localhost

Content-Type: text/xml; charset=utf-8

Content-Length: length

SOAPAction: "http://api.filesanywhere.com/ItemImage"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemImage xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Type>string</Type>
      <Path>string</Path>
      <PageNumber>long</PageNumber>
      <DesiredImageHeight>int</DesiredImageHeight>
      <DesiredImageWidth>int</DesiredImageWidth>
      <ZoomPercent>int</ZoomPercent>
    </ItemImage>
  </soap:Body>
</soap:Envelope>
```

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemImageResponse xmlns="http://api.filesanywhere.com/">
      <ItemImageResult>
        <ErrorMessage>string</ErrorMessage>
        <ImageStream>base64Binary</ImageStream>
      </ItemImageResult>
    </ItemImageResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ItemImage xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Type>string</Type>
      <Path>string</Path>
      <PageNumber>long</PageNumber>
      <DesiredImageHeight>int</DesiredImageHeight>
      <DesiredImageWidth>int</DesiredImageWidth>
      <ZoomPercent>int</ZoomPercent>
    </ItemImage>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ItemImageResponse xmlns="http://api.filesanywhere.com/">
      <ItemImageResult>
        <ErrorMessage>string</ErrorMessage>
        <ImageStream>base64Binary</ImageStream>
      </ItemImageResult>
    </ItemImageResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```

Private Sub getItemImage ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim ItemImageResult As New WebReference.ItemImageResult
    ItemImageResult = FAWAPI.ItemImageResult (Token, ItemType, Path,
    PageNumber, DesiredImageheight, DesiredImageWidth, ZoomPercentage)
    If ItemImageResult.ErrorMessage IsNot Nothing Then
        MsgBox ItemImageResult.ErrorMessage
    End If
    ItemImageResult = Nothing
    FAWAPI = Nothing
End Sub

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
Type	String	Yes	Type can either be a 'file' or a 'folder'
Path	String	No	Full FilesAnywhere path of the Item in the form \UserVolume\FolderName\ItemName. If the path is blank and the type is folder then the folder image is returned else if the type is file the path is required.
PageNumber	long	Yes	For some file types with extensions like jpg, jpeg, gif, png, bmp, wmf the image returned be always for the first page. For items like Microsoft office files or pdf or tif files, the image returned will be of the specified page number.
DesiredImageHeight	integer	Yes	Image returned will have height as passed in DesiredImageHeight
DesiredImageWidth	integer	Yes	Image returned will have width as passed in DesiredImageWidth
ZoomPercentage	Integer	Yes	The image returned can be zoomed in or out by passing value in the ZoomPercentage. For all other possible combinations please refer to the usage section of this function.

Response

ItemImageResult

ItemImageResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ItemImage method was executed. If the ItemImage call was successful, the ErrorMessage is blank or null.
ImageStream	Byte Array	ItemImage image of the item as byte array.

ItemProperties

This method returns the basic item attributes. This includes: the complete folder path, the file size, file and folder count under the folder, create datetime and last updated date. The Date that is returned is always in Eastern Time.

Syntax

ItemPropertiesResult = ItemProperties(Token, Type, Path)

Usage

Use the ItemProperties method to get the basic attributes of the item.

The following parameters are needed to make the ItemProperties call:

- Token (The Token ID returned by the last AccountLogin call).
- Type is the Item type and can either be a 'file' or a 'folder'.
- Path (Name of the item with the FilesAnywhere Path). The Path should be formatted like \UserVolume\FolderName\ItemName, where UserVolume is the FilesAnywhere UserName in whose account the item exists.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

POST /FACORPAPI/FAWAPI.asmx HTTP/1.1

Host: localhost

Content-Type: text/xml; charset=utf-8

Content-Length: length

SOAPAction: "http://api.filesanywhere.com/ItemProperties"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemProperties xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Type>string</Type>
      <Path>string</Path>
    </ItemProperties>
  </soap:Body>
</soap:Envelope>
```

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemPropertiesResponse xmlns="http://api.filesanywhere.com/">
      <ItemsPropertiesResult>
        <ErrorMessage>string</ErrorMessage>
        <Size>string</Size>
        <Count>string</Count>
        <DateCreated>dateTime</DateCreated>
        <DateLastModified>dateTime</DateLastModified>
      </ItemsPropertiesResult>
    </ItemPropertiesResponse>
  </soap:Body>
</soap:Envelope>
```


SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

POST /FACORPAPI/FAWAPI.asmx HTTP/1.1

Host: localhost

Content-Type: text/xml; charset=utf-8

Content-Length: length

SOAPAction: "http://api.filesanywhere.com/ItemProperties"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemProperties xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Type>string</Type>
      <Path>string</Path>
    </ItemProperties>
  </soap:Body>
</soap:Envelope>
```

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemPropertiesResponse xmlns="http://api.filesanywhere.com/">
      <ItemsPropertiesResult>
        <ErrorMessage>string</ErrorMessage>
        <Size>string</Size>
        <Count>string</Count>
        <DateCreated>dateTime</DateCreated>
        <DateLastModified>dateTime</DateLastModified>
      </ItemsPropertiesResult>
    </ItemPropertiesResponse>
  </soap:Body>
</soap:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub getItemProperties ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim ItemPropertiesResult As New WebReference.ItemPropertiesResult
    ItemPropertiesResult = FAWAPI.DeleteItems(Token, ItemType, Path)
    If ItemPropertiesResult.ErrorMessage <> "" Then
        MsgBox ItemPropertiesResult.ErrorMessage
    End If
    ItemPropertiesResult = Nothing
    FAWAPI = Nothing
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
Type	String	Yes	Type can either be a 'file' or a 'folder'
Path	String	Yes	Full FilesAnywhere path of the Item in the form \UserVolume\FolderName\ItemName

Response

ItemPropertiesResult

ItemPropertiesResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ItemProperties method was executed. If the ItemProperties call was successful, the ErrorMessage is blank or null.
Size	String	If the item type is file, size will contain the file size in MB and bytes. If the type is folder the Size is null
Count	string	File and folder count (subfolders are not included in the count), if the type is folder otherwise null.
DateCreated	string	Date and time in Eastern time when the item was created.
DateLastModified	string	Date and time in Eastern time when the item was last modified.

ItemThumbnail

This method returns a thumbnail for the item. For image files and documents indexed for the Online Viewer, the thumbnail will be a small picture of the first page of the file. For folders, a folder icon is returned.

Syntax

ItemThumbnailResult = ItemThumbnail(Token, Type, Path)

Usage

Use the ItemThumbnail method to get the thumbnail image of the item.

The following parameters are needed to make the ItemThumbnail call:

- Token (The Token ID returned by the last AccountLogin call).
- Type is the Item type and can either be a 'file' or a 'folder'.
- Path (Name of the item with the FilesAnywhere Path). The Path should be formatted like \UserVolume\FolderName\ItemName, where UserVolume is the FilesAnywhere UserName in whose account the item exists.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

POST /FACORPAPI/FAWAPI.asmx HTTP/1.1

Host: localhost

Content-Type: text/xml; charset=utf-8

Content-Length: length

SOAPAction: "http://api.filesanywhere.com/ItemThumbnail"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemThumbnail xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Type>string</Type>
      <Path>string</Path>
    </ItemThumbnail>
  </soap:Body>
</soap:Envelope>
```

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemThumbnailResponse xmlns="http://api.filesanywhere.com/">
      <ItemThumbnailResult>
        <ErrorMessage>string</ErrorMessage>
        <ImageStream>base64Binary</ImageStream>
      </ItemThumbnailResult>
    </ItemThumbnailResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

POST /FACORPAPI/FAWAPI.asmx HTTP/1.1

Host: localhost

Content-Type: application/soap+xml; charset=utf-8

Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ItemThumbnail xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Type>string</Type>
      <Path>string</Path>
    </ItemThumbnail>
  </soap12:Body>
</soap12:Envelope>
```

HTTP/1.1 200 OK

Content-Type: application/soap+xml; charset=utf-8

Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ItemThumbnailResponse xmlns="http://api.filesanywhere.com/">
      <ItemThumbnailResult>
        <ErrorMessage>string</ErrorMessage>
        <ImageStream>base64Binary</ImageStream>
      </ItemThumbnailResult>
    </ItemThumbnailResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub getItemThumbnail ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim ItemThumbnailResult As New WebReference.ItemThumbnailResult
    ItemThumbnailResult = FAWAPI.ItemThumbnail(Token, ItemType, Path)
    If ItemThumbnailResult.ErrorMessage IsNot Nothing Then
        MsgBox ItemThumbnailResult.ErrorMessage
    End If
    ItemThumbnailResult = Nothing
    FAWAPI = Nothing
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
Type	String	Yes	Type can either be a 'file' or a 'folder'
Path	String	Yes	Full FilesAnywhere path of the Item in the form \UserVolume\FolderName\ItemName

Logout

This method logs you out from your FilesAnywhere account for the current API session.

Syntax

LogoutResult = Logout(Token)

Usage

Use the Logout method to end your current FilesAnywhere API session. After you call this method, the Token ID received in the last AccountLogin call cannot be used for any subsequent API calls.

The following parameters are needed to make the Logout call:

- Token (The Token ID returned by the last AccountLogin call).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/Logout"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Logout xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </Logout>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <LogoutResponse xmlns="http://api.filesanywhere.com/">
      <LogoutResult>
        <ErrorMessage>string</ErrorMessage>
        <LoggedOut>boolean</LoggedOut>
      </LogoutResult>
    </LogoutResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

POST /fawapi.asmx HTTP/1.1

Host: api.filesanywhere.com

Content-Type: application/soap+xml; charset=utf-8

Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <Logout xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </Logout>
  </soap12:Body>
</soap12:Envelope>
```

HTTP/1.1 200 OK

Content-Type: application/soap+xml; charset=utf-8

Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <LogoutResponse xmlns="http://api.filesanywhere.com/">
      <LogoutResult>
        <ErrorMessage>string</ErrorMessage>
        <LoggedOut>boolean</LoggedOut>
      </LogoutResult>
    </LogoutResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub Logout()  
    Dim FAWAPI As New WebReference.FAWAPI  
    Dim LogoutResult As New WebReference.LogoutResult  
    LogoutResult = FAWAPI.Logout(Token)  
    If not LogoutResult.LoggedOut Or LogoutResult.ErrorMessage <> ""  
    Then  
        MsgBox LogoutResult.ErrorMessage  
    End If  
    LogoutResult = Nothing  
    FAWAPI = Nothing  
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call

Response

LogoutResult

LogoutResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered during the Logout method was executed. If the Logout call was successful, the ErrorMessage is blank
LoggedOut	Boolean	True or False indicating whether the Logout call was a success or a failure

Paste

This method pastes items from the clipboard into a specified folder. Items must exist in the clipboard from a previous `CopypItems` or `CutItems` call. A successful call also empties the clipboard.

Syntax

`PasteResult = Paste(Token, Path)`

Usage

Use the `Paste` method after a `CopypItems` or `CutItems` call to paste the items in the clipboard into a specified folder.

The following parameters are needed to make the `Paste` call:

- `Token` (The `Token ID` returned by the last `AccountLogin` call).
- `Path` (Full `FilesAnywhere Path`). This is the `FilesAnywhere Path` of the folder where the clipboard items will be pasted. The `Path` should be formatted like `\UserVolume\FolderName`, where `UserVolume` is the `FilesAnywhere UserName` in whose account the items will be pasted and `FolderName` is the folder or subfolder where the items will be pasted.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/Paste"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Paste xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
    </Paste>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <PasteResponse xmlns="http://api.filesanywhere.com/">
      <PasteResult>
        <ErrorMessage>string</ErrorMessage>
        <Pasted>boolean</Pasted>
      </PasteResult>
    </PasteResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <Paste xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
    </Paste>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <PasteResponse xmlns="http://api.filesanywhere.com/">
      <PasteResult>
        <ErrorMessage>string</ErrorMessage>
        <Pasted>boolean</Pasted>
      </PasteResult>
    </PasteResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub Paste()  
    Dim FAWAPI As New WebReference.FAWAPI  
    Dim PasteResult As New WebReference.PasteResult  
    PasteResult = FAWAPI.Paste(Token, Path)  
    If not PasteResult.Pasted Or PasteResult.ErrorMessage <> "" Then  
        MsgBox PasteResult.ErrorMessage  
    End If  
    PasteResult = Nothing  
    FAWAPI = Nothing  
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
Path	String	Yes	Full FilesAnywhere path of the folder or subfolder in the form \UserVolume\FolderName

Response

PasteResult

PasteResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the Paste method was executed. If the Paste call was successful, the ErrorMessage is blank
Pasted	Boolean	True or False indicating whether the Paste call was a success or a failure

Renameltem

This method renames an item.

Syntax

RenameltemResult = Renameltem(Token, Path, Type, NewName)

Usage

Use the Renameltem method to rename a file or a folder.

The following parameters are needed to make the Renameltem call:

- Token (The Token ID returned by the last AccountLogin call).
- Path (Full FilesAnywhere Path). This is the FilesAnywhere Path of the item to be renamed. The Path should be formatted like \UserVolume\FolderName\ItemName, where UserVolume is the FilesAnywhere UserName in whose account the item exists.
- Type (Type of the item to be renamed). The Type can be a 'file' or a 'folder'.
- NewName (The new name for the file or the folder).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/RenameItem"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <RenameItem xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
      <Type>string</Type>
      <NewName>string</NewName>
    </RenameItem>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <RenameItemResponse xmlns="http://api.filesanywhere.com/">
      <RenameItemResult>
        <ErrorMessage>string</ErrorMessage>
        <Renamed>boolean</Renamed>
      </RenameItemResult>
    </RenameItemResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <RenameItem xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
      <Type>string</Type>
      <NewName>string</NewName>
    </RenameItem>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <RenameItemResponse xmlns="http://api.filesanywhere.com/">
      <RenameItemResult>
        <ErrorMessage>string</ErrorMessage>
        <Renamed>boolean</Renamed>
      </RenameItemResult>
    </RenameItemResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub RenameItem()  
    Dim FAWAPI As New WebReference.FAWAPI  
    Dim RenameItemResult As New WebReference.RenameItemResult  
    RenameItemResult = FAWAPI.RenameItem(Token, Path, Type, NewName)  
    If not RenameItemResult.renamed Or RenameItemResult.ErrorMessage  
<> "" Then  
        MsgBox RenameItemResult.ErrorMessage  
    End If  
    RenameItemResult = Nothing  
    FAWAPI = Nothing  
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
Path	String	Yes	Full FilesAnywhere path of the Item in the form \UserVolume\FolderName\ItemName
Type	String	Yes	Type can either be a 'file' or a 'folder'
NewName	String	Yes	The new name for the file or folder in question

Response

RenameItemResult

RenameltemResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the Renameltem method was executed. If the Renameltem call was successful, the ErrorMessage is blank
Renamed	Boolean	True or False indicating whether the Renameltem call was a success or a failure

RequestDocIndexing

This function sends the system a request to index the referenced file. The system processes these requests in a queue. Once the document pages are indexed for viewing, the `ItemThumbnailImage` or `ItemImage` function may be called to return the page image(s) in the file.

Syntax

`IndexedItemPropertiesResult= IndexedItemProperties(Token, Path)`

Usage

Use the `RequestDocIndexing` method to send a request to index the referenced file.

The following parameters are needed to make the `RequestDocIndexing` call:

- `Token` (The Token ID returned by the last `AccountLogin` call).
- `ItemsToIndex` is an array of type `Item`. An item is defined by the following parameter:
 - `Path` (Name of the item with the FilesAnywhere Path). The Path should be formatted like `\UserVolume\FolderName\ItemName`, where `UserVolume` is the FilesAnywhere Username in whose account the item exists. The function submits items with a valid document extension as listed below for indexing.

Note: Items with extensions like `jpg`, `jpeg`, `gif`, `png`, `bmp` and `wmf` are always considered as an indexed file and page counts is always `1` and are never submitted for indexing.

`RequestDocIndexing` function should be used only for specific file types like all the Microsoft Office documents, `pdf` and `tif` files. The valid document extensions are `doc`, `ppt`, `xls`, `pdf`, `tif`, `vsd`, `txt`, `rtf`, `wri`, `pub`, `mpp`, `mdi`, `dot`, `xlt`, `wq1`, `wks`, `wk3`, `wk4`, `slk`, `prn`, `dif`, `csv`.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /facorpapi/fawapi.asmx HTTP/1.1
Host: 10.10.10.16
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/RequestDocIndexing"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <RequestDocIndexing xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ItemsToIndex>
        <Item>
          <Path>string</Path>
        </Item>
        <Item>
          <Path>string</Path>
        </Item>
      </ItemsToIndex>
    </RequestDocIndexing>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <RequestDocIndexingResponse xmlns="http://api.filesanywhere.com/">
      <DocIndexingResult>
        <ErrorMessage>string</ErrorMessage>
        <SuccessCount>int</SuccessCount>
      </DocIndexingResult>
    </RequestDocIndexingResponse>
  </soap:Body>
</soap:Envelope>
```


SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /facorpapi/fawapi.asmx HTTP/1.1
Host: 10.10.10.16
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <RequestDocIndexing xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ItemsToIndex>
        <Item>
          <Path>string</Path>
        </Item>
        <Item>
          <Path>string</Path>
        </Item>
      </ItemsToIndex>
    </RequestDocIndexing>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <RequestDocIndexingResponse xmlns="http://api.filesanywhere.com/">
      <DocIndexingResult>
        <ErrorMessage>string</ErrorMessage>
        <SuccessCount>int</SuccessCount>
      </DocIndexingResult>
    </RequestDocIndexingResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub RequestDocIndexing ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim DocIndexingResult As New WebReference.DocIndexingResult
    DocIndexingResult=FAWAPI.RequestDocIndexingResult (Token,
    ItemsToIndex)
    If DocIndexingResult.ErrorMessage IsNot Nothing Then
        MsgBox DocIndexingResult.ErrorMessage
    End If
    DocIndexingResult = Nothing
    FAWAPI = Nothing
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
ItemsToIndex	Array of type Item	Yes	An array of items Path.

An Item has the following properties:

Property Name	Data Type	Required	Description
Path	String	Yes	Full FilesAnywhere path of the Item in the form \UserVolume\FolderName\ItemName. Please refer to the usage section of this function for the list of the items extension that can be passed to this function.

Response**DocIndexingResult**

DocIndexingResults Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ItemThumbnail method was executed. If the ItemThumbnail call was successful, the ErrorMessage is blank or null.
SuccessCount	int	Returns that items count that is submitted for indexing.

SAMLAccountLogin

This method logs you in to your FilesAnywhere account using SAML 2.0 security token and starts an API session. It returns back a Token ID which must be used for subsequent API calls.

Syntax

LoginResult = SAMLAccountLogin(APIkey, EmailAddress, SAMLToken)

Usage

Use the SAMLAccountLogin call to login to your FilesAnywhere account using SAML 2.0 security token and start a client API session. A client application must first login and obtain a Token ID before making other API calls.

The following parameters are needed to make the SAMLAccountLogin call:

- **APIKey** (A Developer API Key. See the “Getting Started” section above, to find out how to obtain a Developer API Key).
- **EmailAddress** (Email address used for Single Sign-On integration). This is the email address configured in Identity provider(Active Directory, OneLogin, etc.) for user’s identity.
- **SAMLToken**. This is the SAML 2.0 security token received from identity provider (Active Directory Federation Service, OneLogin etc.) for FilesAnywhere application after authentication.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: dev.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/SAMLAccountLogin"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SAMLAccountLogin xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <EmailAddress>string</EmailAddress>
      <SAMLToken>string</SAMLToken>
    </SAMLAccountLogin>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SAMLAccountLoginResponse xmlns="http://api.filesanywhere.com/">
      <LoginResult>
        <ErrorMessage>string</ErrorMessage>
        <Token>string</Token>
        <ClientEncryptedPassword>string</ClientEncryptedPassword>
      </LoginResult>
    </SAMLAccountLoginResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: dev.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SAMLAccountLogin xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <EmailAddress>string</EmailAddress>
      <SAMLToken>string</SAMLToken>
    </SAMLAccountLogin>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SAMLAccountLoginResponse xmlns="http://api.filesanywhere.com/">
      <LoginResult>
        <ErrorMessage>string</ErrorMessage>
        <Token>string</Token>
        <ClientEncryptedPassword>string</ClientEncryptedPassword>
      </LoginResult>
    </SAMLAccountLoginResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub SAMLAccountLogin()  
    Dim FAWAPI As New WebReference.FAWAPI  
    Dim LoginResult As New WebReference.LoginResult  
  
    LoginResult = FAWAPI.SAMLAccountLogin(APIKey, EmailAddress,  
SAML20SecurityToken)  
    If LoginResult.ErrorMessage <> "" Then  
        MsgBox LoginResult.ErrorMessage  
    Else  
        Dim Token As String = LoginResult.Token  
    End If  
    LoginResult = Nothing  
    FAWAPI = Nothing  
End Sub
```

Request Arguments

Name	Data Type	Required	Description
APIKey	String	Yes	A Developer API Key. The “Getting Started” section describes how to obtain a Developer API Key
EmailAddress	String	Yes	Email address used for Single Sign-On integration and configured into identity provider settings for user.
SAMLToken	String	Yes	SAML security token received from identity provider after authentication.

Response

LoginResult

LoginResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any encountered while the SAMLAccountLogin method was executed. If the SAMLAccountLogin call was successful, the ErrorMessage is blank.
Token	String	A Token ID to be used in subsequent API calls
ClientEncryptedPassword	String	This will be always blank and not applicable for Single Sign-On authentication.

GoogleOAuthAccountLogin

This method logs you in to your FilesAnywhere account using GoogleOAuth security code and starts an API session. It returns back a Token ID which must be used for subsequent API calls.

Syntax

```
LoginResult = GoogleOAuthAccountLogin(APIkey, EmailAddress, OAuthCode, ApplicationType)
```

Usage

Use the GoogleOAuthAccountLogin call to login to your FilesAnywhere account using GoogleOAuth code and start a client API session. A client application must first login and obtain a Token ID before making other API calls.

The following parameters are needed to make the GoogleOAuthAccountLogin call:

- **APIKey** (A Developer API Key. See the “Getting Started” section above, to find out how to obtain a Developer API Key).
- **EmailAddress** (Email address used for Single Sign-On integration). This is the email address configured in Google account for user’s identity.
- **OAuthCode**. This is the Google authorization code received from Google for FilesAnywhere application after authentication.
- **ApplicationType**. Based on the calling application type parameter value needs to pass. Possible values are as below:
 - **Native Applications:** 1
 - Native application covers all installed applications like Windows app, Android app, iOS app etc.
 - **FilesAnywhere Mobile Web app:** 2

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GoogleOAuthAccountLogin"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GoogleOAuthAccountLogin xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <EmailAddress>string</EmailAddress>
      <GoogleOAuthCode>string</GoogleOAuthCode>
      <ApplicationType>int</ApplicationType>
    </GoogleOAuthAccountLogin>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GoogleOAuthAccountLoginResponse
xmlns="http://api.filesanywhere.com/">
      <LoginResult>
        <ErrorMessage>string</ErrorMessage>
        <Token>string</Token>
        <ClientEncryptedPassword>string</ClientEncryptedPassword>
      </LoginResult>
    </GoogleOAuthAccountLoginResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GoogleOAuthAccountLogin xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <EmailAddress>string</EmailAddress>
      <GoogleOAuthCode>string</GoogleOAuthCode>
      <ApplicationType>int</ApplicationType>
    </GoogleOAuthAccountLogin>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GoogleOAuthAccountLoginResponse
xmlns="http://api.filesanywhere.com/">
      <LoginResult>
        <ErrorMessage>string</ErrorMessage>
        <Token>string</Token>
        <ClientEncryptedPassword>string</ClientEncryptedPassword>
      </LoginResult>
    </GoogleOAuthAccountLoginResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```

Private Sub GoogleOAuthAccountLogin()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim LoginResult As New WebReference.LoginResult

    LoginResult = FAWAPI.GoogleOAuthAccountLogin(APIKey, EmailAddress,
GoogleOAuthCode, AppType)
    If LoginResult.ErrorMessage <> "" Then
        MsgBox LoginResult.ErrorMessage
    Else
        Dim Token As String = LoginResult.Token
    End If
    LoginResult = Nothing
    FAWAPI = Nothing
End Sub

```

Request Arguments

Name	Data Type	Required	Description
APIKey	String	Yes	A Developer API Key. The "Getting Started" section describes how to obtain a Developer API Key
EmailAddress	String	Yes	Email address used for Single Sign-On integration and configured into identity provider settings for user.
GoogleOAuthCode	String	Yes	GoogleOAuth code received from Google after authentication.
ApplicationType	Int	Yes	Value could be: 1 for Native installed applications 2 for FilesAnywhere Mobile web app

Response

LoginResult

LoginResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error encountered while the GoogleOAuthAccountLogin method was executed. If the GoogleOAuthAccountLogin call was successful, the ErrorMessage is blank.
Token	String	A Token ID to be used in subsequent PI calls
ClientEncryptedPassword	String	This will be always blank and not applicable for Single Sign-On authentication.

LDAPAccountLogin

This method logs you in to your FilesAnywhere account using LDAP credentials and starts an API session. It returns back a Token ID which must be used for subsequent API calls.

Syntax

LoginResult = LDAPAccountLogin(APIkey, EmailAddress, UserName, Password)

Usage

Use the LDAPAccountLogin call to login to your FilesAnywhere account using LDAP credentials and start a client API session. A client application must first login and obtain a Token ID before making other API calls.

The following parameters are needed to make the GoogleOAuthAccountLogin call:

- **APIKey** (A Developer API Key. See the “Getting Started” section above, to find out how to obtain a Developer API Key).
- **EmailAddress** (Email address used for Single Sign-On integration). This is the email address configured in LDAP server for user’s identity.
- **UserName** (LDAP UserName). This is the LDAP username using with the user logs in to his FilesAnywhere account.
- **Password** (LDAP Password). This is the user’s LDAP account password.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/LDAPAccountLogin"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <LDAPAccountLogin xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <EmailAddress>string</EmailAddress>
      <UserName>string</UserName>
      <Password>string</Password>
    </LDAPAccountLogin>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <LDAPAccountLoginResponse xmlns="http://api.filesanywhere.com/">
      <LoginResult>
        <ErrorMessage>string</ErrorMessage>
        <Token>string</Token>
        <ClientEncryptedPassword>string</ClientEncryptedPassword>
      </LoginResult>
    </LDAPAccountLoginResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <LDAPAccountLogin xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <EmailAddress>string</EmailAddress>
      <UserName>string</UserName>
      <Password>string</Password>
    </LDAPAccountLogin>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <LDAPAccountLoginResponse xmlns="http://api.filesanywhere.com/">
      <LoginResult>
        <ErrorMessage>string</ErrorMessage>
        <Token>string</Token>
        <ClientEncryptedPassword>string</ClientEncryptedPassword>
      </LoginResult>
    </LDAPAccountLoginResponse>
  </soap12:Body>
</soap12:Envelope>
```


Sample Visual Basic.Net Code

```
Private Sub LDAPAccountLogin()  
    Dim FAWAPI As New WebReference.FAWAPI  
    Dim LoginResult As New WebReference.LoginResult  
  
    LoginResult = FAWAPI.LDAPAccountLogin(APIKey, EmailAddress,  
Username, Password)  
    If LoginResult.ErrorMessage <> "" Then  
        MsgBox LoginResult.ErrorMessage  
    Else  
        Dim Token As String = LoginResult.Token  
    End If  
    LoginResult = Nothing  
    FAWAPI = Nothing  
End Sub
```

Request Arguments

Name	Data Type	Required	Description
APIKey	String	Yes	A Developer API Key. The "Getting Started" section describes how to obtain a Developer API Key
EmailAddress	String	Yes	Email address used for Single Sign-On integration and configured into identity provider settings for user.
Username	String	Yes	LDAP Username. it could be account login or common name of LDAP server.
Password	String	Yes	LDAP password of user

Response

LoginResult

LoginResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error encountered while the LDAPAccountLogin method was executed. If the LDAPAccountLogin call was successful, the ErrorMessage is blank.
Token	String	A Token ID to be used in subsequent PI calls
ClientEncryptedPassword	String	This will be always blank and not applicable for Single Sign-On authentication.

SSOLoginURL

This method provides you login URL of Single Sign-On interface of user, which can be used to take the credential of user.

Syntax

SSOLoginURLResult = SSOLoginURL (APIkey, EmailAddress)

Usage

Use the SSOLoginURL call to get the Login URL and Single Sign-On type for a user, which is used by client application to authenticate the user.

The following parameters are needed to make the SSOLoginURL call:

- **APIKey** (A Developer API Key. See the “Getting Started” section above, to find out how to obtain a Developer API Key).
- **EmailAddress** (Email address used for Single Sign-On integration). This is the email address configured in Identity provider(Active Directory, OneLogin, etc.) for user’s identity.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/SSOLoginURL"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SSOLoginURL xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <EmailAddress>string</EmailAddress>
    </SSOLoginURL>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SSOLoginURLResponse xmlns="http://api.filesanywhere.com/">
      <SSOLoginURL>
        <ErrorMessage>string</ErrorMessage>
        <LoginURL>string</LoginURL>
        <SSOType>string</SSOType>
        <ClientID>string</ClientID>
        <UserName>string</UserName>
      </SSOLoginURL>
    </SSOLoginURLResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: dev.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SSOLoginURL xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <EmailAddress>string</EmailAddress>
    </SSOLoginURL>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SSOLoginURLResponse xmlns="http://api.filesanywhere.com/">
      <SSOLoginURL>
        <ErrorMessage>string</ErrorMessage>
        <LoginURL>string</LoginURL>
        <SSOType>string</SSOType>
        <ClientID>string</ClientID>
        <UserName>string</UserName>
      </SSOLoginURL>
    </SSOLoginURLResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub SAMLAccountLogin()  
    Dim FAWAPI As New WebReference.FAWAPI  
    Dim SSOLoginURLResult As New WebReference.SSOLoginURLResult  
  
    SSOLoginURLResult = FAWAPI.SSOLoginURL(APIKey, EmailAddress)  
    If SSOLoginURLResult.ErrorMessage <> "" Then  
        MsgBox LoginResult.ErrorMessage  
    Else  
        Dim sLoginURL As String = SSOLoginURLResult.LoginURL  
        Dim sSSOType As String = SSOLoginURLResult.SSOType  
        Dim sClientID As String = SSOLoginURLResult.ClientID  
        Dim sUserName As String = SSOLoginURLResult.UserName  
    End If  
    SSOLoginURLResult = Nothing  
    FAWAPI = Nothing  
End Sub
```

Request Arguments

Name	Data Type	Required	Description
APIKey	String	Yes	A Developer API Key. The “Getting Started” section describes how to obtain a Developer API Key
EmailAddress	String	Yes	Email address used for Single Sign-On integration and configured into identity provider settings for user.

Response

SSOLoginURLResult

SSOLoginURLResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any encountered while the SSOLoginURL method was executed. If the SSOLoginURL call was successful, the ErrorMessage is blank.
LoginURL	String	Login URL to be used for authentication with Identity provider of user.
SSOType	String	Type of Single Sign-On configured for User. It could be ADFS, OneLogin, LDAP, GoogleAuth etc.
UserName	String	Filesanywhere username
ClientID	String	ClientID of company in which user belongs to.

SSOLoginWebURL

This method provides you login URL of Single Sign On interface of user, which can be used to take the credential of user on Web applications.

Syntax

SSOLoginURLResult = SSOLoginWebURL (APIkey, EmailAddress)

Usage

Use the SSOLoginWebURL call to get the Login URL and Single Sign-On type for a user, which can be use by Web applications to authenticate the user.

The following parameters are needed to make the SSOLoginWebURL call:

- **APIKey** (A Developer API Key. See the “Getting Started” section above, to find out how to obtain a Developer API Key).
- **EmailAddress** (Email address used for Single Sign On integration). This is the email address configured in Identity provider(Active Directory, OneLogin, etc.) for user’s identity.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /facorpapi/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/SSOLoginWebURL"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SSOLoginWebURL xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <EmailAddress>string</EmailAddress>
    </SSOLoginWebURL>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SSOLoginWebURLResponse xmlns="http://api.filesanywhere.com/">
      <SSOLoginURL>
        <ErrorMessage>string</ErrorMessage>
        <LoginURL>string</LoginURL>
        <SSOType>string</SSOType>
      </SSOLoginURL>
    </SSOLoginWebURLResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /facorpapi/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SSOLoginWebURL xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <EmailAddress>string</EmailAddress>
    </SSOLoginWebURL>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SSOLoginWebURLResponse xmlns="http://api.filesanywhere.com/">
      <SSOLoginURL>
        <ErrorMessage>string</ErrorMessage>
        <LoginURL>string</LoginURL>
        <SSOType>string</SSOType>
      </SSOLoginURL>
    </SSOLoginWebURLResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub SAMLAccountLogin()  
    Dim FAWAPI As New WebReference.FAWAPI  
    Dim SSOLoginURLResult As New WebReference.SSOLoginURLResult  
  
    SSOLoginURLResult = FAWAPI.SSOLoginWebURL(APIKey, EmailAddress)  
    If SSOLoginURLResult.ErrorMessage <> "" Then  
        MsgBox LoginResult.ErrorMessage  
    Else  
        Dim sLoginURL As String = SSOLoginURLResult.LoginURL  
        Dim sSSType As String = SSOLoginURLResult.SSType  
    End If  
    SSOLoginURLResult = Nothing  
    FAWAPI = Nothing  
End Sub
```

Request Arguments

Name	Data Type	Required	Description
APIKey	String	Yes	A Developer API Key. The “Getting Started” section describes how to obtain a Developer API Key
EmailAddress	String	Yes	Email address used for Single Sign On integration and configured into identity provider settings for user.

Response

SSOLoginURLResult

SSOLoginURLResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any encountered while the SSOLoginURL method was executed. If the SSOLoginWebURL call was successful, the ErrorMessage is blank.
LoginURL	String	Login URL to be used for authentication with Identity provider of user.
SSOType	String	Type of Single Sign-On configured for User. It could be ADFS, OneLogin, LDAP, GoogleAuth etc.

SendDropboxLink

This function produces a web URL that anyone, even non-members, can use to upload files to a designated folder in a user account. An email is sent by the system containing an encrypted web link, which when activated by recipients, will prompt them to Upload any files to the specified folder, according to the parameters (options) given when calling the function. If the 'RecipientsEmailAddresses' element is equal to the value 'GENERALLINK', the function will return the FileReceive URL Link without sending an email. This link can be used directly from any location, and the link behaves exactly like the way it does when sent in an email.

Syntax

`SendElinkResult = SendDropboxLink (Token, DropFolderPath, RecipientsEmailAddresses, ShareDays, RecordFileHistoryLog, NotifyDownloadByEmail, SendReadReceipt, LinkInFileAttachment, OverWriteFiles, SubFolderNamePrompt, UserEmailAsSender, EmailSubject, EmailBody, ElinkPassWord)`

Usage

Use the `SendDropboxLink` method to share a FileReceive URL link securely with anyone by sending encrypted web link in an email.

The following parameters are needed to make the `SendDropboxLink` call:

- `Token` (The Token ID returned by the last `AccountLogin` call).
- `DropFolderPath` is the Path (Name of the item with the FilesAnywhere Path). The Path should be formatted like `\UserVolume\FolderName\ItemName`, where `UserVolume` is the FilesAnywhere Username in whose account the item exists .
- `RecipientsEmailAddresses` is the list of email addresses of the non member users with whom the user wants to share the FileReceive Link. The email addresses must be separated by “;”. If the 'RecipientsEmailAddresses' element is equal to the value 'GENERALLINK', the function will return the FileReceive URL Link without sending an email.
- `ShareDays` is the number of days the user wants to share the items with the recipients, after which the link will be expired. `ShareDays` can only have one of the following values:

1, 2, 5, 10, 15, 30, 60 and 365

- **RecordFileHistoryLog** is one of the tracking option through which the user can record the file drops in the file history log. This flag can be set as an uppercase Y (Yes) or N (No).
- **NotifyDownloadByEmail** is one of the tracking option through which the user can notified by email for each item uploaded. This flag can be set as an uppercase Y (Yes) or N (No).
- **SendReadReceipt** is one of the tracking option through which the user can be notified by email when the recipient clicks on the FileReceive Link. This flag can be set as an uppercase Y (Yes) or N (No).
- **LinkInFileAttachment** is the flag that can be set as an uppercase Y (Yes) or N (No). If this flag is set as “Y” then the FileReceive Link is included in the text file which is sent as an attachment in an email.
- **OverWriteFiles** is the flag that can be set as an uppercase Y (Yes) or N (No). If this flag is set as “Y”, file drops will overwrite files of the same name.
- **SubFolderNamePrompt** is the name of the folder which gets created into which the file drops will be saved.
- **ReadOnlyPermission** is the Permission flag that can be set as an uppercase Y (Yes) or N (No). If this flag is set as “Y” then the recipient can only view the items shared by the user and they cannot download any of the item. If this flag is set to “N” then the recipient can download any of the items.
- **UserEmailAsSender** if set to “Y” then the user’s email address appears as the email sender. If set to “N” the FilesAnywhere system email address appears as the email sender.
- **EmailSubject** is the email subject that will appear in the email sent to the recipients.
- **EmailBody** is the message that will appear in the email body that is sent to the recipients.
- **ElinkPassWord** is the password that the recipient is required to enter to view the items. The password is not emailed to the recipient and the user has to inform them directly.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

POST /FACORPAPI/FAWAPI.asmx HTTP/1.1

Host: localhost

Content-Type: text/xml; charset=utf-8

Content-Length: length

SOAPAction: "http://api.filesanywhere.com/SendDropboxLink"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendDropboxLink xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <DropFolderPath>string</DropFolderPath>
      <RecipientsEmailAddresses>string</RecipientsEmailAddresses>
      <ShareDays>int</ShareDays>
      <RecordFileHistoryLog>string</RecordFileHistoryLog>
      <NotifyDownloadByEmail>string</NotifyDownloadByEmail>
      <SendReadReceipt>string</SendReadReceipt>
      <LinkInFileAttachment>string</LinkInFileAttachment>
      <OverWriteFiles>string</OverWriteFiles>
      <SubFolderNamePrompt>string</SubFolderNamePrompt>
      <UserEmailAsSender>string</UserEmailAsSender>
      <EmailSubject>string</EmailSubject>
      <EmailBody>string</EmailBody>
      <ElinkPassWord>string</ElinkPassWord>
    </SendDropboxLink>
  </soap:Body>
</soap:Envelope>
```

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendDropboxLinkResponse xmlns="http://api.filesanywhere.com/">
      <SendDropboxLink>
        <ErrorMessage>string</ErrorMessage>
        <ElinkSent>boolean</ElinkSent>
        <ElinkURLs>
          <ElinkURL>
            <URL>string</URL>
          </ElinkURL>
          <ElinkURL>
            <URL>string</URL>
          </ElinkURL>
        </ElinkURLs>
      </SendDropboxLink>
    </SendDropboxLinkResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

POST /FACORPAPI/FAWAPI.asmx HTTP/1.1

Host: localhost

Content-Type: application/soap+xml; charset=utf-8

Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SendDropboxLink xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <DropFolderPath>string</DropFolderPath>
      <RecipientsEmailAddresses>string</RecipientsEmailAddresses>
      <ShareDays>int</ShareDays>
      <RecordFileHistoryLog>string</RecordFileHistoryLog>
      <NotifyDownloadByEmail>string</NotifyDownloadByEmail>
      <SendReadReceipt>string</SendReadReceipt>
      <LinkInFileAttachment>string</LinkInFileAttachment>
      <OverWriteFiles>string</OverWriteFiles>
      <SubFolderNamePrompt>string</SubFolderNamePrompt>
      <UserEmailAsSender>string</UserEmailAsSender>
      <EmailSubject>string</EmailSubject>
      <EmailBody>string</EmailBody>
      <ElinkPassWord>string</ElinkPassWord>
    </SendDropboxLink>
  </soap12:Body>
</soap12:Envelope>
```

HTTP/1.1 200 OK

Content-Type: application/soap+xml; charset=utf-8

Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SendDropboxLinkResponse xmlns="http://api.filesanywhere.com/">
      <SendDropboxLink>
        <ErrorMessage>string</ErrorMessage>
        <ElinkSent>boolean</ElinkSent>
        <ElinkURLs>
          <ElinkURL>
            <URL>string</URL>
          </ElinkURL>
          <ElinkURL>
            <URL>string</URL>
          </ElinkURL>
        </ElinkURLs>
      </SendDropboxLink>
    </SendDropboxLinkResponse>
  </soap12:Body>
</soap12:Envelope>
```


Sample Visual Basic.Net Code

```

Private Sub SendDropboxLink()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim wsSendElinkResult As New WebReference.SendElinkResult
    wsSendElinkResult=wsFAWAPI.SendDropboxLink(Token, DropFolderPath,
    RecipientsEmailAddresses, ShareDays, RecordFileHistoryLog,
    NotifyDownloadByEmail, SendReadReceipt, LinkInFileAttachment,
    OverWriteFiles, SubFolderNamePrompt, UserEmailAsSender,
    EmailSubject, EmailBody, ElinkPassWord)
    If Not wsSendElinkResult.ElinkSent Or
        wsSendElinkResult.ErrorMessage <> "" Then
        MsgBox wsSendElinkResult.ErrorMessage
    End If
    wsSendElinkResult = Nothing
    FAWAPI = Nothing
End Sub

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
DropFolderPath	String	Yes	Full FilesAnywhere path of the file in the form \UserVolume\FolderName\File Name
RecipientsEmailAddresses	String	Yes	Email Addresses of the recipients
ShareDays	integer	Yes	Number of days the recipient can access the items in the link
RecordFileHistoryLog	String	Yes	Uppercase "Y" or "N" flag. If set to "Y" then the item downloads are recorded in the file history log.
NotifyDownloadByEmail	String	Yes	Uppercase "Y" or "N" flag. If set to "Y" then the user is notified by email for each item uploaded.
SendReadReceipt	String	Yes	Uppercase "Y" or "N" flag. If set to "Y" then the user is notified by email when the recipient clicks on the FileReceive Link.
LinkInFileAttachment	String	Yes	Uppercase "Y" or "N" flag. If set to "Y" then the item share link is included in the text file which is sent as an attachment in the email.

OverWriteFiles	String	Yes	Uppercase "Y" or "N" flag. If this flag is set as "Y" then the file drop will overwrite a file of the same name.
SubFolderNamePrompt	String	Yes	The name of the subfolder which gets created when the user makes a file drop. The file gets saved to this subfolder.
UserEmailAsSender	String	Yes	Uppercase "Y" or "N" flag. If set to "Y" then the user's email address appears as the email sender.
EmailSubject	String	Yes	Text that appears in the email subject.
EmailBody	String	No	Message that appears in the email body.
ElinkPassWord	String	No	It's the password that the recipient is required to enter to view the items. The password is not emailed to the recipient and the user has to inform them directly.

Response**SendDropboxLink**

SendDropboxLink Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the generating and emailing the link to recipients. If the SendDropboxLink call was successful, the ErrorMessage is blank or null.
ElinkSent	Boolean	Returns true if the SendDropboxLink call was successful.
ElinkURLs	An array of type ElinkURL	An array of URLs

A ElinkURL has the following properties:

Property Name	Data Type	Required	Description
URL	String	Yes	A string which is a URL

SendItemsELinkWithDefaults

This function is similar to the `SendFileShareLink` operation, except that it uses the defaults stored in the user profile. It shares items securely with anyone, even non-members, using a web URL link. An email is sent with an encrypted web link, prompting the guest recipients to download the shared items as specified using this function.

Syntax

`SendElinkResult = SendItemsELinkWithDefaults (Token, ELinkItems, RecipientsEmailAddresses, SendEmail, ElinkPassWord)`

Usage

Use the `SendItemsELinkWithDefaults` method to share items securely with anyone by sending encrypted web link in an email using default settings stored in your user profile.

The following parameters are needed to make the `SendItemsELinkWithDefaults` call:

- **Token** (The Token ID returned by the last `AccountLogin` call).
- **ELinkItems** is an array of type `Item`. An item is defined by the following parameter:
 - **Type**. This is the Item type and can either be a 'file' or a 'folder'.
 - **Path** (Name of the item with the FilesAnywhere Path). The Path should be formatted like `\UserVolume\FolderName\ItemName`, where `UserVolume` is the FilesAnywhere Username in whose account the item exists. The function submits items with a valid document extension as listed below for indexing.
- **RecipientsEmailAddresses** is the list of email addresses of the non member users with whom the user wants to share its items. The email addresses must be separated by ";". If the 'RecipientsEmailAddresses' element is equal to the value 'GENERALLINK', the function will return the FileShare URL Link without sending an email.
- **SendEmail** is a flag which can be set as an uppercase Y (Yes) or N (No). If this flag is set as "Y", then an email is sent to the `RecipientsEmailAddresses` with the FileShare link. If it is set as "N", then no email is sent, just a link is generated.

- **ElinkPassWord** is the password that the recipient is required to enter to view the items. The password is not emailed to the recipient and the user has to inform them directly.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/SendItemsELinkWithDefaults"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendItemsELinkWithDefaults xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ELinkItems>
        <Item>
          <Type>string</Type>
          <Path>string</Path>
        </Item>
        <Item>
          <Type>string</Type>
          <Path>string</Path>
        </Item>
      </ELinkItems>
      <RecipientsEmailAddresses>string</RecipientsEmailAddresses>
      <SendEmail>string</SendEmail>
      <ElinkPassWord>string</ElinkPassWord>
    </SendItemsELinkWithDefaults>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendItemsELinkWithDefaultsResponse
xmlns="http://api.filesanywhere.com/">
      <SendElinkResult>
```

```

    <ErrorMessage>string</ErrorMessage>
    <ELinkSent>boolean</ELinkSent>
    <ElinksDetails>
      <ElinkDetails>
        <URL>string</URL>
        <EmailAddress>string</EmailAddress>
        <FileShareID>int</FileShareID>
      </ElinkDetails>
    </ElinkDetails>
    <ElinkDetails>
      <URL>string</URL>
      <EmailAddress>string</EmailAddress>
      <FileShareID>int</FileShareID>
    </ElinkDetails>
  </ElinksDetails>
</SendElinkResult>
</SendItemsElinkWithDefaultsResponse>
</soap:Body>
</soap:Envelope>

```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The [placeholders](#) shown need to be replaced with actual values.

```

POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SendItemsElinkWithDefaults xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ElinkItems>
        <Item>
          <Type>string</Type>
          <Path>string</Path>
        </Item>
        <Item>
          <Type>string</Type>
          <Path>string</Path>
        </Item>
      </ElinkItems>
      <RecipientsEmailAddresses>string</RecipientsEmailAddresses>
      <SendEmail>string</SendEmail>
      <ElinkPassWord>string</ElinkPassWord>
    </SendItemsElinkWithDefaults>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SendItemsELinkWithDefaultsResponse
xmlns="http://api.filesanywhere.com/">
      <SendElinkResult>
        <ErrorMessage>string</ErrorMessage>
        <ElinkSent>boolean</ElinkSent>
        <ElinksDetails>
          <ElinkDetails>
            <URL>string</URL>
            <EmailAddress>string</EmailAddress>
            <FileShareID>int</FileShareID>
          </ElinkDetails>
          <ElinkDetails>
            <URL>string</URL>
            <EmailAddress>string</EmailAddress>
            <FileShareID>int</FileShareID>
          </ElinkDetails>
        </ElinksDetails>
      </SendElinkResult>
    </SendItemsELinkWithDefaultsResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub SendItemsELinkWithDefaults ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim wsSendElinkListResult As New WebReference.SendElinkListResult
    wsSendElinkListResult=wsFAWAPI.SendItemsELinkWithDefaults
    (Token,aItemsDetails, RecipientsEmailAddresses,SendEmail,ElinkPassWord)
    If Not wsSendElinkResult.ElinkSent Or
        wsSendElinkResult.ErrorMessage <> "" Then
        MsgBox wsSendElinkResult.ErrorMessage
    End If
    wsSendElinkResult = Nothing
    FAWAPI = Nothing
End Sub
```


Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
ELinkItems	Array of type Item	Yes	An array of items Path.
RecipientsEmailAddresses	String	Yes	Email Addresses of the recipients
SendEmail	String	Yes	Uppercase "Y" or "N" flag. If set to "Y" then an email is sent to the RecipientsEmailAddresses with the FileShare link. If set to "N", then no email is sent, just a link is generated.
ElinkPassWord	String	No	It's the password that the recipient is required to enter to view the items. The password is not emailed to the recipient and the user has to inform them directly.

An Item has the following properties:

Property Name	Data Type	Required	Description
Type	String	Yes	Type can either be a 'file' or a 'folder'
Path	String	Yes	Full FilesAnywhere path of the Item in the form \UserVolume\FolderName\ItemName. Please refer to the usage section of this function for the list of the items extension that can be passed to this function.

Response**SendElinkListResult**

SendElinkListResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the generating and emailing the link to recipients. If the SendFileShareLink call was successful, the ErrorMessage is blank or null.
ElinkSent	Boolean	Returns true if the items ELink was sent to the recipient's email addresses.
ElinksDetails	An array of type ElinkDetails	An array of ElinkDetails that contains ElinkURLs and Emailaddress

A ElinkDetails has the following properties:

Property Name	Data Type	Required	Description
URL	String	Yes	A string which is a URL
EmailAddress	String	Yes	A string which has an EmailAddress
FileShareID	Integer	Yes	Represents ID of created Fileshare link

SendFileShareLinkWithDefaults

This function is similar to the `SendFileShareLink` operation, except that it uses the defaults stored in the user profile. It shares items securely with anyone, even non-members, using a web URL link. An email is sent with an encrypted web link, prompting the guest recipients to download the shared items as specified using this function.

Syntax

```
SendElinkListResult = SendFileShareLinkWithDefaults(Token, SendElinkDefaultParameter);
```

Usage

Use the `SendFileShareLinkWithDefaults` method to share items securely with anyone by sending encrypted web link in an email using default settings stored in your user profile.

The following parameters are needed to make the `SendItemsElinkWithDefaults` call:

- **Token** (The Token ID returned by the last `AccountLogin` call).
- **SendElinkDefaultParameter**
 1. **ElinkItems** is an array of type `Item`. An item is defined by the following parameter:
 - **Type**. This is the `Item` type and can either be a 'file' or a 'folder'.
 - **Path** (Name of the item with the FilesAnywhere Path). The Path should be formatted like `\UserVolume\FolderName\ItemName`, where `UserVolume` is the FilesAnywhere `UserName` in whose account the item exists. The function submits items with a valid document extension as listed below for indexing.
 2. **RecipientsEmailAddresses** is the list of email addresses of the non member users with whom the user wants to share its items. The email addresses must be separated by ";". If the 'RecipientsEmailAddresses' element is equal to the value 'GENERALLINK', the function will return the FileShare URL Link without sending an email.
 3. **SendEmail** is a flag which can be set as an uppercase Y (Yes) or N (No). If this flag is set as "Y", then an email is sent to the `RecipientsEmailAddresses` with the FileShare link. If it is set as "N", then no email is sent, just a link is generated.

4. **ElinkPassWord** is the password that the recipient is required to enter to view the items. The password is not emailed to the recipient and the user has to inform them directly.
5. **PasswordSecurity** is the security type, which needs to pass as an integer value.

Password security 1 means, created file share link will be protected by password; in this case, user needs to provide value for parameter 'ElinkPassword'

Password security 2 means, created file share link will be protected by OTP verification by email.

Password security 3 means, created file share link will be protected by OTP verification with SMS.

Password security 0 means, created file share link does not protected by password or OTP email or OTP sms.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/SendFileShareLinkWithDefaults"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendFileShareLinkWithDefaults xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Parameters>
        <ELinkItems>
          <Item>
            <Type>string</Type>
            <Path>string</Path>
          </Item>
          <Item>
            <Type>string</Type>
            <Path>string</Path>
          </Item>
        </ELinkItems>
        <RecipientsEmailAddresses>string</RecipientsEmailAddresses>
        <SendEmail>string</SendEmail>
        <ElinkPassWord>string</ElinkPassWord>
        <PasswordSecurity>int</PasswordSecurity>
      </SendFileShareLinkWithDefaults>
    </soap:Body>
  </soap:Envelope>
```

```

    </Parameters>
  </SendFileShareLinkWithDefaults>
</soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendFileShareLinkWithDefaultsResponse
xmlns="http://api.filesanywhere.com/">
      <SendFileShareLinkWithDefaultsResult>
        <ErrorMessage>string</ErrorMessage>
        <ELinkSent>boolean</ELinkSent>
        <ElinksDetails>
          <ElinkDetails>
            <URL>string</URL>
            <EmailAddress>string</EmailAddress>
            <FileShareID>int</FileShareID>
          </ElinkDetails>
          <ElinkDetails>
            <URL>string</URL>
            <EmailAddress>string</EmailAddress>
            <FileShareID>int</FileShareID>
          </ElinkDetails>
        </ElinksDetails>
      </SendFileShareLinkWithDefaultsResult>
    </SendFileShareLinkWithDefaultsResponse>
  </soap:Body>
</soap:Envelope>

```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The [placeholders](#) shown need to be replaced with actual values.

```

POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SendFileShareLinkWithDefaults xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Parameters>
        <ElinkItems>
          <Item>

```

```

        <Type>string</Type>
        <Path>string</Path>
    </Item>
    <Item>
        <Type>string</Type>
        <Path>string</Path>
    </Item>
</ELinkItems>
<RecipientsEmailAddresses>string</RecipientsEmailAddresses>
<SendEmail>string</SendEmail>
<ElinkPassWord>string</ElinkPassWord>
<PasswordSecurity>int</PasswordSecurity>
</Parameters>
</SendFileShareLinkWithDefaults>
</soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
    <soap12:Body>
        <SendFileShareLinkWithDefaultsResponse
xmlns="http://api.filesanywhere.com/">
            <SendFileShareLinkWithDefaultsResult>
                <ErrorMessage>string</ErrorMessage>
                <ELinkSent>boolean</ELinkSent>
                <ElinksDetails>
                    <ElinkDetails>
                        <URL>string</URL>
                        <EmailAddress>string</EmailAddress>
                        <FileShareID>int</FileShareID>
                    </ElinkDetails>
                    <ElinkDetails>
                        <URL>string</URL>
                        <EmailAddress>string</EmailAddress>
                        <FileShareID>int</FileShareID>
                    </ElinkDetails>
                </ElinksDetails>
            </SendFileShareLinkWithDefaultsResult>
        </SendFileShareLinkWithDefaultsResponse>
    </soap12:Body>
</soap12:Envelope>

```

Sample Visual Basic.Net Code

```

Private Sub SendItemsELinkWithDefaults ()
    FAWAPI.ItemDetails[] objfileShareLinkItems = new FAWAPI.ItemDetails[2];
    FAWAPI.ItemDetails objFileShareLinkItem = new FAWAPI.ItemDetails();
    FAWAPI.FAWAPIv2SoapClient objFAWAPI = new FAWAPI.FAWAPIv2SoapClient();
    objFileShareLinkItem = new FAWAPI.ItemDetails();
    objFileShareLinkItem.Path = txtFSPath.Text;

```

```

objFileShareLinkItem.Type = txtFSPath.Text.EndsWith("\\") ? "folder" :
"file";
objfileShareLinkItems[0] = objFileShareLinkItem;
FAWAPI.SendElinkDefaultParameter objParams = new
FAWAPI.SendElinkDefaultParameter();
objParams.ElinkItems = objfileShareLinkItems;
objParams.ElinkPassWord = "Password";
objParams.RecipientsEmailAddresses = "test@domain.com";
objParams.SendEmail = "Y";
objParams.PasswordSecurity = 1;

FAWAPI.SendElinkListResult objResult = objFAWAPI.
SendFileShareLinkWithDefaults (msToken, objParams);
If Not objResult.ErrorMessage <> "" Then
Msgbox objResult.ErrorMessage
End If
objResult = Nothing
FAWAPI = Nothing
End Sub

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
Parameters	SendElinkDefaultParameter	Yes	Parameter which contains required fields to create file share link

Parameters has the following properties:

Property Name	Data Type	Required	Description
ElinkItems	Array of type Item	Yes	An array of items Path.
RecipientsEmailAddresses	String	Yes	Email Addresses of the recipients
SendEmail	String	Yes	Uppercase "Y" or "N" flag. If set to "Y" then an email is sent to the RecipientsEmailAddresses with the FileShare link. If set to "N", then no email is sent, just a link is generated.
ElinkPassWord	String	No	It's the password that the recipient is required to enter to view the items. The password is not emailed to the recipient and the user has to inform them directly.
PasswordSecurity	Int	No	1 means file share link will be protected with

			password, 2 means file share link will be protected with OTP email, 3 means file share link will be protected with OTP SMS, 0 means unprotected file share link will be created.
--	--	--	--

An ElinkItems has the following properties:

Property Name	Data Type	Required	Description
Type	String	Yes	Type can either be a 'file' or a 'folder'
Path	String	Yes	Full FilesAnywhere path of the Item in the form \UserVolume\FolderName\ItemName. Please refer to the usage section of this function for the list of the items extension that can be passed to this function.

Response**SendElinkListResult**

SendElinkListResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the generating and emailing the link to recipients. If the SendFileShareLink call was successful, the ErrorMessage is blank or null.
ElinkSent	Boolean	Returns true if the items ELink was sent to the recipient's email addresses.
ElinksDetails	An array of type ElinkDetails	An array of ElinkDetails that contains ElinkURLs and Emailaddress

A ElinkDetails has the following properties:

Property Name	Data Type	Required	Description
URL	String	Yes	A string which is a URL
EmailAddress	String	Yes	A string which has an EmailAddress
FileShareID	Integer	Yes	Represents ID of created Fileshare link

FaxGetDefaults

This function returns the defaults that were saved from your last fax sent.

Syntax

FaxDefaultsResult = FaxGetDefaults(Token)

Usage

Use the FaxGetDefaults method to retrieve the values you elected to save from your last fax sent. This makes resending faxes more convenient by retrieving cover page, sender and recipient information.

The following parameter is needed to make the FaxGetDefaults call:

- Token (The Token ID returned by the last AccountLogin call).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

POST /FACORPAPI/FAWAPI.asmx HTTP/1.1

Host: localhost

Content-Type: text/xml; charset=utf-8

Content-Length: length

SOAPAction: "http://api.filesanywhere.com/FaxGetDefaults"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <FaxGetDefaults xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </FaxGetDefaults>
  </soap:Body>
</soap:Envelope>
```

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <FaxGetDefaultsResponse xmlns="http://api.filesanywhere.com/">
      <FaxDefaultsResult>
        <ErrorMessage>string</ErrorMessage>
        <RecipientFaxNumbers>string</RecipientFaxNumbers>
        <RecipientNames>string</RecipientNames>
        <RecipientCompanyName>string</RecipientCompanyName>
        <RecipientEmailAddresses>string</RecipientEmailAddresses>
        <RecipientAlternateFaxNumber>string</RecipientAlternateFaxNumber>
        <RecipientVoiceNumber>string</RecipientVoiceNumber>
        <RecipientCoverSheet>string</RecipientCoverSheet>
        <SendersName>string</SendersName>
        <SendersVoiceNumber>string</SendersVoiceNumber>
        <SendersFaxNumber>string</SendersFaxNumber>
        <SendersEmailAddress>string</SendersEmailAddress>
        <SendersCompanyName>string</SendersCompanyName>
        <FaxSensitivityCode>string</FaxSensitivityCode>
        <FaxReferenceNumber>string</FaxReferenceNumber>
        <SaveSentFaxes>string</SaveSentFaxes>
        <SaveCoverPageAsDefault>string</SaveCoverPageAsDefault>
        <SaveRecipientAsDefault>string</SaveRecipientAsDefault>
        <SaveSenderAsDefault>string</SaveSenderAsDefault>
        <UseACoverPage>string</UseACoverPage>
        <CoverPageSubject>string</CoverPageSubject>
      </FaxDefaultsResult>
    </FaxGetDefaultsResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <FaxGetDefaults xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </FaxGetDefaults>
  </soap12:Body>
</soap12:Envelope>
```

HTTP/1.1 200 OK

Content-Type: application/soap+xml; charset=utf-8

Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <FaxGetDefaultsResponse xmlns="http://api.filesanywhere.com/">
      <FaxDefaultsResult>
        <ErrorMessage>string</ErrorMessage>
        <RecipientFaxNumbers>string</RecipientFaxNumbers>
        <RecipientNames>string</RecipientNames>
        <RecipientCompanyName>string</RecipientCompanyName>
        <RecipientEmailAddresses>string</RecipientEmailAddresses>
        <RecipientAlternateFaxNumber>string</RecipientAlternateFaxNumber>
        <RecipientVoiceNumber>string</RecipientVoiceNumber>
        <RecipientCoverSheet>string</RecipientCoverSheet>
        <SendersName>string</SendersName>
        <SendersVoiceNumber>string</SendersVoiceNumber>
        <SendersFaxNumber>string</SendersFaxNumber>
        <SendersEmailAddress>string</SendersEmailAddress>
        <SendersCompanyName>string</SendersCompanyName>
        <FaxSensitivityCode>string</FaxSensitivityCode>
        <FaxReferenceNumber>string</FaxReferenceNumber>
        <SaveSentFaxes>string</SaveSentFaxes>
        <SaveCoverPageAsDefault>string</SaveCoverPageAsDefault>
        <SaveRecipientAsDefault>string</SaveRecipientAsDefault>
        <SaveSenderAsDefault>string</SaveSenderAsDefault>
        <UseACoverPage>string</UseACoverPage>
        <CoverPageSubject>string</CoverPageSubject>
      </FaxDefaultsResult>
    </FaxGetDefaultsResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub FaxGetDefaults ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim wsFaxDefaultsResult As New WebReference.FaxDefaultsResult
    ws FaxDefaultsResult=wsFAWAPI.FaxGetDefaults(Token)
    If Not wsFaxDefaultsResult.ErrorMessage <> "" Then
        MsgBox wsFaxDefaultsResult.ErrorMessage
    End If
    wsFaxDefaultsResult = Nothing
    FAWAPI = Nothing
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call

Response

FaxDefaultsResult

FaxDefaultsResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while retrieving fax defaults. If the FaxGetDefaults call was successful, the ErrorMessage is blank or null.
SaveRecipientAsDefault	String	Returns "Y", "N", or "" from last saved fax. Indicates whether (RecipientNames, RecipientFaxNumbers, RecipientCompanyName, RecipientEmailAddresses, RecipientVoiceNumber, RecipientAlternateFaxNumber) should be saved as defaults for your future faxes.
SaveSentFaxes	String	Returns "Y", "N", or "" from last saved fax. Indicates whether faxes will be saved as tiff files in your FilesAnywhere account.
FaxSensitivityCode	String	Sensitivity of the fax (ex: Confidential, Personal)
RecipientNames	String	Returns ";" delimited list or single name to which the last saved fax was sent.Ex: John Doe;Joe Blow;Murphys Law Or John Doe
RecipientEmailAddresses	String	Returns ";" delimited list or single emailaddress to which the last saved fax was sent.Ex: John.Doe@abc.com;Joe.Blow@abc.com;Murphys.Law@abc.com

		Or John.Doe@abc.com
RecipientFaxNumbers	String	Returns “;” delimited list or single number to which the last saved fax was sent.Ex: 9725551212;8175551212;2145551212 Or 9725551212
RecipientCompanyName	String	Name of fax recipient’s company.
SendersName	String	Name used on the fax for “Sent By”.
SendersEmailAddress	String	“Sent By” email address. This address will receive a delivery confirmation.
SendersFaxNumber	String	“Sent By” fax number.
SendersCompanyName	String	“Sent By” company name.
RecipientCoverSheet	String	Text body of the cover sheet.
RecipientAlternateFaxNumber	String	Number used if the primary fax number is busy..
SaveSenderAsDefault	String	Returns “Y”, “N”, or “” from last saved fax. If “Y” following fields will be saved: (SendersName, SendersVoiceNumber, SendersFaxNumber, SendersEmailAddress, SendersCompanyName) as defaults for your future faxes.
SendersVoiceNumber	String	“Sent By” voice number.
RecipientVoiceNumber	String	Recipient’s voice phone number.
UseACoverPage	String	Returns “Y”, “N”, or “” from last saved fax. If “Y” then faxes will have a cover page containing the information provided in the following fields: (SendersName, SendersVoiceNumber, SendersFaxNumber, SendersEmailAddress, SendersCompanyName, CoverPageSubject, RecipientCoverSheet, FaxSensitivityCode, FaxReferenceNumber)
SaveCoverPageAsDefault	String	Returns “Y”, “N”, or “” from last saved fax. If “Y” then the coversheet information will be saved for future faxes sent.
CoverPageSubject	String	Returns value from last saved fax.
FaxReferenceNumber	String	Returns value from last saved fax.

FaxSend

This function sends a fax using the proprietary FilesAnywhere Web Fax Service.

Syntax

FaxSendResult = FaxSend(Token, FaxFilePaths, RecipientFaxNumbers, RecipientNames, RecipientCompanyName, RecipientEmailAddresses, RecipientAlternateFaxNumber, RecipientVoiceNumber, RecipientCoverSheet, SendersName, SendersVoiceNumber, SendersFaxNumber, SendersEmailAddress, SendersCompanyName, FaxSensitivityCode, FaxReferenceNumber, SaveSentFaxes, SaveCoverPageAsDefault, SaveRecipientAsDefault, SaveSenderAsDefault, UseACoverPage, CoverPageSubject)

Usage

Use the FaxSend method to Fax Your Files directly to any Fax machine. Send faxes directly online from your account to any fax machine in the world! You can fax any Office file (.xls, .doc, etc.), PDF, graphic/photo or text file. You can also browse and view all sent faxes online through a Fax Viewer. Basic outbound faxing is completely free.

The following parameters are needed to make the FaxSend call:

- **Token** (The Token ID returned by the last AccountLogin call).
- **FaxFilePaths** is "*" delimited list of file paths to be faxed. The list should not start or end with "*". So if there is only one file there will be no "*".
- **RecipientFaxNumbers** is ";" delimited list of fax numbers to receive your fax. The list should not start or end with ";". So if there is only one number there will be no ";".
- **RecipientNames** is ";" delimited list of the fax recipient names. The list should not start or end with ";". So if there is only one name there will be no ";". **THE LIST SHOULD MATCH RecipientFaxNumbers ONE TO ONE.**
- **RecipientCompanyName** is the name of fax recipient's company. This is not a list. If provided it will apply to ALL faxes. Leave blank if not in use.
- **RecipientEmailAddresses** is ";" delimited list of the fax recipient email address. If provided recipient will receive a fax delivery notice and a link to download the file. The list should not start or end with ";". So if there is only one email there will be no ";". **THE LIST SHOULD MATCH RecipientFaxNumbers ONE TO ONE.**

- **RecipientAlternateFaxNumber** is used if the primary fax number is busy. If one is provided it will be used for all faxes.
- **RecipientVoiceNumber** is the recipient's voice phone number. If one is provided it will be used for all faxes.
- **RecipientCoverSheet** is the actual text of the cover sheet.
- **SendersName** is the name used on the fax for "Sent By".
- **SendersVoiceNumber** is the "Sent By" voice number.
- **SendersFaxNumber** is the "Sent By" fax number.
- **SendersEmailAddress** is the "Sent By" email address. This address will receive a delivery confirmation.
- **SendersCompanyName** is the "Sent By" company name.
- **FaxSensitivityCode** is the sensitivity of the fax (ex: Confidential, Personal).
- **FaxReferenceNumber** is a field you may want to use for your own tracking.
- **SaveSentFaxes** "Y" or "N" if "Y" then faxes will be saved as tiff files in your FilesAnywhere account.
- **SaveCoverPageAsDefault** "Y" or "N" if "Y" then the coversheet information will be saved for future faxes sent.
- **SaveRecipientAsDefault** "Y" or "N" if "Y" then the following fields will be saved: (RecipientNames,RecipientFaxNumbers,RecipientCompanyName,RecipientEmailAddresses,RecipientVoiceNumber,RecipientAlternateFaxNumber) as defaults for your future faxes.
- **SaveSenderAsDefault** "Y" or "N" if "Y" then the following fields will be saved: (SendersName,SendersVoiceNumber,SendersFaxNumber,SendersEmailAddress,SendersCompanyName) as defaults for your future faxes.
- **UseACoverPage** "Y" or "N" if "Y" then faxes will have a cover page containing the information provided in the following fields: (SendersName,SendersVoiceNumber,SendersFaxNumber,SendersEmailAddress,SendersCompanyName,CoverPageSubject,RecipientCoverSheet,FaxSensitivityCode,FaxReferenceNumber)
- **CoverPageSubject** is the information on the Subject line in the fax cover page.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

POST /FACORPAPI/FAWAPI.asmx HTTP/1.1

Host: localhost

Content-Type: text/xml; charset=utf-8

Content-Length: length

SOAPAction: "http://api.filesanywhere.com/FaxSend"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <FaxSend xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <FaxFilePaths>string</FaxFilePaths>
      <RecipientFaxNumbers>string</RecipientFaxNumbers>
      <RecipientNames>string</RecipientNames>
      <RecipientCompanyName>string</RecipientCompanyName>
      <RecipientEmailAddresses>string</RecipientEmailAddresses>
      <RecipientAlternateFaxNumber>string</RecipientAlternateFaxNumber>
      <RecipientVoiceNumber>string</RecipientVoiceNumber>
      <RecipientCoverSheet>string</RecipientCoverSheet>
      <SendersName>string</SendersName>
      <SendersVoiceNumber>string</SendersVoiceNumber>
      <SendersFaxNumber>string</SendersFaxNumber>
      <SendersEmailAddress>string</SendersEmailAddress>
      <SendersCompanyName>string</SendersCompanyName>
      <FaxSensitivityCode>string</FaxSensitivityCode>
      <FaxReferenceNumber>string</FaxReferenceNumber>
      <SaveSentFaxes>string</SaveSentFaxes>
      <SaveCoverPageAsDefault>string</SaveCoverPageAsDefault>
      <SaveRecipientAsDefault>string</SaveRecipientAsDefault>
      <SaveSenderAsDefault>string</SaveSenderAsDefault>
      <UseACoverPage>string</UseACoverPage>
      <CoverPageSubject>string</CoverPageSubject>
    </FaxSend>
  </soap:Body>
</soap:Envelope>
```

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
```

```
<FaxSendResponse xmlns="http://api.filesanywhere.com/">
  <FaxSendResult>
    <ErrorMessage>string</ErrorMessage>
    <ProcessID>string</ProcessID>
  </FaxSendResult>
</FaxSendResponse>
</soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <FaxSend xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <FaxFilePaths>string</FaxFilePaths>
      <RecipientFaxNumbers>string</RecipientFaxNumbers>
      <RecipientNames>string</RecipientNames>
      <RecipientCompanyName>string</RecipientCompanyName>
      <RecipientEmailAddresses>string</RecipientEmailAddresses>
      <RecipientAlternateFaxNumber>string</RecipientAlternateFaxNumber>
      <RecipientVoiceNumber>string</RecipientVoiceNumber>
      <RecipientCoverSheet>string</RecipientCoverSheet>
      <SendersName>string</SendersName>
      <SendersVoiceNumber>string</SendersVoiceNumber>
      <SendersFaxNumber>string</SendersFaxNumber>
      <SendersEmailAddress>string</SendersEmailAddress>
      <SendersCompanyName>string</SendersCompanyName>
      <FaxSensitivityCode>string</FaxSensitivityCode>
      <FaxReferenceNumber>string</FaxReferenceNumber>
      <SaveSentFaxes>string</SaveSentFaxes>
      <SaveCoverPageAsDefault>string</SaveCoverPageAsDefault>
      <SaveRecipientAsDefault>string</SaveRecipientAsDefault>
      <SaveSenderAsDefault>string</SaveSenderAsDefault>
      <UseACoverPage>string</UseACoverPage>
      <CoverPageSubject>string</CoverPageSubject>
    </FaxSend>
  </soap12:Body>
</soap12:Envelope>
```

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <FaxSendResponse xmlns="http://api.filesanywhere.com/">
      <FaxSendResult>
        <ErrorMessage>string</ErrorMessage>
        <ProcessID>string</ProcessID>
      </FaxSendResult>
    </FaxSendResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub FaxSend()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim wsFaxSendResult As New WebReference.FaxSendResult
    wsFaxSendResult=wsFAWAPI.FaxSend(Token, FaxFilePaths,
    RecipientFaxNumbers, RecipientNames, RecipientCompanyName,
    RecipientEmailAddresses, RecipientAlternateFaxNumber,
    RecipientVoiceNumber, RecipientCoverSheet, SendersName,
    SendersVoiceNumber, SendersFaxNumber, SendersEmailAddress,
    SendersCompanyName, FaxSensitivityCode, FaxReferenceNumber,
    SaveSentFaxes, SaveCoverPageAsDefault, SaveRecipientAsDefault,
    SaveSenderAsDefault, UseACoverPage, CoverPageSubject)
    If Not wsFaxSendResult.ErrorMessage <> "" Then
        MsgBox wsFaxSendResult.ErrorMessage
    End If
    wsFaxSendResult = Nothing
    FAWAPI = Nothing
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
FaxFilePaths	String	Yes	Full FilesAnywhere path list delimited by a * of the files in the form of: \\UserVolume\FolderName\File Name.ext*\UserVolume\Fold

			erName2\FileName2.ext*\UserVolume\FolderName3\FileName3.ext Or \UserVolume\FolderName\FileName.ext
RecipientFaxNumbers	String	Yes	List of recipient fax numbers ex: 9725551212;8175551212;2145551212 Or 9725551212 NOTE: Valid phone code characters are also allowed like () - , # *
RecipientNames	string	Yes	A “;” delimited list of the fax recipient names. The list should not start or end with “;”. So if there is only one name there will be no “;”.THE LIST SHOULD MATCH RecipientFaxNumbers ONE TO ONE. Ex: Joe Blow;John Doe;Murphys Law Or Joe Blow
RecipientCompanyName	String	No	The name of fax recipient’s company. This is not a list. If provided it will apply to ALL faxes. Leave blank if not in use.
RecipientEmailAddresses	String	No	A “;” delimited list of the fax recipient email address. If provided recipient will receive a fax delivery notice and a link to download the file. The list should not start or end with “;”. So if there is only one email there will be no “;”. If provided THE LIST SHOULD MATCH RecipientFaxNumbers ONE TO ONE. Joe.Blow@abc.com;John.Doe@abc.com;Murphys.Law@abc.com Or Joe.Blow@abc.com;;Murphys.Law@abc.com (if you don’t know john doe’s email address) Or Joe.Blow@abc.com
RecipientAlternateFaxNumber	String	No	Used if the primary fax number is busy. If one is provided it will be used for all faxes.

RecipientVoiceNumber	String	No	The recipient's voice phone number. If one is provided it will be used for all faxes.
RecipientCoverSheet	String	No	The actual text of the cover sheet.
SendersName	String	No	The name used on the fax cover sheet for "Sent By".
SendersVoiceNumber	String	No	The "Sent By" voice number.
SendersFaxNumber	String	No	The "Sent By" fax number.
SendersEmailAddress	String	No	The "Sent By" email address. This address will receive a delivery confirmation.
SendersCompanyName	String	No	The "Sent By" company name.
FaxSensitivityCode	String	No	The sensitivity of the fax (ex: Confidential, Personal).
FaxReferenceNumber	String	No	A field you may want to use for your own tracking.
SaveSentFaxes	String	Yes	"Y" or "N" if "Y" then sent faxes will be saved as tiff files in your FilesAnywhere account.
SaveCoverPageAsDefault	String	Yes	"Y" or "N" if "Y" then the coverage information will be saved for future faxes sent. The following fields will be saved: (SendersName,SendersVoiceNumber,SendersFaxNumber,SendersEmailAddress,SendersCompanyName,CoverPageSubject,RecipientCoverSheet,FaxSensitivityCode,FaxReferenceNumber)
SaveRecipientAsDefault	String	Yes	"Y" or "N" if "Y" then the following fields will be saved: (RecipientNames,RecipientFaxNumbers,RecipientCompanyName,RecipientEmailAddresses,RecipientVoiceNumber,RecipientAlternateFaxNumber) as defaults for your future faxes.
SaveSenderAsDefault	String	Yes	"Y" or "N" if "Y" then the following fields will be saved: (SendersName,SendersVoiceNumber,SendersFaxNumber,SendersEmailAddress,SendersCompanyName) as defaults for your future faxes.
UseACoverPage	String	Yes	"Y" or "N" if "Y" then faxes will have a cover page containing the information provided in the following fields: (SendersName,SendersVoiceNumber,SendersFaxNumber,SendersEmailAddress,SendersCompanyName,CoverPageSubject,RecipientCoverSheet,FaxSensitivityCode,FaxReferenceNumber)
CoverPageSubject	String	No	The information on the Subject line in the fax cover page.

Response

FaxSendResult

FaxSendResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while sending the fax. If the FaxSend call was successful, the ErrorMessage is blank or null.
ProcessID	String	Returns a “;” delimited list of integers which are fax’s processids. Ex: 5535;5536;5537 Or 5535

GetFaxStatus

This function returns the status of sent fax.

Syntax

`FaxSentStatus = GetFaxStatus(Token, FaxStatusParameters)`

Usage

Use the `GetFaxStatus` method to retrieve the status of your last fax sent.

The following parameter is needed to make the `FaxGetDefaults` call:

- `Token` (The Token ID returned by the last `AccountLogin` call).
- `FaxStatusParameters` (The `processId` received from `FaxSend` Api method, need to pass in `FaxStatusParameters` object)

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetFaxStatus"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetFaxStatus xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Parameters>
        <ProcessID>int</ProcessID>
      </Parameters>
    </GetFaxStatus>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetFaxStatusResponse xmlns="http://api.filesanywhere.com/">
      <FaxSentStatus>
        <ErrorMessage>string</ErrorMessage>
        <StatusID>string</StatusID>
        <StatusDescription>string</StatusDescription>
        <FaxPages>string</FaxPages>
        <FaxSavedPath>string</FaxSavedPath>
      </FaxSentStatus>
    </GetFaxStatusResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetFaxStatus xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Parameters>
        <ProcessID>int</ProcessID>
      </Parameters>
    </GetFaxStatus>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetFaxStatusResponse xmlns="http://api.filesanywhere.com/">
      <FaxSentStatus>
        <ErrorMessage>string</ErrorMessage>
        <StatusID>string</StatusID>
```



```
<StatusDescription>string</StatusDescription>
<FaxPages>string</FaxPages>
<FaxSavedPath>string</FaxSavedPath>
</FaxSentStatus>
</GetFaxStatusResponse>
</soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub FaxGetDefaults ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim wsFaxSentStatus As New WebReference.FaxSentStatus
    Dim wsFaxStatusParameters As New WebReference.FaxStatusParameters
    wsFaxStatusParameters.ProcessID = ProcessID;

    wsFaxSentStatus=wsFAWAPI.GetFaxStatus (Token, wsFaxStatusParameters)
    If Not wsFaxDefaultsResult.ErrorMessage <> "" Then
        MsgBox wsFaxDefaultsResult.ErrorMessage
    End If
    wsFaxSentStatus = Nothing
    FAWAPI = Nothing
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
Parameters	FaxStatusParameters	Yes	Assign process id which will receive from FaxSend API method

Response

FaxSentStatus

FaxSentStatus Properties

Name	Data Type	Description
StatusID	Int	Status ID repress Fax status in numeric digit 0 - Failed 1 - Waiting for conversion (Scheduled) 2 - OK (Sent successfully) 3 - Warning
StatusDescription	String	Fax status description
FaxPages	Int	Number of pages of sent fax
FaxSavedPath	String	Path where the sent fax document is saved
ErrorMessage	String	A description of the error, if any error was encountered while retrieving fax status. If the FaxSentStatus call was successful, the ErrorMessage is blank or null.

ViewFile

This method returns back a link using which you can view the file.

Syntax

DownloadFileResult = ViewFile(Token, Path)

Usage

Use the ViewFile method to get a temporary link using which you can view a file.

The following parameters are needed to make the ViewFile call:

- Token (The Token ID returned by the last AccountLogin call).
- Path (Name of the file to be viewed along with the FilesAnywhere Path). The Path should be formatted like `\UserVolume\FolderName\FileName`, where Foldername is the name of the folder in which the file exists and UserVolume is the FilesAnywhere UserName in whose account the file exists.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/ViewFile"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ViewFile xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
    </ViewFile>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ViewFileResponse xmlns="http://api.filesanywhere.com/">
      <DownloadFileResult>
        <ErrorMessage>string</ErrorMessage>
        <URL>string</URL>
      </DownloadFileResult>
    </ViewFileResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ViewFile xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
    </ViewFile>
  </soap12:Body>
</soap12:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ViewFileResponse xmlns="http://api.filesanywhere.com/">
      <DownloadFileResult>
        <ErrorMessage>string</ErrorMessage>
        <URL>string</URL>
      </DownloadFileResult>
    </ViewFileResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```

Private Sub ViewFile ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim DownloadFileResult As New WebReference.DownloadFileResult
    DownloadFileResult = FAWAPI.ViewFile(Token, Path)
    If DownloadFileResult.ErrorMessage <> "" Then
        MsgBox DownloadFileResult.ErrorMessage
    End If
    DownloadFileResult = Nothing
    FAWAPI = Nothing
End Sub

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
Path	String	Yes	Full FilesAnywhere path of the file in the form \UserVolume\FolderName\FileName

Response

DownloadFileResult

DownloadFileResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ViewFile method was executed. If the ViewFile call was successful, the ErrorMessage is blank
URL	String	The URL pointing to the file which is to be viewed.

FileProperties

This function allows the users to basic file properties (date and size) for a given file.

Note: use this method instead of "ItemProperties" if you only need the file size and modified date for a particular file, since this method is slightly more efficient at getting that information than ItemProperties, and it returns the date in GMT.

Syntax

FilePropertiesResult = FileProperties (Token, Path)

Usage

Use the FileProperties method to get the basic modified date and size attributes for a given file and path.

The following parameters are needed to make the FileProperties call:

- Token (The Token ID returned by the last AccountLogin call).
- Path (Name of the file with the FilesAnywhere Path). The Path should be formatted like \UserVolume\FolderName\FileName, where UserVolume is the FilesAnywhere UserName in whose account the item exists.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

POST /FACORPAPI/FAWAPI.asmx HTTP/1.1

Host: api.filesanywhere.com

Content-Type: text/xml; charset=utf-8

Content-Length: length

SOAPAction: "http://api.filesanywhere.com/FileProperties"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <FileProperties xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
    </FileProperties>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <FilePropertiesResponse xmlns="http://api.filesanywhere.com/">
      <ItemsPropertiesResult>
        <FileExists>short</FileExists>
        <ErrorMessage>string</ErrorMessage>
        <Size>string</Size>
        <DateLastModifiedInGMT>string</DateLastModifiedInGMT>
        <DateTimelnTicks>long</DateTimelnTicks>
      </ItemsPropertiesResult>
    </FilePropertiesResponse>
  </soap:Body>
</soap:Envelope>
```


SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

POST /FACORPAPI/FAWAPI.asmx HTTP/1.1

Host: api.filesanywhere.com

Content-Type: application/soap+xml; charset=utf-8

Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <FileProperties xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
    </FileProperties>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <FilePropertiesResponse xmlns="http://api.filesanywhere.com/">
      <ItemsPropertiesResult>
        <FileExists>short</FileExists>
        <ErrorMessage>string</ErrorMessage>
        <Size>string</Size>
        <DateLastModifiedInGMT>string</DateLastModifiedInGMT>
        <DateTimelnTicks>long</DateTimelnTicks>
      </ItemsPropertiesResult>
    </FilePropertiesResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub getFileProperties ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim FilePropertiesResult As New WebReference. FilePropertiesResult
    FilePropertiesResult = FAWAPI.FileProperties(Token, Path)
    If FilePropertiesResult.ErrorMessage <> "" Then
        MsgBox FilePropertiesResult.ErrorMessage
        If FilePropertiesResult.FileExists = 0 then
            MsgBox("File Not Found")
        End if
    End If
    FilePropertiesResult = Nothing
    FAWAPI = Nothing
End Sub
```

Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
Path	String	Yes	Full FilesAnywhere path of the File in the form \UserVolume\FolderName\File Name

Response**FilePropertiesResult**

FilePropertiesResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the adding the new user. If the UserAdd call was successful, the ErrorMessage is blank or null.
FileExists	Integer	Returns -1 if there is an error while retrieving the file properties. Returns 0 if File is not found. Returns 1 if File is found.
Size	String	File size in bytes
ModifiedDateTimeInGMT	String	Returns Date when the file was last modified (Modified Datetime) in GMT Date Format: MM/DD/YYYY HH:MM:SS AM/PM
DateTimeInTicks	Long	Returns File DateTime value as the number of ticks (the number of 100-nanosecond intervals) that have elapsed since 12:00:00 midnight, January 1, 0001.

FileAdvancedProperties

This function allows the users to get advanced file properties (date, size and checksum) for a given file.

Note: use this method instead of "FileProperties" if you need the file checksum with size and modified date for a particular file.

Syntax

```
FileAdvancedPropertiesResult = FileAdvancedProperties(Token,  
FileAdvancedPropertiesParameters)
```

Usage

Use the FileAdvancedProperties method to get the file checksum, modified date and size attributes for a given file and path.

The following parameters are needed to make the FileAdvancedProperties call:

- Token (The Token ID returned by the last AccountLogin call).
- FileAdvancedPropertiesParameters (Pass the name of the file with FilesAnywhere path in this object. The Path should be formatted like \UserVolume\FolderName\File Name, where UserVolume is the FilesAnywhere UserName in whose account the item exists.)

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

POST /FACORPAPI/FAWAPI.asmx HTTP/1.1

Host: api.filesanywhere.com

Content-Type: text/xml; charset=utf-8

Content-Length: length

SOAPAction: "http://api.filesanywhere.com/FileAdvancedProperties"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <FileAdvancedProperties xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Parameters>
        <FilePath>string</FilePath>
      </Parameters>
    </FileAdvancedProperties>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <FileAdvancedPropertiesResponse xmlns="http://api.filesanywhere.com/">
      <FileAdvancedPropertiesResult>
        <Checksum>string</Checksum>
        <Size>long</Size>
        <CreatedDate>string</CreatedDate>
        <ModifiedDate>string</ModifiedDate>
        <ErrorMessage>string</ErrorMessage>
      </FileAdvancedPropertiesResult>
    </FileAdvancedPropertiesResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

POST /FACORPAPI/FAWAPI.asmx HTTP/1.1

Host: api.filesanywhere.com

Content-Type: application/soap+xml; charset=utf-8

Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <FileAdvancedProperties xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Parameters>
        <FilePath>string</FilePath>
      </Parameters>
    </FileAdvancedProperties>
  </soap12:Body>
</soap12:Envelope>
```

HTTP/1.1 200 OK

Content-Type: application/soap+xml; charset=utf-8

Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <FileAdvancedPropertiesResponse xmlns="http://api.filesanywhere.com/">
      <FileAdvancedPropertiesResult>
        <Checksum>string</Checksum>
        <Size>long</Size>
        <CreatedDate>string</CreatedDate>
        <ModifiedDate>string</ModifiedDate>
        <ErrorMessage>string</ErrorMessage>
      </FileAdvancedPropertiesResult>
    </FileAdvancedPropertiesResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub FileAdvancedProperties ()

    Dim FAWAPI As New WebReference.FAWAPI
    Dim objAdvancedPropertiesResult As New WebReference.
    FileAdvancedPropertiesResult

    Dim wsFilePropertiesParameters As New
    WebReference.FileAdvancedPropertiesParameters

    wsFilePropertiesParameters.FilePath = FilePath
    `Filesanywhere file path

    objAdvancedPropertiesResult = FAWAPI.FileAdvancedProperties (Token,
    wsFilePropertiesParameters)

    If objAdvancedPropertiesResult.ErrorMessage <> "" Then
        MsgBox objAdvancedPropertiesResult.ErrorMessage
    End If
    objAdvancedPropertiesResult = Nothing
    FAWAPI = Nothing
End Sub
```

Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
Parameters	FileAdvancedPropertiesParameters	Yes	FilePath :- Full FilesAnywhere path of the File in the form \UserVolume\FolderName\File Name

Response

FileAdvancedPropertiesResult

FileAdvancedPropertiesResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while calling the FileAdvancedProperties method. If FileAdvancedProperties call was successful, then ErrorMessage is blank or null.
Checksum	String	Checksum is a calculated value that is used to determine the integrity of data. Checksum serves as a unique identifier for the data (a file, a text string, or a hexadecimal string). If the data changes then so does the checksum value. This makes it easy to verify the integrity of the data.
Size	String	File size in bytes
ModifiedDate	String	Returns Date when the file was last modified (Modified Datetime) in GMT Date Format: MM/DD/YYYY HH:MM:SS AM/PM
CreatedDate	String	Returns Date when the file was created in GMT

Note: For the large file size (more than 1.5 GB), sometime system may take longer time to calculate the file checksum. While system is still calculating the checksum, you may get message “20069005: Please wait while we are calculating the checksum, or try after some time.”

In such case please call FileAdvancedProperties API method again after some time to get the actual checksum details.

ConvertImage

This function converts a saved image, to a new scale and/or rotation, and saves it to the path specified. The function returns a processid which can be queried to get the process status, since the scaling process is performed asynchronously on the server. The new image's dimensions are proportionate to the dimensions of the original image.

Note: ConvertImage only works with JPEG, GIF, PNG, BMP or WMF files. Please use one of these image types, or you can request that we support additional image types.

Image conversion for the entire folder is not supported presently but can be added in future.

Syntax

ConvertImageResult = ConvertImage (Token, Type, ImageSourcePath, ImageDestinationPath, NewImageWidth, NewImageHeight, ZoomPercent, RotateDegree)

Usage

Users can now scale or rotate an existing image and save it to the path specified. If the path passed is blank then the existing image file is overwritten by the new image. The users have to pass new image dimensions (width and height) or zoom percentage and/or rotation degree to scale and/or rotate the image. If the user passes both zoom percentage and new dimensions then the new dimensions are ignored and the image is scaled as per the zoom percentage.

The following parameters are needed to make the ConvertImage call:

- **Token.** (The Token ID returned by the last AccountLogin call).
- **Type.** This is the Item type and must be a 'file'. Image conversion for the entire folder is not supported presently but can be added in future.
- **ImageSourcePath.** (Name of the existing image file to be scaled/rotate along with the FilesAnywhere Path). The Path should be formatted like \UserVolume\FolderName\FileName, where Foldername is the name of the folder in which the file resides and UserVolume is the FilesAnywhere UserName in whose account the file resides. The image file type must be of JPEG, GIF, PNG, BMP or WMF file. Users can request that we support additional image types.
- **ImageDestinationPath.** (Name of the new image file along with the FilesAnywhere Path). The Path should be formatted like \UserVolume\FolderName\FileName, where Foldername is the name of the folder in which the file will be saved and UserVolume is the FilesAnywhere

UserName in whose account the file will be saved. If the folder doesn't exist, then it is created.

- **NewImageWidth.** This is the new width in pixels to which the image is scaled.
- **NewImageHeight.** This is the new height in pixels to which the image is scaled.
- **ZoomPercent** is the percentage of the original image size to which the image is scaled. **NewImageWidth** and **NewImageHeight** are ignored if the **ZoomPercentage** passed is not equal to zero. If the zoom percentage is set to 100 then the new image will have same dimensions as an original image. Zoom percentage value must be between 0 to 1000 (whole number with no symbols)
- **RotateDegree** is the degree by which the original image will be rotated. The rotate degree can only be one of these four values: 0, 90, 180 or 270. If the zoom percentage and the new dimension values are 0, then the image is only rotated and not scaled.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

POST /FACORPAPI/FAWAPI.asmx HTTP/1.1

Host: localhost

Content-Type: text/xml; charset=utf-8

Content-Length: length

SOAPAction: "http://api.filesanywhere.com/ConvertImage"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ConvertImage xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Type>string</Type>
      <ImageSourcePath>string</ImageSourcePath>
      <ImageDestinationPath>string</ImageDestinationPath>
      <NewImageWidth>int</NewImageWidth>
      <NewImageHeight>int</NewImageHeight>
      <ZoomPercent>int</ZoomPercent>
      <RotateDegree>int</RotateDegree>
    </ConvertImage>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ConvertImageResponse xmlns="http://api.filesanywhere.com/">
      <ConvertImageResult>
        <ErrorMessage>string</ErrorMessage>
        <ProcessID>int</ProcessID>
      </ConvertImageResult>
    </ConvertImageResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ConvertImage xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Type>string</Type>
      <ImageSourcePath>string</ImageSourcePath>
      <ImageDestinationPath>string</ImageDestinationPath>
      <NewImageWidth>int</NewImageWidth>
      <NewImageHeight>int</NewImageHeight>
      <ZoomPercent>int</ZoomPercent>
      <RotateDegree>int</RotateDegree>
    </ConvertImage>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ConvertImageResponse xmlns="http://api.filesanywhere.com/">
```

```
<ConvertImageResult>  
  <ErrorMessage>string</ErrorMessage>  
  <ProcessID>int</ProcessID>  
</ConvertImageResult>  
</ConvertImageResponse>  
</soap12:Body>  
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub ConvertImage ()  
    Dim FAWAPI As New WebReference.FAWAPI  
    Dim wsConvertImageResult As New WebReference.ConvertImageResult  
    wsConvertImageResult =  
        wsFAWAPI.ConvertImage(Token, Type, ImageSourcePath,  
        ImageDestinationPath, NewImageWidth, NewImageHeight, ZoomPercent,  
        RotateDegree)  
    If Not wsConvertImageResult.ErrorMessage  
        <> "" Then  
        MsgBox wsConvertImageResult.ErrorMessage  
    End If  
    Dim ProcessID As integer = wsConvertImageResult.ProcessID  
    wsConvertImageResult = Nothing  
    FAWAPI = Nothing  
End Sub
```

Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call
Type	String	Yes	Type must be a 'file'.
ImageSourcePath	String	Yes	Full FilesAnywhere path of the Item in the form \UserVolume\FolderName\ItemName
ImageDestinationPath	String	No	Full FilesAnywhere path of the Item in the form \UserVolume\FolderName\ItemName. If no destination path is passed the original image will be overwritten by the new image.
NewImageWidth	integer	Yes	New width in pixels to which the image is scaled
NewImageHeight	integer	Yes	New height in pixels to which the image is scaled
ZoomPercent	integer	Yes	Percentage of the original image size to which the image is scaled. Zoom percentage value must be between 0 to 1000 (whole number with no symbols)
RotateDegree	integer	Yes	Degree by which the original image will be rotated. The rotate degree can only be one of these four values: 0, 90, 180 or 270.

Response**ConvertImageResult**

ConvertImageResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the submitting the request for image conversion. If the ConvertImage call was successful, the ErrorMessage is blank or null.
ProcessID	integer	Returns process ID in integer which can be queried to get the process status, since the scaling process is performed asynchronously on the server

ConvertImageStatus

The function checks the status of the conversion. This function returns a string: Y = conversion in process, P = pending/queued, C = conversion complete, E = Error.

Note: Image conversion for the entire folder is not supported presently but can be added in future.

Syntax

ConvertImageStatusResult = ConvertImageStatus (Token, ProcessID)

Usage

Users can check the status of the image conversion request submitted through ConvertImage method, by passing the Process ID returned by ConvertImage method.

The following parameters are needed to make the ConvertImage call:

- Token. (The Token ID returned by the last AccountLogin call).
- ProcessID. This is the ID returned by the ConvertImage method by which the user can check the status of the image conversion process.

Status 'P' indicates that the request is in queue and waiting to get processed. Status 'Y' indicates that the request is picked up for processing and is in process. Status 'C' indicates that the request is processed and the image is converted and saved successfully to the new path. Status 'E' indicates that there was some error while processing the request.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

POST /FACORPAPI/FAWAPI.asmx HTTP/1.1

Host: api.filesanywhere.com

Content-Type: text/xml; charset=utf-8

Content-Length: length

SOAPAction: "http://api.filesanywhere.com/ConvertImageStatus"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ConvertImageStatus xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ProcessID>int</ProcessID>
    </ConvertImageStatus>
  </soap:Body>
</soap:Envelope>
```

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ConvertImageStatusResponse xmlns="http://api.filesanywhere.com/">
      <ConvertImageStatusResult>
        <ErrorMessage>string</ErrorMessage>
        <StatusCode>string</StatusCode>
      </ConvertImageStatusResult>
    </ConvertImageStatusResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

POST /FACORPAPI/FAWAPI.asmx HTTP/1.1

Host: api.filesanywhere.com

Content-Type: application/soap+xml; charset=utf-8

Content-Length: length


```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ConvertImageStatus xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ProcessID>int</ProcessID>
    </ConvertImageStatus>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ConvertImageStatusResponse xmlns="http://api.filesanywhere.com/">
      <ConvertImageStatusResult>
        <ErrorMessage>string</ErrorMessage>
        <StatusCode>string</StatusCode>
      </ConvertImageStatusResult>
    </ConvertImageStatusResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub getConvertImageStatus ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim wsConvertImageStatusResult As New
WebReference.ConvertImageStatusResult
    wsConvertImageResult =
    wsFAWAPI.ConvertImageStatus(Token, ProcessID)
    If Not wsConvertImageStatusResult.ErrorMessage
    <> "" Then
        MsgBox wsConvertImageStatusResult.ErrorMessage
    End If
    Dim StatusCode As String = wsConvertImageStatusResult.StatusCode
    wsConvertImageResult = Nothing
    FAWAPI = Nothing
End Sub
```

Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call
ProcessID	Integer	Yes	ID returned by ConvertImage call. This is used to check the status of the image conversion process.

Response**ConvertImageStatusResult**

ConvertImageStatusResult Properties

Name	Data Type	Description
ErrorMessage	String	<p>A description of the error, if any error was encountered while the checking the status of the image conversion process.</p> <p>If the ConvertImageStatus call was successful, the ErrorMessage is blank or null.</p>
StatusCode	String	<p>Code that indicates the progress of the image conversion process.</p> <p>Status 'P' indicates that the request is in queue and waiting to get processed. Status 'Y' indicates that the request is picked up for processing and is in process. Status 'C' indicates that the request is processed and the image is converted and saved successfully to the new path. Status 'E' indicates that there was some error while processing the request.</p>

GetMediaStream

This function returns a streaming URL for a media file. If the file is not ready for streaming then the function submits the job for streaming conversion and provides the status of the submitted job.

Syntax

```
GetMediaStreamResult = GetMediaStream(Token, FilePath, MediaType,  
ResolutionWidth, ProcessId)
```

Usage

User can now stream media files (Audio and Video) per the requested media type. Only video files are converted in to HLS (HTTP Live Streaming) format and this method returns a URL for m3u8 playlist that contains HLS file information. In addition to media files, if the File path provided is for a Document/Image then the API method will return the URL to access the file.

The following parameters are needed for calling GetMediaStream:

- **Token.** The Token ID returned by the last AccountLogin call
- **FilePath.** The media file to be converted into a streaming source. The media file path should be formatted like `\UserVolume\FolderName\FileName` where UserVolume is the FilesAnywhere UserName in which account the media file is present.
- **MediaType.** The MediaTypes supported by GetMediaStream are “webm” and “mp4”. MediaType “mp3” is ignored. If Media type is not provided to GetMediaStream method then “mp4” is the default.
- **ResolutionWidth.** If ResolutionWidth is not provided then the actual resolution of the video will be displayed. To use the actual resolution, submit 0 for this parameter.

Valid ResolutionWidth values are: 0 (for actual size), 320, 400, 480, 640, 854, and 1280.

- **ProcessID.** This is the ID returned by the GetMediaStream method when the job is submitted. Users can check the status of a streaming media conversion process by referencing the ProcessID. For the first call to this method ProcessID is not required.

Input File Types supported by GetMediaStream Method: Method supports below mentioned file types and provides a streaming URL as output:

Video files:

avi,mpeg,mpg,qt,dv,aac,aif,aiff,mov,mp4,3gz,3gp,3g2,m2v,wmv,asf,wma,m4v,wav,flv,m4v,webm,mpeg-1,asf,nsf,seq,mxf,cin,gxf,avs,wve,uv2,fli,flc,cin,wc3,mpeg4,vob,mpeg-4,dvb

Document files:

doc,ppt,xls,pdf,vsd,txt,rtf,wri,pub,dot,xlt,wq1,wks,wk3,wk4,docx,dotx,xlsx,xltx,xlsb,pptx,ppsx,docm,dotm,xlsm,xl tm,pptm,ppam,ppsm,lwp,ans,htm,html,wk5,wki,wku,wpd,psw,pwd,pab,olk,mcw,xlw,pps,log,dxf,svg,vdw,vdx,vs s,vst,vsx,vtx,slk,odt,mpp,mpt,vsdm,vsdx,vssm,vssx,vstm,vstx

Image files: jpg,jpeg,jpe,jfif,gif,png,bmp,wmf,dib,tif,tiff,emf

Music files: mp3,m4a

Autocad files: dwf,dwfx

Flash files: swf

SOAP 1.1

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetMediaStream"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetMediaStream xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <FilePath>string</FilePath>
      <MediaType>string</MediaType>
      <ResolutionWidth>int</ResolutionWidth>
      <ProcessId>int</ProcessId>
    </GetMediaStream>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetMediaStreamResponse xmlns="http://api.filesanywhere.com/">
      <GetMediaStreamResult>
        <ErrorMessage>string</ErrorMessage>
```

```

    <ProcessId>string</ProcessId>
    <Url>string</Url>
    <StatusCode>string</StatusCode>
    <StatusText>string</StatusText>
    <MediaServer>string</MediaServer>
    <WebUrl>string</WebUrl>
    <FileType>string</FileType>
  </GetMediaStreamResult>
</GetMediaStreamResponse>
</soap:Body>
</soap:Envelope>

```

SOAP 1.2

```

POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetMediaStream xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <FilePath>string</FilePath>
      <MediaType>string</MediaType>
      <ResolutionWidth>int</ResolutionWidth>
      <ProcessId>int</ProcessId>
    </GetMediaStream>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetMediaStreamResponse xmlns="http://api.filesanywhere.com/">
      <GetMediaStreamResult>
        <ErrorMessage>string</ErrorMessage>
        <ProcessId>string</ProcessId>
        <Url>string</Url>
        <StatusCode>string</StatusCode>
        <StatusText>string</StatusText>
        <MediaServer>string</MediaServer>
        <WebUrl>string</WebUrl>
        <FileType>string</FileType>
      </GetMediaStreamResult>
    </GetMediaStreamResponse>
  </soap12:Body>
</soap12:Envelope>

```

```
</GetMediaStreamResponse>
</soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub getMediaStream()
    Dim FAWAPI As New WebReference.FAWAPI()
    Dim wsGetMediaStreamResult As
webAPITest.WebReference.GetMediaStreamResult =
FAWAPI.GetMediaStream(Token, FilePath, MediaType, ResolutionWidth,
ProcessId)
    If Not wsGetMediaStreamResult.ErrorMessage <> String.Empty Then
        Response.Write("FileType:= " & wsGetMediaStreamResult.FileType
& "<br>")
        Response.Write("ProcessId:= " &
wsGetMediaStreamResult.ProcessId & "<br>")
        Response.Write("URL:= " & wsGetMediaStreamResult.Url & "<br>")
        Response.Write("StatusCode:= " &
wsGetMediaStreamResult.StatusCode & "<br>")
        Response.Write("StatusText:= " &
wsGetMediaStreamResult.StatusText & "<br>")
        Response.Write("MediaServer:= " &
wsGetMediaStreamResult.MediaServer & "<br>")
        Response.Write("WebURL:= " & wsGetMediaStreamResult.WebUrl &
"<br>")
        Response.Write("Error:= " &
wsGetMediaStreamResult.ErrorMessage & "<br>")
    End If
    wsGetMediaStreamResult = Nothing
    FAWAPI = Nothing
End Sub
```

Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call
FilePath	String	Yes	The full FilesAnywhere path of the media file in the form \UserVolume\FolderName\ItemName
MediaType	String	No	Type of the media file expected after streaming. If not provided default media type considered as "mp4"
ResolutionWidth	Int	No	Width in pixels to which video media to be streamed. Ignored for music files. Default value for int is 0.

ProcessId	Int	No	ID returned by GetMediaStream call. This is used to check the status of the conversion process with next GetMediaStream calls. Default value for int is 0.
-----------	-----	----	--

Response

GetMediaStreamResult

GetMediaStreamResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the submitting the request for media streaming. If the GetMediaStream call was successful, the ErrorMessage will be blank or null.
ProcessID	String	Returns process ID in string which can be queried to get the streaming status, since the streaming process is performed asynchronously on the server.
FileType	String	Media file type available after streaming.
Url	String	FilesAnywhere Url path which indicates streaming is completed.
StatusCode	String	Status code of media streaming process. Status 'P' indicates that the request is in queue and waiting to get processed. Status 'Y' indicates that the request is picked up for processing and is in process. Status 'C' indicates that the request is processed and the conversion completed successfully. Status 'E' indicates that there was some error while processing the request.
StatusText	String	Status of media streaming process.

MediaServer	String	Server name used for Wouwza server.
WebUrl	String	Part of Url which can be used to get complete Url.

Information about GetMediaStream output usage

If “Url” returned by GetMediaStream method is empty string this means either a new job is submitted or job is in progress. Until “Url” is not populated by the GetMediaStream method, GetMediaStream have to be called with the last returned ProcessID.

Once “Url” is populated by GetMediaStream method, media can be used with HTML5 or compatible HLS player.

GetMediaStream response can be verify using below sample steps:

- Using Music files with HTML5:
 - a) Put GetMediaStream response value of WebURL and URL into below tag:

```
<audio controls>
<source src="[[WEBURL]]/[[URL]" type="audio/mp3">
</audio>
```
 - b) Put Step#a updated tag into an HTML file.
 - c) Open the HTML file into any HTML5 supported browser.

- Using Video files with HLS Video Player:
 - a) Build video url from GetMediaStream response value of WebURL and URL as shown below: and play the video in any HLS compatible video player using below url

```
[[WEBURL]]/[[URL]
```
 - b) Play URL from Step#a in any HLS compatible video player.

GetImageStream

This function returns a streamed image for the image location specified. Image types supported are jpg, jpeg, jpe, jfif, gif, png, bmp, wmf, dib, tif, tiff, emf.

Syntax

`GetImageStreamResult = GetImageStream(Token, ImagePath, SizeType, Width, Height, Zoom, RotateDegree, PageNum)`

Usage

User can now stream image file per the requested image path.

SizeType information.

View	Code	Width	Height	Description
Icon	I	76	76	Icon View
List	L	32	32	List View
Hover	H	400	300	Hover View
Custom	C	As specified at method GetImageStream	As specified at method GetImageStream	Custom View

The following parameters are needed for calling GetImageStream:

- **Token.** The Token ID returned by the last AccountLogin call
- **ImagePath.** The image file to be converted into a streamed image. The image file path should be formatted like `\UserVolume\FolderName\File Name` where UserVolume is the FilesAnywhere UserName in which account the image file is present.
- **SizeType.** The SizeTypes supported by GetImageStream are "I", "L", "H" and "C". If SizeType is not provided to method then "L" is the default.
- **Width.** Applicable only if SizeType View is Custom, Otherwise ignored. Specify 0 for default.
- **Height.** Applicable only if SizeType View is Custom, Otherwise ignored. Specify 0 for default.
- **Zoom.** Applicable only if SizeType View is Custom, Otherwise ignored. Specify 0 for default. Zoom value is in percentage.

- **RotateDegree.** Applicable only if SizeType View is Custom ,Otherwise ignored. Valid degrees for rotation are 90, 180, 270. Specify 0 for default.
- **PageNum.** Applicable for tif & tiff images only. Specify 0 for default.

SOAP 1.1

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetImageStream"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetImageStream xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ImagePath>string</ImagePath>
      <SizeType>string</SizeType>
      <Width>int</Width>
      <Height>int</Height>
      <Zoom>int</Zoom>
      <RotateDegree>int</RotateDegree>
      <PageNum>int</PageNum>
    </GetImageStream>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetImageStreamResponse xmlns="http://api.filesanywhere.com/">
      <GetImageResult>
        <ErrorMessage>string</ErrorMessage>
        <ImageBytes>base64Binary</ImageBytes>
      </GetImageResult>
    </GetImageStreamResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetImageStream xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ImagePath>string</ImagePath>
      <SizeType>string</SizeType>
      <Width>int</Width>
      <Height>int</Height>
      <Zoom>int</Zoom>
      <RotateDegree>int</RotateDegree>
      <PageNum>int</PageNum>
    </GetImageStream>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetImageStreamResponse xmlns="http://api.filesanywhere.com/">
      <GetImageResult>
        <ErrorMessage>string</ErrorMessage>
        <ImageBytes>base64Binary</ImageBytes>
      </GetImageResult>
    </GetImageStreamResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub getImageStream()
    Dim FAWAPI As New WebReference.FAWAPI()
```

```

Dim wsGetImageStreamResult As
webAPITest.WebReference.GetImageStreamResult =
FAWAPI.GetMediaStream(Token, ImagePath, SizeType, Width, Height, Zoom,
RotateDegree)
If Not wsGetImageStreamResult.ErrorMessage <> String.Empty Then
    Response.ContentType = "image/jpeg"
    Try
        Response.BinaryWrite(wsGetImageResult.ImageBytes)
    Catch ex As Exception
    End Try
    Response.End()
End If
wsGetImageStreamResult = Nothing
FAWAPI = Nothing
End Sub

```

Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call
ImagePath	String	Yes	The full FilesAnywhere path of the image file in the form \UserVolume\FolderName\ItemName
SizeType	String	No	SizeType of the image file expected after streaming. If not provided default image type considered as "L"
Width	Integer	No	Width in pixels to which image file to be streamed. Ignored if SizeType is not "C". Default value for Width is 32.
Height	Integer	No	Height in pixels to which image file to be streamed. Ignored if SizeType is not "C". Default value for Width is 32.
Zoom	Integer	No	Image zoom expected for image in percentage. Ignored if SizeType is not "C".
RotateDegree	Integer	No	Rotate image by degree provided, possible values 90, 180, 270. Ignored if SizeType is not "C".
PageNum	integer	No	Image number to be strimmed, applicable only if image type is tif or tiff. Default is 1.

GetImageStreamResult Properties

Name	Data Type	Description
------	-----------	-------------

ErrorMessage	String	A description of the error, if any error was encountered while submitting the request for image streaming. If the GetImageStream call was successful, the ErrorMessage will be blank or null.
ImageBytes	Byte()	Returns Byte() on successful image streaming of imagepath provided.

GetTIFFPageCount

This function returns count of pages present at the TIFF image location specified. Image types supported are tif, tiff.

Syntax

GetTIFFPageCountResult = GetTIFFPageCount (Token, ImagePath)

Usage

User can receive page count for the tif or tiff image path.

The following parameters are needed for calling GetTIFFPageCount:

- **Token.** The Token ID returned by the last AccountLogin call.
- **ImagePath.** The image file path for which page count is required. The image file path should be formatted like `\UserVolume\FolderName\FileName` where `UserVolume` is the FilesAnywhere Username in which account the image file is present.

SOAP 1.1

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetTIFFPageCount"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetTIFFPageCount xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ImagePath>string</ImagePath>
    </GetTIFFPageCount>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetTIFFPageCountResponse xmlns="http://api.filesanywhere.com/">
      <GetTIFFPageCountResult>
        <ErrorMessage>string</ErrorMessage>
        <PageCount>int</PageCount>
      </GetTIFFPageCountResult>
    </GetTIFFPageCountResponse>
  </soap:Body>
</soap:Envelope>
```



```
</GetTIFFPageCountResult>
</GetTIFFPageCountResponse>
</soap:Body>
</soap:Envelope>
```

SOAP 1.2

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetTIFFPageCount xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ImagePath>string</ImagePath>
    </GetTIFFPageCount>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetTIFFPageCountResponse xmlns="http://api.filesanywhere.com/">
      <GetTIFFPageCountResult>
        <ErrorMessage>string</ErrorMessage>
        <PageCount>int</PageCount>
      </GetTIFFPageCountResult>
    </GetTIFFPageCountResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub getGetTIFFPageCount ()
    Dim FAWAPI As New WebReference.FAWAPI ()
    Dim wsGetTIFFPageCountResult As
        WebReference.GetTIFFPageCountResult =
            FAWAPI.GetTIFFPageCount (Token, ImagePath)
    If Not wsGetTIFFPageCountResult.ErrorMessage <> String.Empty Then
```

```

        Response.Write("TIFF page count is: " &
        wsGetTIFFPageCountResult.PageCount)
    End If
    wsGetTIFFPageCountResult = Nothing
    FAWAPI = Nothing
End Sub

```

Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call
ImagePath	String	Yes	The full FilesAnywhere path of the image file in the form \UserVolume\FolderName\ItemName

GetTIFFPageCountResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while submitting the request. If the GetTIFFPageCount call was successful, the ErrorMessage will be blank or null.
PageCount	Integer	Returns number of pages found in tiff or tif imagepath provided.

FAWAPIv2 Methods

APNEchoNotification

Syntax

APNEchoNotificationResult = APNEchoNotification(Token,Message)

Usage

Use the APNEchoNotification method to test APNS on a registered device.

The following parameters are needed to make the APNEchoNotification call:

- **Token.** (The Token ID returned by the last AccountLogin call).
- **Message .** (The test message to send to the device currently registered for the logged in user.)

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/APNEchoNotification"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <APNEchoNotification xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Message>string</Message>
    </APNEchoNotification>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <APNEchoNotificationResponse xmlns="http://api.filesanywhere.com/">
```

```
<APNEchoResult>
  <ErrorMessage>string</ErrorMessage>
  <NotificationId>guid</NotificationId>
</APNEchoResult>
</APNEchoNotificationResponse>
</soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <APNEchoNotification xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Message>string</Message>
    </APNEchoNotification>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <APNEchoNotificationResponse xmlns="http://api.filesanywhere.com/">
      <APNEchoResult>
```

```

    <ErrorMessage>string</ErrorMessage>
    <NotificationId>guid</NotificationId>
  </APNEchoResult>
</APNEchoNotificationResponse>
</soap12:Body>
</soap12:Envelope>

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
Message	String	Yes	The test message to send to the device currently registered for the logged in user.

Response

APNEchoNotificationResult

APNEchoNotificationResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the APNEchoNotification method was executed. If the APNEchoNotification call was successful, the ErrorMessage is blank
NotificationId	GUID	The identifier for the notification sent. This can be used in conjunction with APNGetNotificationProperties to determine the status of the notification.

APNGetNotificationProperties

Syntax

APNGetNotificationPropertiesResult =
APNGetNotificationProperties(Token,NotificationId)

Usage

Use the APNGetNotificationProperties method to fetch the status and other properties of an APNS notification.

The following parameters are needed to make the APNRegisterDevice call:

- Token. (The Token ID returned by the last AccountLogin call).
- NotificationId. (The identifier for the notification to retrieve properties for).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/APNGetNotificationProperties"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <APNGetNotificationProperties xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <NotificationId>guid</NotificationId>
    </APNGetNotificationProperties>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
```

```
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <APNGetNotificationPropertiesResponse
xmlns="http://api.filesanywhere.com/">
      <APNSendNotificationResult>
        <ErrorMessage>string</ErrorMessage>
        <APNNotificationId>guid</APNNotificationId>
        <NotificationDate>dateTime</NotificationDate>
        <NotificationPostDate>dateTime</NotificationPostDate>
        <Attempts>int</Attempts>
        <NotificationStatus>string</NotificationStatus>
      </APNSendNotificationResult>
    </APNGetNotificationPropertiesResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <APNGetNotificationProperties xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <NotificationId>guid</NotificationId>
    </APNGetNotificationProperties>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <APNGetNotificationPropertiesResponse
xmlns="http://api.filesanywhere.com/">
      <APNSendNotificationResult>
        <ErrorMessage>string</ErrorMessage>
        <APNNotificationId>guid</APNNotificationId>
        <NotificationDate>dateTime</NotificationDate>
        <NotificationPostDate>dateTime</NotificationPostDate>
        <Attempts>int</Attempts>
        <NotificationStatus>string</NotificationStatus>
      </APNSendNotificationResult>
    </APNGetNotificationPropertiesResponse>
  </soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
NotificationId	String	Yes	The identifier for the notification to retrieve properties for.

Response

APNGetNotificationPropertiesResult

APNGetNotificationPropertiesResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the APNGetNotificationProperties method was executed. If the APNGetNotificationProperties

		call was successful, the ErrorMessage is blank
APNNotificationId	GUID	The identifier for the notification.
NotificationDate	DateTime	The datetime stamp when the notification was generated.
NotificationPostDate	DateTime	The datetime stamp when the notification was sent to APNS services.
Attempts	Integer	The number of attempts to send the notification.
NotificationStatus	String	The status of the notification.

APNRegisterDevice

Syntax

APNRegisterDeviceResult = APNRegisterDevice(Token,DeviceId,DeviceToken)

Usage

Use the APNRegisterDevice method to register an Apple iOS device for push notification services.

The following parameters are needed to make the APNRegisterDevice call:

- Token. (The Token ID returned by the last AccountLogin call).
- DeviceId. (The iOS hardware device id)
- DeviceToken. (The deviceToken returned in didRegisterForRemoteNotificationsWithDeviceToken).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/APNRegisterDevice"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <APNRegisterDevice xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <DeviceId>string</DeviceId>
      <DeviceToken>string</DeviceToken>
    </APNRegisterDevice>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <APNRegisterDeviceResponse xmlns="http://api.filesanywhere.com/">
      <APNRegisterDeviceResult>
        <ErrorMessage>string</ErrorMessage>
      </APNRegisterDeviceResult>
    </APNRegisterDeviceResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <APNRegisterDevice xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <DeviceId>string</DeviceId>
      <DeviceToken>string</DeviceToken>
    </APNRegisterDevice>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <APNRegisterDeviceResponse xmlns="http://api.filesanywhere.com/">
```

```

    <APNRegisterDeviceResult>
      <ErrorMessage>string</ErrorMessage>
    </APNRegisterDeviceResult>
  </APNRegisterDeviceResponse>
</soap12:Body>
</soap12:Envelope>

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
DeviceId	String	Yes	The hardware device ID of the iOS device.
DeviceToken	String	Yes	The deviceToken returned in didRegisterForRemoteNotificationsWithDeviceToken.

Response

APNRegisterDeviceResult

APNRegisterDeviceResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the APNRegisterDevice method was executed. If the APNRegisterDevice call was successful, the ErrorMessage is blank

APNSendNotification

Syntax

APNSendNotificationResult = APNSendNotification(Token,NotificationType)

Usage

Use the APNSendNotification method to send an Apple push notification to an iOS device. This is for testing purposes only and will only send notifications to the device registered for the provided token, using the default values for substitution parameters.

The following parameters are needed to make the APNSendNotification call:

- Token. (The Token ID returned by the last AccountLogin call).
- NotificationType. (The type of notification to be sent to the current iOS device).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/APNSendNotification"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <APNSendNotification xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <NotificationType>FILEACTION_ANNOTATE or FILEACTION_DELETED or
FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANGE or
GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or
```

```
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or ACCOUNTACTION_STORAGEEXCEEDED or
NONE</NotificationType>
</APNSendNotification>
</soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <APNSendNotificationResponse xmlns="http://api.filesanywhere.com/">
      <APNSendNotificationResult>
        <ErrorMessage>string</ErrorMessage>
        <NotificationId>guid</NotificationId>
      </APNSendNotificationResult>
    </APNSendNotificationResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <APNSendNotification xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <NotificationType>FILEACTION_ANNOTATE or FILEACTION_DELETED or
FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANG
```

```
GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or ACCOUNTACTION_STORAGEEXCEEDED or
NONE</NotificationType>
  </APNSendNotification>
</soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <APNSendNotificationResponse xmlns="http://api.filesanywhere.com/">
      <APNSendNotificationResult>
        <ErrorMessage>string</ErrorMessage>
        <NotificationId>guid</NotificationId>
      </APNSendNotificationResult>
    </APNSendNotificationResponse>
  </soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
NotificationType	Enum	Yes	The type of notification to be sent.

Response

APNSendNotificationResult

APNSendNotificationResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the APNSendNotification method was executed. If the APNSendNotification call was successful, the ErrorMessage is blank
NotificationId	Guid	The identifier for the notification message.

APNUnregisterDevice

Syntax

APNUnregisterDeviceResult = APNUnregisterDevice(Token,DeviceId,DeviceToken)

Usage

Use the APNUnregisterDevice method to unregister an Apple iOS device for push notification services.

The following parameters are needed to make the APNRegisterDevice call:

- Token. (The Token ID returned by the last AccountLogin call).
- DeviceId. (The iOS hardware device id)
- DeviceToken. (The deviceToken returned in didRegisterForRemoteNotificationsWithDeviceToken).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/APNUnregisterDevice"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <APNUnregisterDevice xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <DeviceId>string</DeviceId>
      <DeviceToken>string</DeviceToken>
    </APNUnregisterDevice>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
```

```
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <APNUnregisterDeviceResponse xmlns="http://api.filesanywhere.com/">
      <APNUnregisterDeviceResult>
        <ErrorMessage>string</ErrorMessage>
      </APNUnregisterDeviceResult>
    </APNUnregisterDeviceResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <APNUnregisterDevice xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <DeviceId>string</DeviceId>
      <DeviceToken>string</DeviceToken>
    </APNUnregisterDevice>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <APNUnregisterDeviceResponse xmlns="http://api.filesanywhere.com/">
      <APNUnregisterDeviceResult>
```

```
<ErrorMessage>string</ErrorMessage>
</APNUnregisterDeviceResult>
</APNUnregisterDeviceResponse>
</soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
DeviceId	String	Yes	The hardware device ID of the iOS device.
DeviceToken	String	Yes	The deviceToken returned in didRegisterForRemoteNotificationsWithDeviceToken.

Response

APNUnregisterDeviceResult

APNUnregisterDeviceResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the APNUnregisterDevice method was executed. If the APNUnregisterDevice call was successful, the ErrorMessage is blank

APNUpdateRegistration

Syntax

APNUpdateRegistrationResult = APNUpdateRegistration(Token,Badges)

Usage

Use the APNUpdateRegistration method to update the number of unread notifications on the current Apple device.

The following parameters are needed to make the APNUpdateRegistration call:

- Token. (The Token ID returned by the last AccountLogin call).
- Badges. (The number of unread notifications)

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/APNUpdateRegistration"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <APNUpdateRegistration xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Badges>int</Badges>
    </APNUpdateRegistration>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <APNUpdateRegistrationResponse xmlns="http://api.filesanywhere.com/">
      <APNUpdateRegistration>
        <ErrorMessage>string</ErrorMessage>
      </APNUpdateRegistration>
    </APNUpdateRegistrationResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <APNUpdateRegistration xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Badges>int</Badges>
    </APNUpdateRegistration>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <APNUpdateRegistrationResponse xmlns="http://api.filesanywhere.com/">
      <APNUpdateRegistration>
```

```
<ErrorMessage>string</ErrorMessage>
</APNUpdateRegistration>
</APNUpdateRegistrationResponse>
</soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
Badges	Integer	Yes	The number of unread notifications

Response

APNUnregisterDeviceResult

APNUnregisterDeviceResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the APNUnregisterDevice method was executed. If the APNUnregisterDevice call was successful, the ErrorMessage is blank

DeleteMeeting

Syntax

DeleteMeetingResult= DeleteMeeting(Token,Meeting)

Usage

Use the DeleteMeeting method to cancel a scheduled meeting.

The following parameters are needed to make the DeleteMeeting call:

- Token. (The Token ID returned by the last AccountLogin call).
- Meeting. (A populated meeting structure).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/DeleteMeeting"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <DeleteMeeting xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Meeting>
        <MeetingId>long</MeetingId>
        <MeetingName>string</MeetingName>
        <Organizer />
        <Role>None or Organizer or Attendee or OrganizerOrAttendee</Role>
        <MeetingDate>dateTime</MeetingDate>
        <MeetingDescription>string</MeetingDescription>
        <SendReminder>boolean</SendReminder>
        <ReminderLeadTime>None or FiveMinutes or FifteenMinutes or
ThirtyMinutes or FortyFiveMinutes or OneHour</ReminderLeadTime>
        <Attendees>
          <AttendeeId />
        </Attendees>
      </Meeting>
    </DeleteMeeting>
  </soap:Body>
</soap:Envelope>
```

```

        <AttendeeId />
    </Attendees>
    <ReminderSent>boolean</ReminderSent>
</Meeting>
</DeleteMeeting>
</soap:Body>
</soap:Envelope>

```

```

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <DeleteMeetingResponse xmlns="http://api.filesanywhere.com/">
      <DeleteMeetingResult>
        <ErrorMessage>string</ErrorMessage>
        <Deleted>boolean</Deleted>
      </DeleteMeetingResult>
    </DeleteMeetingResponse>
  </soap:Body>
</soap:Envelope>

```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```

POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <DeleteMeeting xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    <Meeting>
      <MeetingId>long</MeetingId>
      <MeetingName>string</MeetingName>
      <Organizer />
      <Role>None or Organizer or Attendee or OrganizerOrAttendee</Role>
      <MeetingDate>dateTime</MeetingDate>
      <MeetingDescription>string</MeetingDescription>
    </Meeting>
  </soap12:Body>
</soap12:Envelope>

```



```

    <SendReminder>boolean</SendReminder>
    <ReminderLeadTime>None or FiveMinutes or FifteenMinutes or
    ThirtyMinutes or FortyFiveMinutes or OneHour</ReminderLeadTime>
    <Attendees>
      <AttendeeId />
      <AttendeeId />
    </Attendees>
    <ReminderSent>boolean</ReminderSent>
  </Meeting>
</DeleteMeeting>
</soap12:Body>
</soap12:Envelope>

```

```

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <DeleteMeetingResponse xmlns="http://api.filesanywhere.com/">
      <DeleteMeetingResult>
        <ErrorMessage>string</ErrorMessage>
        <Deleted>boolean</Deleted>
      </DeleteMeetingResult>
    </DeleteMeetingResponse>
  </soap12:Body>
</soap12:Envelope>

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
Meeting	Meeting	Yes	A populated meeting structure.

A Meeting has the following properties:

Property Name	Data Type	Required	Description
---------------	-----------	----------	-------------

MeetingId	Long	Yes	Unique identifier for the meeting.
MeetingName	String	Yes	String identifier or title for the meeting.
Organizer	UserIdentifier	Yes	User identifier representing the organizer.
Role	Enum	Yes	The role of the current user in the meeting.
MeetingDate	DateTime	Yes	The scheduled start date of the meeting.
DurationInMinutes	Integer	Yes	The planned length of the meeting in minutes.
MeetingDescription	String	Yes	The agenda or discussion plans for the meeting.
SendReminder	Boolean	Yes	Flag indicating if reminder notifications are to be sent to attendees.
ReminderLeadTime	Enum	Yes	The number of minutes before the meeting starts to send reminder notifications.
Attendees	UserIdentifier()	Yes	Array of user identifiers representing attendees. Ignored in DeleteMeeting.
ReminderSent	Boolean	Yes	Flag indicating if reminder notifications have been sent. Ignored in DeleteMeeting.

Response

DeleteMeetingResult

DeleteMeetingResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the GetMeeting method was executed. If the GetMeeting call was successful, the ErrorMessage is blank.
Deleted	Boolean	Flag indicating if the meeting was deleted successfully.

EditDocumentWithZoho

Syntax

`EditDocumentWithZohoResult = EditDocumentWithZoho(Token,Path)`

Usage

The following parameters are needed to make the `EditDocumentWithZoho` call:

- **Token.** (The Token ID returned by the last `AccountLogin` call).
- **Path.** (The path of the document to be edited).

SOAP 1.1

The following is a sample **SOAP 1.1** request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/EditDocumentWithZoho"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <EditDocumentWithZoho xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
    </EditDocumentWithZoho>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <EditDocumentWithZohoResponse xmlns="http://api.filesanywhere.com/">
      <UndoCheckOutFileResult>
        <ErrorMessage>string</ErrorMessage>
        <URL>string</URL>
      </UndoCheckOutFileResult>
    </EditDocumentWithZohoResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <EditDocumentWithZoho xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
    </EditDocumentWithZoho>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <EditDocumentWithZohoResponse xmlns="http://api.filesanywhere.com/">
      <UndoCheckOutFileResult>
        <ErrorMessage>string</ErrorMessage>
        <URL>string</URL>
      </UndoCheckOutFileResult>
    </EditDocumentWithZohoResponse>
  </soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
Path	String	Yes	The path to the document to be edited.

Response

EditDocumentWithZohoResult

EditDocumentWithZohoResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the EditDocumentWithZoho method was executed. If the EditDocumentWithZoho call was successful, the ErrorMessage is blank
URL	String	A URL which can be opened in a browser to edit the document.

EsendFiles

Syntax

EsendFilesResult = EsendFiles(Token,FileList,Recipients,Message)

Usage

Use the EsendFiles method to send files directly to another user's file inbox.

The following parameters are needed to make the EsendFiles call:

- Token. (The Token ID returned by the last AccountLogin call).
- FileList. (The paths of the files to be sent to the recipients).
- Recipients (A list of usernames to receive the files).
- Message (A message to send with the notifications to the recipients).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/EsendFiles"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <EsendFiles xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <FileList>string</FileList>
      <FileList>string</FileList>
      <Recipients>string</Recipients>
      <Message>string</Message>
    </EsendFiles>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ESendFilesResponse xmlns="http://api.filesanywhere.com/">
      <ESendFilesResult>
        <ErrorMessage>string</ErrorMessage>
      </ESendFilesResult>
    </ESendFilesResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ESendFiles xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <FileList>string</FileList>
      <FileList>string</FileList>
      <Recipients>string</Recipients>
      <Message>string</Message>
    </ESendFiles>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ESendFilesResponse xmlns="http://api.filesanywhere.com/">
      <ESendFilesResult>
        <ErrorMessage>string</ErrorMessage>
      </ESendFilesResult>
    </ESendFilesResponse>
  </soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
FileList	String()	Yes	The paths to the files to be sent to the recipients.
Recipients	String	Yes	A comma-separated list of usernames.
Message	String	Yes	A message to include in the notifications to the recipients.

Response

ESendFilesResult

ESendFilesResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ESendFiles method was executed. If the ESendFiles call was successful, the ErrorMessage is blank

GetAccountDetails

Syntax

GetAccountDetailsResult = GetAccountDetails(Token)

Usage

Use the GetAccountDetails method to retrieve a user's account properties, including email address, storage allocation, storage usage, user account type, and account status. If Logged in user client is configured for multiple regions, then API will provide region details (RegionID, Region Name, User site URL) else Region detail will be null.

The following parameters are needed to make the GetAccountDetails call:

- Token. (The Token ID returned by the last AccountLogin call).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetAccountDetails"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetAccountDetails xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </GetAccountDetails>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
```

```

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetAccountDetailsResponse xmlns="http://api.filesanywhere.com/">
      <GetAccountDetailsResult>
        <ErrorMessage>string</ErrorMessage>
        <EmailAddress>string</EmailAddress>
        <CanChangeEmailAddress>boolean</CanChangeEmailAddress>
        <StorageAllocationInBytes>long</StorageAllocationInBytes>
        <StorageAllocationDisplay>string</StorageAllocationDisplay>
        <StorageUsageInBytes>long</StorageUsageInBytes>
        <StorageUsageDisplay>string</StorageUsageDisplay>
        <UserType>string</UserType>
        <AccountStatus>string</AccountStatus>
        <RegionID>int</RegionID>
        <RegionDetails>
          <Name>string</Name>
          <UserURL>string</UserURL>
        </RegionDetails>
      </GetAccountDetailsResult>
    </GetAccountDetailsResponse>
  </soap:Body>
</soap:Envelope>

```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```

POST /v2/fawapi.asmx HTTP/1.1
Host: dev.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetAccountDetails xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </GetAccountDetails>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetAccountDetailsResponse xmlns="http://api.filesanywhere.com/">
      <GetAccountDetailsResult>
        <ErrorMessage>string</ErrorMessage>
        <EmailAddress>string</EmailAddress>
        <CanChangeEmailAddress>boolean</CanChangeEmailAddress>
        <StorageAllocationInBytes>long</StorageAllocationInBytes>
        <StorageAllocationDisplay>string</StorageAllocationDisplay>
        <StorageUsageInBytes>long</StorageUsageInBytes>
        <StorageUsageDisplay>string</StorageUsageDisplay>
        <UserType>string</UserType>
        <AccountStatus>string</AccountStatus>
        <RegionID>int</RegionID>
        <RegionDetails>
          <Name>string</Name>
          <UserURL>string</UserURL>
        </RegionDetails>
      </GetAccountDetailsResult>
    </GetAccountDetailsResponse>
  </soap12:Body>
</soap12:Envelope>
```

HTTP POST

The following is a sample HTTP POST request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /facorpapi/v2/fawapi.asmx/GetAccountDetails HTTP/1.1
Host: dev.filesanywhere.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length

Token=string
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<GetAccountDetailsResultv2 xmlns="http://api.filesanywhere.com/">
  <ErrorMessage>string</ErrorMessage>
  <EmailAddress>string</EmailAddress>
  <CanChangeEmailAddress>boolean</CanChangeEmailAddress>
  <StorageAllocationInBytes>long</StorageAllocationInBytes>
  <StorageAllocationDisplay>string</StorageAllocationDisplay>
  <StorageUsageInBytes>long</StorageUsageInBytes>
  <StorageUsageDisplay>string</StorageUsageDisplay>
  <UserType>string</UserType>
  <AccountStatus>string</AccountStatus>
```

```
<RegionID>int</RegionID>
<RegionDetails>
  <Name>string</Name>
  <UserURL>string</UserURL>
</RegionDetails>
</GetAccountDetailsResultv2>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.

Response

GetAccountDetailsResult

GetAccountDetailsResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the GetAccountDetails method was executed. If the GetAccountDetails call was successful, the ErrorMessage is blank
EmailAddress	String	User Email address
CanChangeEmailAddress	Boolean	User can change his email address
StorageAllocationInBytes	LongInt	Allocated Space in Bytes
StorageAllocationDisplay	String	Allocated space in Text to display with storage unit
StorageUsageInBytes	LogInt	Used space in bytes

StorageUsageDisplay	String	Used space in Text to display with storage unit
UserType	String	P- Paid user/ String.Empty-Free user
AccountStatus	String	A-Active/ I-Inactive/ D-Disabled
RegionID	Int	When Client is configured for multiple region then region ID represent user storage area in region.
RegionDetails	Class	When Client is configured for multiple region then Region details will contain sub attribute with region details (Region Name/ User URL) else RegionDetails will be null

GetAPNSSetting

Syntax

GetAPNSSettingResult = GetAPNSSetting (Token, NotificationType)

Usage

Use the GetAPNSSetting method to determine if a specific notification type is permitted by the logged on user.

The following parameters are needed to make the GetAPNSSetting call:

- Token. (The Token ID returned by the last AccountLogin call).
- NotificationType. (The type of notification.)

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetAPNSSetting"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetAPNSSetting xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <NotificationCategory>FILEACTION_ANNOTATE or FILEACTION_DELETED or
FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANGE or
GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or
```

```
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or ACCOUNTACTION_STORAGEEXCEEDED or  
NONE</NotificationCategory>  
</GetAPNSetting>  
</soap:Body>  
</soap:Envelope>
```

```
HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: length  
  
<?xml version="1.0" encoding="utf-8"?>  
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <GetAPNSettingResponse xmlns="http://api.filesanywhere.com/">  
      <GetAPNSettingResult>  
        <ErrorMessage>string</ErrorMessage>  
        <NotificationType>FILEACTION_ANNOTATE or FILEACTION_DELETED or  
FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or  
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or  
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE  
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or  
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED  
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANGE or  
GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or  
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or  
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or  
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or  
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or ACCOUNTACTION_STORAGEEXCEEDED or  
NONE</NotificationType>  
        <Setting>string</Setting>  
      </GetAPNSettingResult>  
    </GetAPNSettingResponse>  
  </soap:Body>  
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetAPNSSetting xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <NotificationCategory>FILEACTION_ANNOTATE or FILEACTION_DELETED or
FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANGЕ or
GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or ACCOUNTACTION_STORAGEEXCEEDED or
NONE</NotificationCategory>
    </GetAPNSSetting>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetAPNSSettingResponse xmlns="http://api.filesanywhere.com/">
      <GetAPNSSettingResult>
        <ErrorMessage>string</ErrorMessage>
        <NotificationType>FILEACTION_ANNOTATE or FILEACTION_DELETED or
FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE
```



```

or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANGE or
GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or ACCOUNTACTION_STORAGEEXCEEDED or
NONE</NotificationType>
    <Setting>string</Setting>
  </GetAPNSettingResult>
</GetAPNSettingResponse>
</soap12:Body>
</soap12:Envelope>

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
NotificationType	Enum	Yes	The type of notification to receive the enabled setting for.

Response

GetAPNSettingResult

GetAPNSettingResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the GetAPNSetting method was executed. If the GetAPNSetting call was successful, the ErrorMessage is blank
Settings.APNSetting	Enum	The notification type.
Settings.Setting	String	Flag indicating the notification type is enabled or disabled.

GetAPNSettings

Syntax

GetAPNSettingsResult = GetAPNSettings (Token)

Usage

Use the GetAPNSettings method to retrieve a list of notification type and their current settings for the logged on user.

The following parameters are needed to make the GetAPNSettings call:

- Token. (The Token ID returned by the last AccountLogin call).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetAPNSettings"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetAPNSettings xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </GetAPNSettings>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
```

Content-Type: text/xml; charset=utf-8
Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetAPNSettingsResponse xmlns="http://api.filesanywhere.com/">
      <GetAPNSettingResult>
        <ErrorMessage>string</ErrorMessage>
        <Settings>
          <APNSetting>
            <NotificationType>FILEACTION_ANNOTATE or FILEACTION_DELETED or
FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANGE or
GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or ACCOUNTACTION_STORAGEEXCEEDED or
NONE</NotificationType>
            <Setting>string</Setting>
          </APNSetting>
          <APNSetting>
            <NotificationType>FILEACTION_ANNOTATE or FILEACTION_DELETED or
FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANGE or
GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or ACCOUNTACTION_STORAGEEXCEEDED or
NONE</NotificationType>
            <Setting>string</Setting>
          </APNSetting>
        </Settings>
      </GetAPNSettingResult>
    </GetAPNSettingsResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetAPNSettings xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </GetAPNSettings>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetAPNSettingsResponse xmlns="http://api.filesanywhere.com/">
      <GetAPNSettingResult>
        <ErrorMessage>string</ErrorMessage>
        <Settings>
          <APNSetting>
            <NotificationType>FILEACTION_ANNOTATE or FILEACTION_DELETED or
FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTION_APPUPDATE
or PRODUCTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANGE or
GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or ACCOUNTACTION_STORAGEEXCEEDED or
NONE</NotificationType>
            <Setting>string</Setting>
          </APNSetting>
        </Settings>
      </GetAPNSettingResult>
    </GetAPNSettingsResponse>
  </soap12:Body>
</soap12:Envelope>
```

```
<APNSSetting>
  <NotificationType>FILEACTION_ANNOTATE or FILEACTION_DELETED or
FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANGE or
GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or ACCOUNTACTION_STORAGEEXCEEDED or
NONE</NotificationType>
  <Setting>string</Setting>
</APNSSetting>
</Settings>
</GetAPNSettingResult>
</GetAPNSettingsResponse>
</soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.

Response**GetAPNSettingsResult**

GetAPNSettingsResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the GetAPNSettings method was executed. If the GetAPNSettings call was successful, the ErrorMessage is blank
Settings.APNSetting.NotificationType	Enum	The notification type.
Settings.APNSetting.Setting	String	Flag indicating the notification type is enabled or disabled.

GetDropBoxDefaults2

Syntax

GetDropBoxDefaults2Result = GetDropBoxDefaults2(Token)

Usage

Use the GetDropBoxDefaults2 method to retrieve the default settings for creating a new FileReceive Link.

The following parameters are needed to make the GetDropBoxDefaults2 call:

- Token. (The Token ID returned by the last AccountLogin call).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetDropBoxDefaults2"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetDropBoxDefaults2 xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </GetDropBoxDefaults2>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetDropBoxDefaults2Response xmlns="http://api.filesanywhere.com/">
```



```
<GetDropBoxDefaultsResult>
  <ErrorMessage>string</ErrorMessage>
  <ShareDays>int</ShareDays>
  <FlagLog>string</FlagLog>
  <FlagEmail>string</FlagEmail>
  <FlagEmailFromSystem>string</FlagEmailFromSystem>
  <EmailSubject>string</EmailSubject>
  <EmailBody>string</EmailBody>
  <FlagAttachment>string</FlagAttachment>
  <FlagReadReceipt>string</FlagReadReceipt>
  <FlagOverwrite>string</FlagOverwrite>
  <SubFolderNamePrompt>string</SubFolderNamePrompt>
  <ReturnLinkLabel>string</ReturnLinkLabel>
  <ReturnLinkURL>string</ReturnLinkURL>
</GetDropBoxDefaultsResult>
</GetDropBoxDefaults2Response>
</soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetDropBoxDefaults2 xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </GetDropBoxDefaults2>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetDropBoxDefaults2Response xmlns="http://api.filesanywhere.com/">
      <GetDropBoxDefaultsResult>
        <ErrorMessage>string</ErrorMessage>
        <ShareDays>int</ShareDays>
        <FlagLog>string</FlagLog>
        <FlagEmail>string</FlagEmail>
        <FlagEmailFromSystem>string</FlagEmailFromSystem>
        <EmailSubject>string</EmailSubject>
        <EmailBody>string</EmailBody>
        <FlagAttachment>string</FlagAttachment>
        <FlagReadReceipt>string</FlagReadReceipt>
        <FlagOverwrite>string</FlagOverwrite>
        <SubFolderNamePrompt>string</SubFolderNamePrompt>
        <ReturnLinkLabel>string</ReturnLinkLabel>
        <ReturnLinkURL>string</ReturnLinkURL>
      </GetDropBoxDefaultsResult>
    </GetDropBoxDefaults2Response>
  </soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.

Response

GetDropBoxDefaultsResult2v2

GetDropBoxDefaultsResult2v2 Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the GetDropBoxDefaults2 method was executed. If the GetDropBoxDefaults2 call was successful, the ErrorMessage is blank
ShareDays	Integer	The number of days the link will be active.
FlagLog	String	Flag indicating to log access to the link.
FlagEmail	String	Flag indicating to send an email notification of the link to the recipients.
FlagEmailFromSystem	String	Flag indicating to use the system email address or the user's email address.
EmailSubject	String	Subject of the email notification.
FlagAttachment	String	
FlagReadReceipt	String	Flag indicating to send a read-receipt when the notification is read.

FlagOverwrite	String	Flag indicating to allow file overwrites.
SubFolderNamePrompt	String	Prompt used to create a separate partition within the FileReceive Link.
ReturnLinkLabel	String	Text displayed to return the user to a different URL.
ReturnLinkURL	String	The URL to return the user to.

GetELinkDefaults2

Syntax

GetELinkDefaults2Result = GetELinkDefaults2(Token)

Usage

Use the GetELinkDefaults2 method to retrieve the default settings for creating a new FileShare link.

The following parameters are needed to make the GetELinkDefaults2 call:

- Token. (The Token ID returned by the last AccountLogin call).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetELinkDefaults2"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetELinkDefaults2 xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </GetELinkDefaults2>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetELinkDefaults2Response xmlns="http://api.filesanywhere.com/">
```

```

<GetELinkDefaults2Result>
  <ErrorMessage>string</ErrorMessage>
  <ShareDays>int</ShareDays>
  <FlagLog>string</FlagLog>
  <FlagEmail>string</FlagEmail>
  <FlagMediaView>string</FlagMediaView>
  <FlagPlayOptions>string</FlagPlayOptions>
  <DownloadLimit>int</DownloadLimit>
  <FlagViewOnly>string</FlagViewOnly>
  <FlagWatermark>string</FlagWatermark>
  <WatermarkCenter>string</WatermarkCenter>
  <WatermarkBottom>string</WatermarkBottom>
  <FlagEmailFromSystem>string</FlagEmailFromSystem>
  <EmailSubject>string</EmailSubject>
  <EmailBody>string</EmailBody>
  <FlagAttachment>string</FlagAttachment>
  <FlagReadReceipt>string</FlagReadReceipt>
  <FlagCoverPage>string</FlagCoverPage>
  <CoverPageTitle>string</CoverPageTitle>
  <CoverPageMessage>string</CoverPageMessage>
  <ApprovalText>string</ApprovalText>
  <InputPromptText>string</InputPromptText>
  <SignaturePromptText>string</SignaturePromptText>
</GetELinkDefaults2Result>
</GetELinkDefaults2Response>
</soap:Body>
</soap:Envelope>

```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```

POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetELinkDefaults2 xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </GetELinkDefaults2>
  </soap12:Body>
</soap12:Envelope>

```

```

HTTP/1.1 200 OK

```

Content-Type: application/soap+xml; charset=utf-8
Content-Length: [length](#)

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetELinkDefaults2Response xmlns="http://api.filesanywhere.com/">
      <GetELinkDefaults2Result>
        <ErrorMessage>string</ErrorMessage>
        <ShareDays>int</ShareDays>
        <FlagLog>string</FlagLog>
        <FlagEmail>string</FlagEmail>
        <FlagMediaView>string</FlagMediaView>
        <FlagPlayOptions>string</FlagPlayOptions>
        <DownloadLimit>int</DownloadLimit>
        <FlagViewOnly>string</FlagViewOnly>
        <FlagWatermark>string</FlagWatermark>
        <WatermarkCenter>string</WatermarkCenter>
        <WatermarkBottom>string</WatermarkBottom>
        <FlagEmailFromSystem>string</FlagEmailFromSystem>
        <EmailSubject>string</EmailSubject>
        <EmailBody>string</EmailBody>
        <FlagAttachment>string</FlagAttachment>
        <FlagReadReceipt>string</FlagReadReceipt>
        <FlagCoverPage>string</FlagCoverPage>
        <CoverPageTitle>string</CoverPageTitle>
        <CoverPageMessage>string</CoverPageMessage>
        <ApprovalText>string</ApprovalText>
        <InputPromptText>string</InputPromptText>
        <SignaturePromptText>string</SignaturePromptText>
      </GetELinkDefaults2Result>
    </GetELinkDefaults2Response>
  </soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.

Response

GetELinkDefaultsResult2v2

GetELinkDefaultsResult2v2 Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the GetELinkDefaults2 method was executed. If the GetELinkDefaults2 call was successful, the ErrorMessage is blank
ShareDays	Integer	The number of days the link will be active.
FlagLog	String	Flag indicating to log access to the link.
FlagEmail	String	Flag indicating to send an email notification of the link to the recipients.
FlagMediaView	String	Flag indicating the view type to display by default.
FlagPlayOptions	String	Flag indicating the play options.
DownloadLimit	String	The maximum number of times files can be downloaded.
FlagViewOnly	String	Flag specifying whether to allow downloads or not.
FlagWatermark	String	Flag indicating whether or not to generate a watermark.

WatermarkCenter	String	The watermark text for the center.
WatermarkBottom	String	The watermark text for the bottom.
FlagEmailFromSystem	String	Flag indicating to send the notification from the system email account or the user's email account.
EmailSubject	String	Email notification subject.
EmailBody	String	Email notification body.
FlagAttachment	String	
FlagReadReceipt	String	Flag indicating to send a read receipt when the notification email is read.
FlagCoverPage	String	Flag indicating whether or not to use a cover page.
CoverPageTitle	String	The title for the cover page.
CoverPageMessage	String	The message body for the cover page.
ApprovalText	String	Text to display to require approval from the user.
InputPromptText	String	Text to display to retrieve input from the user.
SignaturePromptText	String	Text to display to retrieve a signature from the user.

GetELinkDefaults3

Syntax

GetELinkDefaults3Result = GetELinkDefaults3(Token)

Usage

Use the GetELinkDefaults3 method to retrieve the default settings for creating a new FileShare link.

The following parameters are needed to make the GetELinkDefaults3 call:

- Token. (The Token ID returned by the last AccountLogin call).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetELinkDefaults3"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetELinkDefaults3 xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </GetELinkDefaults3>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetELinkDefaults3Response xmlns="http://api.filesanywhere.com/">
      <GetELinkDefaults3Result>
        <ErrorMessage>string</ErrorMessage>
      </GetELinkDefaults3Result>
    </GetELinkDefaults3Response>
  </soap:Body>
</soap:Envelope>
```

```

    <ShareDays>int</ShareDays>
    <FlagLog>string</FlagLog>
    <FlagEmail>string</FlagEmail>
    <FlagMediaView>string</FlagMediaView>
    <FlagPlayOptions>string</FlagPlayOptions>
    <DownloadLimit>int</DownloadLimit>
    <FlagViewOnly>string</FlagViewOnly>
    <FlagWatermark>string</FlagWatermark>
    <WatermarkCenter>string</WatermarkCenter>
    <WatermarkBottom>string</WatermarkBottom>
    <FlagEmailFromSystem>string</FlagEmailFromSystem>
    <EmailSubject>string</EmailSubject>
    <EmailBody>string</EmailBody>
    <FlagAttachment>string</FlagAttachment>
    <FlagReadReceipt>string</FlagReadReceipt>
    <FlagCoverPage>string</FlagCoverPage>
    <CoverPageTitle>string</CoverPageTitle>
    <CoverPageMessage>string</CoverPageMessage>
    <ApprovalText>string</ApprovalText>
    <InputPromptText>string</InputPromptText>
    <SignaturePromptText>string</SignaturePromptText>
    <ShowEmailAddress>boolean</ShowEmailAddress>
    <ShowName>boolean</ShowName>
    <ShowComment>boolean</ShowComment>
    <IncludeEmailInWatermark>Boolean</IncludeEmailInWatermark>
<IncludeFileShareIdInWatermark>Boolean</IncludeFileShareIdInWatermark>
    <IncludeIPInWatermark>Boolean</IncludeIPInWatermark>
    <IncludeTimeStampInWatermark>Boolean</IncludeTimeStampInWatermark>
    <WatermarkColor>string</WatermarkColor>
    <WatermarkSize>int</WatermarkSize>
  </GetELinkDefaults3Result>
</GetELinkDefaults3Response>
</soap:Body>
</soap:Envelope>

```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```

POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetELinkDefaults3 xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>

```

```
</GetELinkDefaults3>  
</soap12:Body>  
</soap12:Envelope>
```

```
HTTP/1.1 200 OK  
Content-Type: application/soap+xml; charset=utf-8  
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>  
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">  
  <soap12:Body>  
    <GetELinkDefaults3Response xmlns="http://api.filesanywhere.com/">  
      <GetELinkDefaults3Result>  
        <ErrorMessage>string</ErrorMessage>  
        <ShareDays>int</ShareDays>  
        <FlagLog>string</FlagLog>  
        <FlagEmail>string</FlagEmail>  
        <FlagMediaView>string</FlagMediaView>  
        <FlagPlayOptions>string</FlagPlayOptions>  
        <DownloadLimit>int</DownloadLimit>  
        <FlagViewOnly>string</FlagViewOnly>  
        <FlagWatermark>string</FlagWatermark>  
        <WatermarkCenter>string</WatermarkCenter>  
        <WatermarkBottom>string</WatermarkBottom>  
        <FlagEmailFromSystem>string</FlagEmailFromSystem>  
        <EmailSubject>string</EmailSubject>  
        <EmailBody>string</EmailBody>  
        <FlagAttachment>string</FlagAttachment>  
        <FlagReadReceipt>string</FlagReadReceipt>  
        <FlagCoverPage>string</FlagCoverPage>  
        <CoverPageTitle>string</CoverPageTitle>  
        <CoverPageMessage>string</CoverPageMessage>  
        <ApprovalText>string</ApprovalText>  
        <InputPromptText>string</InputPromptText>  
        <SignaturePromptText>string</SignaturePromptText>  
        <ShowEmailAddress>boolean</ShowEmailAddress>  
        <ShowName>boolean</ShowName>  
        <ShowComment>boolean</ShowComment>  
        <IncludeEmailInWatermark>boolean</IncludeEmailInWatermark>  
<IncludeFileShareIdInWatermark>boolean</IncludeFileShareIdInWatermark>  
        <IncludeIPInWatermark>boolean</IncludeIPInWatermark>  
        <IncludeTimeStampInWatermark>boolean</IncludeTimeStampInWatermark>  
        <WatermarkColor>string</WatermarkColor>  
        <WatermarkSize>int</WatermarkSize>  
      </GetELinkDefaults3Result>  
    </GetELinkDefaults3Response>  
  </soap12:Body>  
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.

Response

GetELinkDefaultsResult3v2

GetELinkDefaultsResult3v2 Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the GetELinkDefaults2 method was executed. If the GetELinkDefaults2 call was successful, the ErrorMessage is blank
ShareDays	Integer	The number of days the link will be active.
FlagLog	String	Flag indicating to log access to the link.
FlagEmail	String	Flag indicating to send an email notification of the link to the recipients.
FlagMediaView	String	Flag indicating the view type to display by default.
FlagPlayOptions	String	Flag indicating the play options.
DownloadLimit	String	The maximum number of times files can be downloaded.
FlagViewOnly	String	Flag specifying whether to allow downloads or not.
FlagWatermark	String	Flag indicating whether or not to generate a watermark.
WatermarkCenter	String	The watermark text for the center.
WatermarkBottom	String	The watermark text for the bottom.

FlagEmailFromSystem	String	Flag indicating to send the notification from the system email account or the user's email account.
EmailSubject	String	Email notification subject.
EmailBody	String	Email notification body.
FlagAttachment	String	
FlagReadReceipt	String	Flag indicating to send a read receipt when the notification email is read.
FlagCoverPage	String	Flag indicating whether or not to use a cover page.
CoverPageTitle	String	The title for the cover page.
CoverPageMessage	String	The message body for the cover page.
ApprovalText	String	Text to display to require approval from the user.
InputPromptText	String	Text to display to retrieve input from the user.
SignaturePromptText	String	Text to display to retrieve a signature from the user.
ShowEmailAddress	Boolean	Flag indicating to show the name of the owner of the fileshare.
ShowName	Boolean	Flag indicating to show the email address of the owner of the fileshare.
ShowComment	Boolean	Flag indicating to show the comments against each fileshare items.
IncludeEmailInWatermark	Boolean	Flag indicating to show the email address of recipient in watermark.
IncludeFileShareIdInWatermark	Boolean	Flag indicating to Include FileShare ID in watermark text.
IncludeIPInWatermark	Boolean	Flag indicating to Include recipient's IP address into watermark text.
IncludeTimeStampInWatermark	Boolean	Flag indicating to include Downloaded DateTime stamp of a file into Watermark.
WatermarkColor	String	<p>Possible values for this parameters are 'RD', 'GY', 'GN', 'BE', 'BK', 'WE'</p> <p>RD = Red, GY = Grey, GN = Green, BK = Black, WE = White, BE = Blue</p>

WatermarkSize	Integer	Values should be {8, 10, 12, 14, 16, 18, 20, 22, 24} in pixel format
---------------	---------	--

GetFolderProperties

Syntax

`GetFolderPropertiesResult = GetFolderProperties(Token,FolderPath)`

Usage

Use the `GetFolderProperties` method to retrieve the properties for a folder.

The following parameters are needed to make the `GetFolderProperties` call:

- `Token`. (The Token ID returned by the last `AccountLogin` call).
- `FolderPath`. (The path to the folder to retrieve properties for).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetFolderProperties"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetFolderProperties xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <FolderPath>string</FolderPath>
    </GetFolderProperties>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
```



```

<soap:Body>
  <GetFolderPropertiesResponse xmlns="http://api.filesanywhere.com/">
    <GetFolderPropertiesResult>
      <ErrorMessage>string</ErrorMessage>
      <Comments>string</Comments>
      <FolderCreated>dateTime</FolderCreated>
      <VC_FLAG_FILE_VERSION>boolean</VC_FLAG_FILE_VERSION>
      <VC_FLAG_FILE_CHECKOUT>boolean</VC_FLAG_FILE_CHECKOUT>
      <VC_FLAG_FILE_DELETE>boolean</VC_FLAG_FILE_DELETE>
      <VC_FLAG_DAILY_EMAIL>boolean</VC_FLAG_DAILY_EMAIL>
      <VC_FLAG_UPDATE_EMAIL>boolean</VC_FLAG_UPDATE_EMAIL>
      <VC_FLAG_OWNER_EMAIL>boolean</VC_FLAG_OWNER_EMAIL>
      <VC_FLAG_GROUPSHARE_EMAIL>boolean</VC_FLAG_GROUPSHARE_EMAIL>
      <VC_EMAIL_TO>string</VC_EMAIL_TO>
      <VersionKeepCount>int</VersionKeepCount>
    </GetFolderPropertiesResult>
  </GetFolderPropertiesResponse>
</soap:Body>
</soap:Envelope>

```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```

POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetFolderProperties xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <FolderPath>string</FolderPath>
    </GetFolderProperties>
  </soap12:Body>
</soap12:Envelope>

```

```

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

```

```

<?xml version="1.0" encoding="utf-8"?>

```

```
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetFolderPropertiesResponse xmlns="http://api.filesanywhere.com/">
      <GetFolderPropertiesResult>
        <ErrorMessage>string</ErrorMessage>
        <Comments>string</Comments>
        <FolderCreated>dateTime</FolderCreated>
        <VC_FLAG_FILE_VERSION>boolean</VC_FLAG_FILE_VERSION>
        <VC_FLAG_FILE_CHECKOUT>boolean</VC_FLAG_FILE_CHECKOUT>
        <VC_FLAG_FILE_DELETE>boolean</VC_FLAG_FILE_DELETE>
        <VC_FLAG_DAILY_EMAIL>boolean</VC_FLAG_DAILY_EMAIL>
        <VC_FLAG_UPDATE_EMAIL>boolean</VC_FLAG_UPDATE_EMAIL>
        <VC_FLAG_OWNER_EMAIL>boolean</VC_FLAG_OWNER_EMAIL>
        <VC_FLAG_GROUPSHARE_EMAIL>boolean</VC_FLAG_GROUPSHARE_EMAIL>
        <VC_EMAIL_TO>string</VC_EMAIL_TO>
        <VersionKeepCount>int</VersionKeepCount>
      </GetFolderPropertiesResult>
    </GetFolderPropertiesResponse>
  </soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
FolderPath	String	Yes	The path to the folder to retrieve properties for.

Response

GetFolderPropertiesResult

GetFolderPropertiesResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the <code>GetFolderProperties</code> method was executed. If the <code>GetFolderProperties</code> call was successful, the <code>ErrorMessage</code> is blank
Comments	String	Comments attached to the folder.
FolderCreated	DateTime	The datetime stamp the folder was created.
VC_FLAG_FILE_VERSION	Boolean	Flag indicating if file version control history is enabled for the folder.
VC_FLAG_FILE_CHECKOUT	Boolean	Flag indicating if file check-in/checkout is enabled for the folder.
VC_FLAG_FILE_DELETE	Boolean	Flag indicating if disable delete is enabled for the folder.
VC_FLAG_DAILY_EMAIL	Boolean	Flag indicating if file activity for the folder is emailed daily.
VC_FLAG_UPDATE_EMAIL	Boolean	Flag indicating if updates to the

		folder are emailed.
VC_FLAG_OWNER_EMAIL	Boolean	Flag indicating to send emails to the owner.
VC_FLAG_GROUPSHARE_EMAIL	Boolean	Flag indicating to send emails to the members of the GroupShare.
VC_EMAIL_TO	String	Recipient addresses for email notifications.
VersionKeepCount	Integer	The number of versions to keep for version control history.

GetMeeting

Syntax

GetMeetingResult= GetMeeting(Token,Role,OrganizerOrAttendee,MeetingId)

Usage

Use the GetMeeting method to retrieve the properties of a scheduled meeting.

The following parameters are needed to make the GetMeeting call:

- Token. (The Token ID returned by the last AccountLogin call).
- Role. (The role of the OrganizerOrAttendee parameter).
- OrganizerOrAttendee. (The user identifier for the meeting organizer, or attendee).
- MeetingId. (The unique identifier for the meeting).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetMeeting"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetMeeting xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Role>None or Organizer or Attendee or OrganizerOrAttendee</Role>
      <OrganizerOrAttendee>
        <ClientId>int</ClientId>
        <Username>string</Username>
      </OrganizerOrAttendee>
      <MeetingId>long</MeetingId>
    </GetMeeting>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetMeetingResponse xmlns="http://api.filesanywhere.com/">
      <GetMeetingResult>
        <ErrorMessage>string</ErrorMessage>
        <Meeting>
          <MeetingId>long</MeetingId>
          <MeetingName>string</MeetingName>
          <Organizer />
          <Role>None or Organizer or Attendee or
OrganizerOrAttendee</Role>
          <MeetingDate>dateTime</MeetingDate>
          <MeetingDescription>string</MeetingDescription>
        </Meeting>
      </GetMeetingResult>
    </GetMeetingResponse>
  </soap:Body>
</soap:Envelope>
```

```
<SendReminder>boolean</SendReminder>
<ReminderLeadTime>None or FiveMinutes or FifteenMinutes or
ThirtyMinutes or FortyFiveMinutes or OneHour</ReminderLeadTime>
<Attendees>
  <AttendeeId xsi:nil="true" />
  <AttendeeId xsi:nil="true" />
</Attendees>
<ReminderSent>boolean</ReminderSent>
</Meeting>
</GetMeetingResult>
</GetMeetingResponse>
</soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetMeeting xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Role>None or Organizer or Attendee or OrganizerOrAttendee</Role>
      <OrganizerOrAttendee>
        <ClientId>int</ClientId>
        <Username>string</Username>
      </OrganizerOrAttendee>
      <MeetingId>long</MeetingId>
    </GetMeeting>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetMeetingResponse xmlns="http://api.filesanywhere.com/">
      <GetMeetingResult>
        <ErrorMessage>string</ErrorMessage>
        <Meeting>
          <MeetingId>long</MeetingId>
          <MeetingName>string</MeetingName>
          <Organizer />
          <Role>None or Organizer or Attendee or
OrganizerOrAttendee</Role>
          <MeetingDate>dateTime</MeetingDate>
          <MeetingDescription>string</MeetingDescription>
          <SendReminder>boolean</SendReminder>
          <ReminderLeadTime>None or FiveMinutes or FifteenMinutes or
ThirtyMinutes or FortyFiveMinutes or OneHour</ReminderLeadTime>
          <Attendees>
            <AttendeeId xsi:nil="true" />
            <AttendeeId xsi:nil="true" />
          </Attendees>
          <ReminderSent>boolean</ReminderSent>
        </Meeting>
      </GetMeetingResult>
    </GetMeetingResponse>
  </soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
Role	Enum	Yes	The role of the OrganizerOrAttendee parameter.
OrganizerOrAttendee	UserIdentifier	No	The user identifier for the meeting organizer or attendee. If null, then the current user is used.
MeetingId	Long	Yes	Unique identifier for the meeting.

Response

GetMeetingResult

GetMeetingResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the GetMeeting method was executed. If the GetMeeting call was successful, the ErrorMessage is blank.
Meeting	Meeting	A populated meeting structure.

A Meeting has the following properties:

Property Name	Data Type	Required	Description
MeetingId	Long	Yes	Unique identifier for the meeting.

MeetingName	String	Yes	String identifier or title for the meeting.
Organizer	UserIdentifier	Yes	User identifier representing the organizer.
Role	Enum	Yes	The role of the current user in the meeting.
MeetingDate	DateTime	Yes	The scheduled start date of the meeting.
DurationInMinutes	Integer	Yes	The planned length of the meeting in minutes.
MeetingDescription	String	Yes	The agenda or discussion plans for the meeting.
SendReminder	Boolean	Yes	Flag indicating if reminder notifications are to be sent to attendees.
ReminderLeadTime	Enum	Yes	The number of minutes before the meeting starts to send reminder notifications.
Attendees	UserIdentifier()	Yes	Array of user identifiers representing attendees.
ReminderSent	Boolean	Yes	Flag indicating if reminder notifications have been sent.

GetNotifications

Syntax

GetNotificationsResult = GetNotifications(Token,DeviceId)

Usage

Use the GetNotifications method to retrieve non-APNS user notifications.

The following parameters are needed to make the GetNotifications call:

- Token. (The Token ID returned by the last AccountLogin call).
- DeviceId. (The physical device ID used in RegisterDevice).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetNotifications"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetNotifications xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <DeviceId>string</DeviceId>
    </GetNotifications>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
```

```
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetNotificationsResponse xmlns="http://api.filesanywhere.com/">
      <GetNotificationsResult>
        <ErrorMessage>string</ErrorMessage>
        <Data>string</Data>
      </GetNotificationsResult>
    </GetNotificationsResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetNotifications xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <DeviceId>string</DeviceId>
    </GetNotifications>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
```

Content-Length: `length`

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetNotificationsResponse xmlns="http://api.filesanywhere.com/">
      <GetNotificationsResult>
        <ErrorMessage>string</ErrorMessage>
        <Data>string</Data>
      </GetNotificationsResult>
    </GetNotificationsResponse>
  </soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
DeviceId	String	Yes	The physical device ID used in RegisterDevice.

Response

GetNotificationsResult

GetNotificationsResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the GetNotifications method was executed. If the GetNotifications call was successful, the ErrorMessage is blank
Data	XML	Notification data.

GetNotificationSettingForDevice

Syntax

GetNotificationSettingForDeviceResult =
GetNotificationSettingForDevice(Token,DeviceId,NotificationType)

Usage

Use the `GetNotificationSettingForDevice` method to retrieve the setting for a specific notification type for a given device.

The following parameters are needed to make the `GetNotificationSettingForDevice` call:

- `Token`. (The Token ID returned by the last `AccountLogin` call).
- `DeviceId`. (The physical device ID used in `RegisterDevice`).
- `NotificationType`. (The type of notification).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetNotificationSettingForDevice"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetNotificationSettingForDevice
xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <DeviceId>string</DeviceId>
      <NotificationType>NONE or FILEACTION_ANNOTATE or FILEACTION_DELETED
or FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANG or
GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or
```

```
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or  
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or  
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or  
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or  
ACCOUNTACTION_STORAGEEXCEEDED</NotificationType>  
  </GetNotificationSettingForDevice>  
</soap:Body>  
</soap:Envelope>
```

```
HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: length  
  
<?xml version="1.0" encoding="utf-8"?>  
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <GetNotificationSettingForDeviceResponse  
xmlns="http://api.filesanywhere.com/">  
      <GetAPNSettingResult>  
        <ErrorMessage>string</ErrorMessage>  
        <Allowed>boolean</Allowed>  
      </GetAPNSettingResult>  
    </GetNotificationSettingForDeviceResponse>  
  </soap:Body>  
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1  
Host: api.filesanywhere.com  
Content-Type: application/soap+xml; charset=utf-8  
Content-Length: length  
  
<?xml version="1.0" encoding="utf-8"?>  
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">  
  <soap12:Body>  
    <GetNotificationSettingForDevice  
xmlns="http://api.filesanywhere.com/">  
      <Token>string</Token>  
      <DeviceId>string</DeviceId>  
      <NotificationType>NONE or FILEACTION_ANNOTATE or FILEACTION_DELETED  
or FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or  
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
```

```
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE  
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or  
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED  
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANG or  
GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or  
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or  
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or  
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or  
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or  
ACCOUNTACTION_STORAGEEXCEEDED</NotificationType>  
  </GetNotificationSettingForDevice>  
</soap12:Body>  
</soap12:Envelope>
```

```
HTTP/1.1 200 OK  
Content-Type: application/soap+xml; charset=utf-8  
Content-Length: length  
  
<?xml version="1.0" encoding="utf-8"?>  
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">  
  <soap12:Body>  
    <GetNotificationSettingForDeviceResponse  
xmlns="http://api.filesanywhere.com/">  
      <GetAPNSettingResult>  
        <ErrorMessage>string</ErrorMessage>  
        <Allowed>boolean</Allowed>  
      </GetAPNSettingResult>  
    </GetNotificationSettingForDeviceResponse>  
  </soap12:Body>  
</soap12:Envelope>
```


Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
DeviceId	String	Yes	The physical device ID used in RegisterDevice.
NotificationType	Enum	Yes	The notification type to retrieve the setting for.

Response

GetNotificationSettingForDeviceResult

GetNotificationSettingForDeviceResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the GetNotificationSettingForDevice method was executed. If the GetNotificationSettingForDevice call was successful, the ErrorMessage is blank
Allowed	Boolean	Whether the notification type is allowed or not.

GetNotificationSettingsForDevice

Syntax

**GetNotificationSettingsForDeviceResult =
GetNotificationSettingsForDevice(Token,DeviceId)**

Usage

Use the **GetNotificationSettingsForDevice** method to retrieve the notification settings for a given device.

The following parameters are needed to make the **GetNotificationSettingsForDevice** call:

- **Token.** (The Token ID returned by the last **AccountLogin** call).
- **DeviceId.** (The physical device ID used in **RegisterDevice**).

SOAP 1.1

The following is a sample **SOAP 1.1** request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction:
"http://api.filesanywhere.com/GetNotificationSettingsForDevice"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetNotificationSettingsForDevice
xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <DeviceId>string</DeviceId>
    </GetNotificationSettingsForDevice>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
```

```
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetNotificationSettingsForDeviceResponse
xmlns="http://api.filesanywhere.com/">
      <GetNotificationSettingForDeviceResult>
        <ErrorMessage>string</ErrorMessage>
        <AllowedTypes>
          <Value>long</Value>
          <Types>
            <AllowedType xsi:nil="true" />
            <AllowedType xsi:nil="true" />
          </Types>
        </AllowedTypes>
      </GetNotificationSettingForDeviceResult>
    </GetNotificationSettingsForDeviceResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetNotificationSettingsForDevice
xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <DeviceId>string</DeviceId>
    </GetNotificationSettingsForDevice>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetNotificationSettingsForDeviceResponse
xmlns="http://api.filesanywhere.com/">
      <GetNotificationSettingForDeviceResult>
        <ErrorMessage>string</ErrorMessage>
        <AllowedTypes>
          <Value>long</Value>
          <Types>
            <AllowedType xsi:nil="true" />
            <AllowedType xsi:nil="true" />
          </Types>
        </AllowedTypes>
      </GetNotificationSettingForDeviceResult>
    </GetNotificationSettingsForDeviceResponse>
  </soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
DeviceId	String	Yes	The physical device ID used in RegisterDevice.

Response**GetNotificationSettingsForDeviceResult**

GetNotificationSettingsForDeviceResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the GetNotificationSettingsForDevice method was executed. If the GetNotificationSettingsForDevice call was successful, the ErrorMessage is blank
AllowedTypes.Value	Long	The numeric representation of the allowed notification types.
AllowedTypes.AllowedType.Type	Enum	The allowed notification type.
AllowedTypes.AllowedType.Value	Long	The numeric representation of the allowed notification type.

GroupShareAdd2

Syntax

GroupShareAdd2Result =

GroupShareAdd2(Token,UserAccountsWithWhomToShareFolder,SharedFolderName,SharedFolderAccessLevel,NotifyByEmail,Notes)

Usage

Use the GroupShareAdd2 method to create a new GroupShare.

The following parameters are needed to make the GroupShareAdd2 call:

- Token. (The Token ID returned by the last AccountLogin call).
- UserAccountsWithWhomToShareFolder. (List of users to share with).
- SharedFolderName. (The path to the folder to share).
- SharedFolderAccessLevel. (The type of access to grant to the users).
- NotifyByEmail. (Flag indicating to send an email notification to the users).
- Notes. (Notes field for GroupShare).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GroupShareAdd2"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GroupShareAdd2 xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>

      <UserAccountsWithWhomToShareFolder>string</UserAccountsWithWhomToShareFolder>
    </GroupShareAdd2>
  </soap:Body>
</soap:Envelope>
```

```
<SharedFolderName>string</SharedFolderName>
<SharedFolderAccessLevel>string</SharedFolderAccessLevel>
<NotifyByEmail>string</NotifyByEmail>
<Notes>string</Notes>
</GroupShareAdd2>
</soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GroupShareAdd2Response xmlns="http://api.filesanywhere.com/">
      <GroupShareAdd2Result>
        <ErrorMessage>string</ErrorMessage>
        <Users>
          <GroupShareAdd2ResultItem>
            <ErrorMessage>string</ErrorMessage>
            <Username>string</Username>
            <UserAdded>boolean</UserAdded>
          </GroupShareAdd2ResultItem>
          <GroupShareAdd2ResultItem>
            <ErrorMessage>string</ErrorMessage>
            <Username>string</Username>
            <UserAdded>boolean</UserAdded>
          </GroupShareAdd2ResultItem>
        </Users>
        <GroupShareCreated>boolean</GroupShareCreated>
      </GroupShareAdd2Result>
    </GroupShareAdd2Response>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GroupShareAdd2 xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>

<UserAccountsWithWhomToShareFolder>string</UserAccountsWithWhomToShareFolder>
      <SharedFolderName>string</SharedFolderName>
      <SharedFolderAccessLevel>string</SharedFolderAccessLevel>
      <NotifyByEmail>string</NotifyByEmail>
      <Notes>string</Notes>
    </GroupShareAdd2>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GroupShareAdd2Response xmlns="http://api.filesanywhere.com/">
      <GroupShareAdd2Result>
        <ErrorMessage>string</ErrorMessage>
        <Users>
          <GroupShareAdd2ResultItem>
            <ErrorMessage>string</ErrorMessage>
            <Username>string</Username>
            <UserAdded>boolean</UserAdded>
          </GroupShareAdd2ResultItem>
          <GroupShareAdd2ResultItem>
            <ErrorMessage>string</ErrorMessage>
            <Username>string</Username>
            <UserAdded>boolean</UserAdded>
        </Users>
      </GroupShareAdd2Result>
    </GroupShareAdd2Response>
  </soap12:Body>
</soap12:Envelope>
```



```

        </GroupShareAdd2ResultItem>
    </Users>
    <GroupShareCreated>boolean</GroupShareCreated>
</GroupShareAdd2Result>
</GroupShareAdd2Response>
</soap12:Body>
</soap12:Envelope>

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
UserAccountsWithWhomToShareFolder	String	Yes	Semi-colon-separated list of usernames to share with.
SharedFolderName	String	Yes	The path to the folder to share.
SharedFolderAccessLevel	String	Yes	The permissions for the users to the GroupShare.
NotifyByEmail	String	Yes	Flag indicating to send email notification to users.
Notes	String	Yes	Notes field.

Response

GroupShareAdd2Result

GroupShareAdd2Result Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the GroupShareAdd2 method was executed. If the GroupShareAdd2 call was successful, the ErrorMessage is blank.

Users.ErrorMessage	String	Error message pertaining to one user.
Users.Username	String	The username.
Users.UserAdded	Boolean	Flag indicating if the user was successfully added.
GroupShareCreated	Boolean	Flag indicating if the GroupShare was created.

GroupShareRemove

Syntax

GroupShareRemoveResult =
GroupShareRemove(Token,SharedFolderName,UserAccountsToRemove,SendEmail)

Usage

Use the **GroupShareRemove** method to remove users from a **GroupShare**.

The following parameters are needed to make the **GroupShareRemove** call:

- **Token**. (The Token ID returned by the last **AccountLogin** call).
- **SharedFolderName**. (The path to the folder to share).
- **UserAccountsToRemove**. (List of usernames to remove).
- **SendEmail**. (Flag indicating to send email notification to users removed).

SOAP 1.1

The following is a sample **SOAP 1.1** request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GroupShareRemove"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GroupShareRemove xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <SharedFolderName>string</SharedFolderName>
      <UserAccountsToRemove>string</UserAccountsToRemove>
      <SendEmail>boolean</SendEmail>
    </GroupShareRemove>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GroupShareRemoveResponse xmlns="http://api.filesanywhere.com/">
      <GroupShareRemoveResult>
        <ErrorMessage>string</ErrorMessage>
        <Users>
          <GroupShareRemoveResultItem>
            <ErrorMessage>string</ErrorMessage>
            <Username>string</Username>
            <UserRemoved>boolean</UserRemoved>
          </GroupShareRemoveResultItem>
          <GroupShareRemoveResultItem>
            <ErrorMessage>string</ErrorMessage>
            <Username>string</Username>
            <UserRemoved>boolean</UserRemoved>
          </GroupShareRemoveResultItem>
        </Users>
      </GroupShareRemoveResult>
    </GroupShareRemoveResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GroupShareRemove xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <SharedFolderName>string</SharedFolderName>
      <UserAccountsToRemove>string</UserAccountsToRemove>
      <SendEmail>boolean</SendEmail>
    </GroupShareRemove>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GroupShareRemoveResponse xmlns="http://api.filesanywhere.com/">
      <GroupShareRemoveResult>
        <ErrorMessage>string</ErrorMessage>
        <Users>
          <GroupShareRemoveResultItem>
            <ErrorMessage>string</ErrorMessage>
            <Username>string</Username>
            <UserRemoved>boolean</UserRemoved>
          </GroupShareRemoveResultItem>
          <GroupShareRemoveResultItem>
            <ErrorMessage>string</ErrorMessage>
            <Username>string</Username>
            <UserRemoved>boolean</UserRemoved>
          </GroupShareRemoveResultItem>
        </Users>
      </GroupShareRemoveResult>
    </GroupShareRemoveResponse>
  </soap12:Body>
```

```
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
SharedFolderName	String	Yes	The path to the shared folder.
UserAccountsToRemove	String	Yes	Semi-colon-separated list of usernames to remove from the GroupShare.
SendEmail	Boolean	Yes	Flag indicating to send an email notification to the user's removed from the GroupShare.

Response

GroupShareRemoveResult

GroupShareRemoveResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the GroupShareRemove method was executed. If the GroupShareRemove call was successful, the ErrorMessage is blank.
Users.ErrorMessage	String	Error message pertaining to one user.
Users.Username	String	The username.
Users.UserRemoved	Boolean	Flag indicating if the user was successfully removed.

ItemComments

Syntax

`ItemCommentsResult = ItemComments(Token,ItemPath)`

Usage

Use the `ItemComments` method to list comments for an item.

The following parameters are needed to make the `ItemComments` call:

- `Token`. (The Token ID returned by the last `AccountLogin` call).
- `ItemPath`. (The path to the file or folder).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/ItemComments"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemComments xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ItemPath>string</ItemPath>
    </ItemComments>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemCommentsResponse xmlns="http://api.filesanywhere.com/">
```

```
<ItemCommentsListResult>
  <ErrorMessage>string</ErrorMessage>
  <CommentList>
    <ItemComment>
      <ID>long</ID>
      <Comment>string</Comment>
      <CreatedBy>string</CreatedBy>
      <CreatedOn>dateTime</CreatedOn>
      <ParentID>long</ParentID>
      <Username>string</Username>
    </ItemComment>
    <ItemComment>
      <ID>long</ID>
      <Comment>string</Comment>
      <CreatedBy>string</CreatedBy>
      <CreatedOn>dateTime</CreatedOn>
      <ParentID>long</ParentID>
      <Username>string</Username>
    </ItemComment>
  </CommentList>
</ItemCommentsListResult>
</ItemCommentsResponse>
</soap:Body>
</soap:Envelope>
```


SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ItemComments xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ItemPath>string</ItemPath>
    </ItemComments>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ItemCommentsResponse xmlns="http://api.filesanywhere.com/">
      <ItemCommentsListResult>
        <ErrorMessage>string</ErrorMessage>
        <CommentList>
          <ItemComment>
            <ID>long</ID>
            <Comment>string</Comment>
            <CreatedBy>string</CreatedBy>
            <CreatedOn>dateTime</CreatedOn>
            <ParentID>long</ParentID>
            <Username>string</Username>
          </ItemComment>
          <ItemComment>
            <ID>long</ID>
            <Comment>string</Comment>
            <CreatedBy>string</CreatedBy>
            <CreatedOn>dateTime</CreatedOn>
            <ParentID>long</ParentID>
            <Username>string</Username>
          </ItemComment>
        </CommentList>
      </ItemCommentsListResult>
    </ItemCommentsResponse>
  </soap12:Body>
</soap12:Envelope>
```

```

        </ItemComment>
    </CommentList>
</ItemCommentsListResult>
</ItemCommentsResponse>
</soap12:Body>
</soap12:Envelope>

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
ItemPath	String	Yes	The path to the file or folder.

Response

ItemCommentsResult

ItemCommentsResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ItemComments method was executed. If the ItemComments call was successful, the ErrorMessage is blank.
CommentList.ID	Long	Comment ID.
CommentList.Comment	String	The comment.
CommentList.CreatedBy	String	The name of the person who created the comment.
CommentList.CreatedOn	DateTime	The datetime stamp the comment was created.
CommentList.ParentID	Long	The parent comment the

		comment applies to.
Username	String	The username of the person who created the comment.

ItemCommentsWithThumbnail

Syntax

**ItemCommentsResult =
ItemCommentsWithThumbnail(Token,ItemPath,ThumbnailSize)**

Usage

Use the `ItemCommentsWithThumbnail` method to list comments, including the profile image of the commenter for an item.

The following parameters are needed to make the `ItemCommentsWithThumbnail` call:

- **Token.** (The Token ID returned by the last `AccountLogin` call).
- **ItemPath.** (The path to the file or folder).
- **ThumbnailSize.** (The size of the image to be specified as `Small` or `Medium` or `Large`)

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/ItemCommentsWithThumbnail"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemCommentsWithThumbnail xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ItemPath>string</ItemPath>
      <ThumbnailSize>Small or Medium or Large</ThumbnailSize>
    </ItemCommentsWithThumbnail>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemCommentsWithThumbnailResponse
xmlns="http://api.filesanywhere.com/">
```

```
<ItemCommentsListResult>
  <ErrorMessage>string</ErrorMessage>
  <CommentList>
    <ItemComment>
      <ID>long</ID>
      <Comment>string</Comment>
      <CreatedBy>string</CreatedBy>
      <CreatedOn>dateTime</CreatedOn>
      <ParentID>long</ParentID>
      <Username>string</Username>
      <Thumbnail>base64Binary</Thumbnail>
    </ItemComment>
    <ItemComment>
      <ID>long</ID>
      <Comment>string</Comment>
      <CreatedBy>string</CreatedBy>
      <CreatedOn>dateTime</CreatedOn>
      <ParentID>long</ParentID>
      <Username>string</Username>
      <Thumbnail>base64Binary</Thumbnail>
    </ItemComment>
  </CommentList>
</ItemCommentsListResult>
</ItemCommentsWithThumbnailResponse>
</soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```

<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ItemCommentsWithThumbnail xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ItemPath>string</ItemPath>
      <ThumbnailSize>Small or Medium or Large</ThumbnailSize>
    </ItemCommentsWithThumbnail>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ItemCommentsWithThumbnailResponse
xmlns="http://api.filesanywhere.com/">
      <ItemCommentsListResult>
        <ErrorMessage>string</ErrorMessage>
        <CommentList>
          <ItemComment>
            <ID>long</ID>
            <Comment>string</Comment>
            <CreatedBy>string</CreatedBy>
            <CreatedOn>dateTime</CreatedOn>
            <ParentID>long</ParentID>
            <Username>string</Username>
            <Thumbnail>base64Binary</Thumbnail>
          </ItemComment>
          <ItemComment>
            <ID>long</ID>
            <Comment>string</Comment>
            <CreatedBy>string</CreatedBy>
            <CreatedOn>dateTime</CreatedOn>
            <ParentID>long</ParentID>
            <Username>string</Username>
            <Thumbnail>base64Binary</Thumbnail>
          </ItemComment>
        </CommentList>
      </ItemCommentsListResult>
    </ItemCommentsWithThumbnailResponse>
  </soap12:Body>
</soap12:Envelope>

```

Sample Visual Basic.Net Code

```

Private Sub ItemCommentsWithThumbnail ()
  Dim FAWAPI As New WebReference.FAWAPI

```

```

Dim ItemCommentsResult As WebReference.ItemCommentsResult
ItemCommentsResult = FAWAPI.ItemCommentsWithThumbnail(
    Token, ItemPath, ThumbnailSize)
If ItemCommentsResult.ErrorMessage <> "" Then
    MsgBox ItemCommentsResult.ErrorMessage
End If
ItemCommentsResult = Nothing
FAWAPI = Nothing
End Sub

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
ItemPath	String	Yes	The path to the file or folder.
ThumbnailSize	Enum	Yes	The size of the image to be specified as Small or Medium or Large

Response

ItemCommentsResult

ItemCommentsResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ItemComments method was executed. If the ItemComments call was successful, the ErrorMessage is blank.
CommentList.ID	Long	Comment ID.
CommentList.Comment	String	The comment.
CommentList.CreatedBy	String	The name of the person who created the comment.

CommentList.CreatedOn	DateTime	The datetime stamp the comment was created.
CommentList.ParentID	Long	The parent comment the comment applies to.
Username	String	The username of the person who created the comment.
Thumbnail	Base64Binary	The image stream of the profile image of the person who created the comment.

ListGroupShareUsers

Syntax

ListGroupShareUsersResult = ListGroupShareUsers(Token,SharedFolderName)

Usage

Use the ListGroupShareUsers method to list users who have access to a GroupShare.

The following parameters are needed to make the ListGroupShareUsers call:

- Token. (The Token ID returned by the last AccountLogin call).
- SharedFolderName. (The path to the shared folder).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/ListGroupShareUsers"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ListGroupShareUsers xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <SharedFolderName>string</SharedFolderName>
    </ListGroupShareUsers>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
```

Content-Length: `length`

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ListGroupShareUsersResponse xmlns="http://api.filesanywhere.com/">
      <ListGroupShareUsersResult>
        <ErrorMessage>string</ErrorMessage>
        <GroupShareUsers>
          <GroupShareUser>
            <Username>string</Username>
            <Permission>string</Permission>
          </GroupShareUser>
          <GroupShareUser>
            <Username>string</Username>
            <Permission>string</Permission>
          </GroupShareUser>
        </GroupShareUsers>
      </ListGroupShareUsersResult>
    </ListGroupShareUsersResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ListGroupShareUsers xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <SharedFolderName>string</SharedFolderName>
    </ListGroupShareUsers>
  </soap12:Body>
</soap12:Envelope>
```

```

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ListGroupShareUsersResponse xmlns="http://api.filesanywhere.com/">
      <ListGroupShareUsersResult>
        <ErrorMessage>string</ErrorMessage>
        <GroupShareUsers>
          <GroupShareUser>
            <Username>string</Username>
            <Permission>string</Permission>
          </GroupShareUser>
          <GroupShareUser>
            <Username>string</Username>
            <Permission>string</Permission>
          </GroupShareUser>
        </GroupShareUsers>
      </ListGroupShareUsersResult>
    </ListGroupShareUsersResponse>
  </soap12:Body>
</soap12:Envelope>

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
SharedFolderName	String	Yes	The path to the GroupShare folder.

Response

ListGroupShareUsersResult

ListGroupShareUsersResult Properties

Name	Data Type	Description
------	-----------	-------------

ErrorMessage	String	A description of the error, if any error was encountered while the ListGroupShareUsers method was executed. If the ListGroupShareUsers call was successful, the ErrorMessage is blank.
GroupShareUsers.Username	String	The username of the user who has access to the GroupShare.
GroupShareUsers.Permission	String	The permission the user has been granted to the GroupShare.

ListItems2

This method lists all items in the specified path, including the user's inbox and version history folders.

Syntax

ListItemsResult = ListItems2(Token, Path, PageSize, PageNum)

Usage

Use the ListItems2 method to get a list of all items in a folder.

The following parameters are needed to make the ListItems2 call:

- Token (The Token ID returned by the last AccountLogin call).
- Path (Full FilesAnywhere Path). This is the FilesAnywhere Path of the folder for which the item list is being requested. The Path should be formatted like `\UserVolume\FolderName`, where `UserVolume` is the FilesAnywhere Username for which the item list is being requested and `FolderName` is the folder or subfolder if any for which the item list is being requested.
- PageSize. (The maximum number of items to return).
- PageNum. (The page number of results to return).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/ListItems2"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ListItems2 xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
      <PageSize>int</PageSize>
      <PageNum>int</PageNum>
    </ListItems2>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ListItems2Response xmlns="http://api.filesanywhere.com/">
      <ListItemsResult>
        <ErrorMessage>string</ErrorMessage>
        <Items>
          <Item>
            <Type>string</Type>
            <Name>string</Name>
            <Size>string</Size>
            <Path>string</Path>
            <DateLastModified>string</DateLastModified>
          </Item>
          <Item>
            <Type>string</Type>
            <Name>string</Name>
            <Size>string</Size>
            <Path>string</Path>
            <DateLastModified>string</DateLastModified>
          </Item>
        </Items>
        <PageNum>int</PageNum>
        <PageCount>int</PageCount>
        <ClipboardIsEmpty>boolean</ClipboardIsEmpty>
      </ListItemsResult>
    </ListItems2Response>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ListItems2 xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
      <PageSize>int</PageSize>
      <PageNum>int</PageNum>
    </ListItems2>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ListItems2Response xmlns="http://api.filesanywhere.com/">
      <ListItemsResult>
        <ErrorMessage>string</ErrorMessage>
        <Items>
          <Item>
            <Type>string</Type>
            <Name>string</Name>
            <Size>string</Size>
            <Path>string</Path>
            <DateLastModified>string</DateLastModified>
          </Item>
          <Item>
            <Type>string</Type>
            <Name>string</Name>
            <Size>string</Size>
            <Path>string</Path>
            <DateLastModified>string</DateLastModified>
          </Item>
        </Items>
      </ListItemsResult>
    </ListItems2Response>
  </soap12:Body>
</soap12:Envelope>
```

```

</Items>
<PageNum>int</PageNum>
<PageCount>int</PageCount>
<ClipboardIsEmpty>boolean</ClipboardIsEmpty>
</ListItemsResult>
</ListItems2Response>
</soap12:Body>
</soap12:Envelope>

```

Sample Visual Basic.Net Code

```

Private Sub ListItems ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim ListItemsResult As WebReference.ListItemsResult
    ListItemsResult = FAWAPI.ListItems2(Token, Path, 20, 1)
    If ListItemsResult.ErrorMessage <> "" Then
        MsgBox ListItemsResult.ErrorMessage
    End If
    ListItemsResult = Nothing
    FAWAPI = Nothing
End Sub

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
Path	String	Yes	Full FilesAnywhere path of the folder or subfolder in the form \UserVolume\FolderName
PageSize	Integer	Yes	The maximum number of items to return. If PageSize is passed as 0 then API will consider default value as 10000 for PageSize. It will return default 10000 items for input path when PageSize passed as 0.
PageNum	Integer	Yes	The page of results to return. When PageNum is passed as 0 then default API will return the item(s) for PageNum = 1.

Response

ListItemsResult

ListItemsResult

Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ListItems method was executed. If the ListItems call was successful, the ErrorMessage is blank
Items	Array of type Item	An array of Items
PageNum	Int	Current page Number for which items are returned.
PageCount	Int	Total number of pages for input PageSize.

An Item has the following properties:

Property Name	Data Type	Required	Description
Type	String	Yes	Type can either be a 'file' or a 'folder'
Path	String	Yes	Full FilesAnywhere path of the Item in the form \UserVolume\FolderName\ItemName

ListLogActivities

Syntax

ListLogActivitiesResult =
ListLogActivities(Token,StartDate,EndDate,FolderOrFile,ActivityType,SubType)

Usage

Use the ListLogActivities method to list activities to a file or folder.

The following parameters are needed to make the ListLogActivities call:

- Token. (The Token ID returned by the last AccountLogin call).
- StartDate. (The date marking the beginning of the activity list returned).
- EndDate. (The date marking the end of the activity list returned).
- FolderOrFile. (The path to the folder or file to return activities for).
- ActivityType. (Filter of activity types to return).
- SubType. (Filter of activity sub types to return).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/ListLogActivities"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ListLogActivities xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
```

```

    <StartDate>date</StartDate>
    <EndDate>date</EndDate>
    <FolderOrFile>string</FolderOrFile>
    <ActivityType>Undefined or AccessControl or FormDelete or Copy or
DeleteFolderFile or RestoreData or AdminAddUser or PurgeUser or
FileVersion or AdminEditUser or GroupShare or CopyIndexing or
SendLoginDetails or Move or CreateFolder or GeneralNotification or
ImageToPDFConversion or Download or RenameFolderFile or
AdminSettingsChanged or Upload or ChangePwd or ResetAccount or
SetAccountStatus or ZohoRequest</ActivityType>
    <SubType>
      <fileVersionActivity>Undefined or Add or Update or
Delete</fileVersionActivity>
      <formActivity>Undefined or DeleteEFormData or
DeleteEFormDataAndForm</formActivity>
      <groupShareActivity>Undefined or Add or Update or
Delete</groupShareActivity>
      <uploadActivity>Undefined or API or Classic or DropBoxClassic or
DropBoxFlash or Flash or Java or LocalClassic or UltimateUploader or Zoho
or Form</uploadActivity>
    </SubType>
  </ListLogActivities>
</soap:Body>
</soap:Envelope>

```

```

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ListLogActivitiesResponse xmlns="http://api.filesanywhere.com/">
      <ListLogActivitiesResult>
        <ErrorMessage>string</ErrorMessage>
        <Items>
          <ListLogActivitiesItem>
            <ActionDate>dateTime</ActionDate>
            <ActivityType>Undefined or AccessControl or FormDelete or Copy
or DeleteFolderFile or RestoreData or AdminAddUser or PurgeUser or
FileVersion or AdminEditUser or GroupShare or CopyIndexing or
SendLoginDetails or Move or CreateFolder or GeneralNotification or
ImageToPDFConversion or Download or RenameFolderFile or
AdminSettingsChanged or Upload or ChangePwd or ResetAccount or
SetAccountStatus or ZohoRequest</ActivityType>
            <AppUserName>string</AppUserName>
            <SourcePath>string</SourcePath>
            <DestinationPath>string</DestinationPath>
            <SubActivity xsi:nil="true" />
            <SourceItemType>Folder or File</SourceItemType>

```

```
        <Size>long</Size>
        <Comments>string</Comments>
    </ListLogActivitiesItem>
</ListLogActivitiesItem>
    <ActionDate>dateTime</ActionDate>
    <ActivityType>Undefined or AccessControl or FormDelete or Copy
or DeleteFolderFile or RestoreData or AdminAddUser or PurgeUser or
FileVersion or AdminEditUser or GroupShare or CopyIndexing or
SendLoginDetails or Move or CreateFolder or GeneralNotification or
ImageToPDFConversion or Download or RenameFolderFile or
AdminSettingsChanged or Upload or ChangePwd or ResetAccount or
SetAccountStatus or ZohoRequest</ActivityType>
    <AppUserName>string</AppUserName>
    <SourcePath>string</SourcePath>
    <DestinationPath>string</DestinationPath>
    <SubActivity xsi:nil="true" />
    <SourceItemType>Folder or File</SourceItemType>
    <Size>long</Size>
    <Comments>string</Comments>
</ListLogActivitiesItem>
</Items>
</ListLogActivitiesResult>
</ListLogActivitiesResponse>
</soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ListLogActivities xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <StartDate>date</StartDate>
      <EndDate>date</EndDate>
      <FolderOrFile>string</FolderOrFile>
      <ActivityType>Undefined or AccessControl or FormDelete or Copy or
DeleteFolderFile or RestoreData or AdminAddUser or PurgeUser or
FileVersion or AdminEditUser or GroupShare or CopyIndexing or
SendLoginDetails or Move or CreateFolder or GeneralNotification or
ImageToPDFConversion or Download or RenameFolderFile or
AdminSettingsChanged or Upload or ChangePwd or ResetAccount or
SetAccountStatus or ZohoRequest</ActivityType>
      <SubType>
        <fileVersionActivity>Undefined or Add or Update or
Delete</fileVersionActivity>
        <formActivity>Undefined or DeleteEFormData or
DeleteEFormDataAndForm</formActivity>
        <groupShareActivity>Undefined or Add or Update or
Delete</groupShareActivity>
        <uploadActivity>Undefined or API or Classic or DropBoxClassic or
DropBoxFlash or Flash or Java or LocalClassic or UltimateUploader or Zoho
or Form</uploadActivity>
      </SubType>
    </ListLogActivities>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
```

Content-Length: `length`

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ListLogActivitiesResponse xmlns="http://api.filesanywhere.com/">
      <ListLogActivitiesResult>
        <ErrorMessage>string</ErrorMessage>
        <Items>
          <ListLogActivitiesItem>
            <ActionDate>dateTime</ActionDate>
            <ActivityType>Undefined or AccessControl or FormDelete or Copy
or DeleteFolderFile or RestoreData or AdminAddUser or PurgeUser or
FileVersion or AdminEditUser or GroupShare or CopyIndexing or
SendLoginDetails or Move or CreateFolder or GeneralNotification or
ImageToPDFConversion or Download or RenameFolderFile or
AdminSettingsChanged or Upload or ChangePwd or ResetAccount or
SetAccountStatus or ZohoRequest</ActivityType>
            <AppUserName>string</AppUserName>
            <SourcePath>string</SourcePath>
            <DestinationPath>string</DestinationPath>
            <SubActivity xsi:nil="true" />
            <SourceItemType>Folder or File</SourceItemType>
            <Size>long</Size>
            <Comments>string</Comments>
          </ListLogActivitiesItem>
          <ListLogActivitiesItem>
            <ActionDate>dateTime</ActionDate>
            <ActivityType>Undefined or AccessControl or FormDelete or Copy
or DeleteFolderFile or RestoreData or AdminAddUser or PurgeUser or
FileVersion or AdminEditUser or GroupShare or CopyIndexing or
SendLoginDetails or Move or CreateFolder or GeneralNotification or
ImageToPDFConversion or Download or RenameFolderFile or
AdminSettingsChanged or Upload or ChangePwd or ResetAccount or
SetAccountStatus or ZohoRequest</ActivityType>
            <AppUserName>string</AppUserName>
            <SourcePath>string</SourcePath>
            <DestinationPath>string</DestinationPath>
            <SubActivity xsi:nil="true" />
            <SourceItemType>Folder or File</SourceItemType>
            <Size>long</Size>
            <Comments>string</Comments>
          </ListLogActivitiesItem>
        </Items>
      </ListLogActivitiesResult>
    </ListLogActivitiesResponse>
  </soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
StartDate	DateTime	Yes	The date marking the start of the log activities returned.
EndDate	DateTime	Yes	The date marking the end of the log activities returned.
FolderOrFile	String	Yes	The file path or folder base path to return activities for.
ActivityType	Enum	No	Filter for types of activities to return.
SubType	Object	No	Filter for sub types of activities to return.

Response

ListLogActivitiesResult

ListLogActivitiesResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ListLogActivities method was executed. If the ListLogActivities call was successful, the ErrorMessage is blank.
Items.ActionDate	DateTime	The datetime stamp the action was performed.
Items.ActivityType	Enum	The type of activity performed.
Items.AppUserName	String	The username who performed the action.
Items.SourcePath	String	The source path of the file or folder acted upon.

Items.DestinationPath	String	The destination path of the file or folder acted upon.
Items.SubActivity	Object	The activity sub-type.
Items.SourceItemtype	Enum	The source file or folder type.
Items.Size	Long	The size of the file, or 0 for folder.
Items.Comments	String	The comments associated with the action.

ListMeetings

Syntax

ListMeetingsResult= ListMeetings(Token,Role,OrganizerOrAttendee,FromDate,ToDate,
,SortBy,SortOrder)

Usage

Use the ListMeetings method to list the meetings a user is attending.

The following parameters are needed to make the ListMeetings call:

- Token. (The Token ID returned by the last AccountLogin call).
- Role. (A filter used to distinguish meetings where the user is an organizer, an attendee, or either).
- OrganizerOrAttendee. (The user identifier for the meeting organizer or attendee).
- FromDate. (Date range start filter).
- ToDate. (Date range end filter).
- SortBy. (The meeting property to sort the list results).
- SortOrder. (Ascending or descending sort).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/ListMeetings"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ListMeetings xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Role>None or Organizer or Attendee or OrganizerOrAttendee</Role>
      <OrganizerOrAttendee>
        <ClientId>int</ClientId>
        <Username>string</Username>
      </OrganizerOrAttendee>
      <FromDate>dateTime</FromDate>
      <ToDate>dateTime</ToDate>
      <SortBy>MeetingName or MeetingDate or Role or SendReminders or
RemindersSent</SortBy>
      <SortOrder>ASC or DESC</SortOrder>
    </ListMeetings>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ListMeetingsResponse xmlns="http://api.filesanywhere.com/">
      <ListMeetingsResult>
        <ErrorMessage>string</ErrorMessage>
        <Meetings>
          <MeetingFromList>
            <MeetingId>long</MeetingId>
            <MeetingName>string</MeetingName>
            <Organizer xsi:nil="true" />
            <Role>None or Organizer or Attendee or
OrganizerOrAttendee</Role>
            <MeetingDate>dateTime</MeetingDate>
            <MeetingDescription>string</MeetingDescription>
            <SendReminder>boolean</SendReminder>
            <ReminderLeadTime>None or FiveMinutes or FifteenMinutes or
ThirtyMinutes or FortyFiveMinutes or OneHour</ReminderLeadTime>
            <ReminderSent>boolean</ReminderSent>
          </MeetingFromList>
          <MeetingFromList>
            <MeetingId>long</MeetingId>
            <MeetingName>string</MeetingName>
            <Organizer xsi:nil="true" />
```

```

    <Role>None or Organizer or Attendee or
    OrganizerOrAttendee</Role>
    <MeetingDate>dateTime</MeetingDate>
    <MeetingDescription>string</MeetingDescription>
    <SendReminder>boolean</SendReminder>
    <ReminderLeadTime>None or FiveMinutes or FifteenMinutes or
    ThirtyMinutes or FortyFiveMinutes or OneHour</ReminderLeadTime>
    <ReminderSent>boolean</ReminderSent>
  </MeetingFromList>
</Meetings>
</ListMeetingsResult>
</ListMeetingsResponse>
</soap:Body>
</soap:Envelope>

```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```

POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ListMeetings xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Role>None or Organizer or Attendee or OrganizerOrAttendee</Role>
      <OrganizerOrAttendee>
        <ClientId>int</ClientId>
        <Username>string</Username>
      </OrganizerOrAttendee>
      <FromDate>dateTime</FromDate>
      <ToDate>dateTime</ToDate>
      <SortBy>MeetingName or MeetingDate or Role or SendReminders or
      RemindersSent</SortBy>
      <SortOrder>ASC or DESC</SortOrder>
    </ListMeetings>
  </soap12:Body>
</soap12:Envelope>

```

```

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ListMeetingsResponse xmlns="http://api.filesanywhere.com/">
      <ListMeetingsResult>
        <ErrorMessage>string</ErrorMessage>
        <Meetings>
          <MeetingFromList>
            <MeetingId>long</MeetingId>
            <MeetingName>string</MeetingName>
            <Organizer xsi:nil="true" />
            <Role>None or Organizer or Attendee or
OrganizerOrAttendee</Role>
            <MeetingDate>dateTime</MeetingDate>
            <MeetingDescription>string</MeetingDescription>
            <SendReminder>boolean</SendReminder>
            <ReminderLeadTime>None or FiveMinutes or FifteenMinutes or
ThirtyMinutes or FortyFiveMinutes or OneHour</ReminderLeadTime>
            <ReminderSent>boolean</ReminderSent>
          </MeetingFromList>
          <MeetingFromList>
            <MeetingId>long</MeetingId>
            <MeetingName>string</MeetingName>
            <Organizer xsi:nil="true" />
            <Role>None or Organizer or Attendee or
OrganizerOrAttendee</Role>
            <MeetingDate>dateTime</MeetingDate>
            <MeetingDescription>string</MeetingDescription>
            <SendReminder>boolean</SendReminder>
            <ReminderLeadTime>None or FiveMinutes or FifteenMinutes or
ThirtyMinutes or FortyFiveMinutes or OneHour</ReminderLeadTime>
            <ReminderSent>boolean</ReminderSent>
          </MeetingFromList>
        </Meetings>
      </ListMeetingsResult>
    </ListMeetingsResponse>
  </soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
Role	Enum	Yes	Filter to distinguish meetings the user is an organizer for, an attendee of, or either.
OrganizerOrAttendee	UserIdentifier	No	The user identifier to search for. If null, then the current user is used.
FromDate	DateTime	No	Date range start filter.
ToDate	DateTime	No	Date range end filter.
SortBy	Enum	Yes	The meeting property to sort the results by.
SortOrder	Enum	Yes	Ascending or descending sort.

Response**ListMeetingsResult**

ListMeetingsResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ListMeetings method was executed. If the ListMeetings call was successful, the ErrorMessage is blank.
Meetings	MeetingFromList()	An array of populated MeetingFromList structures.

A MeetingFromList has the following properties:

Property Name	Data Type	Required	Description
MeetingId	Long	Yes	Unique identifier for the meeting.
MeetingName	String	Yes	String identifier or title for the meeting.
Organizer	UserIdentifier	Yes	User identifier representing the organizer.
Role	Enum	Yes	The role of the current user in the meeting.
MeetingDate	DateTime	Yes	The scheduled start date of the meeting.
DurationInMinutes	Integer	Yes	The planned length of the meeting in minutes.
MeetingDescription	String	Yes	The agenda or discussion plans for the meeting.
SendReminder	Boolean	Yes	Flag indicating if reminder notifications are to be sent to attendees.
ReminderLeadTime	Enum	Yes	The number of minutes before the meeting starts to send reminder notifications.
ReminderSent	Boolean	Yes	Flag indicating if reminder notifications have been sent.

ListUserVolumes2

This method lists all user volumes which are accessible from the user's FilesAnywhere account. This includes the user's own account, inbox, version history, and other GroupShares if they exist.

Syntax

ListUserVolumesResult = ListUserVolumes2(Token)

Usage

Use the ListUserVolumes2 method to get a list of all shares to the logged in user. The list also includes the current user's private storage, inbox, and version history directories.

The following parameters are needed to make the ListUserVolumes2 call:

- Token (The Token ID returned by the last AccountLogin call).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/ListUserVolumes2"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ListUserVolumes2 xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </ListUserVolumes2>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ListUserVolumes2Response xmlns="http://api.filesanywhere.com/">
      <ListUserVolumesResult>
        <ErrorMessage>string</ErrorMessage>
        <UserVolumes>
          <UserVolumev2>
            <Username>string</Username>
            <Permission>string</Permission>
            <Path>string</Path>
          </UserVolumev2>
          <UserVolumev2>
            <Username>string</Username>
            <Permission>string</Permission>
            <Path>string</Path>
          </UserVolumev2>
        </UserVolumes>
      </ListUserVolumesResult>
    </ListUserVolumes2Response>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ListUserVolumes2 xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </ListUserVolumes2>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```



```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ListUserVolumes2Response xmlns="http://api.filesanywhere.com/">
      <ListUserVolumesResult>
        <ErrorMessage>string</ErrorMessage>
        <UserVolumes>
          <UserVolumev2>
            <Username>string</Username>
            <Permission>string</Permission>
            <Path>string</Path>
          </UserVolumev2>
          <UserVolumev2>
            <Username>string</Username>
            <Permission>string</Permission>
            <Path>string</Path>
          </UserVolumev2>
        </UserVolumes>
      </ListUserVolumesResult>
    </ListUserVolumes2Response>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub ListUserVolumes ()
  Dim FAWAPI As New WebReference.FAWAPI
  Dim ListUserVolumesResult As WebReference.ListUserVolumesv2Result
  ListUserVolumesResult = FAWAPI.ListUserVolumes2 (Token)
  If ListUserVolumesResult.ErrorMessage <> "" Then
    MsgBox ListUserVolumesResult.ErrorMessage
  End If
  ListUserVolumesResult = Nothing
  FAWAPI = Nothing
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call

Response

ListUserVolumesResult

ListUserVolumesResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ListUserVolumes method was executed. If the ListUserVolumes call was successful, the ErrorMessage is blank
UserVolumes	Array of type UserVolumev2	An array of UserVolumesv2

A UserVolumev2 has the following properties:

Property Name	Data Type	Required	Description
Username	String	Yes	This is the FilesAnywhere account Username
Permission	String	Yes	This can have the values 'R' for (Read only), 'F' for (Full Access) and 'C' for (Create and Update)
Path	String	Yes	The path to the volume shared to the user which can be used in ListItems and other methods.

MoveItems

Syntax

MoveItemsResult= MoveItems(Token,SrcPath,DstPath)

Usage

Use the MoveItems method to move one or more files and/or folders.

The following parameters are needed to make the MoveItems call:

- Token. (The Token ID returned by the last AccountLogin call).
- SrcPath. (List of paths to files or folders to move).
- DstPath. (The path to the folder to move the source paths to).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/MoveItems"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <MoveItems xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <SrcPath>string</SrcPath>
      <SrcPath>string</SrcPath>
      <DstPath>string</DstPath>
    </MoveItems>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <MoveItemsResponse xmlns="http://api.filesanywhere.com/">
      <MoveItemsResult>
        <ErrorMessage>string</ErrorMessage>
        <MovedItems>
          <MovedItem>
            <ErrorMessage>string</ErrorMessage>
            <SrcPath>string</SrcPath>
            <DstPath>string</DstPath>
            <Moved>boolean</Moved>
          </MovedItem>
          <MovedItem>
            <ErrorMessage>string</ErrorMessage>
            <SrcPath>string</SrcPath>
            <DstPath>string</DstPath>
            <Moved>boolean</Moved>
          </MovedItem>
        </MovedItems>
      </MoveItemsResult>
    </MoveItemsResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <MoveItems xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <SrcPath>string</SrcPath>
      <SrcPath>string</SrcPath>
      <DstPath>string</DstPath>
    </MoveItems>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <MoveItemsResponse xmlns="http://api.filesanywhere.com/">
      <MoveItemsResult>
        <ErrorMessage>string</ErrorMessage>
        <MovedItems>
          <MovedItem>
            <ErrorMessage>string</ErrorMessage>
            <SrcPath>string</SrcPath>
            <DstPath>string</DstPath>
            <Moved>boolean</Moved>
          </MovedItem>
          <MovedItem>
            <ErrorMessage>string</ErrorMessage>
            <SrcPath>string</SrcPath>
            <DstPath>string</DstPath>
            <Moved>boolean</Moved>
          </MovedItem>
        </MovedItems>
      </MoveItemsResult>
    </MoveItemsResponse>
  </soap12:Body>
</soap12:Envelope>
```

```

    </MoveItemsResult>
  </MoveItemsResponse>
</soap12:Body>
</soap12:Envelope>

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
SrcPath	String()	Yes	Paths to files and/or folders to be moved.
DstPath	String	Yes	The folder path to move the source items to.

Response

MoveItemsResultv2

MoveItemsResultv2 Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the MoveItems method was executed. If the MoveItems call was successful, the ErrorMessage is blank.
MovedItems.ErrorMessage	String	Error message pertaining to one specific file.
MovedItems.SrcPath	String	The path of the source file or folder.
MovedItems.DstPath	String	The path of the destination file or folder.
Items.Moved	Boolean	Flag indicating the item was

		moved successfully.
--	--	---------------------

RegisterDevice

Syntax

RegisterDeviceResult= RegisterDevice(Token,DeviceId,DeviceToken)

Usage

Use the RegisterDevice method to register a device for notifications.

The following parameters are needed to make the RegisterDevice call:

- Token. (The Token ID returned by the last AccountLogin call).
- DeviceId. (Physical ID of the device).
- DeviceToken. (Logical ID for the device or session).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/RegisterDevice"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <RegisterDevice xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <DeviceId>string</DeviceId>
      <DeviceToken>string</DeviceToken>
    </RegisterDevice>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
```



```
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <RegisterDeviceResponse xmlns="http://api.filesanywhere.com/">
      <RegisterDeviceResult>
        <ErrorMessage>string</ErrorMessage>
        <MinimumSyncCheckFrequency>long</MinimumSyncCheckFrequency>
        <MinimumNotificationCheckFrequency>long</MinimumNotificationCheckFrequency
      >
      </RegisterDeviceResult>
    </RegisterDeviceResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <RegisterDevice xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <DeviceId>string</DeviceId>
      <DeviceToken>string</DeviceToken>
    </RegisterDevice>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <RegisterDeviceResponse xmlns="http://api.filesanywhere.com/">
      <RegisterDeviceResult>
        <ErrorMessage>string</ErrorMessage>
        <MinimumSyncCheckFrequency>long</MinimumSyncCheckFrequency>

<MinimumNotificationCheckFrequency>long</MinimumNotificationCheckFrequency
>
      </RegisterDeviceResult>
    </RegisterDeviceResponse>
  </soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
DeviceId	String	Yes	The physical ID of the device.
DeviceToken	String	Yes	The logical ID of the device or session.

Response

RegisterDeviceResult

RegisterDeviceResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the RegisterDevice method was executed. If the RegisterDevice call was successful, the ErrorMessage is blank.
MinimumSyncCheckFrequency	Long	The minimum number of seconds to wait between

		polling intervals for synchronization.
MinimumNotificationCheckFrequency	Long	The minimum number of seconds to wait between polling intervals for notifications.

ScheduleMeeting

Syntax

ScheduleMeetingResult=
ScheduleMeeting(Token,Organizer,MeetingName,MeetingDate,DurationInMinutes,MeetingDescription,SendReminder,ReminderLeadTime,Attendees)

Usage

Use the ScheduleMeeting method to schedule a meeting between 2 or more users.

The following parameters are needed to make the ScheduleMeeting call:

- Token. (The Token ID returned by the last AccountLogin call).
- Organizer. (The user identifier for the meeting organizer).
- MeetingName. (A string identifier or title for the meeting).
- MeetingDate. (The scheduled start date and time for the meeting).
- DurationInMinutes. (The planned length of the meeting in minutes).
- MeetingDescription. (Agenda or planned discussion).
- SendReminder. (Flag indicating to notify attendees of the meeting).
- ReminderLeadTime.(The number of minutes before the meeting starts to send the reminder).
- Attendees. (User identifiers for all parties invited to attend the meeting).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/ScheduleMeeting"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ScheduleMeeting xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Organizer />
      <MeetingName>string</MeetingName>
      <MeetingDate>dateTime</MeetingDate>
      <DurationInMinutes>int</DurationInMinutes>
      <MeetingDescription>string</MeetingDescription>
      <SendReminder>boolean</SendReminder>
      <ReminderLeadTime>None or FiveMinutes or FifteenMinutes or
ThirtyMinutes or FortyFiveMinutes or OneHour</ReminderLeadTime>
      <Attendees>
        <Attendee />
        <Attendee />
      </Attendees>
    </ScheduleMeeting>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ScheduleMeetingResponse xmlns="http://api.filesanywhere.com/">
      <ScheduleMeetingResult>
        <ErrorMessage>string</ErrorMessage>
        <Meeting>
          <MeetingId>long</MeetingId>
          <MeetingName>string</MeetingName>
          <Organizer />
          <Role>None or Organizer or Attendee or
OrganizerOrAttendee</Role>
          <MeetingDate>dateTime</MeetingDate>
          <DurationInMinutes>int</DurationInMinutes>
          <MeetingDescription>string</MeetingDescription>
          <SendReminder>boolean</SendReminder>
          <ReminderLeadTime>None or FiveMinutes or FifteenMinutes or
ThirtyMinutes or FortyFiveMinutes or OneHour</ReminderLeadTime>
          <Attendees>
            <AttendeeId xsi:nil="true" />
            <AttendeeId xsi:nil="true" />
          </Attendees>
        </Meeting>
      </ScheduleMeetingResult>
    </ScheduleMeetingResponse>
  </soap:Body>
</soap:Envelope>
```

```
        <ReminderSent>boolean</ReminderSent>
    </Meeting>
</ScheduleMeetingResult>
</ScheduleMeetingResponse>
</soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ScheduleMeeting xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Organizer />
      <MeetingName>string</MeetingName>
      <MeetingDate>dateTime</MeetingDate>
      <DurationInMinutes>int</DurationInMinutes>
      <MeetingDescription>string</MeetingDescription>
      <SendReminder>boolean</SendReminder>
      <ReminderLeadTime>None or FiveMinutes or FifteenMinutes or
ThirtyMinutes or FortyFiveMinutes or OneHour</ReminderLeadTime>
      <Attendees>
        <Attendee />
        <Attendee />
      </Attendees>
    </ScheduleMeeting>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ScheduleMeetingResponse xmlns="http://api.filesanywhere.com/">
      <ScheduleMeetingResult>
        <ErrorMessage>string</ErrorMessage>
        <Meeting>
          <MeetingId>long</MeetingId>
          <MeetingName>string</MeetingName>
          <Organizer />
          <Role>None or Organizer or Attendee or
OrganizerOrAttendee</Role>
          <MeetingDate>dateTime</MeetingDate>
          <DurationInMinutes>int</DurationInMinutes>
          <MeetingDescription>string</MeetingDescription>
          <SendReminder>boolean</SendReminder>
          <ReminderLeadTime>None or FiveMinutes or FifteenMinutes or
ThirtyMinutes or FortyFiveMinutes or OneHour</ReminderLeadTime>
          <Attendees>
            <AttendeeId xsi:nil="true" />
            <AttendeeId xsi:nil="true" />
          </Attendees>
          <ReminderSent>boolean</ReminderSent>
        </Meeting>
      </ScheduleMeetingResult>
    </ScheduleMeetingResponse>
  </soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
Organizer	OrganizerId	No	The user identifier for the meeting organizer. If null, then the current user is the organizer.
MeetingName	String	Yes	String identifier or title of the meeting.
MeetingDate	DateTime	Yes	The scheduled start date and time of the meeting.
DurationInMinutes	Integer	Yes	The planned length of the meeting in

			minutes.
MeetingDescription	String	Yes	Agenda or discussion plans for the meeting.
SendReminder	Boolean	Yes	Flag indicating to send reminder notifications for the meeting to attendees.
ReminderLeadTime	Enum	Yes	The number of minutes before the meeting starts to send reminder notifications.
Attendees	AttendeeId()	Yes	Array of user identifiers representing users invited to the meeting.

Response

ScheduleMeetingResult

ScheduleMeetingResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the ScheduleMeeting method was executed. If the ScheduleMeeting call was successful, the ErrorMessage is blank.
Meeting	Meeting	A populated meeting structure.

A Meeting has the following properties:

Property Name	Data Type	Required	Description
MeetingId	Long	Yes	Unique identifier for the meeting.
MeetingName	String	Yes	String identifier or title for the meeting.
Organizer	UserIdentifier	Yes	User identifier representing the organizer.

Role	Enum	Yes	The role of the current user in the meeting.
MeetingDate	DateTime	Yes	The scheduled start date of the meeting.
DurationInMinutes	Integer	Yes	The planned length of the meeting in minutes.
MeetingDescription	String	Yes	The agenda or discussion plans for the meeting.
SendReminder	Boolean	Yes	Flag indicating if reminder notifications are to be sent to attendees.
ReminderLeadTime	Enum	Yes	The number of minutes before the meeting starts to send reminder notifications.
Attendees	UserIdentifier()	Yes	Array of user identifiers representing attendees.
ReminderSent	Boolean	Yes	Flag indicating if reminder notifications have been sent.

SearchFiles

Syntax

SearchFilesResult=
SearchFiles(Token,Path,MP,ItemName,TextInFile,Comments,Tag,SizeUnit,SizeFrom,SizeTo,DateFrom,DateTo,CheckFolderName,CheckSubFolders,CheckIndex,SortBy,SortOrder,StartPage,PageSize)

Usage

Use the SearchFiles method to search for files and folders by properties or content.

The following parameters are needed to make the SearchFiles call:

- **Token.** (The Token ID returned by the last AccountLogin call).
- **Path.** (The base path in which to search).
- **MP.** (Model Param).
- **ItemName.** (The name of the file or folder to search for).
- **TextInFile.** (Content within the file to search for).
- **Comments.** (Comments to search for).
- **Tag.** (Tag to search for).
- **SizeUnit.** (The unit of measurement for size filters).
- **SizeFrom.** (The least size in units of SizeUnit to search for).
- **SizeTo.** (The greatest size in units of SizeUnit to search for).
- **DateFrom.** (The least date to search for).
- **DateTo.** (The greatest date to search for).
- **CheckFolderName.** (Flag indicating to check the base path).
- **CheckSubFolders.** (Flag indicating to perform a recursive search).
- **CheckIndex.** (Flag indicating to search the index file).

- **SortBy.** (Indicates the desired sort criteria for the results).
- **SortOrder.** (Indicates ascending or descending ordering of results).
- **StartPage.** (Indicates the current page number in the results).
- **PageSize.** (Indicates the size of paged results).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/SearchFiles"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SearchFiles xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
      <MP>string</MP>
      <ItemName>string</ItemName>
      <TextInFile>string</TextInFile>
      <Comments>string</Comments>
      <Tag>string</Tag>
      <SizeUnit>None or KiloBytes or MegaBytes or GigaBytes</SizeUnit>
      <SizeFrom>long</SizeFrom>
      <SizeTo>long</SizeTo>
      <DateFrom>dateTime</DateFrom>
      <DateTo>dateTime</DateTo>
      <CheckFolderName>boolean</CheckFolderName>
      <CheckSubFolders>boolean</CheckSubFolders>
      <CheckIndex>boolean</CheckIndex>
      <SortBy>NAME or TYPETEXT or MODIFIEDDATE or SIZE</SortBy>
      <SortOrder>ASC or DESC</SortOrder>
      <StartPage>int</StartPage>
      <PageSize>int</PageSize>
    </SearchFiles>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SearchFilesResponse xmlns="http://api.filesanywhere.com/">
      <SearchFilesResult>
        <ErrorMessage>string</ErrorMessage>
        <Items>
          <CAddListItem>
            <Folder>string</Folder>
            <Name>string</Name>
            <Parameter>string</Parameter>
            <TotalSize>long</TotalSize>
            <FileID>string</FileID>
            <SortField>string</SortField>
          </CAddListItem>
          <CAddListItem>
            <Folder>string</Folder>
            <Name>string</Name>
            <Parameter>string</Parameter>
            <TotalSize>long</TotalSize>
            <FileID>string</FileID>
            <SortField>string</SortField>
          </CAddListItem>
        </Items>
      </SearchFilesResult>
    </SearchFilesResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SearchFiles xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
      <MP>string</MP>
      <ItemName>string</ItemName>
      <TextInFile>string</TextInFile>
      <Comments>string</Comments>
      <Tag>string</Tag>
      <SizeUnit>None or KiloBytes or MegaBytes or GigaBytes</SizeUnit>
      <SizeFrom>long</SizeFrom>
      <SizeTo>long</SizeTo>
      <DateFrom>dateTime</DateFrom>
      <DateTo>dateTime</DateTo>
      <CheckFolderName>boolean</CheckFolderName>
      <CheckSubFolders>boolean</CheckSubFolders>
      <CheckIndex>boolean</CheckIndex>
      <SortBy>NAME or TYPETEXT or MODIFIEDDATE or SIZE</SortBy>
      <SortOrder>ASC or DESC</SortOrder>
      <StartPage>int</StartPage>
      <PageSize>int</PageSize>
    </SearchFiles>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SearchFilesResponse xmlns="http://api.filesanywhere.com/">
      <SearchFilesResult>
        <ErrorMessage>string</ErrorMessage>
        <Items>
```

```

    <CAddListItem>
      <Folder>string</Folder>
      <Name>string</Name>
      <Parameter>string</Parameter>
      <TotalSize>long</TotalSize>
      <FileID>string</FileID>
      <SortField>string</SortField>
    </CAddListItem>
  <CAddListItem>
    <Folder>string</Folder>
    <Name>string</Name>
    <Parameter>string</Parameter>
    <TotalSize>long</TotalSize>
    <FileID>string</FileID>
    <SortField>string</SortField>
  </CAddListItem>
</Items>
</SearchFilesResult>
</SearchFilesResponse>
</soap12:Body>
</soap12:Envelope>

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
Path	String	Yes	The base path in which to search.
MP	String	Yes	Model Param.
ItemName	String	Yes	The name of the file or folder to search for.
TextInFile	String	Yes	The text to search for within the files.
Comments	String	Yes	The comments to search for.
Tag	String	Yes	The tag to search for.
SizeUnit	Enum	Yes	The unit of measure for size parameters.
SizeFrom	Long	No	The least size in units of SizeUnit to search for.
SizeTo	Long	No	The greatest size in units of SizeUnit to search for.

DateFrom	DateTime	No	The least date to search for.
DateTo	DateTime	No	The greatest date to search for.
CheckFolderName	Boolean	Yes	Flag indicating to check the provided Path.
CheckSubFolders	Boolean	Yes	Flag indicating to perform a recursive search.
CheckIndex	Boolean	Yes	Flag indicating to search the index file.
SortBy	Enum	Yes	Indicates the desired sort criteria for results.
SortOrder	Enum	Yes	Indicates ascending or descending sort ordering of results.
StartPage	Integer	Yes	Indicates the current page number within the paged results.
PageSize	Integer	Yes	Indicates the size of the paged results.

Response

SearchFilesResult

SearchFilesResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the SearchFiles method was executed. If the SearchFiles call was successful, the ErrorMessage is blank.
Items	Object()	The results of the search.

How to search for files in the GroupShare Folders?

To search for files in all the GroupShare folders, user has to perform below steps

- a) Call ListUserVolumns2 API function which will return all the root folder names including shared folders.
- b) Call SearchFiles API function to search files in each GroupShare folder path that were returned in the above function call and club the results.

SendDropboxLink2

Syntax

SendElinkResult=

SendDropboxLink2(Token,DropFolderPath,RecipientEmailAddresses,ShareDays,RecordFileHistoryLog,NotifyDownloadByEmail,SendReadReceipt,LinkInFileAttachment,OverWriteFiles,SubFolderNamePrompt,UserEmailAsSender,EmailSubject,EmailBody,ElinkPassword,RedirectURL,RedirectText,DisplayName,DisplayEmailAddress)

Usage

Use the SendDropboxLink2 method to create and send a FileReceive Link.

The following parameters are needed to make the SendDropboxLink2 call:

- Token. (The Token ID returned by the last AccountLogin call).
- DropFolderPath. (The path to the folder where the FileReceive Link will exist).
- RecipientEmailAddresses. (List of email addresses to send the FileReceive Link to).
- ShareDays. (Number of days the FileReceive Link will be active).
- RecordFileHistoryLog. (Flag indicating to log file activity).
- NotifyDownloadByEmail. (Flag indicating an email notification to be sent to recipients).
- SendReadReceipt. (Flag indicating a read receipt should be generated for the email notification when it is read).
- LinkInFileAttachment. ()
- OverWriteFiles. (Flag indicating files can be overwritten).
- SubFolderNamePrompt. (Text displayed to create a user partition in the FileReceive Link for each user).

- **UserEmailAsSender.** (Flag indicating to send email notifications from the system email account or the user's email account).
- **EmailSubject.** (The subject line of the email notification).
- **EmailBody.** (The body of the email notification).
- **ElinkPassWord.** (The password assigned to the FileReceive Link).
- **RedirectURL.** (The URL to return the user to).
- **RedirectText.** (The text displayed to redirect the user back to a provided URL).
- **DisplayName.** (Flag indicating to display the FileReceive Link owner's name).
- **DisplayEmailAddress.** (Flag indicating to display the FileReceive Link owner's email address).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/SendDropboxLink2"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendDropboxLink2 xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <DropFolderPath>string</DropFolderPath>
      <RecipientsEmailAddresses>string</RecipientsEmailAddresses>
      <ShareDays>int</ShareDays>
      <RecordFileHistoryLog>string</RecordFileHistoryLog>
      <NotifyDownloadByEmail>string</NotifyDownloadByEmail>
      <SendReadReceipt>string</SendReadReceipt>
      <LinkInFileAttachment>string</LinkInFileAttachment>
      <OverWriteFiles>string</OverWriteFiles>
      <SubFolderNamePrompt>string</SubFolderNamePrompt>
      <UserEmailAsSender>string</UserEmailAsSender>
      <EmailSubject>string</EmailSubject>
      <EmailBody>string</EmailBody>
      <ElinkPassWord>string</ElinkPassWord>
    </SendDropboxLink2>
  </soap:Body>
</soap:Envelope>
```

```
<RedirectURL>string</RedirectURL>
<RedirectText>string</RedirectText>
<DisplayName>boolean</DisplayName>
<DisplayEmailAddress>boolean</DisplayEmailAddress>
</SendDropboxLink2>
</soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendDropboxLink2Response xmlns="http://api.filesanywhere.com/">
      <SendDropboxLink3>
        <ErrorMessage>string</ErrorMessage>
        <ELinkSent>boolean</ELinkSent>
        <ELinkURLs>
          <ELinkURL>
            <URL>string</URL>
          </ELinkURL>
          <ELinkURL>
            <URL>string</URL>
          </ELinkURL>
        </ELinkURLs>
      </SendDropboxLink3>
    </SendDropboxLink2Response>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SendDropboxLink2 xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <DropFolderPath>string</DropFolderPath>
      <RecipientsEmailAddresses>string</RecipientsEmailAddresses>
      <ShareDays>int</ShareDays>
      <RecordFileHistoryLog>string</RecordFileHistoryLog>
      <NotifyDownloadByEmail>string</NotifyDownloadByEmail>
      <SendReadReceipt>string</SendReadReceipt>
      <LinkInFileAttachment>string</LinkInFileAttachment>
      <OverWriteFiles>string</OverWriteFiles>
      <SubFolderNamePrompt>string</SubFolderNamePrompt>
      <UserEmailAsSender>string</UserEmailAsSender>
      <EmailSubject>string</EmailSubject>
      <EmailBody>string</EmailBody>
      <ElinkPassWord>string</ElinkPassWord>
      <RedirectURL>string</RedirectURL>
      <RedirectText>string</RedirectText>
      <DisplayName>boolean</DisplayName>
      <DisplayEmailAddress>boolean</DisplayEmailAddress>
    </SendDropboxLink2>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SendDropboxLink2Response xmlns="http://api.filesanywhere.com/">
      <SendDropboxLink3>
        <ErrorMessage>string</ErrorMessage>
        <ELinkSent>boolean</ELinkSent>
        <ELinkURLs>
          <ElinkURL>
            <URL>string</URL>
          </ElinkURL>
          <ElinkURL>
            <URL>string</URL>
          </ElinkURL>
        </ELinkURLs>
      </SendDropboxLink3>
    </SendDropboxLink2Response>
  </soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
DropFolderPath	String	Yes	The folder path where the DrobBoxLink will exist.
RecipientEmailAddresses	String	Yes	Semi-colon-separated list of email addresses to send the FileReceive Link to.
ShareDays	Integer	Yes	The number of days the FileReceive Link will remain active.
RecordFileHistoryLog	String	Yes	Flag indicating to log activity.
NotifyDownloadByEmail	String	Yes	Flag indicating to send email notification to recipients.
SendReadReceipt	String	Yes	Flag indicating a read receipt should be generated for the email notification when it is read.
LinkInFileAttachment	String	Yes	
OverWriteFiles	String	Yes	Flag indicating files can be overwritten.
SubFolderNamePrompt	String	Yes	Text to display to create a user partition for each recipient.
UserEmailAsSender	String	Yes	Flag indicating the email notification should be sent from the system email account or the user's email account.
EmailSubject	String	Yes	Email subject in the notification.
EmailBody	String	Yes	Email body in the notification.
ElinkPassWord	String	Yes	The password assigned to the FileReceive Link.
RedirectURL	String	Yes	URL to redirect the users to.
RedirectText	String	Yes	Text displayed to redirect the users to a URL.
DisplayName	Boolean	Yes	Flag indicating to show the user's name.

DisplayEmailAddress	Boolean	Yes	Flag indicating to show the user's email address.
---------------------	---------	-----	---

Response**SendELinkResult**

SendELinkResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the SendDropboxLink2 method was executed. If the SendDropboxLink2 call was successful, the ErrorMessage is blank.
ELinkSent	Boolean	Flag indicating the link was sent to the recipients.
ELinkURLs.URL	String	URL created for each user.

SetAccountDetails

Syntax

SetAccountDetails =
SetAccountDetails(Token,EmailAddress,OldPassword,NewPassword)

Usage

Use the **SetAccountDetails** method to set account properties.

The following parameters are needed to make the **SetAccountDetails** call:

- **Token.** (The Token ID returned by the last **AccountLogin** call).
- **EmailAddress.** (The user's email address).
- **OldPassword.** (The user's current password).
- **NewPassword.** (The user's new password).

SOAP 1.1

The following is a sample **SOAP 1.1** request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/SetAccountDetails"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetAccountDetails xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <EmailAddress>string</EmailAddress>
      <OldPassword>string</OldPassword>
      <NewPassword>string</NewPassword>
    </SetAccountDetails>
  </soap:Body>
```

```
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetAccountDetailsResponse xmlns="http://api.filesanywhere.com/">
      <SetAccountDetailsResult>
        <ErrorMessage>string</ErrorMessage>
      </SetAccountDetailsResult>
    </SetAccountDetailsResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SetAccountDetails xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <EmailAddress>string</EmailAddress>
      <OldPassword>string</OldPassword>
      <NewPassword>string</NewPassword>
    </SetAccountDetails>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
```



```
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SetAccountDetailsResponse xmlns="http://api.filesanywhere.com/">
      <SetAccountDetailsResult>
        <ErrorMessage>string</ErrorMessage>
      </SetAccountDetailsResult>
    </SetAccountDetailsResponse>
  </soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
EmailAddress	String	Yes	The user's email address.
OldPassword	String	Yes	The user's current password.
NewPassword	String	No	A new password for the user.

Response

SetAccountDetailsResultv2

SetAccountDetailsResultv2 Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the SetAccountDetails method was executed. If the SetAccountDetails call was successful, the ErrorMessage is blank

SetAPNSSetting

Syntax

SetAPNSSettingResult = SetAPNSSetting (Token,NotificationType,Setting)

Usage

Use the SetAPNSSetting method to retrieve a list of notification type and their current settings for the logged on user.

The following parameters are needed to make the SetAPNSSetting call:

- Token. (The Token ID returned by the last AccountLogin call).
- NotificationType (The type of notification, listed in the NotificationTypes enum)
- Setting ("Y" := on/enabled, "N" :=off/disabled)

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/SetAPNSSetting"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetAPNSSetting xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <NotificationType>FILEACTION_ANNOTATE or FILEACTION_DELETED or
FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANGE or
GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or
```

```
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or  
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or  
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or ACCOUNTACTION_STORAGEEXCEEDED or  
NONE</NotificationType>  
  <Settings>string</Settings>  
</SetAPNSetting>  
</soap:Body>  
</soap:Envelope>
```

```
HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>  
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <SetAPNSettingResponse xmlns="http://api.filesanywhere.com/">  
      <SetAPNSettingsResult>  
        <ErrorMessage>string</ErrorMessage>  
        <NotificationType>FILEACTION_ANNOTATE or FILEACTION_DELETED or  
FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or  
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or  
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE  
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or  
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED  
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANGE or  
GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or  
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or  
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or  
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or  
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or ACCOUNTACTION_STORAGEEXCEEDED or  
NONE</NotificationType>  
        <Setting>string</Setting>  
      </SetAPNSettingsResult>  
    </SetAPNSettingResponse>  
  </soap:Body>  
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SetAPNSSetting xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <NotificationType>FILEACTION_ANNOTATE or FILEACTION_DELETED or
FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANG
e or GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or ACCOUNTACTION_STORAGEEXCEEDED or
NONE</NotificationType>
      <Settings>string</Settings>
    </SetAPNSSetting>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SetAPNSSettingResponse xmlns="http://api.filesanywhere.com/">
      <SetAPNSSettingsResult>
        <ErrorMessage>string</ErrorMessage>
        <NotificationType>FILEACTION_ANNOTATE or FILEACTION_DELETED or
FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
```

```

CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANG or
GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or ACCOUNTACTION_STORAGEEXCEEDED or
NONE</NotificationType>
    <Setting>string</Setting>
  </SetAPNSettingsResult>
</SetAPNSettingResponse>
</soap12:Body>
</soap12:Envelope>

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
NotificationType	Enum	Yes	The type of notification for the setting to take effect.
Setting	String	Yes	Flag whether the notification is to be enabled or disabled.

Response

SetAPNSettingResult

SetAPNSettingResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the SetAPNSetting method was executed. If the SetAPNSetting call was successful, the ErrorMessage is blank
NotificationType	Enum	The notification type.

Setting	String	Flag indicating the notification type is enabled or disabled.
---------	--------	---

SetAPNSettings

Syntax

SetAPNSettingsResult = SetAPNSettings(Token,Settings)

Usage

Use the SetAPNSettings method to retrieve a list of notification type and their current settings for the logged on user.

The following parameters are needed to make the SetAPNSetting call:

- Token. (The Token ID returned by the last AccountLogin call).
- Settings List of Setting type consisting of NotificationType and Setting

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/SetAPNSettings"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetAPNSettings xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Settings>
        <NotificationType>FILEACTION_ANNOTATE or FILEACTION_DELETED or
FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANGE or
GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or
```



```
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or ACCOUNTACTION_STORAGEEXCEEDED or
NONE</NotificationType>
  <Setting>string</Setting>
</Settings>
<Settings>
  <NotificationType>FILEACTION_ANNOTATE or FILEACTION_DELETED or
FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANGE or
GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or ACCOUNTACTION_STORAGEEXCEEDED or
NONE</NotificationType>
  <Setting>string</Setting>
</Settings>
</SetAPNSSettings>
</soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetAPNSSettingsResponse xmlns="http://api.filesanywhere.com/">
      <SetAPNSSettingResult>
        <ErrorMessage>string</ErrorMessage>
        <Settings>
          <APNSSetting>
            <NotificationType>FILEACTION_ANNOTATE or FILEACTION_DELETED or
FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANGE or
GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or ACCOUNTACTION_STORAGEEXCEEDED or
NONE</NotificationType>
            <Setting>string</Setting>
```

```

    </APNSSetting>
    <APNSSetting>
      <NotificationType>FILEACTION_ANNOTATE or FILEACTION_DELETED or
FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANG
e or GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or ACCOUNTACTION_STORAGEEXCEEDED or
NONE</NotificationType>
      <Setting>string</Setting>
    </APNSSetting>
  </Settings>
</SetAPNSSettingResult>
</SetAPNSSettingsResponse>
</soap:Body>
</soap:Envelope>

```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```

POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SetAPNSSettings xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Settings>
        <NotificationType>FILEACTION_ANNOTATE or FILEACTION_DELETED or
FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANG
e or GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or

```

```

FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or ACCOUNTACTION_STORAGEEXCEEDED or
NONE</NotificationType>
    <Setting>string</Setting>
</Settings>
<Settings>
    <NotificationType>FILEACTION_ANNOTATE or FILEACTION_DELETED or
FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANGE or
GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or ACCOUNTACTION_STORAGEEXCEEDED or
NONE</NotificationType>
    <Setting>string</Setting>
</Settings>
</SetAPNSettings>
</soap12:Body>
</soap12:Envelope>

```

```

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SetAPNSettingsResponse xmlns="http://api.filesanywhere.com/">
      <SetAPNSettingResult>
        <ErrorMessage>string</ErrorMessage>
        <Settings>
          <APNSetting>
            <NotificationType>FILEACTION_ANNOTATE or FILEACTION_DELETED or
FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANGE or
GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or ACCOUNTACTION_STORAGEEXCEEDED or
NONE</NotificationType>

```

```
        <Setting>string</Setting>
    </APNSSetting>
    <APNSSetting>
        <NotificationType>FILEACTION_ANNOTATE or FILEACTION_DELETED or
FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANG or
GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or ACCOUNTACTION_STORAGEEXCEEDED or
NONE</NotificationType>
        <Setting>string</Setting>
    </APNSSetting>
</Settings>
</SetAPNSSettingResult>
</SetAPNSSettingsResponse>
</soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
Settings.NotificationType	Enum	Yes	The type of notification for the setting to take effect.
Settings.Setting	String	Yes	Flag whether the notification is to be enabled or disabled.

Response

SetAPNSettingsResult

SetAPNSettingsResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the SetAPNSetting method was executed. If the SetAPNSetting call was successful, the ErrorMessage is blank
Settings.APNSetting.NotificationType	Enum	The notification type.
Settings.APNSetting.Setting	String	Flag indicating the notification type is enabled or disabled.

SetDropBoxDefaults

Syntax

`SetDropBoxDefaultsResult = SetDropBoxDefaults`
(Token,ShareDays,FlagLog,FlagEmail,FlagEmailFromSystem,EmailSubject,EmailBody,FlagAttachment,FlagReadReceipt,FlagOverwrite,SubFolderNamePrompt,ReturnLinkLabel,ReturnLinkURL)

Usage

Use the `SetDropBoxDefaults` method to set the default settings for creating a FileReceive Link.

The following parameters are needed to make the `SetDropBoxDefaults` call:

- **Token.** (The Token ID returned by the last `AccountLogin` call).
- **ShareDays.** (The number of days the FileReceive Link will remain active).
- **FlagLog.** (Flag indicating logging of activity to FileReceive Link).
- **FlagEmail.** (Flag indicating an email notification should be sent to recipients).
- **FlagEmailFromSystem.** (Flag indicating the email notification should be sent from the system email account or the user's email account).
- **EmailSubject.** (Email notification subject).
- **EmailBody.** (Email notification text).
- **FlagAttachment.** ()
- **FlagReadReceipt.** (Flag indicating a read receipt should be generated when email notifications are read).
- **FlagOverwrite.** (Flag indicating files in the FileReceive Link can be overwritten).
- **SubFolderNamePrompt.** (Text to display when partitioning a FileReceive Link for each user).
- **ReturnLinkLabel.** (Text to display to return the user to another URL).
- **ReturnLinkURL.** (URL to return the user to).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/SetDropBoxDefaults"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetDropBoxDefaults xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ShareDays>int</ShareDays>
      <FlagLog>string</FlagLog>
      <FlagEmail>string</FlagEmail>
      <FlagEmailFromSystem>string</FlagEmailFromSystem>
      <EmailSubject>string</EmailSubject>
      <EmailBody>string</EmailBody>
      <FlagAttachment>string</FlagAttachment>
      <FlagReadReceipt>string</FlagReadReceipt>
      <FlagOverwrite>string</FlagOverwrite>
      <SubFolderNamePrompt>string</SubFolderNamePrompt>
      <ReturnLinkLabel>string</ReturnLinkLabel>
      <ReturnLinkURL>string</ReturnLinkURL>
    </SetDropBoxDefaults>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetDropBoxDefaultsResponse xmlns="http://api.filesanywhere.com/">
      <SetDropBoxDefaultsResult>
        <ErrorMessage>string</ErrorMessage>
        <Updated>boolean</Updated>
      </SetDropBoxDefaultsResult>
    </SetDropBoxDefaultsResponse>
  </soap:Body>
</soap:Envelope>
```

```
    </SetDropBoxDefaultsResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SetDropBoxDefaults xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ShareDays>int</ShareDays>
      <FlagLog>string</FlagLog>
      <FlagEmail>string</FlagEmail>
      <FlagEmailFromSystem>string</FlagEmailFromSystem>
      <EmailSubject>string</EmailSubject>
      <EmailBody>string</EmailBody>
      <FlagAttachment>string</FlagAttachment>
      <FlagReadReceipt>string</FlagReadReceipt>
      <FlagOverwrite>string</FlagOverwrite>
      <SubFolderNamePrompt>string</SubFolderNamePrompt>
      <ReturnLinkLabel>string</ReturnLinkLabel>
      <ReturnLinkURL>string</ReturnLinkURL>
    </SetDropBoxDefaults>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SetDropBoxDefaultsResponse xmlns="http://api.filesanywhere.com/">
      <SetDropBoxDefaultsResult>
        <ErrorMessage>string</ErrorMessage>
        <Updated>boolean</Updated>
      </SetDropBoxDefaultsResult>
    </SetDropBoxDefaultsResponse>
  </soap12:Body>
</soap12:Envelope>
```



```

    </SetDropBoxDefaultsResult>
  </SetDropBoxDefaultsResponse>
</soap12:Body>
</soap12:Envelope>

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
ShareDays	Integer	Yes	The number of days the FileReceive Link will remain active.
FlagLog	String	Yes	Flag indicating logging of activity to FileReceive Link.
FlagEmail	String	Yes	Flag indicating an email notification should be sent to recipients.
FlagEmailFromSystem	String	Yes	Flag indicating the email notification should be sent from the system email account or the user's email account.
EmailSubject	String	Yes	Email notification subject.
EmailBody	String	Yes	Email notification text.
FlagAttachment	String	Yes	
FlagReadReceipt	String	Yes	Flag indicating a read receipt should be generated when an email notification is read.
FlagOverwrite	String	Yes	Flag indicating files in the FileReceive link can be overwritten.
SubFolderNamePrompt	String	Yes	Text displayed when partitioning a FileReceive Link for each user.
ReturnLinkLabel	String	Yes	Text displayed to return a user to another URL.
ReturnLinkURL	String	Yes	URL to return user to.

Response**SetDropBoxDefaultsResultv2**

SetDropBoxDefaultsResultv2 Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the SetAPNSSetting method was executed. If the SetAPNSSetting call was successful, the ErrorMessage is blank
Updated	Boolean	Indicates the default settings were updated.

SetDropBoxDefaults2

Syntax

SetDropBoxDefaultsResult = SetDropBoxDefaults2
(Token,ShareDays,FlagLog,FlagEmail,FlagEmailFromSystem,EmailSubject,EmailBody,FlagAttachment,FlagReadReceipt,FlagOverwrite,SubFolderNamePrompt,ReturnLinkLabel,ReturnLinkURL, ShowName, ShowEmailAddress)

Usage

Use the **SetDropBoxDefaults** method to set the default settings for creating a **FileReceive Link**.

The following parameters are needed to make the **SetDropBoxDefaults** call:

- **Token.** (The Token ID returned by the last **AccountLogin** call).
- **ShareDays.** (The number of days the **FileReceive Link** will remain active).
- **FlagLog.** (Flag indicating logging of activity to **FileReceive Link**).
- **FlagEmail.** (Flag indicating an email notification should be sent to recipients).
- **FlagEmailFromSystem.** (Flag indicating the email notification should be sent from the system email account or the user's email account).
- **EmailSubject.** (Email notification subject).
- **EmailBody.** (Email notification text).
- **FlagAttachment.** ()
- **FlagReadReceipt.** (Flag indicating a read receipt should be generated when email notifications are read).
- **FlagOverwrite.** (Flag indicating files in the **FileReceive Link** can be overwritten).
- **SubFolderNamePrompt.** (Text to display when partitioning a **FileReceive Link** for each user).
- **ReturnLinkLabel.** (Text to display to return the user to another URL).
- **ReturnLinkURL.** (URL to return the user to).

- **ShowName.** (Flag indicating to show the name of the owner of the FileReceive Link).
- **ShowEmailAddress.** (Flag indicating to show the email address of the owner of the FileReceive Link).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/SetDropBoxDefaults2"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetDropBoxDefaults2 xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ShareDays>int</ShareDays>
      <FlagLog>string</FlagLog>
      <FlagEmail>string</FlagEmail>
      <FlagEmailFromSystem>string</FlagEmailFromSystem>
      <EmailSubject>string</EmailSubject>
      <EmailBody>string</EmailBody>
      <FlagAttachment>string</FlagAttachment>
      <FlagReadReceipt>string</FlagReadReceipt>
      <FlagOverwrite>string</FlagOverwrite>
      <SubFolderNamePrompt>string</SubFolderNamePrompt>
      <ReturnLinkLabel>string</ReturnLinkLabel>
      <ReturnLinkURL>string</ReturnLinkURL>
      <ShowName>boolean</ShowName>
      <ShowEmailAddress>boolean</ShowEmailAddress>
    </SetDropBoxDefaults2>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
```

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetDropBoxDefaults2Response xmlns="http://api.filesanywhere.com/">
      <SetDropBoxDefaultsResult>
        <ErrorMessage>string</ErrorMessage>
        <Updated>boolean</Updated>
      </SetDropBoxDefaultsResult>
    </SetDropBoxDefaults2Response>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SetDropBoxDefaults2 xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ShareDays>int</ShareDays>
      <FlagLog>string</FlagLog>
      <FlagEmail>string</FlagEmail>
      <FlagEmailFromSystem>string</FlagEmailFromSystem>
      <EmailSubject>string</EmailSubject>
      <EmailBody>string</EmailBody>
      <FlagAttachment>string</FlagAttachment>
      <FlagReadReceipt>string</FlagReadReceipt>
      <FlagOverwrite>string</FlagOverwrite>
      <SubFolderNamePrompt>string</SubFolderNamePrompt>
      <ReturnLinkLabel>string</ReturnLinkLabel>
      <ReturnLinkURL>string</ReturnLinkURL>
      <ShowName>boolean</ShowName>
      <ShowEmailAddress>boolean</ShowEmailAddress>
    </SetDropBoxDefaults2>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
```

Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SetDropBoxDefaults2Response xmlns="http://api.filesanywhere.com/">
      <SetDropBoxDefaultsResult>
        <ErrorMessage>string</ErrorMessage>
        <Updated>boolean</Updated>
      </SetDropBoxDefaultsResult>
    </SetDropBoxDefaults2Response>
  </soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
ShareDays	Integer	Yes	The number of days the FileReceive Link will remain active.
FlagLog	String	Yes	Flag indicating logging of activity to FileReceive Link.
FlagEmail	String	Yes	Flag indicating an email notification should be sent to recipients.
FlagEmailFromSystem	String	Yes	Flag indicating the email notification should be sent from the system email account or the user's email account.
EmailSubject	String	Yes	Email notification subject.
EmailBody	String	Yes	Email notification text.
FlagAttachment	String	Yes	
FlagReadReceipt	String	Yes	Flag indicating a read receipt should be generated when an email notification is read.
FlagOverwrite	String	Yes	Flag indicating files in the FileReceive Link can be overwritten.
SubFolderNamePrompt	String	Yes	Text displayed when partitioning a FileReceive

			Link for each user.
ReturnLinkLabel	String	Yes	Text displayed to return a user to another URL.
ReturnLinkURL	String	Yes	URL to return user to.
ShowName	Boolean	Yes	Flag indicating to show the name of the owner of the FileReceive Link.
ShowEmailAddress	Boolean	Yes	Flag indicating to show the email address of the owner of the FileReceive Link.

Response

SetDropBoxDefaultsResultv2

SetDropBoxDefaultsResultv2 Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the SetAPNSetting method was executed. If the SetAPNSetting call was successful, the ErrorMessage is blank
Updated	Boolean	Indicates the default settings were updated.

SetELinkDefaults

Syntax

SetELinkDefaultsResult = SetELinkDefaults
(Token,ShareDays,FlagLog,FlagEmail,FlagMediaView,FlagPlayOptions,DownloadLimit,FlagViewOnly,FlagWatermark,WatermarkCenter,WatermarkBottom,FlagEmailFromSystem,EmailSubject,EmailBody,FlagReadReceipt,FlagCoverPage,CoverPageTitle,CoverPageMessage,ApprovalText,InputPromptText,SignaturePromptText)

Usage

Use the SetELinkDefaults method to set the default settings for creating a FileShare link.

The following parameters are needed to make the SetELinkDefaults call:

- **Token.** (The Token ID returned by the last AccountLogin call).
- **ShareDays.** (The number of days the FileReceive Link will remain active).
- **FlagLog.** (Flag indicating logging of activity to FileReceive Link).
- **FlagEmail.** (Flag indicating an email notification should be sent to recipients).
- **FlagMediaView.** (Flag indicating the type of view for the FileShare).
- **FlagPlayOptions.** (Flag indicating the play options for the FileShare).
- **DownloadLimit.** (The maximum number of downloads for the FileShare).
- **FlagViewOnly.** (Flag specifying if files can be downloaded or view-only).
- **FlagWatermark.** (Flag indicating the FileShare has a watermark).
- **WatermarkCenter.** (Text displayed in the center watermark).
- **WatermarkBottom.** (Text displayed in the bottom watermark).
- **FlagEmailFromSystem.** (Flag indicating the email notification should be sent from the system email account or the user's email account).
- **EmailSubject.** (Email notification subject).
- **EmailBody.** (Email notification text).

- **FlagReadReceipt.** (Flag indicating a read receipt should be generated when email notifications are read).
- **FlagCoverPage.** (Flag indicating the FileShare has a cover page).
- **CoverPageTitle.** (Title text for the cover page).
- **CoverPageMessage.** (Message text for the cover page).
- **ApprovalText.** (Text displayed when obtaining approval from the user).
- **InputPromptText.** (Text displayed when obtaining input from the user).
- **SignaturePromptText.** (Text displayed when obtaining a signature from the user).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/SetELinkDefaults"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetELinkDefaults xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ShareDays>int</ShareDays>
      <FlagLog>string</FlagLog>
      <FlagEmail>string</FlagEmail>
      <FlagMediaView>string</FlagMediaView>
      <FlagPlayOptions>string</FlagPlayOptions>
      <DownloadLimit>int</DownloadLimit>
      <FlagViewOnly>string</FlagViewOnly>
      <FlagWatermark>string</FlagWatermark>
      <WatermarkCenter>string</WatermarkCenter>
      <WatermarkBottom>string</WatermarkBottom>
      <FlagEmailFromSystem>string</FlagEmailFromSystem>
      <EmailSubject>string</EmailSubject>
      <EmailBody>string</EmailBody>
      <FlagReadReceipt>string</FlagReadReceipt>
    </SetELinkDefaults>
  </soap:Body>
</soap:Envelope>
```

```

    <FlagCoverPage>string</FlagCoverPage>
    <CoverPageTitle>string</CoverPageTitle>
    <CoverPageMessage>string</CoverPageMessage>
    <ApprovalText>string</ApprovalText>
    <InputPromptText>string</InputPromptText>
    <SignaturePromptText>string</SignaturePromptText>
  </SetELinkDefaults>
</soap:Body>
</soap:Envelope>

```

```

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetELinkDefaultsResponse xmlns="http://api.filesanywhere.com/">
      <SetELinkDefaultsResult>
        <ErrorMessage>string</ErrorMessage>
        <Updated>boolean</Updated>
      </SetELinkDefaultsResult>
    </SetELinkDefaultsResponse>
  </soap:Body>
</soap:Envelope>

```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```

POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SetELinkDefaults xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ShareDays>int</ShareDays>
      <FlagLog>string</FlagLog>
      <FlagEmail>string</FlagEmail>
    </SetELinkDefaults>
  </soap12:Body>
</soap12:Envelope>

```

```

<FlagMediaView>string</FlagMediaView>
<FlagPlayOptions>string</FlagPlayOptions>
<DownloadLimit>int</DownloadLimit>
<FlagViewOnly>string</FlagViewOnly>
<FlagWatermark>string</FlagWatermark>
<WatermarkCenter>string</WatermarkCenter>
<WatermarkBottom>string</WatermarkBottom>
<FlagEmailFromSystem>string</FlagEmailFromSystem>
<EmailSubject>string</EmailSubject>
<EmailBody>string</EmailBody>
<FlagReadReceipt>string</FlagReadReceipt>
<FlagCoverPage>string</FlagCoverPage>
<CoverPageTitle>string</CoverPageTitle>
<CoverPageMessage>string</CoverPageMessage>
<ApprovalText>string</ApprovalText>
<InputPromptText>string</InputPromptText>
<SignaturePromptText>string</SignaturePromptText>
</SetELinkDefaults>
</soap12:Body>
</soap12:Envelope>

```

```

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SetELinkDefaultsResponse xmlns="http://api.filesanywhere.com/">
      <SetELinkDefaultsResult>
        <ErrorMessage>string</ErrorMessage>
        <Updated>boolean</Updated>
      </SetELinkDefaultsResult>
    </SetELinkDefaultsResponse>
  </soap12:Body>
</soap12:Envelope>

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
ShareDays	Integer	Yes	The number of days the FileReceive Link will remain active.

FlagLog	String	Yes	Flag indicating logging of activity to FileReceive Link.
FlagEmail	String	Yes	Flag indicating an email notification should be sent to recipients.
FlagMediaView	String	Yes	Flag indicating the type of view for the FileShare.
FlagPlayOptions	String	Yes	Flag specifying the play options for the FileShare.
DownloadLimit	Integer	Yes	The maximum number of downloads for the FileShare.
FlagViewOnly	String	Yes	Flag indicating files can be downloaded or view only.
FlagWatermark	String	Yes	Flag indicating the FileShare has a cover page.
WatermarkCenter	String	Yes	Text displayed in the center watermark.
WatermarkBottom	String	Yes	Text displayed in the bottom watermark.
FlagEmailFromSystem	String	Yes	Flag indicating the email notification should be sent from the system email account or the user's email account.
EmailSubject	String	Yes	Email notification subject.
EmailBody	String	Yes	Email notification text.
FlagReadReceipt	String	Yes	Flag indicating a read receipt should be generated when an email notification is read.
FlagCoverPage	String	Yes	Flag indicating the FileShare has a cover page.
CoverPageTitle	String	Yes	Title text for the cover page.
CoverPageMessage	String	Yes	Message text for the cover page.
ApprovalText	String	Yes	Text displayed to obtain approval from the user.
InputPromptText	String	Yes	Text displayed to obtain input from the user.
SignaturePromptText	String	Yes	Text displayed to obtain a signature from the user.

Response

SetELinkDefaultsResultv2

SetELinkDefaultsResultv2 Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the SetELinkDefaults method was executed. If the SetELinkDefaults call was successful, the ErrorMessage is blank
Updated	Boolean	Indicates the default settings were updated.

SetELinkDefaults2

Syntax

SetELinkDefaultsResult = SetELinkDefaults2

(Token,ShareDays,FlagLog,FlagEmail,FlagMediaView,FlagPlayOptions,DownloadLimit,FlagViewOnly,FlagWatermark,WatermarkCenter,WatermarkBottom,FlagEmailFromSystem,EmailSubject,EmailBody,FlagReadReceipt,FlagCoverPage,CoverPageTitle,CoverPageMessage,ApprovalText,InputPromptText,SignaturePromptText,ShowName,ShowEmailAddress)

Usage

Use the SetELinkDefaults method to set the default settings for creating a FileShare link.

The following parameters are needed to make the SetELinkDefaults call:

- Token. (The Token ID returned by the last AccountLogin call).
- ShareDays. (The number of days the FileReceive Link will remain active).
- FlagLog. (Flag indicating logging of activity to FileReceive Link).
- FlagEmail. (Flag indicating an email notification should be sent to recipients).
- FlagMediaView. (Flag indicating the type of view for the FileShare).
- FlagPlayOptions. (Flag indicating the play options for the FileShare).
- DownloadLimit. (The maximum number of downloads for the FileShare).
- FlagViewOnly. (Flag specifying if files can be downloaded or view-only).
- FlagWatermark. (Flag indicating the FileShare has a watermark).
- WatermarkCenter. (Text displayed in the center watermark).
- WatermarkBottom. (Text displayed in the bottom watermark).
- FlagEmailFromSystem. (Flag indicating the email notification should be sent from the system email account or the user's email account).
- EmailSubject. (Email notification subject).

- EmailBody. (Email notification text).
- FlagReadReceipt. (Flag indicating a read receipt should be generated when email notifications are read).
- FlagCoverPage. (Flag indicating the FileShare has a cover page).
- CoverPageTitle. (Title text for the cover page).
- CoverPageMessage. (Message text for the cover page).
- ApprovalText. (Text displayed when obtaining approval from the user).
- InputPromptText. (Text displayed when obtaining input from the user).
- SignaturePromptText. (Text displayed when obtaining a signature from the user).
- ShowName. (Flag indicating to show the name of the owner of the fileshare).
- ShowEmailAddress. (Flag indicating to show the email address of the owner of the fileshare).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/SetELinkDefaults2"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetELinkDefaults2 xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ShareDays>int</ShareDays>
      <FlagLog>string</FlagLog>
      <FlagEmail>string</FlagEmail>
      <FlagMediaView>string</FlagMediaView>
      <FlagPlayOptions>string</FlagPlayOptions>
      <DownloadLimit>int</DownloadLimit>
      <FlagViewOnly>string</FlagViewOnly>
    </SetELinkDefaults2>
  </soap:Body>
</soap:Envelope>
```

```
<FlagWatermark>string</FlagWatermark>
<WatermarkCenter>string</WatermarkCenter>
<WatermarkBottom>string</WatermarkBottom>
<FlagEmailFromSystem>string</FlagEmailFromSystem>
<EmailSubject>string</EmailSubject>
<EmailBody>string</EmailBody>
<FlagReadReceipt>string</FlagReadReceipt>
<FlagCoverPage>string</FlagCoverPage>
<CoverPageTitle>string</CoverPageTitle>
<CoverPageMessage>string</CoverPageMessage>
<ApprovalText>string</ApprovalText>
<InputPromptText>string</InputPromptText>
<SignaturePromptText>string</SignaturePromptText>
<ShowName>boolean</ShowName>
<ShowEmailAddress>boolean</ShowEmailAddress>
</SetELinkDefaults2>
</soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetELinkDefaults2Response xmlns="http://api.filesanywhere.com/">
      <SetELinkDefaultsResult>
        <ErrorMessage>string</ErrorMessage>
        <Updated>boolean</Updated>
      </SetELinkDefaultsResult>
    </SetELinkDefaults2Response>
  </soap:Body>
</soap:Envelope>
```


SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SetELinkDefaults2 xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ShareDays>int</ShareDays>
      <FlagLog>string</FlagLog>
      <FlagEmail>string</FlagEmail>
      <FlagMediaView>string</FlagMediaView>
      <FlagPlayOptions>string</FlagPlayOptions>
      <DownloadLimit>int</DownloadLimit>
      <FlagViewOnly>string</FlagViewOnly>
      <FlagWatermark>string</FlagWatermark>
      <WatermarkCenter>string</WatermarkCenter>
      <WatermarkBottom>string</WatermarkBottom>
      <FlagEmailFromSystem>string</FlagEmailFromSystem>
      <EmailSubject>string</EmailSubject>
      <EmailBody>string</EmailBody>
      <FlagReadReceipt>string</FlagReadReceipt>
      <FlagCoverPage>string</FlagCoverPage>
      <CoverPageTitle>string</CoverPageTitle>
      <CoverPageMessage>string</CoverPageMessage>
      <ApprovalText>string</ApprovalText>
      <InputPromptText>string</InputPromptText>
      <SignaturePromptText>string</SignaturePromptText>
      <ShowName>boolean</ShowName>
      <ShowEmailAddress>boolean</ShowEmailAddress>
    </SetELinkDefaults2>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
```

Content-Length: `length`

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SetELinkDefaults2Response xmlns="http://api.filesanywhere.com/">
      <SetELinkDefaultsResult>
        <ErrorMessage>string/ErrorMessage>
        <Updated>boolean/Updated>
      </SetELinkDefaultsResult>
    </SetELinkDefaults2Response>
  </soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
ShareDays	Integer	Yes	The number of days the FileReceive Link will remain active.
FlagLog	String	Yes	Flag indicating logging of activity to FileReceive Link.
FlagEmail	String	Yes	Flag indicating an email notification should be sent to recipients.
FlagMediaView	String	Yes	Flag indicating the type of view for the FileShare.
FlagPlayOptions	String	Yes	Flag specifying the play options for the FileShare.
DownloadLimit	Integer	Yes	The maximum number of downloads for the FileShare.
FlagViewOnly	String	Yes	Flag indicating files can be downloaded or view only.
FlagWatermark	String	Yes	Flag indicating the FileShare has a cover page.
WatermarkCenter	String	Yes	Text displayed in the center watermark.
WatermarkBottom	String	Yes	Text displayed in the bottom watermark.
FlagEmailFromSystem	String	Yes	Flag indicating the email notification should be

			sent from the system email account or the user's email account.
EmailSubject	String	Yes	Email notification subject.
EmailBody	String	Yes	Email notification text.
FlagReadReceipt	String	Yes	Flag indicating a read receipt should be generated when an email notification is read.
FlagCoverPage	String	Yes	Flag indicating the FileShare has a cover page.
CoverPageTitle	String	Yes	Title text for the cover page.
CoverPageMessage	String	Yes	Message text for the cover page.
ApprovalText	String	Yes	Text displayed to obtain approval from the user.
InputPromptText	String	Yes	Text displayed to obtain input from the user.
SignaturePromptText	String	Yes	Text displayed to obtain a signature from the user.
ShowName	Boolean	Yes	Flag indicating to show the name of the owner of the fileshare.
ShowEmailAddress	Boolean	Yes	Flag indicating to show the email address of the owner of the fileshare.

Response

SetELinkDefaultsResultv2

SetELinkDefaultsResultv2 Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the SetELinkDefaults2 method was executed. If the SetELinkDefaults2 call was successful, the ErrorMessage is blank
Updated	Boolean	Indicates the default settings were updated.

SetNotificationSettingForDevice

Syntax

SetNotificationSettingForDeviceResult =
SetNotificationSettingForDevice(Token,DeviceId,NotificationType,Allow)

Usage

Use the `SetNotificationSettingForDevice` method to enable or disable a specific notification type on a given device.

The following parameters are needed to make the `SetNotificationSettingForDevice` call:

- `Token`. (The Token ID returned by the last `AccountLogin` call).
- `DeviceId`. (The physical device Id).
- `NotificationType`. (The type of notification to allow or disable).
- `Allow`. (Flag indicating if the notification type is allowed or disabled).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/SetNotificationSettingForDevice"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetNotificationSettingForDevice
xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <DeviceId>string</DeviceId>
      <NotificationType>NONE or FILEACTION_ANNOTATE or FILEACTION_DELETED
or FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
```

```
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE  
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or  
TESTINGACTION_ECHO or ESENDACTION_FILERECEIVED or EMAILACTION_FILERECEIVED  
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANGE or  
GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or  
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or  
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or  
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or  
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or  
ACCOUNTACTION_STORAGEEXCEEDED</NotificationType>  
  <Allow>boolean</Allow>  
</SetNotificationSettingForDevice>  
</soap:Body>  
</soap:Envelope>
```

```
HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: length  
  
<?xml version="1.0" encoding="utf-8"?>  
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <SetNotificationSettingForDeviceResponse  
xmlns="http://api.filesanywhere.com/">  
      <SetAPNSettingResult>  
        <ErrorMessage>string</ErrorMessage>  
      </SetAPNSettingResult>  
    </SetNotificationSettingForDeviceResponse>  
  </soap:Body>  
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1  
Host: api.filesanywhere.com  
Content-Type: application/soap+xml; charset=utf-8  
Content-Length: length  
  
<?xml version="1.0" encoding="utf-8"?>
```

```
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SetNotificationSettingForDevice
xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <DeviceId>string</DeviceId>
      <NotificationType>NONE or FILEACTION_ANNOTATE or FILEACTION_DELETED
or FILEACTION_EDIT or FILEACTION_MOVED or FILEACTION_RENAMED or
FILEACTION_UPDATED or FILEACTION_UPLOAD or CHECKACTION_CHECKIN or
CHECKACTION_CHECKOUT or FAXACTION_FAXRECEIVED or PRODUCTACTION_APPUPDATE
or PRODUCTACTION_APPRATINGREMINDER or LIFECYCLEACTION_EXPIRATION or
TESTINGACTION_ECHO or ESENDATION_FILERECEIVED or EMAILACTION_FILERECEIVED
or GROUPSHAREACTION_CREATED or GROUPSHAREACTION_PERMISSIONSCHANGE or
GROUPSHAREACTION_REMOVED or DROPBOXACTION_FILEUPLOAD or
DROPBOXACTION_LINKACCESS or FILESHAREACTION_FILEDOWNLOAD or
FILESHAREACTION_LINKACCESS or ACCOUNTACTION_CREDITCARDDECLINED or
ACCOUNTACTION_CREDITCARDEXPIRED or ACCOUNTACTION_PASSWORDCHANGE or
ACCOUNTACTION_SIXTYDAYSWITHOUTLOGIN or
ACCOUNTACTION_STORAGEEXCEEDED</NotificationType>
      <Allow>boolean</Allow>
    </SetNotificationSettingForDevice>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SetNotificationSettingForDeviceResponse
xmlns="http://api.filesanywhere.com/">
      <SetAPNSettingResult>
        <ErrorMessage>string</ErrorMessage>
      </SetAPNSettingResult>
    </SetNotificationSettingForDeviceResponse>
  </soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
DeviceId	String	Yes	The physical device Id.
NotificationType	Enum	Yes	The type of notification to allow or disable.
Allow	Boolean	Yes	Flag indicating whether to allow or disable the notification type.

Response**SetNotificationSettingsForDeviceResult**

SetNotificationSettingsForDeviceResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the SetNotificationSettingForDevice method was executed. If the SetNotificationSettingForDevice call was successful, the ErrorMessage is blank

SetNotificationSettingsForDevice

Syntax

SetNotificationSettingsForDeviceResult =
SetNotificationSettingsForDevice(Token,DeviceId,AllowedNotificationTypes)

Usage

Use the SetNotificationSettingsForDevice method to indicate which notification types are enabled for a given device.

The following parameters are needed to make the SetNotificationSettingsForDevice call:

- Token. (The Token ID returned by the last AccountLogin call).
- DeviceId. (The physical device Id).
- AllowedNotificationTypes. (The numeric representation of the types of notification to allow).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction:
"http://api.filesanywhere.com/SetNotificationSettingsForDevice"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetNotificationSettingsForDevice
xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <DeviceId>string</DeviceId>
      <AllowedNotificationTypes>long</AllowedNotificationTypes>
    </SetNotificationSettingsForDevice>
  </soap:Body>
</soap:Envelope>
```



```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetNotificationSettingsForDeviceResponse
xmlns="http://api.filesanywhere.com/">
      <SetAPNSettingResult>
        <ErrorMessage>string</ErrorMessage>
      </SetAPNSettingResult>
    </SetNotificationSettingsForDeviceResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SetNotificationSettingsForDevice
xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <DeviceId>string</DeviceId>
      <AllowedNotificationTypes>long</AllowedNotificationTypes>
    </SetNotificationSettingsForDevice>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SetNotificationSettingsForDeviceResponse
xmlns="http://api.filesanywhere.com/">
      <SetAPNSettingResult>
        <ErrorMessage>string</ErrorMessage>
      </SetAPNSettingResult>
    </SetNotificationSettingsForDeviceResponse>
  </soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
DeviceId	String	Yes	The physical device Id.
AllowedNotificationTypes	Long	Yes	The numeric representation of the notification types to allow.

Response

SetNotificationSettingsForDeviceResult

SetNotificationSettingsForDeviceResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the SetNotificationSettingsForDevice method was executed. If the SetNotificationSettingsForDevice call was successful, the ErrorMessage is blank

StopSharingGroupShare

Syntax

StopSharingGroupShareResult = StopSharingGroupShare
(Token,SharedFolderName,SendEmail)

Usage

Use the StopSharingGroupShare method to stop sharing a GroupShare.

The following parameters are needed to make the StopSharingGroupShare call:

- **Token.** (The Token ID returned by the last AccountLogin call).
- **SharedFolderName.** (Path to the folder being shared).
- **SendEmail.** (Flag indicating an email notification should be sent to GroupShare members).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/StopSharingGroupShare"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <StopSharingGroupShare xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <SharedFolderName>string</SharedFolderName>
      <SendEmail>boolean</SendEmail>
    </StopSharingGroupShare>
  </soap:Body>
```

```
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <StopSharingGroupShareResponse xmlns="http://api.filesanywhere.com/">
      <StopSharingGroupShareResult>
        <ErrorMessage>string</ErrorMessage>
        <Users>
          <GroupShareRemoveResultItem>
            <ErrorMessage>string</ErrorMessage>
            <Username>string</Username>
            <UserRemoved>boolean</UserRemoved>
          </GroupShareRemoveResultItem>
          <GroupShareRemoveResultItem>
            <ErrorMessage>string</ErrorMessage>
            <Username>string</Username>
            <UserRemoved>boolean</UserRemoved>
          </GroupShareRemoveResultItem>
        </Users>
        <GroupShareRemoved>boolean</GroupShareRemoved>
      </StopSharingGroupShareResult>
    </StopSharingGroupShareResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <StopSharingGroupShare xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <SharedFolderName>string</SharedFolderName>
      <SendEmail>boolean</SendEmail>
    </StopSharingGroupShare>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <StopSharingGroupShareResponse xmlns="http://api.filesanywhere.com/">
      <StopSharingGroupShareResult>
        <ErrorMessage>string</ErrorMessage>
        <Users>
          <GroupShareRemoveResultItem>
            <ErrorMessage>string</ErrorMessage>
            <Username>string</Username>
            <UserRemoved>boolean</UserRemoved>
          </GroupShareRemoveResultItem>
          <GroupShareRemoveResultItem>
            <ErrorMessage>string</ErrorMessage>
            <Username>string</Username>
            <UserRemoved>boolean</UserRemoved>
          </GroupShareRemoveResultItem>
        </Users>
        <GroupShareRemoved>boolean</GroupShareRemoved>
      </StopSharingGroupShareResult>
    </StopSharingGroupShareResponse>
  </soap12:Body>
```

```
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
SharedFolderName	String	Yes	The path to the folder being shared.
SendEmail	Boolean	Yes	Flag indicating an email notification should be sent to GroupShare members.

Response

StopSharingGroupShareResult

StopSharingGroupShareResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the StopSharingGroupShare method was executed. If the StopSharingGroupShare call was successful, the ErrorMessage is blank
Users.ErrorMessage	String	Error message pertaining to the removal of a single user.
Users.Username	String	The username of a user removed from the GroupShare.
Users.UserRemoved	Boolean	Indicates if the user was removed from the GroupShare.
GroupShareRemoved	Boolean	Indicates if the GroupShare was removed.

UndoCheckout

Syntax

UndoCheckoutResult = UndoCheckout (Token,ItemPath,Comments)

Usage

Use the UndoCheckOut method to undo a file check-out.

The following parameters are needed to make the UndoCheckOut call:

- **Token.** (The Token ID returned by the last AccountLogin call).
- **ItemPath.** (Path to the item that is checked-out).
- **Comments.** (Comments text attached to the undo checkout action).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/UndoCheckout"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UndoCheckout xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ItemPath>string</ItemPath>
      <Comments>string</Comments>
    </UndoCheckout>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
```

```
Content-Type: text/xml; charset=utf-8  
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>  
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <UndoCheckoutResponse xmlns="http://api.filesanywhere.com/">  
      <UndoCheckOutFileResult>  
        <ErrorMessage>string</ErrorMessage>  
      </UndoCheckOutFileResult>  
    </UndoCheckoutResponse>  
  </soap:Body>  
</soap:Envelope>
```


SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <UndoCheckout xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ItemPath>string</ItemPath>
      <Comments>string</Comments>
    </UndoCheckout>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <UndoCheckoutResponse xmlns="http://api.filesanywhere.com/">
      <UndoCheckOutFileResult>
        <ErrorMessage>string</ErrorMessage>
      </UndoCheckOutFileResult>
    </UndoCheckoutResponse>
  </soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
ItemPath	String	Yes	The path to the file that is checked-out.
Comments	String	Yes	Comments text attached to the undo-checkout action.

Response**UndoCheckoutResult**

UndoCheckoutResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the UndoCheckout method was executed. If the UndoCheckout call was successful, the ErrorMessage is blank

UnregisterDevice

Syntax

UnregisterDeviceResult = UnregisterDevice(Token,DeviceId,DeviceToken)

Usage

Use the UnregisterDevice method to remove a device registration for notifications.

The following parameters are needed to make the UnregisterDevice call:

- Token. (The Token ID returned by the last AccountLogin call).
- DeviceId. (The physical device Id).
- DeviceToken. (The logical device Id used in RegisterDevice).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/UnregisterDevice"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UnregisterDevice xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <DeviceId>string</DeviceId>
      <DeviceToken>string</DeviceToken>
    </UnregisterDevice>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
```

```
Content-Type: text/xml; charset=utf-8  
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>  
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <UnregisterDeviceResponse xmlns="http://api.filesanywhere.com/">  
      <UnregisterDeviceResult>  
        <ErrorMessage>string</ErrorMessage>  
      </UnregisterDeviceResult>  
    </UnregisterDeviceResponse>  
  </soap:Body>  
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <UnregisterDevice xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <DeviceId>string</DeviceId>
      <DeviceToken>string</DeviceToken>
    </UnregisterDevice>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <UnregisterDeviceResponse xmlns="http://api.filesanywhere.com/">
      <UnregisterDeviceResult>
        <ErrorMessage>string</ErrorMessage>
      </UnregisterDeviceResult>
    </UnregisterDeviceResponse>
  </soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
DeviceId	String	Yes	The physical device Id.
DeviceToken	String	Yes	The logical device Id used in RegisterDevice.

Response**UnregisterDeviceResult**

UnregisterDeviceResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the UnregisterDevice method was executed. If the UnregisterDevice call was successful, the ErrorMessage is blank

UpdateMeeting

Syntax

UpdateMeetingResult= UpdateMeeting(Token,Meeting)

Usage

Use the UpdateMeeting method to modify the properties of a scheduled meeting.

The following parameters are needed to make the UpdateMeeting call:

- Token. (The Token ID returned by the last AccountLogin call).
- Meeting. (A populated meeting structure).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/UpdateMeeting"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UpdateMeeting xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Meeting>
        <MeetingId>long</MeetingId>
        <MeetingName>string</MeetingName>
        <Organizer />
        <Role>None or Organizer or Attendee or OrganizerOrAttendee</Role>
        <MeetingDate>dateTime</MeetingDate>
        <MeetingDescription>string</MeetingDescription>
        <SendReminder>boolean</SendReminder>
        <ReminderLeadTime>None or FiveMinutes or FifteenMinutes or
ThirtyMinutes or FortyFiveMinutes or OneHour</ReminderLeadTime>
        <Attendees>
          <AttendeeId />
          <AttendeeId />
        </Attendees>
      </Meeting>
    </UpdateMeeting>
  </soap:Body>
</soap:Envelope>
```

```
<ReminderSent>boolean</ReminderSent>
</Meeting>
</UpdateMeeting>
</soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UpdateMeetingResponse xmlns="http://api.filesanywhere.com/">
      <UpdateMeetingResult>
        <ErrorMessage>string</ErrorMessage>
        <Updated>boolean</Updated>
      </UpdateMeetingResult>
    </UpdateMeetingResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <UpdateMeeting xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Meeting>
        <MeetingId>long</MeetingId>
        <MeetingName>string</MeetingName>
        <Organizer />
        <Role>None or Organizer or Attendee or OrganizerOrAttendee</Role>
        <MeetingDate>dateTime</MeetingDate>
```



```

    <MeetingDescription>string</MeetingDescription>
    <SendReminder>boolean</SendReminder>
    <ReminderLeadTime>None or FiveMinutes or FifteenMinutes or
    ThirtyMinutes or FortyFiveMinutes or OneHour</ReminderLeadTime>
    <Attendees>
      <AttendeeId />
      <AttendeeId />
    </Attendees>
    <ReminderSent>boolean</ReminderSent>
  </Meeting>
</UpdateMeeting>
</soap12:Body>
</soap12:Envelope>

```

```

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <UpdateMeetingResponse xmlns="http://api.filesanywhere.com/">
      <UpdateMeetingResult>
        <ErrorMessage>string</ErrorMessage>
        <Updated>boolean</Updated>
      </UpdateMeetingResult>
    </UpdateMeetingResponse>
  </soap12:Body>
</soap12:Envelope>

```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
Meeting	Meeting	Yes	A populated meeting structure.

A Meeting has the following properties:

Property Name	Data Type	Required	Description
MeetingId	Long	Yes	Unique identifier for the meeting.
MeetingName	String	Yes	String identifier or title for the meeting.
Organizer	UserIdentifier	Yes	User identifier representing the organizer.
Role	Enum	Yes	The role of the current user in the meeting.
MeetingDate	DateTime	Yes	The scheduled start date of the meeting.
DurationInMinutes	Integer	Yes	The planned length of the meeting in minutes.
MeetingDescription	String	Yes	The agenda or discussion plans for the meeting.
SendReminder	Boolean	Yes	Flag indicating if reminder notifications are to be sent to attendees.
ReminderLeadTime	Enum	Yes	The number of minutes before the meeting starts to send reminder notifications.
Attendees	UserIdentifier()	Yes	Array of user identifiers representing attendees.
ReminderSent	Boolean	Yes	Flag indicating if reminder notifications have been sent. Ignored in UpdateMeeting.

Response

UpdateMeetingResult

UpdateMeetingResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the GetMeeting method was executed. If the GetMeeting call was successful, the ErrorMessage is blank.
Updated	Boolean	Flag indicating if the meeting was updated successfully.

UploadFileWithCheckin

Syntax

**UploadFileWithCheckinResult =
UploadFileWithCheckin(Token,Path,FileData,Comments)**

Usage

Use the UploadFileWithCheckin method to upload a file and check the file in.

The following parameters are needed to make the UploadFileWithCheckin call:

- **Token.** (The Token ID returned by the last AccountLogin call).
- **Path.** (The path to the file being uploaded).
- **FileData.** (The file contents).
- **Comments.** (Comments attached to the file check-in action).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/UploadFileWithCheckin"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UploadFileWithCheckin xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
      <FileData>base64Binary</FileData>
      <Comments>string</Comments>
    </UploadFileWithCheckin>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
```

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UploadFileWithCheckinResponse xmlns="http://api.filesanywhere.com/">
      <UploadFileResult>
        <ErrorMessage>string</ErrorMessage>
        <Uploaded>boolean</Uploaded>
      </UploadFileResult>
    </UploadFileWithCheckinResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <UploadFileWithCheckin xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Path>string</Path>
      <FileData>base64Binary</FileData>
      <Comments>string</Comments>
    </UploadFileWithCheckin>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <UploadFileWithCheckinResponse xmlns="http://api.filesanywhere.com/">
      <UploadFileResult>
        <ErrorMessage>string</ErrorMessage>
        <Uploaded>boolean</Uploaded>
      </UploadFileResult>
    </UploadFileWithCheckinResponse>
  </soap12:Body>
</soap12:Envelope>
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
Path	String	Yes	The path the file being uploaded.
FileData	Byte()	Yes	The file contents.
Comments	String	Yes	Comments attached to the file check-in action.

Response

UploadFileResult

UploadFileResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the UploadFileWithCheckin method was executed. If the UploadFileWithCheckin call was successful, the ErrorMessage is blank

Uploaded	Boolean	Indicates if the file was uploaded successfully.
----------	---------	--

UserLookup

Syntax

UserLookupResult =
UserLookup(Token,SearchText,TeamId,SortBy,SortOrder,StartPage,PageSize,AutoComplete)

Usage

Use the UserLookup method to list or find users based on search criteria.

The following parameters are needed to make the UserLookup call:

- Token. (The Token ID returned by the last AccountLogin call).
- SearchText. (The search pattern used to filter user results).
- TeamId. (A team identifier to restrict the search to).
- SortBy. (Sort criteria for results).
- SortOrder. (Ascending or descending order of SortBy).
- StartPage. (The current page within the paged results).
- PageSize. (The number of entries per page in the paged results).
- AutoComplete. (Flag indicating a shortened search should be used for purposes of rapid auto completion).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/UserLookup"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UserLookup xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <SearchText>string</SearchText>
      <TeamId>string</TeamId>
      <SortBy>ID or IMAGEICON or USERNAME or FIRSTNAME or LASTNAME or
EMAILADDRESS or COMPANYNAME</SortBy>
      <SortOrder>ASC or DESC</SortOrder>
      <StartPage>int</StartPage>
      <PageSize>int</PageSize>
      <AutoComplete>boolean</AutoComplete>
    </UserLookup>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UserLookupResponse xmlns="http://api.filesanywhere.com/">
      <UserLookupResult>
        <ErrorMessage>string</ErrorMessage>
        <Items>
          <UserLookupItem>
            <ID>string</ID>
            <IMAGEICON>string</IMAGEICON>
            <FIRSTNAME>string</FIRSTNAME>
            <LASTNAME>string</LASTNAME>
            <EMAILADDRESS>string</EMAILADDRESS>
            <COMPANYNAME>string</COMPANYNAME>
            <USERNAME>string</USERNAME>
          </UserLookupItem>
          <UserLookupItem>
            <ID>string</ID>
            <IMAGEICON>string</IMAGEICON>
            <FIRSTNAME>string</FIRSTNAME>
```

```
<LASTNAME>string</LASTNAME>
<EMAILADDRESS>string</EMAILADDRESS>
<COMPANYNAME>string</COMPANYNAME>
<USERNAME>string</USERNAME>
</UserLookupItem>
</Items>
</UserLookupResult>
</UserLookupResponse>
</soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <UserLookup xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <SearchText>string</SearchText>
      <TeamId>string</TeamId>
      <SortBy>ID or IMAGEICON or USERNAME or FIRSTNAME or LASTNAME or
EMAILADDRESS or COMPANYNAME</SortBy>
      <SortOrder>ASC or DESC</SortOrder>
      <StartPage>int</StartPage>
      <PageSize>int</PageSize>
      <AutoComplete>boolean</AutoComplete>
    </UserLookup>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
```



```
<soap12:Body>
  <UserLookupResponse xmlns="http://api.filesanywhere.com/">
    <UserLookupResult>
      <ErrorMessage>string</ErrorMessage>
      <Items>
        <UserLookupItem>
          <ID>string</ID>
          <IMAGEICON>string</IMAGEICON>
          <FIRSTNAME>string</FIRSTNAME>
          <LASTNAME>string</LASTNAME>
          <EMAILADDRESS>string</EMAILADDRESS>
          <COMPANYNAME>string</COMPANYNAME>
          <USERNAME>string</USERNAME>
        </UserLookupItem>
        <UserLookupItem>
          <ID>string</ID>
          <IMAGEICON>string</IMAGEICON>
          <FIRSTNAME>string</FIRSTNAME>
          <LASTNAME>string</LASTNAME>
          <EMAILADDRESS>string</EMAILADDRESS>
          <COMPANYNAME>string</COMPANYNAME>
          <USERNAME>string</USERNAME>
        </UserLookupItem>
      </Items>
    </UserLookupResult>
  </UserLookupResponse>
</soap12:Body>
</soap12:Envelope>
```

Sample VB.Net Code

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
SearchText	String	Yes	The search pattern used to filter user results.
TeamId	String	Yes	A team identifier to restrict the search to.
SortBy	Enum	Yes	Sort criteria for results.
SortOrder	Enum	Yes	Ascending or descending order of SortBy.
StartPage	Integer	Yes	The current page within the paged results.
PageSize	Integer	Yes	The number of entries per page in the paged results.
AutoComplete	Boolean	Yes	Flag indicating a shortened search should be used for purposes of rapid auto completion.

Response

UserLookupResult

UserLookupResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the UserLookup method was executed. If the UserLookup call was successful, the ErrorMessage is blank
Items.ID	String	User identifier
Items.IMAGEICON	String	URL to icon image for the user.
Items.FIRSTNAME	String	First name of the user.
Items.LASTNAME	String	Last name of the user.
Items.EMAILADDRESS	String	Email address of the user.
ITEMS.COMPANYNAME	String	The company name for the user.
ITEMS.USERNAME	String	The username of the user.

GetProfileImage

This method returns the profile image for an user, when the login token and image size are passed as parameters. The image size returned back can have either of 3 sizes: small (24 x 24), medium (80 x 80), or large (225 x 300).

Syntax

```
ItemImageResult = GetProfileImage(Token, ProfileImageSize)
```

Usage

Use the `GetProfileImage` method to get the profile image of an user.

The following parameters are needed to make the `GetProfileImage` call:

- `Token` (The Token ID returned by the last `AccountLogin` call).
- `ProfileImageSize` is the size of the image to be specified as a string value which can be either "small", "medium" or "large".

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```

POST /FACORPAPI/v2/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetProfileImage"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetProfileImage xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ProfileImageSize>Small or Medium or Large</ProfileImageSize>
    </GetProfileImage>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetProfileImageResponse xmlns="http://api.filesanywhere.com/">
      <GetProfileImageResult>
        <ErrorMessage>string</ErrorMessage>
        <ImageStream>base64Binary</ImageStream>
      </GetProfileImageResult>
    </GetProfileImageResponse>
  </soap:Body>
</soap:Envelope>
  
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /FACORPAPI/v2/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetProfileImage xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ProfileImageSize>Small or Medium or Large</ProfileImageSize>
    </GetProfileImage>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetProfileImageResponse xmlns="http://api.filesanywhere.com/">
      <GetProfileImageResult>
        <ErrorMessage>string</ErrorMessage>
        <ImageStream>base64Binary</ImageStream>
      </GetProfileImageResult>
    </GetProfileImageResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub GetProfileImage ()
  Dim FAWAPI As New WebReference.FAWAPI
  Dim ProfileImageResult As New WebReference.ItemImageResult
  ProfileImageResult = FAWAPI.GetProfileImage(Token, ProfileImageSize)
  If ProfileImageResult.ErrorMessage IsNot Nothing Then
    MsgBox ProfileImageResult.ErrorMessage
  End If
  ProfileImageResult = Nothing
  FAWAPI = Nothing
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
ProfileImageSize	String	Yes	The size of the image. This can be passed in as small, medium or large.

AccountLogin2

This method logs you in to your FilesAnywhere account and starts an API session. It returns back a Token ID which must be used for subsequent API calls.

Syntax

```
LoginResult = AccountLogin(APIkey, OrgID, UserName, Password, DeviceID, AllowedIPList, ClientEncryptParam)
```

Usage

Use the AccountLogin call to login to your FilesAnywhere account and start a client API session. A client application must first login and obtain a Token ID before making other API calls.

The following parameters are needed to make the AccountLogin call:

- **APIKey** (A Developer API Key. See the “Getting Started” section above, to find out how to obtain a Developer API Key).
- **OrgID** (Client ID). The OrgID can have the following values:
 - 0 for a FilesAnywhere Web account.
 - 50 for a FilesAnywhere WebAdvanced account.
 - The Private Site Client ID for a FilesAnywhere Private Site.
- **UserName** (FilesAnywhere UserName). This is the FilesAnywhere username using with the user logs in to his FilesAnywhere account.
- **Password** (FilesAnywhere Password). This is the user’s FilesAnywhere account password
- **DeviceID**. This is a unique identifier specific to the device. For example, this will be the SecureUUID on the iOS.
- **AllowedIPList** (List of allowed IPs). This parameter is optional. It is a comma separated list of IPs which may be allowed to make subsequent API calls with the same Token ID which was passed back in this AccountLogin call.
- **ClientEncryptParam** (An optional parameter which can be used to get back an encrypted password which can be used for subsequent logins).
 - If no password encryption is desired, the parameter should be passed as blank.
 - If the string “ENCRYPTEDNO” is passed as the value of this parameter, the LoginResult returns back an encrypted password. This encrypted password can be used in subsequent logins using the AccountLogin API method.
 - When an encrypted password (returned back as explained above) is used to login using the AccountLogin method, the value of this parameter should be passed as “ENCRYPTEDYES”.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: 64.156.56.142
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/AccountLogin2"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <AccountLogin2 xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <OrgID>long</OrgID>
      <UserName>string</UserName>
      <Password>string</Password>
      <DeviceId>string</DeviceId>
      <AllowedIPList>string</AllowedIPList>
      <ClientEncryptParam>string</ClientEncryptParam>
    </AccountLogin2>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <AccountLogin2Response xmlns="http://api.filesanywhere.com/">
      <LoginResult>
        <ErrorMessage>string</ErrorMessage>
        <Token>string</Token>
        <ClientEncryptedPassword>string</ClientEncryptedPassword>
        <AllowCaching>boolean</AllowCaching>
        <ClearCache>boolean</ClearCache>
      </LoginResult>
    </AccountLogin2Response>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: 64.156.56.142
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <AccountLogin2 xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <OrgID>long</OrgID>
      <UserName>string</UserName>
      <Password>string</Password>
      <DeviceId>string</DeviceId>
      <AllowedIPList>string</AllowedIPList>
      <ClientEncryptParam>string</ClientEncryptParam>
    </AccountLogin2>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <AccountLogin2Response xmlns="http://api.filesanywhere.com/">
      <LoginResult>
        <ErrorMessage>string</ErrorMessage>
        <Token>string</Token>
        <ClientEncryptedPassword>string</ClientEncryptedPassword>
        <AllowCaching>boolean</AllowCaching>
        <ClearCache>boolean</ClearCache>
      </LoginResult>
    </AccountLogin2Response>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```

Private Sub AccountLogin2 ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim LoginResult As New WebReference.LoginResult2

    LoginResult = FAWAPI.AccountLogin(APIKey, OrgID, UserName,
    Password, DeviceID, AllowedIPList, ClientEncryptParam)
    If LoginResult.ErrorMessage <> "" Then
        MsgBox LoginResult.ErrorMessage
    Else
        Dim Token As String = LoginResult.Token
    End If
    LoginResult = Nothing
    FAWAPI = Nothing
End Sub

```

Request Arguments

Name	Data Type	Required	Description
APIKey	String	Yes	A Developer API Key. The "Getting Started" section describes how to obtain a Developer API Key
OrgID	Long	Yes	The Client ID
UserName	String	Yes	FilesAnywhere Username
Password	String	Yes	FilesAnywhere Password
DeviceID	String	Yes	The Device ID
AllowedIPList	String	No	Comma separated list of IPs to be allowed to make subsequent API calls with the same Token ID passed back by this AccountLogin call
ClientEncryptParam	String	No	<p>A parameter which can be blank or have the following values: "ENCRYPTEDYES", "ENCRYPTEDNO".</p> <p>When it has the value "ENCRYPTEDNO", it means that the password being passed in the AccountLogin method is not encrypted, but, the LoginResult should pass back an encrypted password which will be used in subsequent AccountLogin calls.</p> <p>When an encrypted password is passed to the AccountLogin method, this parameter should be</p>

			assigned the string value "ENCRYPTEDYES". This tells the system, that the password being passed is an encrypted password and it will decrypt it before using it.
--	--	--	--

GetGroupShareUserProfileImage

This method returns the profile image of an user's GroupShare users, when the login token and image size, and username are passed as parameters. The image size returned back can have either of 3 sizes: small (24 x 24), medium (80 x 80), or large (225 x 300).

Syntax

ItemImageResult = GetGroupShareUserProfileImage(Token, ProfileImageSize, Username)

Usage

Use the GetGroupShareUserProfileImage method to get the profile image of an user.

The following parameters are needed to make the GetProfileImage call:

- Token (The Token ID returned by the last AccountLogin call).
- ProfileImageSize is the size of the image to be specified as a string value which can be either "small", "medium" or "large".
- Username is the username of the user whose profile image is to be returned.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```

POST /FACORPAPI/v2/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetGroupShareUserProfileImage"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetGroupShareUserProfileImage xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ProfileImageSize>Small or Medium or Large</ProfileImageSize>
      <UserName>string</UserName>
    </GetGroupShareUserProfileImage>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetGroupShareUserProfileImageResponse
xmlns="http://api.filesanywhere.com/">
      <GetGroupShareUserProfileImageResult>
        <ErrorMessage>string</ErrorMessage>
        <ImageStream>base64Binary</ImageStream>
      </GetGroupShareUserProfileImageResult>
    </GetGroupShareUserProfileImageResponse>
  </soap:Body>
</soap:Envelope>
  
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```

POST /FACORPAPI/v2/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetGroupShareUserProfileImage xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <ProfileImageSize>Small or Medium or Large</ProfileImageSize>
      <UserName>string</UserName>
    </GetGroupShareUserProfileImage>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetGroupShareUserProfileImageResponse
xmlns="http://api.filesanywhere.com/">
      <GetGroupShareUserProfileImageResult>
        <ErrorMessage>string</ErrorMessage>
        <ImageStream>base64Binary</ImageStream>
      </GetGroupShareUserProfileImageResult>
    </GetGroupShareUserProfileImageResponse>
  </soap12:Body>
</soap12:Envelope>

```

Sample Visual Basic.Net Code

```
Private Sub GetGroupShareUserProfileImage ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim ProfileImageResult As New WebReference.ItemImageResult
    ProfileImageResult = FAWAPI.GetGroupShareUserProfileImage (Token,
ProfileImageSize, Username)
    If ProfileImageResult.ErrorMessage IsNot Nothing Then
        MsgBox ProfileImageResult.ErrorMessage
    End If
    ProfileImageResult = Nothing
    FAWAPI = Nothing
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
ProfileImageSize	String	Yes	The size of the image. This can be passed in as small, medium or large.
Username	String	Yes	Username of the user's image is is required to be returned.

GetID3Tag

This method returns the title, artist and album name of a MP3 file.

Syntax

```
GetID3TagResult = GetID3Tag(Token, FilePath)
```

Usage

Use the GetID3Tag method to get the title, artist and album of a MP3 file..

The following parameters are needed to make the GetProfileImage call:

- Token (The Token ID returned by the last AccountLogin call).
- FilePath is the path to the MP3 file. This is the full FilesAnywhere path of the file in the form `\UserVolume\FolderName\FileName`

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /FACORPAPI/v2/FAWAPI.asmx HTTP/1.1

Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetID3Tag"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetID3Tag xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <FilePath>string</FilePath>
    </GetID3Tag>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetID3TagResponse xmlns="http://api.filesanywhere.com/">
      <ID3InfoResult>
        <Title>string</Title>
        <Artist>string</Artist>
        <Album>string</Album>
        <ErrorMessage>string</ErrorMessage>
      </ID3InfoResult>
    </GetID3TagResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /FACORPAPI/v2/FAWAPI.asmx HTTP/1.1

Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetID3Tag xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <FilePath>string</FilePath>
    </GetID3Tag>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetID3TagResponse xmlns="http://api.filesanywhere.com/">
      <ID3InfoResult>
        <Title>string</Title>
        <Artist>string</Artist>
        <Album>string</Album>
        <ErrorMessage>string</ErrorMessage>
      </ID3InfoResult>
    </GetID3TagResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub GetID3Tag ()
    Dim FAWAPI As New WebReference.FAWAPI
    Dim ID3TagResult As New WebReference.GetID3TagResult
    ID3TagResult = FAWAPI. GetID3Tag (Token, FilePath)
    If ID3TagResult.ErrorMessage IsNot Nothing Then
        MsgBox ID3TagResult.ErrorMessage
    End If
    ID3TagResult = Nothing
    FAWAPI = Nothing
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
FilePath	String	Yes	The full FilesAnywhere path of the MP3 file in the form \UserVolume\FolderName\ItemName

GetAPIURL

This function returns the API URL.

Syntax

```
GetAPIURLResult = GetAPIURL(APIKey,ClientId)
```

Usage

Use the GetAPIURL method to get the API URL of a specific client.

The following parameters are needed to make the GetAPIURL call:

- **APIKey** (A Developer API Key. See the “Getting Started” section above, to find out how to obtain a Developer API Key).
- **ClientId** for which you want to get the API URL

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /FACORPAPI/v1/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetAPIURL"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetAPIURL xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <ClientId>int</ClientId>
    </GetAPIURL>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetAPIURLResponse xmlns="http://api.filesanywhere.com/">
      <GetAPIURLResult>
        <ErrorMessage>string</ErrorMessage>
        <APIURL>string</APIURL>
      </GetAPIURLResult>
    </GetAPIURLResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /FACORPAPI/v1/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetAPIURL xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <ClientId>int</ClientId>
    </GetAPIURL>
  </soap12:Body>
</soap12:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetAPIURLResponse xmlns="http://api.filesanywhere.com/">
      <GetAPIURLResult>
        <ErrorMessage>string</ErrorMessage>
        <APIURL>string</APIURL>
      </GetAPIURLResult>
    </GetAPIURLResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```

Private Sub GetAPIURL ()
    Dim FAWAPI As New WebReference.FAWAPI ()
    Dim APIURLResult As New WebReference.GetAPIURLResult
    APIURLResult = FAWAPI.GetAPIURL (APIKey, ClientId)
    If APIURLResult.ErrorMessage.Length > 0 Then
        MsgBox (APIURLResult.ErrorMessage)
    Else
        MsgBox (APIURLResult.APIURL)
    End If
    APIURLResult = Nothing
    FAWAPI = Nothing
End Sub

```

Request Arguments

Name	Data Type	Required	Description
APIKey	String	Yes	A Developer API Key. The “Getting Started” section describes how to obtain a Developer API Key
ClientId	Integer	Yes	Client Id for which you want to know API URL

Response

GetAPIURLResult

GetAPIURLResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the GetAPIURL method was executed. If the GetAPIURL call was successful, the ErrorMessage is blank
APIURL	String	If the GetAPIURL call was successful, APIURL contains the API URL of given ClientId

GetClientID

Provides you the ClientID for a Company code.

Syntax

```
GetClientIDResult = GetClientID(APIKey, Parameters)
```

Usage

Use the GetClientID method to get the ClientID of a specific Company code.

The following parameters are needed to make the GetClientID call:

- APIKey (A Developer API Key. See the “Getting Started” section above, to find out how to obtain a Developer API Key).
- Parameters is the object which contains the following property to hold the required parameters value to get the Client ID:
 - CompanyCode: Company code of the client for which you want to get the ClientID.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetClientID"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetClientID xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <Parameters>
        <CompanyCode>string</CompanyCode>
      </Parameters>
    </GetClientID>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetClientIDResponse xmlns="http://api.filesanywhere.com/">
      <GetClientIDResult>
        <ErrorMessage>string</ErrorMessage>
        <ClientID>int</ClientID>
      </GetClientIDResult>
    </GetClientIDResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetClientID xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <Parameters>
        <CompanyCode>string</CompanyCode>
      </Parameters>
    </GetClientID>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetClientIDResponse xmlns="http://api.filesanywhere.com/">
      <GetClientIDResult>
        <ErrorMessage>string</ErrorMessage>
        <ClientID>int</ClientID>
      </GetClientIDResult>
    </GetClientIDResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub GetClientID()
    Dim FAWAPI As New WebReference.FAWAPI()
    Dim objGetClientIDParameter As New FAWAPI.GetClientIDParameter()
    objGetClientIDParameter.CompanyCode = sCompanyCode
    Dim objClientIDResult As WebReference.GetClientIDResult =
FAWAPI.GetClientID(APIKey, objGetClientIDParameter)
    If objClientIDResult.ErrorMessage.Length > 0 Then
        MsgBox(objClientIDResult.ErrorMessage)
    Else
        MsgBox(objClientIDResult.ClientID)
    End If
    objClientIDResult = Nothing
    FAWAPI = Nothing
End Sub
```

Request Arguments

Name	Data Type	Required	Description
APIKey	String	Yes	A Developer API Key. The "Getting Started" section describes how to obtain a Developer API Key
Parameters	Object	Yes	Object of class for all the parameters

Parameters Properties

Name	Data Type	Required	Description
CompanyCode	String	Yes	CompanyCode for which you want to get the Client Id.

Response

GetClientIDResult

GetClientIDResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the GetAPIURL method was executed. If the GetClientID call was successful, the ErrorMessage is blank
ClientID	Int	If the GetClientID call was successful, ClientID contains the ClientID of given Company code.

GetCompanyDetails

Provides you the company details for a given Domain name.

Syntax

GetCompanyDetailsResult = GetCompanyDetails (APIKey, Parameters)

Usage

Use the GetCompanyDetails method to get the Company detail(s) (Company code and ClientID) of a specific Domain name.

The following parameters are needed to make the GetCompanyDetails call:

- **APIKey** (A Developer API Key. See the “Getting Started” section above, to find out how to obtain a Developer API Key).
- **Parameters** is the object which contains the following property to hold the required parameters value to get the company detail:
 - **DomainName**: Domain name of the client for which you want to get the company details.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```

POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetCompanyDetails"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetCompanyDetails xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <Parameters>
        <DomainName>string</DomainName>
      </Parameters>
    </GetCompanyDetails>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetCompanyDetailsResponse xmlns="http://api.filesanywhere.com/">
      <GetCompanyDetailsResult>
        <ErrorMessage>string</ErrorMessage>
        <CompanyDetails>
          <CompanyDetail>
            <ClientID>int</ClientID>
            <CompanyCode>string</CompanyCode>
          </CompanyDetail>
          <CompanyDetail>
            <ClientID>int</ClientID>
            <CompanyCode>string</CompanyCode>
          </CompanyDetail>
        </CompanyDetails>
      </GetCompanyDetailsResult>
    </GetCompanyDetailsResponse>
  </soap:Body>
</soap:Envelope>
  
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetCompanyDetails xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <Parameters>
        <DomainName>string</DomainName>
      </Parameters>
    </GetCompanyDetails>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetCompanyDetailsResponse xmlns="http://api.filesanywhere.com/">
      <GetCompanyDetailsResult>
        <ErrorMessage>string</ErrorMessage>
        <CompanyDetails>
          <CompanyDetail>
            <ClientID>int</ClientID>
            <CompanyCode>string</CompanyCode>
          </CompanyDetail>
          <CompanyDetail>
            <ClientID>int</ClientID>
            <CompanyCode>string</CompanyCode>
          </CompanyDetail>
        </CompanyDetails>
      </GetCompanyDetailsResult>
    </GetCompanyDetailsResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```

Private Sub GetComapnyDetails ()
    Dim FAWAPI As New WebReference.FAWAPI ()
    Dim objGetCompanyDetailsParameter As New FAWAPI.
GetCompanyDetailsParameter()
    objGetCompanyDetailsParameter.DomainName = sDomainName
    Dim objGetCompanyDetailsResult As WebReference. GetCompanyDetailsResult
= FAWAPI.GetCompanyDetails (APIKey, objGetCompanyDetailsParameter)
    If objGetCompanyDetailsResult.ErrorMessage.Length > 0 Then
        MsgBox (objGetCompanyDetailsResult.ErrorMessage)
    Else
        MsgBox (objGetCompanyDetailsResult.ClientID)
        MsgBox (objGetCompanyDetailsResult.CompanyCode)

    End If
    objGetCompanyDetailsResult = Nothing
    FAWAPI = Nothing
End Sub

```

Request Arguments

Name	Data Type	Required	Description
APIKey	String	Yes	A Developer API Key. The “Getting Started” section describes how to obtain a Developer API Key
Parameters	Object	Yes	Object of class for all the parameters

Parameters Properties

Name	Data Type	Required	Description
DomainName	String	Yes	DomainName for which you want to get the Client Id and CompanyCode

Response

GetCompanyDetailsResult

GetCompanyDetailsResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the GetAPIURL method was executed. If the GetCompanyDetails call was

		successful, the ErrorMessage is blank
CompanyDetails	Company Detail Array	If GetComapnyDetails call was successful Array of CompanyDetail will return.

ComapnyDetail Properties

Name	Data Type	Description
ClientID	Int	ClientID of company.
CompanyCode	String	CompanyCode of company.

GetStorageLocation

This function returns the storage locations for use with FTP, SFTP and WebDAV.

Syntax

```
GetStorageLocationResult = GetStorageLocation()
```

Usage

Use the `GetStorageLocation` function to get storage locations for FilesAnywhere FTP, SFTP and WebDAV servers.

The following parameter is needed to make the `GetStorageLocation` call:

- **APIKey** (A Developer API Key. See the “Getting Started” section above, to find out how to obtain a Developer API Key).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /FACORPAPI/v1/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetStorageLocation"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetStorageLocation xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
    </GetStorageLocation>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetStorageLocationResponse xmlns="http://api.filesanywhere.com/">
      <GetStorageLocationResult>
        <ErrorMessage>string</ErrorMessage>
        <StorageLocations>
          <StorageLocation>
            <URLType>string</URLType>
            <SiteURL>string</SiteURL>
          </StorageLocation>
          <StorageLocation>
            <URLType>string</URLType>
            <SiteURL>string</SiteURL>
          </StorageLocation>
        </StorageLocations>
      </GetStorageLocationResult>
    </GetStorageLocationResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /FACORPAPI/v1/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetStorageLocation xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
    </GetStorageLocation>
  </soap12:Body>
</soap12:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetStorageLocationResponse xmlns="http://api.filesanywhere.com/">
      <GetStorageLocationResult>
        <ErrorMessage>string</ErrorMessage>
        <StorageLocations>
          <StorageLocation>
            <URLType>string</URLType>
            <SiteURL>string</SiteURL>
          </StorageLocation>
          <StorageLocation>
            <URLType>string</URLType>
            <SiteURL>string</SiteURL>
          </StorageLocation>
        </StorageLocations>
      </GetStorageLocationResult>
    </GetStorageLocationResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```

Private Sub GetStorageLocation ()
    Dim FAWAPI As New WebReference.FAWAPI ()
    Dim StorageLocationResult As New
WebReference.GetStorageLocationResult

    StorageLocationResult = FAWAPI.GetStorageLocation (APIKey)
    If StorageLocationResult.ErrorMessage.Length > 0 Then
        MsgBox (StorageLocationResult.ErrorMessage)
    Else
        For Each sl As WebReference.StorageLocation In
StorageLocationResult.StorageLocations
            'Use StorageLocation type to fetch details
            MsgBox (sl.URLType & ": " & sl.SiteURL)
        Next
    End If
    StorageLocationResult = Nothing
    FAWAPI = Nothing
End Sub

```

Request Arguments

Name	Data Type	Required	Description
APIKey	String	Yes	A Developer API Key. The "Getting Started" section describes how to obtain a Developer API Key

Response

GetStorageLocationResult

GetStorageLocationResult PropertiesName	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the GetStorageLocation method was executed. If the GetStorageLocation call was successful, the ErrorMessage is blank
StorageLocations	List of type StorageLocation	If the GetStorageLocation call was successful, StorageLocations contains a list of type StorageLocation

StorageLocation

StorageLocation Properties

Name	Data Type	Description
URLType	String	Type of storage location
SiteURL	String	Storage location URL of specific storage type

GetFolderAccessKey

This function returns the folder access key for a folder belonging to a specified user.

Syntax

```
GetFolderAccessKeyResult = GetFolderAccessKey(Token, UserName, Password, FolderPath)
```

Usage

Use the `GetFolderAccessKey` function to get or generate the unique folder access key for a folder belonging to a specified user.

The following parameters are needed to make the `GetFolderAccessKey` call:

- **Token** (The Token ID returned by the last `AccountLogin` call).
- **UserName** (FilesAnywhere Username). This is the FilesAnywhere username using which the user logs in to the FilesAnywhere account.
- **Password** (FilesAnywhere Password). This is the user's FilesAnywhere account password.
- **FolderPath** is the path of a folder or a folder name.
 - User can pass the full FilesAnywhere path of the folder in the form `\UserVolume\FolderNameA\FolderNameB\`
 - User can pass only the folder name (without `\`). In this case, the API will create the folder in the first writable folder it finds.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetFolderAccessKey"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetFolderAccessKey xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <UserName>string</UserName>
      <Password>string</Password>
      <FolderPath>string</FolderPath>
    </GetFolderAccessKey>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetFolderAccessKeyResponse xmlns="http://api.filesanywhere.com/">
      <GetFolderAccessKeyResult>
        <ErrorMessage>string</ErrorMessage>
        <FolderAccessKey>string</FolderAccessKey>
      </GetFolderAccessKeyResult>
    </GetFolderAccessKeyResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.


```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetFolderAccessKey xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <UserName>string</UserName>
      <Password>string</Password>
      <FolderPath>string</FolderPath>
    </GetFolderAccessKey>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetFolderAccessKeyResponse xmlns="http://api.filesanywhere.com/">
      <GetFolderAccessKeyResult>
        <ErrorMessage>string</ErrorMessage>
        <FolderAccessKey>string</FolderAccessKey>
      </GetFolderAccessKeyResult>
    </GetFolderAccessKeyResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub GetFolderAccessKey()  
    Dim FAWAPI As New WebReference.FAWAPI  
    Dim FolderAccessKeyResult  
        As New WebReference.GetFolderAccessKeyResult  
  
    FolderAccessKeyResult = FAWAPI.GetFolderAccessKeyResult (Token,  
UserName, Password, FolderPath)  
    If FolderAccessKeyResult.ErrorMessage IsNot Nothing Then  
        MsgBox FolderAccessKeyResult.ErrorMessage  
    Else  
        MsgBox FolderAccessKeyResult.FolderAccessKey  
    End If  
    FolderAccessKeyResult = Nothing  
    FAWAPI = Nothing  
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
UserName	String	Yes	FilesAnywhere username using which the user logs in to the FilesAnywhere account.
Password	String	Yes	User's FilesAnywhere account password.
FolderPath	String	Yes	The full FilesAnywhere path of the folder in the form \UserVolume\FolderNameA\FolderNameB\ OR Only Folder Name (withour \). In this case, the API will create the folder in the first writable folder it finds.

Response**GetFolderAccessKeyResult**

GetFolderAccessKeyResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the <code>GetFolderAccessKey</code> method was executed. If the <code>GetFolderAccessKey</code> call was successful, the <code>ErrorMessage</code> is blank
FolderAccessKey	String	Unique ID of Folder which can be used to upload files into associated folder for user.

UploadFileWithFolderAccessKey

Upload a file using a folder access key.

Syntax

```
UploadFileResult = UploadFileWithFolderAccessKey(APIKey, FolderAccessKey,
FileName, FileData)
```

Usage

Use the `UploadFileWithFolderAccessKey` function to upload file to a user's FilesAnywhere account using the `FolderAccessKey` that the user has shared with you.

The following parameters are needed to make the `UploadFileWithFolderAccessKey` call:

- **APIKey** (A Developer API Key. See the "Getting Started" section above, to find out how to obtain a Developer API Key).
- **FolderAccessKey**. Unique Access Key of a folder belonging to a specified user which can be generated using `GetFolderAccessKey` API.
- **FileName** (Name of the file to be uploaded).
- **FileData** (The content of the file in binary format). The data read from the file must be converted to the XML base64Binary format.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/UploadFileWithFolderAccessKey"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UploadFileWithFolderAccessKey xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <FolderAccessKey>string</FolderAccessKey>
      <FileName>string</FileName>
      <FileData>base64Binary</FileData>
    </UploadFileWithFolderAccessKey>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UploadFileWithFolderAccessKeyResponse
xmlns="http://api.filesanywhere.com/">
      <UploadFileResult>
        <ErrorMessage>string</ErrorMessage>
        <Uploaded>boolean</Uploaded>
      </UploadFileResult>
    </UploadFileWithFolderAccessKeyResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <UploadFileWithFolderAccessKey xmlns="http://api.filesanywhere.com/">
      <APIKey>string</APIKey>
      <FolderAccessKey>string</FolderAccessKey>
      <FileName>string</FileName>
      <FileData>base64Binary</FileData>
    </UploadFileWithFolderAccessKey>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <UploadFileWithFolderAccessKeyResponse
xmlns="http://api.filesanywhere.com/">
      <UploadFileResult>
        <ErrorMessage>string</ErrorMessage>
        <Uploaded>boolean</Uploaded>
      </UploadFileResult>
    </UploadFileWithFolderAccessKeyResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub GetFolderAccessKey()  
    Dim FAWAPI As New WebReference.FAWAPI  
    Dim UploadFileResults  
        As New WebReference.UploadFileResult  
  
    UploadFileResults = FAWAPI.UploadFileWithFolderAccessKey  
(APIKey, FolderAccessKey, FileName, FileData)  
    If UploadFileResults.ErrorMessage IsNot Nothing Then  
        MsgBox UploadFileResults.ErrorMessage  
    Else  
        MsgBox UploadFileResults.Uploaded  
    End If  
    UploadFileResults = Nothing  
    FAWAPI = Nothing  
End Sub
```

Request Arguments

Name	Data Type	Required	Description
APIKey	String	Yes	A Developer API Key. See the "Getting Started" section above, to find out how to obtain a Developer API Key
FolderAccessKey	String	Yes	Unique Access Key of a folder belonging to a specified user.
FileName	String	Yes	Name of the file to be uploaded.
FileData	Base64Binary	Yes	The contents of the file as binary data in XML base64Binary format

Response**UploadFileResult**

UploadFileResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the UploadFile method was executed. If the UploadFile call was successful, the ErrorMessage is blank
Uploaded	Boolean	True or False indicating whether the UploadFile call was a success or a failure User will not get any progress update of upload by using this method.

UpdateUserProfile

Update user profile details.

Note: If the following fields are left blank, the system will erase any existing content on the profile page: First Name, Last Name, Company Name, Job Title, Phone, Cell Phone, Address, City, State, Zip and Country. If PrimaryEmail or SecondaryEmail are left blank, then the system will not make any changes to the existing primary or secondary email address.

Syntax

UpdateUserProfileResult = UpdateUserProfile(Token, Parameters)

Usage

Use the UpdateUserProfile function to update the user profile details.

The following parameters are needed to make the UpdateUserProfile call:

- Token (The Token ID returned by the last AccountLogin call).
- Parameters is the object which contains the following properties to hold the required values to set the user profile details:
 - FirstName: First Name
 - LastName: Last Name
 - CompanyName: Company Name
 - JobTitle: Job Title
 - Phone: Phone Number
 - CellPhone: Cell Phone Number
 - Address
 - State
 - City
 - State
 - Zip

- Country
- PrimaryEmail: Primary Email
- SecondaryEmail: Secondary Email

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/UpdateUserProfile"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UpdateUserProfile xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Parameters>
        <FirstName>string</FirstName>
        <LastName>string</LastName>
        <CompanyName>string</CompanyName>
        <JobTitle>string</JobTitle>
        <Phone>string</Phone>
        <CellPhone>string</CellPhone>
        <Address>string</Address>
        <City>string</City>
        <State>string</State>
        <Zip>string</Zip>
        <Country>string</Country>
        <PrimaryEmail>string</PrimaryEmail>
        <SecondaryEmail>string</SecondaryEmail>
      </Parameters>
    </UpdateUserProfile>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UpdateUserProfileResponse xmlns="http://api.filesanywhere.com/">
```

```
<UpdateUserProfileResult />
</UpdateUserProfileResponse>
</soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <UpdateUserProfile xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Parameters>
        <FirstName>string</FirstName>
        <LastName>string</LastName>
        <CompanyName>string</CompanyName>
        <JobTitle>string</JobTitle>
        <Phone>string</Phone>
        <CellPhone>string</CellPhone>
        <Address>string</Address>
        <City>string</City>
        <State>string</State>
        <Zip>string</Zip>
        <Country>string</Country>
        <PrimaryEmail>string</PrimaryEmail>
        <SecondaryEmail>string</SecondaryEmail>
      </Parameters>
    </UpdateUserProfile>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <UpdateUserProfileResponse xmlns="http://api.filesanywhere.com/">
```

```
<UpdateUserProfileResult />
</UpdateUserProfileResponse>
</soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub UpdateUserProfile()
    Dim UpdateUserProfileParameter As New
    FAWAPI.UpdateUserProfileParameter()
    UpdateUserProfileParameter.FirstName = sFirstName
    UpdateUserProfileParameter.LastName = sLastName
    UpdateUserProfileParameter.CompanyName = sCompanyName
    UpdateUserProfileParameter.JobTitle = dJobTitle
    UpdateUserProfileParameter.Phone = sPhone
    UpdateUserProfileParameter.CellPhone = sCellPhone
    UpdateUserProfileParameter.Address = sAddress
    UpdateUserProfileParameter.City = sCity
    UpdateUserProfileParameter.State = sState
    UpdateUserProfileParameter.Zip = sZip
    UpdateUserProfileParameter.Country = sCountry
    UpdateUserProfileParameter.PrimaryEmail = sPrimaryEmail
    UpdateUserProfileParameter.SecondaryEmail = sSecondaryEmail
    Dim UpdateUserProfileResult As FAWAPI.UpdateUserProfileResult =
    FAWAPI.UpdateUserProfile(sToken, UpdateUserProfileParameter)
    If UpdateUserProfileResult.ErrorMessage.Length > 0 Then
        MsgBox UpdateUserProfileResult.ErrorMessage
    End If
    UpdateUserProfileResult = Nothing
    FAWAPI = Nothing
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
Parameters	Object	Yes	Object of class for all the parameters

Parameters Properties

Name	Data Type	Required	Description
FirstName	String	No	User's first name
LastName	String	No	User's last name
CompanyName	String	No	User's company name

JobTitle	String	No	Job tilte
Phone	String	No	User's phone number
CellPhone	String	No	User's cell phone number
Address	String	No	Address
City	String	No	City
State	String	No	State
Zip	String	No	Zip
Country	String	No	Country
PrimaryEmail	String	No	User's primary email address
SecondaryEmail	String	No	User's secondary email address

Response

UpdateUserProfilResult

UpdateUserProfilResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the UpdateUserProfile method was executed. If the UpdateUserProfile call was successful, the ErrorMessage is blank

GetClientDetails

Get Client details. (Note: As of now this function returns back just the client email address. More items will be added to the output as the need arises and the documentation will be updated accordingly.)

Syntax

GetClientDetailsResult = GetClientDetails (Token)

Usage

Use the GetClientDetails function to get client specific details.

The following parameters are needed to make the GetClientDetails call:

- Token (The Token ID returned by the last AccountLogin call).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/GetClientDetails"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetClientDetails xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </GetClientDetails>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetClientDetailsResponse xmlns="http://api.filesanywhere.com/">
```

```
<GetClientDetailsResult>
  <EmailAddress>string</EmailAddress>
</GetClientDetailsResult>
</GetClientDetailsResponse>
</soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetClientDetails xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
    </GetClientDetails>
  </soap12:Body>
</soap12:Envelope>
</soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <GetClientDetailsResponse xmlns="http://api.filesanywhere.com/">
      <GetClientDetailsResult>
        <EmailAddress>string</EmailAddress>
      </GetClientDetailsResult>
    </GetClientDetailsResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample Visual Basic.Net Code

```
Private Sub GetClientDetails ()
    Dim GetClientDetailsResult As FAWAPI.GetClientDetailsResult =
    FAWAPI.GetClientDetails(sToken)
    If GetClientDetailsResult.ErrorMessage.Length > 0 Then
        MsgBox GetClientDetailsResult.ErrorMessage
    Else
        MsgBox GetClientDetailsResult.EmailAddress
    End If
    GetClientDetailsResult = Nothing
    FAWAPI = Nothing
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call

Response

GetClientDetailsResult

GetClientDetailsResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the GetClientDetails method was executed. If the GetClientDetails call was successful, the ErrorMessage is blank
EmailAddress	String	Client's system email address

SetProfileImage

Set user profile image.

Syntax

SetProfileImageResult = SetProfileImage (Token, Parameters)

Usage

Use the `SetProfileImage` function to upload and set the user profile image (maximum size 10MB).

The following parameters are needed to make the `SetProfileImage` call:

- **Token** (The Token ID returned by the last `AccountLogin` call).
- **Parameters** is the object which contains the following properties to hold the required values to set the user profile image:
 - **FileName**: Image File Name
 - **FileData**: Image content as binary data

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/SetProfileImage"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetProfileImage xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Parameters>
        <FileName>string</FileName>
        <FileData>base64Binary</FileData>
      </Parameters>
    </SetProfileImage>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetProfileImageResponse xmlns="http://api.filesanywhere.com/">
      <SetProfileImageResult />
    </SetProfileImageResponse>
```

```

</soap:Body>
</soap:Envelope>

```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The **placeholders** shown need to be replaced with actual values.

```

POST /FACORPAPI/FAWAPI.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SetProfileImage xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Parameters>
        <FileName>string</FileName>
        <FileData>base64Binary</FileData>
      </Parameters>
    </SetProfileImage>
  </soap12:Body>
</soap12:Envelope>

```

```

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SetProfileImageResponse xmlns="http://api.filesanywhere.com/">
      <SetProfileImageResult />
    </SetProfileImageResponse>
  </soap12:Body>
</soap12:Envelope>

```

Sample Visual Basic.Net Code

```

Private Sub SetProfileImage()
    Dim SetProfileImageParameter As New
    FAWAPI.SetProfileImageParameter()
    SetProfileImageParameter.FileName = sFileName
    SetProfileImageParameter.FileData = bFileData
    Dim SetProfileImageResult As FAWAPI.SetProfileImageResult =
    FAWAPI.SetProfileImage(sToken, SetProfileImageParameter)

```

```
If SetProfileImageResult.ErrorMessage.Length > 0 Then
    MsgBox SetProfileImageResult.ErrorMessage
End If
SetProfileImageResult = Nothing
FAWAPI = Nothing
End Sub
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
Parameters	Object	Yes	Object of class for all the parameters

Parameters Properties

Name	Data Type	Required	Description
FileName	String	Yes	Image file name
FileData	Base64Binary	Yes	Image content as binary data

Response

SetProfileImageResult

SetProfileImageResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the SetProfileImage method was executed. If the SetProfileImage call was successful, the ErrorMessage is blank

SendFileShareLink

Shares files and folders securely with anyone, even non-members, using a web URL link. An email is sent by the system containing an encrypted web link, which when clicked by the recipient, will prompt them to download the shared files and folders, or view them online, according to the parameters (options) given when calling the

function. If the 'RecipientEmails' parameter is equal to the value 'GENERALLINK', the function will return the FileShare URL Link without sending an email. This link can be used directly from any location, and the link behaves exactly like the way it does when sent in an email.

Syntax

`SendFileShareLinkResult= SendFileShareLink(Token, FileShareLinkItems,RecipientEmails,FolderView,ShareDays,RecordFileHistoryLog,NotifyDownloadByEmail,SendReadReceipt,LinkInFileAttachment,ReadOnlyPermission,DownloadLimit,DisplayWatermark,WatermarkAtCenter,WatermarkAtBottom,UserEmailAsSender,EmailSubject,EmailBody,LinkPassWord,FlagCoverPage,CoverPageTitle,CoverPageMessage,ApprovalText,InputPromptMessage,SignaturePrompt,PlayOptions,DisplayUserName,DisplayEmailAddress,DisplayComments)`

Usage

Use the `SendFileShareLink` method to create and send a FileShare link.

The following parameters are needed to make the `SendFileShareLink` call:

- `Token`. (The Token ID returned by the last `AccountLogin` call).
- `FileShareLinkItems`. (Array of items to include in the FileShare).
- `RecipientEmails`. (List of email addresses to send the FileShare link to).
- `FolderView`. (Flag indicating the type of view to display).
- `ShareDays`. (Number of days the FileShare link will be active).
- `RecordFileHistoryLog`. (Flag indicating to log file activity).
- `NotifyDownloadByEmail`. (Flag indicating an email notification to be sent to recipients).
- `SendReadReceipt`. (Flag indicating a read receipt should be generated for the email notification when it is read).
- `LinkInFileAttachment`. ()
- `ReadOnlyPermission`. (Flag indicating files can be downloaded or only viewed).
- `DownloadLimit`. (Maximum number of downloads for the FileShare).
- `DisplayWatermark`. (Flag indicating the file should be watermarked).

- **WatermarkAtCenter.** (Text displayed in the center watermark).
- **WatermarkAtBottom.** (Text displayed in the bottom watermark).
- **UserEmailAsSender.** (Flag indicating to send email notifications from the system email account or the user's email account).
- **EmailSubject.** (The subject line of the email notification).
- **EmailBody.** (The body of the email notification).
- **LinkPassWord.** (The password assigned to the FileShare Link).
- **FlagCoverPage.** (Flag indicating the FileShare has a cover page).
- **CoverPageTitle.** (Title text for the cover page).
- **CoverPageMessage.** (Message text for the cover page).
- **ApprovalText.** (Text to display to obtain the user's approval).
- **InputPromptMessage.** (Text to display to obtain input from the user).
- **SignaturePrompt.** (Text to display to obtain the user's signature).
- **PlayOptions.** (Flag indicating the play options for the FileShare).
- **DisplayUserName.** (Flag indicating to display the user's name).
- **DisplayEmail.** (Flag indicating to display the user's email address).
- **DisplayComments.** (Flag indicating to display the comments).
- **AddEmailInWaterMark** (Include recipient's email address into watermark)
- **AddFileShareIDInWaterMark** (Include FileShare ID into watermark)
- **AddIPInWaterMark** (Include IP address into watermark)
- **AddTimeStampInWaterMark** (Include downloaded item's timestamp into watermark)
- **WaterMarkSize** (Watermark size in pixel, possible values for watermark size are {8, 10, 12, 14, 16, 18, 20, 22, 24})

- **WaterMarkColor** (Color of watermark text, possible values for watermark colors are RD for Red, GN for Green, GY for Grey, WE for White, BE for Blue, BK for Black)
- **PasswordSecurity** (File share link will be protected by password or OTP email or OTP SMS or none if provided value as either 1 or 2 or 3 or 0 respectively)

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/SendFileShareLink"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendFileShareLink xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Parameters>
        <FileShareLinkItems>
          <Item>
            <Type>string</Type>
            <Path>string</Path>
          </Item>
          <Item>
            <Type>string</Type>
            <Path>string</Path>
          </Item>
        </FileShareLinkItems>
        <RecipientEmails>string</RecipientEmails>
        <FolderView>string</FolderView>
        <ShareDays>int</ShareDays>
        <RecordFileHistoryLog>string</RecordFileHistoryLog>
        <NotifyDownloadByEmail>string</NotifyDownloadByEmail>
        <SendReadReceipt>string</SendReadReceipt>
        <LinkInFileAttachment>string</LinkInFileAttachment>
        <ReadOnlyPermission>string</ReadOnlyPermission>
        <DownloadLimit>int</DownloadLimit>
        <DisplayWatermark>string</DisplayWatermark>
        <WatermarkAtCenter>string</WatermarkAtCenter>
        <WatermarkAtBottom>string</WatermarkAtBottom>
        <UserEmailAsSender>string</UserEmailAsSender>
        <EmailSubject>string</EmailSubject>
        <EmailBody>string</EmailBody>
      </Parameters>
    </SendFileShareLink>
  </soap:Body>
</soap:Envelope>
```

```

    <LinkPassword>string</LinkPassword>
    <FlagCoverPage>string</FlagCoverPage>
    <CoverPageTitle>string</CoverPageTitle>
    <CoverPageMessage>string</CoverPageMessage>
    <ApprovalText>string</ApprovalText>
    <InputPromptMessage>string</InputPromptMessage>
    <SignaturePrompt>string</SignaturePrompt>
    <PlayOptions>string</PlayOptions>
    <DisplayUserName>boolean</DisplayUserName>
    <DisplayEmail>boolean</DisplayEmail>
    <DisplayComments>boolean</DisplayComments>
    <AddEmailInWaterMark>boolean</AddEmailInWaterMark>
    <AddFileShareIDInWaterMark>boolean</AddFileShareIDInWaterMark>
    <AddIPInWaterMark>boolean</AddIPInWaterMark>
    <AddTimeStampInWaterMark>boolean</AddTimeStampInWaterMark>
    <WaterMarkSize>int</WaterMarkSize>
    <WaterMarkColor>string</WaterMarkColor>
    <PasswordSecurity>int</PasswordSecurity>
  </Parameters>
</SendFileShareLink>
</soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

```

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendFileShareLinkResponse xmlns="http://api.filesanywhere.com/">
      <SendFileShareLinkResult />
    </SendFileShareLinkResponse>
  </soap:Body>
</soap:Envelope>

```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The [placeholders](#) shown need to be replaced with actual values.

```

POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SendFileShareLink xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Parameters>

```

```

<FileShareLinkItems>
  <Item>
    <Type>string</Type>
    <Path>string</Path>
  </Item>
  <Item>
    <Type>string</Type>
    <Path>string</Path>
  </Item>
</FileShareLinkItems>
<RecipientEmails>string</RecipientEmails>
<FolderView>string</FolderView>
<ShareDays>int</ShareDays>
<RecordFileHistoryLog>string</RecordFileHistoryLog>
<NotifyDownloadByEmail>string</NotifyDownloadByEmail>
<SendReadReceipt>string</SendReadReceipt>
<LinkInFileAttachment>string</LinkInFileAttachment>
<ReadOnlyPermission>string</ReadOnlyPermission>
<DownloadLimit>int</DownloadLimit>
<DisplayWatermark>string</DisplayWatermark>
<WatermarkAtCenter>string</WatermarkAtCenter>
<WatermarkAtBottom>string</WatermarkAtBottom>
<UserEmailAsSender>string</UserEmailAsSender>
<EmailSubject>string</EmailSubject>
<EmailBody>string</EmailBody>
<LinkPassword>string</LinkPassword>
<FlagCoverPage>string</FlagCoverPage>
<CoverPageTitle>string</CoverPageTitle>
<CoverPageMessage>string</CoverPageMessage>
<ApprovalText>string</ApprovalText>
<InputPromptMessage>string</InputPromptMessage>
<SignaturePrompt>string</SignaturePrompt>
<PlayOptions>string</PlayOptions>
<DisplayUserName>boolean</DisplayUserName>
<DisplayEmail>boolean</DisplayEmail>
<DisplayComments>boolean</DisplayComments>
<AddEmailInWaterMark>boolean</AddEmailInWaterMark>
<AddFileShareIDInWaterMark>boolean</AddFileShareIDInWaterMark>
<AddIPInWaterMark>boolean</AddIPInWaterMark>
<AddTimeStampInWaterMark>boolean</AddTimeStampInWaterMark>
<WaterMarkSize>int</WaterMarkSize>
<WaterMarkColor>string</WaterMarkColor>
<PasswordSecurity>Int</PasswordSecurity>
</Parameters>
</SendFileShareLink>
</soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>

```



```

<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <SendFileShareLinkResponse xmlns="http://api.filesanywhere.com/">
      <SendFileShareLinkResult />
    </SendFileShareLinkResponse>
  </soap12:Body>
</soap12:Envelope>

```

Sample C#.Net Code

```

private void CreateFSLink_Click()
{
    try {
        wndOneLoginClient.FAWAPI.FAWAPIv2SoapClient objFAWAPI = new
FAWAPI.FAWAPIv2SoapClient();
        FAWAPI.SendFileShareLinkParameter objParams = new
FAWAPI.SendFileShareLinkParameter();
        wndOneLoginClient.FAWAPI.ItemDetails objElinkItems = new
FAWAPI.ItemDetails();
        objElinkItems.Path = txtFSPath.Text;
        objElinkItems.Type = "file";
        List<wndOneLoginClient.FAWAPI.ItemDetails> objListElinkItems = new
List<FAWAPI.ItemDetails>();
        if (txtWM.Text == "")
        {
            MessageBox.Show("Please provide watermark text");
            return;
        }

        objListElinkItems.Add(objElinkItems);
        objParams.FileShareLinkItems = objListElinkItems.ToArray();
        objParams.RecipientEmails = txtReceipient.Text;
        objParams.EmailSubject = txtFSSubject.Text;
        objParams.ShareDays = Convert.ToInt32(txtFSSharedDays.Text);
        objParams.RecordFileHistoryLog = "Y";
        objParams.NotifyDownloadByEmail = "Y";
        objParams.ReadOnlyPermission = "N"; //ReadOnlyPermission value can
only be 'Y' for Yes, 'N' for No or 'P' for Allow Printing."
        objParams.DisplayWatermark = "N";
        objParams.UserEmailAsSender = "Y";
        objParams.SendReadReceipt = "N";
        objParams.LinkInFileAttachment = "N";
        objParams.DisplayWatermark = rbWMNo.Checked == true ? "N" : "Y";
        if (cmdWMPos.SelectedIndex == 0)
        {
            objParams.WatermarkAtCenter = txtWM.Text;
        }
        else
            objParams.WatermarkAtBottom = txtWM.Text;

        objParams.WaterMarkColor = txtColor.Text;
    }
}

```

```
objParams.WaterMarkSize = Convert.ToInt16(txtSize.Text);
objParams.AddEmailInWaterMark = chkEmail.Checked;
objParams.AddFileShareIDInWaterMark = chkFSID.Checked;
objParams.AddIPInWaterMark = chkWMIP.Checked;
objParams.AddTimeStampInWaterMark = chkDwnlTS.Checked;

FAWAPI.SendFileShareLinkResult objFSLink =
objFAWAPI.SendFileShareLink(msToken, objParams);
if (!string.IsNullOrEmpty(objFSLink.ErrorMessage))
{
    txtError.Text = objFSLink.ErrorMessage;
    txtError.ForeColor = Color.Red;
    return;
}

txtError.Text = objFSLink.ELinkURLs[0].URL;
}catch(Exception ex)
{
    txtError.Text = ex.Message;
}
}
```

Request Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
FileShareLinkItems	Object	Yes	Array of items to be included in the FileShare.
RecipientEmails	String	Yes	Semi-colon-separated list of email addresses to send the FileShare link to.
FolderView	String	No	Flag options for the default view of the FileShare.
ShareDays	Integer	Yes	The number of days the FileShare link will remain active.
RecordFileHistoryLog	String	Yes	Flag indicating to log activity.
NotifyDownloadByEmail	String	Yes	Flag indicating to send email notification to recipients.
SendReadReceipt	String	Yes	Flag indicating a read receipt should be generated for the email notification when it is read.
LinkInFileAttachment	String	Yes	
ReadOnlyPermission	String	Yes	Flag indicating whether the file share allows downloads or is view-only.
DownloadLimit	Integer	No	Maximum number of downloads for the FileShare.
DisplayWatermark	String	Yes	Flag indicating to display a watermark.
WatermarkAtCenter	String	No	Text displayed in center watermark.
WatermarkAtBottom	String	No	Text displayed in bottom watermark.
UserEmailAsSender	String	Yes	Flag indicating the email notification should be sent from the system email account or the user's email account.
EmailSubject	String	Yes	Email subject in the notification.
EmailBody	String	Yes	Email body in the notification.

LinkPassWord	String	No	The password assigned to the FileShare link.
FlagCoverPage	String	No	Flag indicating the FileShare link has a cover page.
CoverPageTitle	String	No	Title text for the cover page.
CoverPageMessage	String	No	Message text for the cover page.
ApprovalText	String	No	Text displayed to obtain approval from the user.
InputPromptMessage	String	No	Text displayed to obtain input from the user.
SignaturePrompt	String	No	Text displayed to obtain a signature from the user.
PlayOptions	String	No	Flag options for playing the contents of the FileShare link.
DisplayUserName	Boolean	No	Flag indicating to display the user's name.
DisplayEmail	Boolean	No	Flag indicating to display the user's email address.
DisplayComments	Boolean	No	Flag indicating to display the comments
AddEmailInWaterMark	Boolean	No	Flag indicating to include recipient's email address into watermark text.
AddFileShareIDInWaterMark	Boolean	No	Flag indicating to include FileShare ID into watermark text.
AddIPInWaterMark	Boolean	No	Flag indicating to include IP address into watermark text
AddTimeStampInWaterMark	Boolean	No	Flag indicating to include downloaded item's date time stamp into watermark text
WaterMarkSize	Integer	No	Size of watermark text in pixels, possible values are {8, 10, 12, 14, 16, 18, 20, 22, 24}. Default value will be 8.
WaterMarkColor	String	No	Color of watermark text. Possible values are RD for Red, GN for Green, GY for Grey, WE for White, BE for Blue, BK for Black
PasswordSecurity	Int	No	Password security value is 0 means

			unprotected file share link will be created, 1 means file share link will be protected by password, 2 means file share link will be protected by OTP email, 3 means file share link will be protected by OTP SMS.
--	--	--	---

Response

SendFileShareLinkResult

SendFileShareLinkResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the SendFileShareLink method was executed. If the SendFileShareLink call was successful, the ErrorMessage is blank.
ELinkSent	Boolean	Flag indicating the link was sent to the recipients.
ELinkURLs.URL	String	URL created for each recipients.
ELinkURLs.FileShareID	Integer	FileShare ID created for each recipients

AdminCreateFolder

This function allows the Private Site Administrator to Create a new folder into user account.

*NOTE: This function is only available for **Private Sites** (not for individual or workgroup FilesAnywhere accounts). Please contact FilesAnywhere Support if you are interested in the Private Site offering and wish to use this function to maintain users.*

Syntax

AdminCreateFolderResult = AdminCreateFolder(AuthParameters, AdminCreateFolderParameters)

Usage

Private Site Administrator can use the AdminCreateFolder method to create new folder into User account.

The following parameters are needed to make the AdminCreateFolder call:

- **AuthParameters:** It consist of below sub values to authenticate the Admin account:

Parameter Name	Description
APIKey	A Developer API Key. See the “Getting Started” section above, to find out how to obtain a Developer API Key
OrgID	The OrgID is the Private Site Client ID for a FilesAnywhere Private Site.
AdminUserID	Private Site Administrator ID
AdminPassword	Private Site Administrator password.

- **AdminCreateFolderParameters;** It consist of below sub parameters which are needed to create folder into user account:

Parameter Name	Description
Path	This is the FilesAnywhere Path of the folder inside which the new folder will be created. The Path should be formatted like <code>\UserVolume\FolderName\</code> , where <code>UserVolume</code> is the FilesAnywhere Username in whose account the folder will be created and <code>FolderName</code> is the folder or subfolder where the new folder will be created. Path should be end with “\”
FolderName	Name of the new folder

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/AdminCreateFolder"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <AdminCreateFolder xmlns="http://api.filesanywhere.com/">
      <AdminAuthParameters>
        <APIKey>string</APIKey>
        <OrgID>long</OrgID>
        <AdminUserID>string</AdminUserID>
        <AdminPassword>string</AdminPassword>
      </AdminAuthParameters>
      <Parameters>
        <Path>string</Path>
        <FolderName>string</FolderName>
      </Parameters>
    </AdminCreateFolder>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
```

```

    <AdminCreateFolderResponse xmlns="http://api.filesanywhere.com/">
      <AdminCreateFolderResult>
        <FolderName>string</FolderName>
      </AdminCreateFolderResult>
    </AdminCreateFolderResponse>
  </soap:Body>
</soap:Envelope>

```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The [placeholders](#) shown need to be replaced with actual values.

```

POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <AdminCreateFolder xmlns="http://api.filesanywhere.com/">
      <AdminAuthParameters>
        <APIKey>string</APIKey>
        <OrgID>long</OrgID>
        <AdminUserID>string</AdminUserID>
        <AdminPassword>string</AdminPassword>
      </AdminAuthParameters>
      <Parameters>
        <Path>string</Path>
        <FolderName>string</FolderName>
      </Parameters>
    </AdminCreateFolder>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <AdminCreateFolderResponse xmlns="http://api.filesanywhere.com/">
      <AdminCreateFolderResult>
        <FolderName>string</FolderName>
      </AdminCreateFolderResult>
    </AdminCreateFolderResponse>
  </soap12:Body>
</soap12:Envelope>

```

Sample C#.Net Code

```
private void CreateFolder()
{
    FAWAPI.AdminAuthParameters objAuthParameters = new
FAWAPI.AdminAuthParameters();
    objAuthParameters.OrgID = Convert.ToInt64(txtOrgID.Text);
    objAuthParameters.APIKey = Program.API_KEY;
    objAuthParameters.AdminUserID = txtAdminUserName.Text;
    objAuthParameters.AdminPassword = txtPassword.Text;

    FAWAPI.AdminCreateFolderParameter objAdminCreateFolder = new
FAWAPI.AdminCreateFolderParameter();
    objAdminCreateFolder.Path = txtFolderPath.Text;
    objAdminCreateFolder.FolderName = txtNewFolderName.Text;

    wndOneLoginClient.FAWAPI.FAWAPIv2SoapClient objFAWAPI = new
FAWAPI.FAWAPIv2SoapClient();
    FAWAPI.AdminCreateFolderResult objAdminCreateFolderResult =
objFAWAPI.AdminCreateFolder(objAuthParameters, objAdminCreateFolder);
    if (!String.IsNullOrEmpty(objAdminCreateFolderResult.ErrorMessage))
    {
        txtErrorMsg.Text = objAdminCreateFolderResult.ErrorMessage;
        return;
    }
    txtErrorMsg.Text = String.Format("Folder {0} created successfully",
objAdminCreateFolderResult.FolderName);
}
```

Arguments

- **AuthParameters:** It consist of below sub values to authenticate the Admin account:

Name	Data Type	Required	Description
APIKey	String	Yes	A Developer API Key. See the “Getting Started” section above, to find out how to obtain a Developer API Key
OrgID	Long	Yes	The OrgID is the Private Site Client ID for a FilesAnywhere Private Site.
AdminUserID	String	Yes	Private Site Administrator ID
AdminPassword	String	Yes	Private Site Administrator password.

- **AdminCreateFolderParameters;** It consist of below sub parameters which are needed to create folder into user account:

Name	Data Type	Required	Description
Path	String	Yes	This is the FilesAnywhere Path of the folder inside which the new folder will be created. The Path should be formatted like <code>\UserVolume\FolderName\</code> , where <code>UserVolume</code> is the FilesAnywhere <code>UserName</code> in whose account the folder will be created and <code>FolderName</code> is the folder or subfolder where the new folder will be created. Path should be end with “\”
FolderName	String	Yes	Name of the new folder

Response

AdminCreateFolderResult

AdminCreateFolderResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the creating new folder. If the AdminCreateFolder call was successful, the ErrorMessage will be blank or null.
FolderName	Boolean	Returns newly created folder name, if it successfully created..

AdminCreateGroupShare

This function allows the Private Site Administrator to Create or update a Groupshare i.e. Share a Folder of one user account to other user account.

*NOTE: This function is only available for **Private Sites** (not for individual or workgroup FilesAnywhere accounts). Please contact FilesAnywhere Support if you are interested in the Private Site offering and wish to use this function to maintain users.*

Syntax

AdminCreateGroupShareResult = AdminCreateGroupShare (AuthParameters, AdminGroupShare)

Usage

Private Site Administrator can use the AdminCreateGroupShare method to create new GroupShare. Using this Admin can share a folder of one User account to other User account with different permissions.

The following parameters are needed to make the AdminCreateGroupShare call:

- **AuthParameters:** It consist of below sub values to authenticate the Admin account:

Parameter Name	Description
APIKey	A Developer API Key. See the “Getting Started” section above, to find out how to obtain a Developer API Key
OrgID	The OrgID is the Private Site Client ID for a FilesAnywhere

	Private Site.
AdminUserID	Private Site Administrator ID
AdminPassword	Private Site Administrator password.

Parameters; It consist of below sub parameters which are needed to create Groupshare:

Parameter Name	Description
FolderPath	<p>the user's personal folder path that the admin wants to share to another user specified in the SharedToUserName field.</p> <p>The FolderPath should start with “\” and should end with “\”. For Example it should be “\USER1\xyz\1234\”.</p>
Permission	<p>the permission flag set for the SharedFolder. Following are the access levels that can be set for the shared folder:</p> <p>F - Full Permission. The user can upload, delete and download files.</p> <p>A - Master access permission. Receipt user can do all as to Full permission, as well as he can share this folder to other users also.</p> <p>R - Read-only Permission. The user can only view and download files from the shared folder.</p> <p>C - Create and Update Permissions.</p> <p>P - Preview only, User can Preview the file from Shared folder, User cannot upload/download the File into Shared folder.</p> <p>U - Upload only, User can only upload the files into Shared folder.</p>
SharedByUserName	FilesAnywhere account username who wants to share his folder to other user
SharedToUserName	FilesAnywhere Account UserName with whom the admin wants to share personal folder of SharedByUserName
ExpirationDate	Represent expiry date of Groupshare. If no date supplied then GroupShare will never expire else after the specified

date GroupShare will not be accessible to recipient user

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/AdminCreateGroupShare"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <AdminCreateGroupShare xmlns="http://api.filesanywhere.com/">
      <AdminAuthParameters>
        <APIKey>string</APIKey>
        <OrgID>long</OrgID>
        <AdminUserID>string</AdminUserID>
        <AdminPassword>string</AdminPassword>
      </AdminAuthParameters>
      <Parameters>
        <FolderPath>string</FolderPath>
        <Permission>string</Permission>
        <SharedByUsername>string</SharedByUsername>
        <SharedToUsername>string</SharedToUsername>
        <ExpirationDate>dateTime</ExpirationDate>
      </Parameters>
    </AdminCreateGroupShare>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <AdminCreateGroupShareResponse xmlns="http://api.filesanywhere.com/">
      <AdminCreateGroupShareResult>
        <FolderPath>string</FolderPath>
      </AdminCreateGroupShareResult>
    </AdminCreateGroupShareResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <AdminCreateGroupShare xmlns="http://api.filesanywhere.com/">
      <AdminAuthParameters>
        <APIKey>string</APIKey>
        <OrgID>long</OrgID>
        <AdminUserID>string</AdminUserID>
        <AdminPassword>string</AdminPassword>
      </AdminAuthParameters>
      <Parameters>
        <FolderPath>string</FolderPath>
        <Permission>string</Permission>
        <SharedByUsername>string</SharedByUsername>
        <SharedToUsername>string</SharedToUsername>
        <ExpirationDate>dateTime</ExpirationDate>
      </Parameters>
    </AdminCreateGroupShare>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <AdminCreateGroupShareResponse xmlns="http://api.filesanywhere.com/">
      <AdminCreateGroupShareResult>
        <FolderPath>string</FolderPath>
      </AdminCreateGroupShareResult>
    </AdminCreateGroupShareResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample C#.Net Code

```
private void CreateGroupShare()
{
    FAWAPI.AdminAuthParameters objAuthParameters = new
FAWAPI.AdminAuthParameters();
    objAuthParameters.OrgID = Convert.ToInt64(txtOrgID.Text);
    objAuthParameters.APIKey = Program.API_KEY;
    objAuthParameters.AdminUserID = txtAdminUserName.Text;
    objAuthParameters.AdminPassword = txtPassword.Text;

    FAWAPI.AdminGroupShareParameter objAdminGroupShare = new
FAWAPI.AdminGroupShareParameter();
    objAdminGroupShare.SharedToUsername = txtSharedToUserName.Text;
    objAdminGroupShare.SharedByUsername = txtSharedByUsername.Text;
    objAdminGroupShare.FolderPath = txtFolderToShare.Text;
    objAdminGroupShare.Permission = cmbPermission.SelectedValue.ToString();
    if (dtExpirationDate.Checked)
        objAdminGroupShare.ExpirationDate = dtExpirationDate.Value;

    wndOneLoginClient.FAWAPI.FAWAPIv2SoapClient objFAWAPI = new
FAWAPI.FAWAPIv2SoapClient();
    FAWAPI.AdminCreateGroupShareResult objAdminCreateGroupShareResult =
objFAWAPI.AdminCreateGroupShare (objAuthParameters, objAdminGroupShare);
    if (!String.IsNullOrEmpty(objAdminCreateGroupShareResult.ErrorMessage))
    {
        txtErrorMsg.Text = objAdminCreateGroupShareResult.ErrorMessage;
        return;
    }
    txtErrorMsg.Text = String.Format("Folder {0} GroupShared successfully",
objAdminCreateGroupShareResult.FolderPath);
}
```

Arguments

- **AuthParameters:** It consist of below sub values to authenticate the Admin account:

Name	DataType	Required	Description
APIKey	String	Yes	A Developer API Key. See the “Getting Started” section above, to find out how to obtain a Developer API Key
OrgID	Long	Yes	The OrgID is the Private Site Client ID for a FilesAnywhere Private Site.
AdminUserID	String	Yes	Private Site Administrator ID
AdminPassword	String	Yes	Private Site Administrator password.

- **Parameters;** It consist of below sub parameters which are needed to create GroupShare:

Name	DataType	Required	Description						
FolderPath	String	Yes	<p>the user’s personal folder path that the admin wants to share to another user specified in the SharedToUserName field.</p> <p>The FolderPath should start with “\” and should end with “\”. For Example it should be “\USER1\xyz\1234\”.</p>						
Permission	String	Yes	<p>the permission flag set for the SharedFolder. Following are the access levels that can be set for the shared folder:</p> <table border="1" data-bbox="938 1562 1464 1879"> <thead> <tr> <th>Access level</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>F</td> <td>Full Permission. The user can upload, delete and download files.</td> </tr> <tr> <td>A</td> <td>Master access permission. Receipt</td> </tr> </tbody> </table>	Access level	Description	F	Full Permission. The user can upload, delete and download files.	A	Master access permission. Receipt
Access level	Description								
F	Full Permission. The user can upload, delete and download files.								
A	Master access permission. Receipt								

			<p>user can do all as to Full permission, as well as he can share this folder to other users also.</p>
			<p>R Read-only Permission. The user can only view and download files from the shared folder.</p>
			<p>C Create and Update Permissions. The user can add/modify/download, but cannot delete or move data.</p>
			<p>P Preview only, User can Preview the file from Shared folder, User cannot upload/download the File into Shared folder.</p>
			<p>U Upload only, User can only upload the files into Shared folder.</p>
SharedByUsername	String	Yes	FilesAnywhere account username who wants to share his folder to other user
SharedToUserName	String	Yes	FilesAnywhere Account UserName with whom the admin wants to share personal folder of SharedByUsername
ExpirationDate	DateTime	No	Represent expiry date of Groupshare. If no date supplied then GroupShare will never expire else after the specified date GroupShare will not be accessible to recipient user

Response

AdminCreateGroupShareResult

AdminCreateGroupShareResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while the creating GroupShare. If the AdminCreateGropuShare call was successful, the ErrorMessage will be blank or null.
FolderName	Boolean	Returns newly created folder Path, if it successfully shared to specified user.

AdminUserAdd

This function allows the Private Site Administrator add a new user to the Private Site.

*NOTE: This function is only available for **Private Sites** (not for individual or workgroup FilesAnywhere accounts). Please contact FilesAnywhere Support if you are interested in the Private Site offering and wish to use this function to maintain users.*

Syntax

AdminUserAddResult = AdminUserAdd (AuthParameters, Parameters)

Usage

Private Site Administrator can use the AdminUserAdd method to add new user (SSO/ Non SSO), set GroupShare folders (map to a folder in a different account), define advanced access options for that user to access the folder through SFTP or WebDAV and enable remote backup feature.

The following parameters are needed to make the AdminUserAdd call:

- **AuthParameters:** It consist of below sub values to authenticate the Admin account:

Parameter Name	Description

APIKey	A Developer API Key. See the “Getting Started” section above, to find out how to obtain a Developer API Key
OrgID	The OrgID is the Private Site Client ID for a FilesAnywhere Private Site.
AdminUserID	Private Site Administrator ID
AdminPassword	Private Site Administrator password.

Parameters; It consist of below sub parameters which are needed to create new User:

Parameter Name	Description
UserDetail	Consist of various properties of user e.g. UserName/ Password/ Division/ Region/ Personalstorage etc...
PersonallInformaiton	Consist of user personal details e.g. Firstname/Lastname/ Address/ City etc...
AdvancedUserDetail	Consist of user advanced properties e.g. advanced access FTP/SFTP/WEBDAV etc...
SSO	Consist of SSO related information
ShareFolder	Consist of user Shared folder details

Each of the above parameter consist of sub properties for various inputs to create new user.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/AdminUserAdd"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <AdminUserAdd xmlns="http://api.filesanywhere.com/">
      <AdminAuthParameters>
```

```

    <APIKey>string</APIKey>
    <OrgID>long</OrgID>
    <AdminUserID>string</AdminUserID>
    <AdminPassword>string</AdminPassword>
  </AdminAuthParameters>
  <Parameters>
    <PersonalInformation>
      <FirstName>string</FirstName>
      <LastName>string</LastName>
      <Email>string</Email>
      <CompanyName>string</CompanyName>
      <JobTitle>string</JobTitle>
      <Phone>string</Phone>
      <CellPhone>string</CellPhone>
      <Address>string</Address>
      <City>string</City>
      <State>string</State>
      <ZipCode>string</ZipCode>
      <Country>string</Country>
      <Notes>string</Notes>
    </PersonalInformation>
    <SSO>
      <SSOAuthenticationType>NonSSO or SSOOnly or
MixedSSO</SSOAuthenticationType>
      <UserPrincipalName>string</UserPrincipalName>
    </SSO>
    <UserDetail>
      <UserName>string</UserName>
      <Password>string</Password>
      <DoNotEmailDetailsToUser>boolean</DoNotEmailDetailsToUser>
      <DivisionCode>string</DivisionCode>
      <RegionName>string</RegionName>
      <PersonalStorage>boolean</PersonalStorage>
      <StorageSize>int</StorageSize>
    </UserDetail>
    <ShareFolder>
      <FolderPath>string</FolderPath>
      <Permission>string</Permission>
      <SharedByUsername>string</SharedByUsername>
      <SharedToUsername>string</SharedToUsername>
      <ExpirationDate>dateTime</ExpirationDate>
    </ShareFolder>
    <AdvancedUserDetail>
      <LockUserProfile>boolean</LockUserProfile>
      <LogAllDownloadActivity>boolean</LogAllDownloadActivity>
      <PurgeFileDays>int</PurgeFileDays>
      <EnableFTP>boolean</EnableFTP>
      <EnableSFTP>boolean</EnableSFTP>
      <EnableWEBDAV>boolean</EnableWEBDAV>
      <FTPHomeFolderPath>string</FTPHomeFolderPath>
      <SFTPHomeFolderPath>string</SFTPHomeFolderPath>
      <WEBDAVHomeFolderPath>string</WEBDAVHomeFolderPath>
    </AdvancedUserDetail>
  </Parameters>

```

```

    </AdminUserAdd>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <AdminUserAddResponse xmlns="http://api.filesanywhere.com/">
      <AdminUserAddResult>
        <UserName>string</UserName>
      </AdminUserAddResult>
    </AdminUserAddResponse>
  </soap:Body>
</soap:Envelope>

```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The [placeholders](#) shown need to be replaced with actual values.

```

POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <AdminUserAdd xmlns="http://api.filesanywhere.com/">
      <AdminAuthParameters>
        <APIKey>string</APIKey>
        <OrgID>long</OrgID>
        <AdminUserID>string</AdminUserID>
        <AdminPassword>string</AdminPassword>
      </AdminAuthParameters>
      <Parameters>
        <PersonalInformation>
          <FirstName>string</FirstName>
          <LastName>string</LastName>
          <Email>string</Email>
          <CompanyName>string</CompanyName>
          <JobTitle>string</JobTitle>
          <Phone>string</Phone>
          <CellPhone>string</CellPhone>
          <Address>string</Address>
          <City>string</City>
          <State>string</State>
          <ZipCode>string</ZipCode>

```

```

        <Country>string</Country>
        <Notes>string</Notes>
    </PersonalInformation>
    <SSO>
        <SSOAuthenticationType>NonSSO or SSOOnly or
MixedSSO</SSOAuthenticationType>
        <UserPrincipalName>string</UserPrincipalName>
    </SSO>
    <UserDetail>
        <UserName>string</UserName>
        <Password>string</Password>
        <DoNotEmailDetailsToUser>boolean</DoNotEmailDetailsToUser>
        <DivisionCode>string</DivisionCode>
        <RegionName>string</RegionName>
        <PersonalStorage>boolean</PersonalStorage>
        <StorageSize>int</StorageSize>
    </UserDetail>
    <ShareFolder>
        <FolderPath>string</FolderPath>
        <Permission>string</Permission>
        <SharedByUsername>string</SharedByUsername>
        <SharedToUsername>string</SharedToUsername>
        <ExpirationDate>dateTime</ExpirationDate>
    </ShareFolder>
    <AdvancedUserDetail>
        <LockUserProfile>boolean</LockUserProfile>
        <LogAllDownloadActivity>boolean</LogAllDownloadActivity>
        <PurgeFileDays>int</PurgeFileDays>
        <EnableFTP>boolean</EnableFTP>
        <EnableSFTP>boolean</EnableSFTP>
        <EnableWEBDAV>boolean</EnableWEBDAV>
        <FTPHomeFolderPath>string</FTPHomeFolderPath>
        <SFTPHomeFolderPath>string</SFTPHomeFolderPath>
        <WEBDAVHomeFolderPath>string</WEBDAVHomeFolderPath>
    </AdvancedUserDetail>
    </Parameters>
</AdminUserAdd>
</soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <AdminUserAddResponse xmlns="http://api.filesanywhere.com/">
      <AdminUserAddResult>
        <UserName>string</UserName>
      </AdminUserAddResult>
    </AdminUserAddResponse>
  </soap12:Body>

```

```
</soap12:Envelope>
```

Sample C#.Net Code

```
private void AddUser ()
{
    FAWAPI.AdminAuthParameters objAuthParameters = new
FAWAPI.AdminAuthParameters();
    objAuthParameters.OrgID = Convert.ToInt64(txtOrgID.Text);
    objAuthParameters.APIKey = Program.API_KEY;
    objAuthParameters.AdminUserID = txtAdminUserName.Text;
    objAuthParameters.AdminPassword = txtPassword.Text;

    FAWAPI.AdminUserAddParameter objAdminUserAdd = new
FAWAPI.AdminUserAddParameter();

    FAWAPI.CUserDetail objUserDetail = new FAWAPI.CUserDetail();
    objUserDetail.UserName = txtUserName.Text;
    objUserDetail.Password = txtUserPassword.Text;

    objUserDetail.DivisionCode = txtDivision.Text;
    objUserDetail.RegionName = txtRegion.Text;

    objUserDetail.PersonalStorage = chkPersonalStorage.Checked;
    objUserDetail.StorageSize = Convert.ToInt32(lstStorageGB.SelectedItem);
    objAdminUserAdd.UserDetail = objUserDetail;

    if (chkShareFolder.Checked)
    {
        FAWAPI.AdminGroupShareParameter objSharedFolder = new
FAWAPI.AdminGroupShareParameter();
        objSharedFolder.SharedByUsername = txtUserAddShareByUsername.Text;
        objSharedFolder.FolderPath = txtUserAddShareFolderPath.Text;
        objSharedFolder.Permission =
cmbUserAddSharePermission.SelectedValue.ToString();
        if (dtUserAddExpirationDate.Checked)
            objSharedFolder.ExpirationDate = dtUserAddExpirationDate.Value;
        objAdminUserAdd.ShareFolder = objSharedFolder;
    }

    FAWAPI.CPersonalInformation objPersonalDetail = new
FAWAPI.CPersonalInformation();
    objPersonalDetail.FirstName = txtFirstName.Text;
    objPersonalDetail.LastName = txtLastName.Text;
    objPersonalDetail.Email = txtEmail.Text;
    objPersonalDetail.Phone = txtPhone.Text;
    objPersonalDetail.CellPhone = txtCellPhone.Text;
    objPersonalDetail.Notes = txtNotes.Text;
    objPersonalDetail.CompanyName = txtCompanyName.Text;
    objPersonalDetail.JobTitle = txtJobTitle.Text;
    objPersonalDetail.Address = txtAddress.Text;
    objPersonalDetail.City = txtCity.Text;
    objPersonalDetail.State = txtState.Text;
    objPersonalDetail.ZipCode = txtZip.Text;
    objPersonalDetail.Country = txtCountry.Text;
    objAdminUserAdd.PersonalInformation = objPersonalDetail;
}
```



```

        FAWAPI.CAdvanceUserDetail objAdvancedUserDetail = new
FAWAPI.CAdvanceUserDetail();
        objAdvancedUserDetail.LockUserProfile = chkLockUserProfile.Checked;
        objAdvancedUserDetail.PurgeFileDays =
String.IsNullOrEmpty(txtPurgeFileDays.Text) ? 0 :
Convert.ToInt16(txtPurgeFileDays.Text);
        objAdvancedUserDetail.LogAllDownloadActivity = chkLogAllDownload.Checked;
        objAdvancedUserDetail.EnableFTP = chkFTP.Checked;
        objAdvancedUserDetail.EnableSFTP = chkSFTP.Checked;
        objAdvancedUserDetail.EnableWEBDAV = chkWEBDAV.Checked;
        objAdvancedUserDetail.FTPHomeFolderPath = txtFTP.Text;
        objAdvancedUserDetail.SFTPHomeFolderPath = txtSFTP.Text;
        objAdvancedUserDetail.WEBDAVHomeFolderPath = txtWEBDAV.Text;
        objAdminUserAdd.AdvancedUserDetail = objAdvancedUserDetail;

        if (cmbSSOType.SelectedValue.ToString() != "0")
        {
            FAWAPI.CSSO objSSO = new FAWAPI.CSSO();
            switch (cmbSSOType.SelectedValue.ToString())
            {
                case "1":
                    objSSO.SSOAuthenticationType = FAWAPI.SSOAuthType.SSOOnly;
                    break;
                case "2":
                    objSSO.SSOAuthenticationType = FAWAPI.SSOAuthType.MixedSSO;
                    break;
            }
            objSSO.UserPrincipalName = txtUPN.Text;
            objAdminUserAdd.SSO = objSSO;
        }

        wndOneLoginClient.FAWAPI.FAWAPIv2SoapClient objFAWAPI = new
FAWAPI.FAWAPIv2SoapClient();
        FAWAPI.AdminUserAddResult objAdminUserAddResult =
objFAWAPI.AdminUserAdd(objAuthParameters, objAdminUserAdd);

        if (!String.IsNullOrEmpty(objAdminUserAddResult.ErrorMessage))
        {
            txtErrorMsg.Text = objAdminUserAddResult.ErrorMessage;
            return;
        }
        txtErrorMsg.Text = String.Format("User {0} Added successfully",
objAdminUserAddResult.UserName);
    }

```

Arguments

- **AuthParameters:** It consist of below sub values to authenticate the Admin account:

Name	DataType	Required	Description
APIKey	String	Yes	A Developer API Key. See the “Getting Started” section above, to find out how to obtain a Developer API Key
OrgID	Long	Yes	The OrgID is the Private Site Client ID for a FilesAnywhere Private Site.
AdminUserID	String	Yes	Private Site Administrator ID
AdminPassword	String	Yes	Private Site Administrator password.

- **Parameters;** It consist of below sub parameters which are needed to create GroupShare:

Name	DataType	Required	Description
UserDetail	Class	Yes	Consist of various properties of user e.g. UserName/ Password/ Division/ Region/ Personalstorage etc...
Personallnformaiton	Class	Yes	Consist of user personal details e.g. Firstname/Lastname/ Address/ City etc...
AdvancedUserDetail	Class	No	Consist of user advanced properties e.g. advanced access FTP/SFTP/WEBDAV etc...
SSO	Class	No	Consist of SSO related information
ShareFolder	Class	No	Consist of user Shared folder details

Each of the above class type object consist of various sub attributes discussed as below:

Name	DataType	Required	Description
UserDetail			
UserName	String	Yes	Unique alphanumeric ID that the user is required to enter to login into his/her account.
Password	String	No	Password the user is required to enter to access its account. If the password field leave blank, then user will get reset password link in welcome email else the input password will apply to user. When user is pure SSO then supplied password will not be applicable.
DoNotEmailDetailsToUser	Boolean	Yes	User will receive welcome email or not.
DivisionCode	String	Yes	Division to which the new user is added. It is the code that was assigned to a Division in the Organization.
RegionName	String	Yes	Region which decides storage area to which user file/folder will be placed. When account is not on different regions then default value needs to be passed as "Default Region"
PersonalStorage	Boolean	Yes	{true/false} Personal storage for user will be enabled or not. If this flag is set as "true" then personal storage space is available to the user. If the flag is set as "false" then personal storage space not assigned to the user. If the user is

			assigned the Group Share folder the user will see only the GroupShare folders.
StorageSize	Int	Yes	Storage Size for this user in gigabytes (GB). The storage size needs to be at least 1 GB, or you can choose "Disable Personal Space". StorageSize sets the storage capacity for the user in its own personal storage account.
PersonallInformation			
FirstName	String	Yes	First name of the new user
LastName	String	Yes	Last name of the new user
Email	String	Yes	New user's email address. All the FilesAnywhere emails including the user password and its other account details will be sent to this email address.
CompanyName	String	No	
JobTitle	String	No	
Phone	String	No	Phone is the new user's phone number.
CellPhone	String	No	CellPhone is the new user's cell phone number.
Address	String	No	UserAddress is the new user's residence/office address.
City	String	No	
State	String	No	
ZipCode	String	No	
Country	String	No	
Notes	String	No	Note is the memo that the administrator wants to attach to the new user for future reference.

ShareFolder															
FolderPath	String	Yes	<p>the user's personal folder path that the admin wants to share to another user specified in the SharedToUserName field.</p> <p>The FolderPath should start with “\” and should end with “\”. For Example it should be “\USER1\xyz\1234\”.</p>												
Permission	String	Yes	<p>the permission flag set for the SharedFolder. Following are the access levels that can be set for the shared folder:</p> <table border="1"> <thead> <tr> <th>Access level</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>F</td> <td>Full Permission. The user can upload, delete and download files.</td> </tr> <tr> <td>A</td> <td>Master access permission. Recipient user can do all as to Full permission, as well as he can share this folder to other users also.</td> </tr> <tr> <td>R</td> <td>Read-only Permission. The user can only view and download files from the shared folder.</td> </tr> <tr> <td>C</td> <td>Create and Update Permissions.</td> </tr> <tr> <td>P</td> <td>Preview only, User can Preview the file from Shared folder, User cannot</td> </tr> </tbody> </table>	Access level	Description	F	Full Permission. The user can upload, delete and download files.	A	Master access permission. Recipient user can do all as to Full permission, as well as he can share this folder to other users also.	R	Read-only Permission. The user can only view and download files from the shared folder.	C	Create and Update Permissions.	P	Preview only, User can Preview the file from Shared folder, User cannot
Access level	Description														
F	Full Permission. The user can upload, delete and download files.														
A	Master access permission. Recipient user can do all as to Full permission, as well as he can share this folder to other users also.														
R	Read-only Permission. The user can only view and download files from the shared folder.														
C	Create and Update Permissions.														
P	Preview only, User can Preview the file from Shared folder, User cannot														

			<table border="1"> <tr> <td></td> <td>upload/download the File into Shared folder.</td> </tr> <tr> <td>U</td> <td>Upload only, User can only upload the files into Shared folder.</td> </tr> </table>		upload/download the File into Shared folder.	U	Upload only, User can only upload the files into Shared folder.
	upload/download the File into Shared folder.						
U	Upload only, User can only upload the files into Shared folder.						
SharedByUsername	String	Yes	FilesAnywhere account username who wants to share his folder to new user				
SharedToUserName	String	Yes	With AdminUserAdd api this parameter does not need to supply, as API will treat this parameter value as new Username.				
ExpirationDate	DateTime	No	Represent expiry date of Groupshare. If no date supplied then GroupShare will never expire else after the specified date GroupShare will not be accessible to recipient user				
AdvanceUserDetail							
LockUserProfile	Boolean	Yes	LockUserProfile is the flag that can be set as true/false. If this flag is set as "true" then the user cannot change its password and cannot update its profile and default settings.				
LogAllDownloadActivity	Boolean	Yes	{true/false} If the value is set as true then all download Activity will be recorded else not.				
PurgeFileDays	Int	No	Automatically purges files older than the specified number of days from the user accounts. If the purge day is set to 0 then the files are not purged from the user accounts.				

EnableFTP	Boolean	Yes	{true/false} If this flag is set as "true" the user can access personal folder and/or shared folder through FTP.	
EnableSFTP	Boolean	Yes	{true/false} If this flag is set as "true" the user can access personal folder and/or shared folder through SFTP.	
EnableWEBDAV	Boolean	Yes	{true/false} If this flag is set as "true" the user can access personal folder and/or shared folder through WebDAV	
FTPHomeFolderPath	String	Yes	If EnableFTP flag is set to true then FolderPath will be set as FTP root	
SFTPHomeFolderPath	String	Yes	If EnableSFTP flag is set to true then FolderPath will be set as SFTP root	
WEBDAVHomeFolderPath	String	Yes	If EnableWEBDAV flag is set to true then FolderPath will be set as WEBDAV root	
SSO				
SSOAuthType	Enum	No	Value	Description
			0-Non SSO	If value is set as 0 then user will be created as Non SSO user
			1-Pure SSO	If value is set as 1 then user will be created as SSO only.
			2-Mixed SSO	If value is set as 2 then user will be created as Mixed SSO

UserPrincipalName	String	No	<p>When SSO user is created then Unique identity of Identity provider can be supplied here.</p> <p>When the UserPrincipalName field remains blank then email address will be used to authenticate the user with SSO parameter.</p>

Response

AdminUserAddResult

AdminUserAddResult Properties

Name	Data Type	Description
ErrorMessage	String	<p>A description of the error, if any error was encountered while the creating GroupShare.</p> <p>If the AdminUseradd call was successful, the ErrorMessage will be blank or null.</p>
UserName	String	Returns newly created Username, if user successfully created.

UpdateFolderProperties

Syntax

UpdateFolderPropertiesResult = UpdateFolderProperties (Token, Parameters)

Usage

Sets folder properties such as file version history and check-in/check-out and enable the notifications on various operations on folder.

The following parameters are needed to make the UpdateFolderProperties call:

Parameter Name	Description
Token	The Token ID returned by the last AccountLogin call
Parameters	It contains various attributes to apply on folder as property. All attributes are declared below.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/UpdateFolderProperties"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UpdateFolderProperties xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Parameters>
        <FolderPath>string</FolderPath>
        <Comments>string</Comments>
        <VCFlagFileVersion>boolean</VCFlagFileVersion>
        <VCFlagFileCheckinCheckOut>boolean</VCFlagFileCheckinCheckOut>
        <VCFlagDisableDelete>boolean</VCFlagDisableDelete>
        <VCFlagDailyEmail>boolean</VCFlagDailyEmail>

        <VCFlagAlwaysSendEmailAllUsers>boolean</VCFlagAlwaysSendEmailAllUsers>

        <VCFlagUploadCheckinNotification>boolean</VCFlagUploadCheckinNotification>
        <VCFlagCheckOutNotification>boolean</VCFlagCheckOutNotification>
        <VCFlagOwnerMail>boolean</VCFlagOwnerMail>
        <VCEmailTo>string</VCEmailTo>
        <VCFlagGroupShareMail>boolean</VCFlagGroupShareMail>
        <RemoveChildProperties>boolean</RemoveChildProperties>
        <VersionKeepCount>int</VersionKeepCount>
        <SaveFlag>DetachFromParent or InheritFromParent or
SaveProperties</SaveFlag>
        <FolderPropertiesNotifications>
          <Comment>boolean</Comment>
          <Delete>boolean</Delete>
          <Move>boolean</Move>
```

```

    <Copy>boolean</Copy>
    <Rename>boolean</Rename>
    <Create>boolean</Create>
  </FolderPropertiesNotifications>
</Parameters>
</UpdateFolderProperties>
</soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UpdateFolderPropertiesResponse xmlns="http://api.filesanywhere.com/">
      <UpdateFolderPropertiesResult>
        <Updated>boolean</Updated>
      </UpdateFolderPropertiesResult>
    </UpdateFolderPropertiesResponse>
  </soap:Body>
</soap:Envelope>

```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The [placeholders](#) shown need to be replaced with actual values.

```

POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <UpdateFolderProperties xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Parameters>
        <FolderPath>string</FolderPath>
        <Comments>string</Comments>
        <VCFlagFileVersion>boolean</VCFlagFileVersion>
        <VCFlagFileCheckinCheckOut>boolean</VCFlagFileCheckinCheckOut>
        <VCFlagDisableDelete>boolean</VCFlagDisableDelete>
        <VCFlagDailyEmail>boolean</VCFlagDailyEmail>

        <VCFlagAlwaysSendEmailAllUsers>boolean</VCFlagAlwaysSendEmailAllUsers>

        <VCFlagUploadCheckinNotification>boolean</VCFlagUploadCheckinNotification>
        <VCFlagCheckOutNotification>boolean</VCFlagCheckOutNotification>
        <VCFlagOwnerMail>boolean</VCFlagOwnerMail>
      </Parameters>
    </UpdateFolderProperties>
  </soap12:Body>
</soap12:Envelope>

```

```
<VCEmailTo>string</VCEmailTo>
<VCFlagGroupShareMail>boolean</VCFlagGroupShareMail>
<RemoveChildProperties>boolean</RemoveChildProperties>
<VersionKeepCount>int</VersionKeepCount>
<SaveFlag>DetachFromParent or InheritFromParent or
SaveProperties</SaveFlag>
  <FolderPropertiesNotifications>
    <Comment>boolean</Comment>
    <Delete>boolean</Delete>
    <Move>boolean</Move>
    <Copy>boolean</Copy>
    <Rename>boolean</Rename>
    <Create>boolean</Create>
  </FolderPropertiesNotifications>
</Parameters>
</UpdateFolderProperties>
</soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <UpdateFolderPropertiesResponse xmlns="http://api.filesanywhere.com/">
      <UpdateFolderPropertiesResult>
        <Updated>boolean</Updated>
      </UpdateFolderPropertiesResult>
    </UpdateFolderPropertiesResponse>
  </soap12:Body>
</soap12:Envelope>
```

Sample C#.Net Code

```
void UpdateFolderProperties()
{
    FAWAPI.FAWAPIv2SoapClient objFAWAPI = new FAWAPI.FAWAPIv2SoapClient();
    FAWAPI.UpdateFolderPropertiesParameter objParameters = new
FAWAPI.UpdateFolderPropertiesParameter();
    objParameters.FolderPath = txtFolderPath.Text;
    objParameters.RemoveChildProperties = chkChildProperties.Checked;
    objParameters.VCFlagFileCheckinCheckout = chkCheckinCheckout.Checked;
    objParameters.VCFlagFileVersion = chkFileVersion.Checked;
    objParameters.VersionKeepCount =
Convert.ToInt16(cmbVersion.SelectedItem.ToString());
    objParameters.VCFlagDisableDelete = chkDisableDelete.Checked;

    objParameters.VCFlagDailyEmail = chkDailyReport.Checked;
    objParameters.VCFlagUploadCheckinNotification = chkCheckinReport.Checked;
    objParameters.VCFlagAlwaysSendEmailAllUsers = chkSendmailtoAllUsers.Checked;
    objParameters.VCFlagCheckoutNotification = chkCheckoutReport.Checked;

    objParameters.VCFlagOwnerMail = chkFolderOwner.Checked;
    objParameters.VCFlagGroupShareMail = chkGroupshareUsers.Checked;
    objParameters.Comments = txtComments.Text;
    objParameters.VCEmailTo = txtEmailRecp.Text;

    FAWAPI.CFolderPropertiesNotification objNotifications = new
FAWAPI.CFolderPropertiesNotification();
    objNotifications.Comment = chkCommentNotification.Checked;
    objNotifications.Delete = chkDeleteNotification.Checked;
    objNotifications.Move = chkMoveNotification.Checked;
    objNotifications.Copy = chkCopyNotification.Checked;
    objNotifications.Rename = chkRenameNotification.Checked;
    objNotifications.Create = chkCreateFolderNotification.Checked;
    objParameters.FolderPropertiesNotifications = objNotifications;

    switch (cmbSaveProperties.SelectedValue.ToString())
    {
        case "68":
            objParameters.SaveFlag =
FAWAPI.FolderPropertiesSaveFlags.DetachFromParent;
            break;
        case "73":
            objParameters.SaveFlag =
FAWAPI.FolderPropertiesSaveFlags.InheritFromParent;
            break;
        case "83":
            objParameters.SaveFlag =
FAWAPI.FolderPropertiesSaveFlags.SaveProperties;
            break;
    }

    FAWAPI.UpdateFolderPropertiesResult objUpdateFolderProperties =
objFAWAPI.UpdateFolderProperties(txtToken.Text, objParameters);
    if (!String.IsNullOrEmpty(objUpdateFolderProperties.ErrorMessage))
    {
        txtError.Text = objUpdateFolderProperties.ErrorMessage;
        return;
    }
}
```

```
        txtError.Text = objUpdateFolderProperties.Updated.ToString() ;  
    }
```

Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call
Parameters	Class	Yes	Represent various properties to be apply on folder, It consist of below sub attributes using that user can apply different properties on folder

- Parameters; It consist of below sub parameters which are needed to call UpdateFolderProperties:

Name	Data Type	Required	Description
FolderPath	String	Yes	The path of the folder to set properties for
Comments	String	No	Comments to attach to the folder.
VCFlagFileVersion	Boolean	Yes	Flag indicating the folder has version history control enabled.
VCFlagFileCheckinCheckOut	Boolean	Yes	Flag indicating the folder has check-in/checkout enabled.
VCFlagDisableDelete	Boolean	Yes	Flag indicating if delete is disabled for the folder.
VCFlagDailyEmail	Boolean	Yes	Flag indicating if a daily email of activity is sent.
VCFlagAlwaysSendEmailAllUsers	Boolean	Yes	Flag indicating notification goes to all users (OwnerUser/GroupShareUseractivity).
VCFlagUploadCheckinNotification	Boolean	Yes	Flag indicating that Notification email sent when File will be Uploaded or Checked-in.
VCFlagCheckOutNotification	Boolean	Yes	Flag indicating that Notification email sent when File will be Checkout
VCFlagOwnerMail	Boolean	Yes	Flag indicating that folder owner user should receive the notifications applied on folder.

VCEmailTo	String	No	Additional email addresses delimited by ; on which you would like to receive the notifications.
VCFlagGroupShareMail	Boolean	Yes	Flag indicating that Groupshare users of folder should receive the notifications applied on folder.
RemoveChildProperties	Boolean	Yes	Flag indicating that all properties from child folders will be removed.
VersionKeepCount	Int	No	Represent the number of Version wants to keep when FlagFileVersion enabled on folder.
SaveFlag	Enum	Yes	Represent how the properties would apply to folder based on the value: <code>DetachFromParent</code> <code>InheritFromParent</code> <code>SaveProperties</code>
FolderPropertiesNotifications	Class	No	Contains below sub properties to apply the notification on various events
FolderPropertiesNotificaitons			
Comment	Boolean	Yes	User will get notification when a new comment will be added to file/folder
Delete	Boolean	Yes	User will get notification when a folder/file will be deleted
Move	Boolean	Yes	User will get notification when a folder/file will be moved
Copy	Boolean	Yes	User will get notification when a folder/file will be copied
Rename	Boolean	Yes	User will get notification when a folder/file will be renamed
Create	Boolean	Yes	User will get notification when a folder will be created

Response

UpdateFolderPropertiesResult

UpdateFolderPropertiesResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while updating the folder properties. If the UpdateFolderProperties call was successful, the ErrorMessage will be blank or null.
Updated	Boolean	Returns true when properties apply to folder get updated successfully else returns false. newly created Username, if user successfully created.

FileAgingRule

Syntax

FileAgingRuleResult = FileAgingRule (Token, Parameters)

Usage

Create or Edit FileLifeCycle and FileArchive Rule on a folder.

The following parameters are needed to make the FileAgingRule call:

Parameter Name	Description
Token	The Token ID returned by the last AccountLogin call
Parameters	It contains various attributes to define the FileAgingRule on a folder. All attributes are declared below.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
```



```
SOAPAction: "http://api.filesanywhere.com/FileAgingRule"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <FileAgingRule xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Parameters>
        <AgingID>int</AgingID>
        <RuleName>string</RuleName>
        <FolderPath>string</FolderPath>
        <DeleteEmptySubFolder>boolean</DeleteEmptySubFolder>
        <Pattern>string</Pattern>
        <DeleteDays>int</DeleteDays>
        <StartDate>dateTime</StartDate>
        <ArchiveLocation>string</ArchiveLocation>
        <AgingType>FileLifeCycle or FileArchiveRule</AgingType>
        <AgingNotification>
          <NotificationAdvanceDays>int</NotificationAdvanceDays>
          <NotificationtoAllGroupShareUsers>boolean</NotificationtoAllGroupShareUser
s>
          <ListGSUsers>
            <string>string</string>
            <string>string</string>
          </ListGSUsers>
          <AdditionalRecipient>string</AdditionalRecipient>
          <FolderOwner>boolean</FolderOwner>
        </AgingNotification>
        <Enable>boolean</Enable>
      </Parameters>
    </FileAgingRule>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <FileAgingRuleResponse xmlns="http://api.filesanywhere.com/">
      <FileAgingRuleResult>
        <AgingID>int</AgingID>
      </FileAgingRuleResult>
    </FileAgingRuleResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The [placeholders](#) shown need to be replaced with actual values.

```

POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <FileAgingRule xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Parameters>
        <AgingID>int</AgingID>
        <RuleName>string</RuleName>
        <FolderPath>string</FolderPath>
        <DeleteEmptySubFolder>boolean</DeleteEmptySubFolder>
        <Pattern>string</Pattern>
        <DeleteDays>int</DeleteDays>
        <StartDate>dateTime</StartDate>
        <ArchiveLocation>string</ArchiveLocation>
        <AgingType>FileLifeCycle or FileArchiveRule</AgingType>
        <AgingNotification>
          <NotificationAdvanceDays>int</NotificationAdvanceDays>
          <NotificationtoAllGroupShareUsers>boolean</NotificationtoAllGroupShareUser
          s>
          <ListGSUsers>
            <string>string</string>
            <string>string</string>
          </ListGSUsers>
          <AdditionalReceipient>string</AdditionalReceipient>
          <FolderOwner>boolean</FolderOwner>
        </AgingNotification>
        <Enable>boolean</Enable>
      </Parameters>
    </FileAgingRule>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <FileAgingRuleResponse xmlns="http://api.filesanywhere.com/">
      <FileAgingRuleResult>

```

```
<AgingID>int</AgingID>  
</FileAgingRuleResult>  
</FileAgingRuleResponse>  
</soap12:Body>  
</soap12:Envelope>
```

Sample C#.Net Code

```

void FileAging()
{
    FAWAPI.FAWAPIv2SoapClient objFAWAPI = new FAWAPI.FAWAPIv2SoapClient();
    FAWAPI.AgingRuleParameter objAgingRule = null;
    if (chkAgingRule.Checked)
    {
        objAgingRule = new FAWAPI.AgingRuleParameter();
        objAgingRule.AgingID = Convert.ToInt16(txtAgingID.Text);
        objAgingRule.RuleName = txtAgingRuleName.Text;
        objAgingRule.FolderPath = txtAgingFolderPath.Text;
        objAgingRule.DeleteEmptySubFolder = chkDeleteEmptyFolders.Checked;
        objAgingRule.Pattern = txtFilePattern.Text;
        objAgingRule.DeleteDays = Convert.ToInt32(txtRetDays.Text);
        objAgingRule.Enable = chkEnable.Checked;
        if (dtAgingStartDate.Checked)
            objAgingRule.StartDate = dtAgingStartDate.Value;
        if (rbtnArchiveRule.Checked)
            objAgingRule.AgingType = FAWAPI.AgingType.FileArchiveRule;
        if (rbtnLifeCycle.Checked)
            objAgingRule.AgingType = FAWAPI.AgingType.FileLifeCycle;
        objAgingRule.ArchiveLocation = txtArchiveLocation.Text;

        //Notification Parameters
        FAWAPI.CAgingNotification objAgingNotificaitons = null;
        if (chkNotifications.Checked) {
            objAgingNotificaitons = new FAWAPI.CAgingNotification();
            objAgingNotificaitons.NotificationAdvanceDays =
Convert.ToInt16(txtAgingNotificationDays.Text);
            objAgingNotificaitons.NotificationtoAllGroupShareUsers =
rbtnGSAllUsers.Checked;
            if (rbtnGSSelectedUsers.Checked)
            {
                List<string> lstUsers = new List<string>();
                foreach (var item in txtUserList.Text.Split(';'))
                {
                    lstUsers.Add(item);
                }
                objAgingNotificaitons.ListGSUsers = lstUsers.ToArray();
            }
            objAgingNotificaitons.FolderOwner = chkFolderOwnerAging.Checked;
            objAgingNotificaitons.AdditionalRecipient = txtAdditionalEmail.Text;
        }
        objAgingRule.AgingNotification = objAgingNotificaitons;
    }

    FAWAPI.AgingRuleResult objAgingRuleResult =
objFAWAPI.FileAgingRule(txtToken.Text, objAgingRule);
    if (!String.IsNullOrEmpty(objAgingRuleResult.ErrorMessage))
    {
        txtError.Text = objAgingRuleResult.ErrorMessage;
        return;
    }
    txtError.Text = String.Format("Aging ID {0} Added/Edited successfully",
objAgingRuleResult.AgingID);
}

```

Arguments

Name	DataType	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call
Parameters	Class	Yes	Represent various properties to be used to create or edit FileAging rule on a folder.

- Parameters; It consist of below sub parameters which are needed to call FileAgingRule API:

Name	DataType	Required	Description
AgingID	Integer	Yes	To create new rule AgingID needs to pass as 0 else for edit operation respective rule AgingID needs to pass.
RuleName	String	Yes	Aging rule name
FolderPath	String	Yes	Folder on which Rule needs to be define.
DeleteEmptySubFolder	Boolean	Yes	Flag indicating the after Archiveal or LifeCycle of files if sub folder(s) are empty then remove it.
Pattern	String	Yes	FileName and extension to include in AgingRule e.g. Mark all the files then Pattern would be *.* Specific extension files then pattern would be *.doc
DeleteDays	Integer	Yes	Files older than specified days will be Deleted or Archived, value should be >0 and <= 9999 days.
StartDate	Date	Yes	Value indicates date onwards rule will start its execution.
AgingType	Enum	Yes	Value indicates Aging rule type, value could be FileLifeCycle or FileArchiveRule
ArchiveLocation	String	Yes	FolderPath where file will be archived When Aging type is FileArchiveRule. When AgingType is FileLifeCycle then value will not

			consider.
Enable	Boolean	Yes	Flag indicating that Rule is enabled or not.
AgingNotification	Class	Yes	Object represents on Archival and Life cycle notification settings. Contains below properties
AgingNotification			
NotificationAdvanceDays	Integer	Yes	Value represents how many days before system should send Archival/LifeCycle reminder mail.
NotificationtoAllGroupShare Users	Boolean	Yes	All Groupshare users will get notification when it is set to True. When it is set to False then only Selected GroupShare user will get notification.
ListGSUsers	List	String	List of Groupshare users which will receive the notifications when NotificationtoAllGroupShareUsers is set to False.
AdditionalRecipient	String	Yes	List of email addresses delimited by ;
FolderOwner	Boolean	Yes	Flag indicates that owner of folder will receive the notifications of Aging rule execution or not.

Response

FileAgingRuleResult

FileAgingRuleResult Properties

Name	Data Type	Description
ErrorMessage	String	A description of the error, if any error was encountered while updating the folder properties. If the FileAgingRule call was successful, the ErrorMessage will be blank or null.
AgingID	Integer	Returns the AgingID of inserted

		or updated rule.
--	--	------------------

FileAgingRuleRemove

Syntax

FileAgingRuleRemoveResult = FileAgingRuleRemove (Token, Parameters)

Usage

Removes File LifeCycle or Archive rules.

The following parameters are needed to make the FileAgingRule call:

Parameter Name	Description
Token	The Token ID returned by the last AccountLogin call
Parameters	It contains various attributes to remove the FileAging rule. All attributes are declared below.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://api.filesanywhere.com/FileAgingRuleRemove"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <FileAgingRuleRemove xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Parameters>
        <ListAgingID>
          <int>int</int>
          <int>int</int>
        </ListAgingID>
      </Parameters>
    </FileAgingRuleRemove>
  </soap:Body>
```

```

</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <FileAgingRuleRemoveResponse xmlns="http://api.filesanywhere.com/">
      <FileAgingRuleRemoveResult />
    </FileAgingRuleRemoveResponse>
  </soap:Body>
</soap:Envelope>

```

SOAP 1.2

The following is a sample SOAP 1.2 request and response. The [placeholders](#) shown need to be replaced with actual values.

```

POST /v2/fawapi.asmx HTTP/1.1
Host: api.filesanywhere.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <FileAgingRuleRemove xmlns="http://api.filesanywhere.com/">
      <Token>string</Token>
      <Parameters>
        <ListAgingID>
          <int>int</int>
          <int>int</int>
        </ListAgingID>
      </Parameters>
    </FileAgingRuleRemove>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <FileAgingRuleRemoveResponse xmlns="http://api.filesanywhere.com/">
      <FileAgingRuleRemoveResult />
    </FileAgingRuleRemoveResponse>
  </soap12:Body>

```



```
</soap12:Envelope>
```

Sample C#.Net Code

```

void RemoveAgingRule ()
{
    FAWAPI.FAWAPIv2SoapClient objFAWAPI = new FAWAPI.FAWAPIv2SoapClient();
    FAWAPI.AgingRuleRemoveParameter objAgingRule = null;
    if (chkFileAgingRemove.Checked)
    {
        objAgingRule = new FAWAPI.AgingRuleRemoveParameter();
        objAgingRule.ListAgingID =
Array.ConvertAll(txtAgingIDRemove.Text.Split(';').ToArray(), int.Parse);
    }
    FAWAPI.AgingRuleRemoveResult objAgingRuleRemoveResult =
objFAWAPI.FileAgingRuleRemove(txtToken.Text, objAgingRule);
    if (!String.IsNullOrEmpty(objAgingRuleRemoveResult.ErrorMessage))
    {
        txtError.Text = objAgingRuleRemoveResult.ErrorMessage;
        return;
    }
    txtError.Text = String.Format("Aging ID removed successfully");
}
}

```

Arguments

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call
Parameters	Class	Yes	It contains various attributes to remove the FileAging rule. All attributes are declared below.

- Parameters; It consist of below sub parameters which are needed to call FileAgingRule API:

Name	Data Type	Required	Description
ListAgingID	Integer List	Yes	List of AgingID needs to supply which needs to be deleted.

Response

FileAgingRuleRemoveResult

FileAgingRuleRemoveResult Properties

Name	Data Type	Description
------	-----------	-------------

ErrorMessage	String	A description of the error, if any error was encountered while updating the folder properties. If the FileAgingRule call was successful, the ErrorMessage will be blank or null.
AgingID	Integer	Returns the AgingID of inserted or updated rule.

Introducing the FilesAnywhere REST API

REST API Service URL

<https://restapi.filesanywhere.com/FAWAPI.svc>

REST API Operations

The following operations are supported. For a formal definition, please review the [Service Description](#).

- **DownloadFile**
Provides a Stream from which a file can be downloaded.
- **ZipDownloadFileOrFolder**
Provides a Stream from which a zip file, containing a file or a folder, can be downloaded.
- **DownloadWithCheckout**
Performs Checkout on a file and then provides a Stream from which the file can be downloaded.
- **DownloadFromFileshare**
Provides a Stream from which all files in a fileshare link can be downloaded.
- **PreUploadValidation**
To upload a file first we need to call this method, which will return UploadKey.
- **UploadItem OR ChunkUpload**
Once user has a valid upload key, user can start the upload of file using 'UploadItem' method or 'ChunkUpload' method. We recommend to use 'ChunkUpload' method for larger file size, i.e. more than 256MB file.

- **ResumeUpload**
While chunk upload is in progress, user can pause and resume the upload service again.

Let's Get Started!

This section helps you in getting started with using the REST API. It tells you what you need in order to start using the REST API in your development environment. It includes the following steps:

- Sign up for a FilesAnywhere account using FAWAPI webservice.
- Use obtained token to invoke the REST API methods with required parameters

Sign up for a Filesanywhere account using FAWAPI

Using FAWAPI web service, invoke AccountLogin method, which will return Token number. Description of AccountLogin method can be found in below URL. <https://api.filesanywhere.com/v2/fawapi.asmx?op=AccountLogin>

Use this token to invoke all REST API methods.

REST API Calls

DownloadFile

This method returns back a stream using which you can download a file.

Syntax

HTTPs Request

POST: /DownloadFile

HOST: <https://restapi.filesanywhere.com/FAWAPI.svc>

Filesanywhere-API-Arg : {"ItemPath":"","Token":""}

Invoke this Host URL, and post the argument ItemPath and Token.

Usage

Use the `DownloadFile` method to get a stream using which you can download a file.

The following parameters are needed to make the `DownloadFile` call:

- Token (The Token ID returned by the last `AccountLogin` call).
- Path (Name of the file to be downloaded along with the its full Path on FilesAnywhere). The Path should be formatted like `\UserVolume\FullPathWithFolderName\FileName`, where `FullPathWithFoldername` is the name of the folder in which the file exists along with its full path and `UserVolume` is the FilesAnywhere Username in whose account the file exists.

Sample C#.Net Code

```
private void DownloadRequest(string sToken, string FilePath)
{
    String swCFURL = "https://restapi.filesanywhere.com/FAWAPI.svc/DownloadFile";

    //DownloadRequest is an object which contains Token and ItemPath as a
    //properties which serialized and passed to REST API.
    DownloadRequest drequest = new DownloadRequest
    {
        Token = sToken,
        ItemPath = FilePath
    };

    var request = (HttpWebRequest)WebRequest.Create(swCFURL);
    request.ContentType = "text/json";
    request.Method = "POST";

    DataContractJsonSerializer ser = new
    DataContractJsonSerializer(typeof(DownloadRequest));
    if (drequest != null)
    {
        using (var streamWriter = new
    StreamWriter(request.GetRequestStream()))
        {
            MemoryStream mem = new MemoryStream();
            ser.WriteObject(mem, drequest);
            string data = Encoding.UTF8.GetString(mem.ToArray(), 0,
(int)mem.Length);
            streamWriter.Write(data);
        }
    }
    HttpResponseMessage HttpResponseMessage = null;
    try
    {
        HttpResponseMessage = (HttpWebResponse)request.GetResponse();
        if (HttpResponse.StatusCode == HttpStatusCode.OK)
        {
```

```

        using (Stream iStream = HttpResponseMessage.GetResponseStream())
        {
            string sDestPath = "D:\\\" + Path.GetFileName(FilePath);
            FileStream destStream = File.Create(sDestPath);
            iStream.CopyTo(destStream);
            destStream.Close();
        }
    }
    else
    { //Error Message
        txtOutput.Text = new
StreamReader(HttpResponse.GetResponseStream()).ReadToEnd();
    }

    }
    catch (WebException error)
    {
        HttpResponseMessage = (HttpWebResponse)error.Response;
    }
}
public class DownloadRequest
{
    public string ItemPath { get; set; }
    public string Token { get; set; }
}

```

Request Arguments

Request arguments should be passed as a JSON serialized string e.g.:

```

{"ItemPath":"\\PATNIDEV\\test.docx","Token":"E9BVY5YK8C1Y2ZCS97KKWTJ095IP9XVT7Z597E20C910"}

```

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
ItemPath	String	Yes	Full path of the file on FilesAnywhere in the form \\UserVolume\FullPathWithFolderName\FileName

Response

If the file is available to be downloaded, the response will be a HttpStatus code 200 OK.

In case of status code OK, user can save the Stream to a local destination path, as mentioned in above sample code.

In case of validation error, it will return status code 406 (Not acceptable), and in case of warning exception it will return 417 (Expectation failed)

To get the error message, the stream needs to be read into a string object, as in the c# code below:

```
Error = new StreamReader(HttpResponse.GetResponseStream()).ReadToEnd();
```

ZipDownloadFileOrFolder

This method returns back a stream from which a zip file containing a file or a folder can be downloaded. If the return HttpStatus code is 202 (Accepted), it means that the zip process is in progress, and, this method can be called repeatedly until the zip processing is complete. Making subsequent calls with the same parameters will query the status of the initial zip request process.

Syntax

HTTP Request

POST: /ZipDownloadFileOrFolder

HOST: <https://restapi.filesanywhere.com/FAWAPI.svc>

Filesanywhere-API-Arg : {"ItemPath":"","Token":""}

Invoke this Host URL, and post the argument ItemPath and Token.

Usage

Use the ZipDownloadFileOrFolder method to download a zip file containing a file or a folder.

The following parameters are needed to make the ZipDownloadFileOrFolder call:

- Token (The Token ID returned by the last AccountLogin call).
- Path (The path of the file/folder to be downloaded.)

Sample C#.Net Code

```
private void ZipDownloadRequest(string sToken, string FilePath)
{
    String sWCFURL = "
https://restapi.filesanywhere.com/FAWAPI.svc/ZipDownloadFileOrFolder"

    //DownloadRequest is an object which contains Token and ItemPath as a
    //properties which serialized and passed to REST API.
```

```
DownloadRequest drequest = new DownloadRequest
{
    Token = sToken,
    ItemPath = FilePath
};

var request = (HttpRequest)WebRequest.Create(sWCFURL);
request.ContentType = "text/json";
request.Method = "POST";

DataContractJsonSerializer ser = new
DataContractJsonSerializer(typeof(DownloadRequest));
if (drequest != null)
{
    using (var streamWriter = new
StreamWriter(request.GetRequestStream()))
    {
        MemoryStream mem = new MemoryStream();
        ser.WriteObject(mem, drequest);
        string data = Encoding.UTF8.GetString(mem.ToArray(), 0,
(int)mem.Length);
        streamWriter.Write(data);
    }
    HttpResponseMessage HttpResponseMessage = null;

do
{
    _response = (HttpResponse)request.GetResponse();

    if (_response.StatusCode == HttpStatusCode.OK)
    {
        using (Stream iStream = _response.GetResponseStream())
        {
            if (iStream == null)
                return;
            string sDestPath = sDestPath = "D:\\FileName.zip";
            FileStream destStream = File.Create(sDestPath);
            iStream.CopyTo(destStream);
            destStream.Close();
            txtOutput.Text = "Zip file downloaded successfully at
location " + txtDestPath.Text;
        }
        return;
    }
    else if (_response.StatusCode == HttpStatusCode.Accepted)
    {
        txtOutput.Text = "Processing Zip download id =" + new
StreamReader(_response.GetResponseStream()).ReadToEnd();
    }
    else
    {
        txtOutput.Text = "Error =" + new
StreamReader(_response.GetResponseStream()).ReadToEnd();
        break;
    }
}
```



```

    }
    request = CreateRequest(sWCFURL1);

    } while (_response.StatusCode == HttpStatusCode.OK);
    _response = null;
}
public class DownloadRequest
{
    public string ItemPath { get; set; }
    public string Token { get; set; }
}

```

Request Arguments

Request arguments should be passed as a JSON serialized string e.g.:

```
{"ItemPath":"\\PATNIDEV\\test\\","Token":"E9BVY5YK8C1Y2ZCS97KKWTJ095IP9XVT7Z597E20C910"}
```

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
ItemPath	String	Yes	Full path of the file or folder on FilesAnywhere in the form \UserVolume\FullPathWithFolderName\FileName Or \UserVolume\FullPathWithFolderName\

Response

If the file is available to be downloaded, the response will be a HttpStatusCode code 200 OK.

In case of status code OK, user can save the Stream to a local destination path, as mentioned in above c# code.

If zipping is in progress, then it will return status code 202 (Accepted).

In case of validation error, it will return status code 406 (Not acceptable), and in case of warning exception it will return 417 (Expectation failed)

To get the error message, the stream needs to be read into a string object, as in the c# code below:

```
Error = new StreamReader(HttpResponse.GetResponseStream()).ReadToEnd();
```

DownloadWithCheckout

This method performs checkout on a file and then returns back a stream using which you can download the file.

Syntax

HTTPs Request

POST: /DownloadWithCheckout

HOST: <https://restapi.filesanywhere.com/FAWAPI.svc>

Filesanywhere-API-Arg :

```
{"Comments":"","EmailReport":true,"ItemPath":"","Recipients":"","Token":""}
```

Invoke this Host URL, and post the argument ItemPath, Token, Comments, EmailReport, Recipients.

Usage

Use the DownloadWithCheckout method to checkout a file and get a stream using which you can download the file.

The following parameters are needed to make the DownloadFile call:

- Token (The Token ID returned by the last AccountLogin call).
- Path (The path of the file to be downloaded).
- Comments. (Checkout comments)
- EmailReport. (Flag indicating whether to send an email notification).
- AdditionalRecipients. (Additional recipients to send the notification to).

Sample C#.Net Code

```
private void DownloadRequest(string sToken, string FilePath)
{
    String swCFURL =
    "https://restapi.filesanywhere.com/FAWAPI.svc/DownloadWithCheckout"
```

```
    //DownloadRequest is an object which contains Token and ItemPath as a
    //properties which serialized and passed to REST API.
```

```
    DownloadCheckoutRequest drequest = new DownloadCheckoutRequest
```

```

        {
            Token = sToken,
            ItemPath = txtFilePath.Text,
            Comments = txtComments.Text,
            EmailReport = chkSendEmail.Checked,
            Recipients = txtRecipients.Text
        };

var request = (HttpRequest)WebRequest.Create(sWCURL);
request.ContentType = "text/json";
request.Method = "POST";

DataContractJsonSerializer ser = new
DataContractJsonSerializer(typeof(DownloadCheckoutRequest));
if (drequest != null)
{
    using (var streamWriter = new
StreamWriter(request.GetRequestStream()))
    {
        MemoryStream mem = new MemoryStream();
        ser.WriteObject(mem, drequest);
        string data = Encoding.UTF8.GetString(mem.ToArray(), 0,
(int)mem.Length);
        streamWriter.Write(data);
    }
}
HttpWebResponse HttpResponse = null;
try
{
    HttpResponse = (HttpWebResponse)request.GetResponse();
    if (HttpResponse.StatusCode == HttpStatusCode.OK)
    {
        using (Stream iStream = HttpResponse.GetResponseStream())
        {
            string sDestPath = "D:\\\" + Path.GetFileName(FilePath);
            FileStream destStream = File.Create(sDestPath);
            iStream.CopyTo(destStream);
            destStream.Close();
        }
    }
    else
    {
        //Error Message
        txtOutput.Text = new
StreamReader(HttpResponse.GetResponseStream()).ReadToEnd();
    }
}
catch (WebException error)
{
    HttpResponse = (HttpWebResponse)error.Response;
}
}
public class DownloadCheckoutRequest
{
    public string ItemPath { get; set; }
    public string Token { get; set; }
}

```

```

        public string Comments { get; set; }
        public Boolean EmailReport { get; set; }
        public string Recipients { get; set; }
    }

```

Request Arguments

Request arguments should be passed as an JSON serialize string e.g.:

```
{"Comments":"","EmailReport":true,"ItemPath":"","Recipients":"","Token":""}
```

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned by the last AccountLogin call.
ItemPath	String	Yes	The path to the file to be downloaded.
Comments	String	No	Checkout comments to be logged with the file activity.
EmailReport	Boolean	Yes	Flag indicating whether to send email notification of the activity.
AdditionalRecipients	String	Yes	Additional recipients to send the notification to.

Response

If the file is available to be downloaded, the response will be a HttpStatus code 200 OK.

In case of status code OK, user can save the Stream to a local destination path, as mentioned in above c# code.

In case of validation error, it will return status code 406 (Not acceptable), and in case of warning exception it will return 417 (Expectation failed)

To get the error message, the stream needs to be read into a string object, as in the c# code below:

```
Error = new StreamReader(HttpResponse.GetResponseStream()).ReadToEnd();
```

DownloadFromFileShare

This method allows you to download one or more files or folders from a FileShare link.

Syntax

HTTP Request

POST: /DownloadFromFileShare

HOST: <https://restapi.filesanywhere.com/FAWAPI.svc>

Filesanywhere-API-Arg : {"FSToken":"","FileIDList":"","PathList":""}

Invoke this Host URL, and post the argument FSToken, FileIDList and PathList.

Usage

This method allows you to download one or more files or folders (* delimited list, also must start with *) from a FileShare link. You will get back a stream for downloading.

If there is a folder path in the list or more than one file to be downloaded, they will be zipped and the zip file will be streamed.

The following parameters are needed to make the DownloadFromFileShare call:

- **FSToken** (The FileShare Token ID returned by the last FileShareGetHeaderInfo call).
- **FileIDList** is a "*" delimited list of File IDs returned by the FileShareGetItems call. The list should start with a "*" and NOT end with one.
- **PathList** is a "*" delimited list of file paths to be downloaded. The list should start with a "*" and NOT end with one. *NOTE: the FileIDList and PathList should match one to one. The combination of FileID and Path make up the unique key of the item to be downloaded.*

Sample C#.Net Code

```
private string DownloadFSFile(string sFSToken, string sFileID, string sPath){  
String swCFURL = " https://restapi.filesanywhere.com/FAWAPI.svc/DownloadFromFileShare"
```

```
        FSDownloadRequest drequest = new FSDownloadRequest  
        {  
            FSToken = sFSToken,  
            FileIDList = sFileID,  
            PathList = sPath  
        };  
  
        var request = (HttpWebRequest)WebRequest.Create(sURL);  
        request.ContentType = "text/json";  
        request.Method = "POST";  
        request.CookieContainer = new CookieContainer();  
  
        DataContractJsonSerializer ser = new  
        DataContractJsonSerializer(typeof(FSDownloadRequest));  
        if (drequest != null)
```

```
        {
            using (var streamWriter = new
StreamWriter(request.GetRequestStream()))
            {
                MemoryStream mem = new MemoryStream();
                ser.WriteObject(mem, drequest);
                string data = Encoding.UTF8.GetString(mem.ToArray(), 0,
(int)mem.Length);
                streamWriter.Write(data);
            }
        }
    }

    HttpResponseMessage httpResponse = null;
    try
    {
        httpResponse = (HttpResponse)request.GetResponse();
        if (httpResponse.StatusCode == HttpStatusCode.OK)
        {
            using (Stream inputStream = httpResponse.GetResponseStream())
            {
                string fileName = Path.GetFileName(sPath);
                if (sPath.Contains("*"))
                {
                    fileName = "TestFile.zip";
                }
                string sDestPath = "D:\\\" + fileName;
                FileStream destStream = File.Create(sDestPath);
                inputStream.CopyTo(destStream);
                destStream.Close();
            }
        }
        else if (httpResponse.StatusCode == HttpStatusCode.Accepted)
        {
            txtOutput.Text = "Processing Zip download id =" + new
StreamReader(httpResponse.GetResponseStream()).ReadToEnd();
        }
        else
        {
            txtOutput.Text = "Error =" + new
StreamReader(httpResponse.GetResponseStream()).ReadToEnd();
        }
    }
    catch (WebException error)
    {
        httpResponse = (HttpResponse)error.Response;
        sError = error.Response.ToString();
    }
}
```

Request Arguments

Request arguments should be passed as a JSON serialized string e.g.:

```
{"FSToken":"b101793b","FileIDList":"3500851","PathList":"\\PATNIDEV\\0013Dbox\\ImagePDF.docx"}
```

Name	Data Type	Required	Description
FSToken	String	Yes	The FSToken ID returned back in the last FileShareGetHeaderInfo call
FileIDList	String	Yes	List of FileIDs returned by the FileShareGetItems call delimited by a * in the form of: *12*332*12*222 Or *12 NOTE: These are NOT unique ids. The combination of FileID and Path make up the key.
PathList	String	Yes	List of Paths returned by the FileShareGetItems call delimited by a * in the form of: *\username\folder\file.ext*\username\folder2\ Or *\username\folder\file.ext NOTE: the FileIDList and PathList must match one to one. The key for a file is the FileID and the Path.

Response

If the file share files are available to be downloaded, the response will be a HttpStatus code 200 OK.

In case of status code OK, user can save the Stream to a local destination path, as mentioned in above c# code.

If file share zip is in progress, then it will return status code 202 (Accepted).

In case of validation error, it will return status code 406 (Not acceptable), and in case of warning exception it will return 417 (Expectation failed)

To get the error message, the stream needs to be read into a string object, as in the c# code below:

```
Error = new StreamReader(HttpResponse.GetResponseStream()).ReadToEnd();
```

DownloadFormData

This method allows you to download form data in excel or csv format

Syntax

HTTP Request

POST: / DownloadFormData

HOST: <https://restapi.filesanywhere.com/FAWAPI.svc>

Filesanywhere-API-Arg :

```
{"Token":"","FormID":"","FilterParams":{"FromDataID":"","UptoDataID":"","DateType":"","StartDate":"","EndDate":""},"ExportFields":"","ExportType:"}
```

Invoke this Host URL, and post the arguments Token, FormID, FilterParams, ExportFields and ExportType.

Usage

This method allows you to download form data of given form ID. You will get back a form data as a stream for downloading as either Excel (.xlsx) or CSV (.csv) file format

The following parameters are needed to make the DownloadFormData call:

- Token (API Token ID returned by the last AccountLogin call).
- FormID is the form ID for which you want to download the data
- FilterParams is a custom class containing different search filter combinations to narrow the data search results. Refer “Request Arguments” section for more details
- ExportFields is a list of form field names for which you want download the data instead of all fields. If ExportFields parameter is null or empty then API call will return all field data.
- ExportType specifies download file formats either Excel (.xlsx) or CSV (.csv)

Sample C#.Net Code

```
private string DownloadFormData(string sToken, int iFormID, int iFromDataID, int iUptoDataID, FServiceManage.CCommonConstantSearchByDateType objSearchByDateType, string sStartDate, string sEndDate, List<string> lstFieldNames, FServiceManage.CCommonConstantExportType objExportType, string sParentFolderPath)
{
    String sWCFURL =
    "https://restapi.filesanywhere.com/FAWAPI.svc/DownloadFormData";
    string sError = string.Empty;

    FServiceManage.FormDataRequest.SearchFilters objSearchFilters = new
    FServiceManage.FormDataRequest.SearchFilters
    {
```



```

        FromDataID = iFromDataID,
        UptoDataID = iUptoDataID,
        DateType = objSearchByDateType,
        StartDate = sStartDate,
        EndDate = sEndDate
    };

    FServiceManage.FormDataRequest objFormDataRequest = new
FServiceManage.FormDataRequest
    {
        Token = sToken,
        FormID = iFormID,
        FilterParams = objSearchFilters,
        ExportFields = lstFieldNames.Count > 0 ? lstFieldNames.ToArray() :
null,
        ExportType = objExportType
    };

    var request = (HttpRequest)WebRequest.Create(SWCFURL);
    request.ContentType = "text/json";
    request.Method = "POST";
    request.CookieContainer = new CookieContainer();

    DataContractJsonSerializer objDataContractJsonSerializer = new
DataContractJsonSerializer(typeof(FServiceManage.FormDataRequest));
    if (objFormDataRequest != null)
    {
        using (var streamWriter = new
StreamWriter(request.GetRequestStream()))
        {
            MemoryStream objMemoryStream = new MemoryStream();
            objDataContractJsonSerializer.WriteObject(objMemoryStream,
objFormDataRequest);
            string sData = Encoding.UTF8.GetString(objMemoryStream.ToArray(),
0, (int)objMemoryStream.Length);
            streamWriter.Write(sData);
        }
    }

    HttpResponseMessage httpResponse = null;
    try
    {
        httpResponse = GetHttpResponse(request);
        System.Threading.Thread.Sleep(500); //Give some pause and display
progress to user
        if (httpResponse.StatusCode == HttpStatusCode.OK)
        {
            using (Stream iStream = httpResponse.GetResponseStream())
            {
                if (!Directory.Exists(sParentFolderPath))
                {
                    Directory.CreateDirectory(sParentFolderPath);
                }
                string sDestPath = sParentFolderPath +
string.Format("{0}_{1}.{2}",

```

```

Path.GetFileNameWithoutExtension(Path.GetRandomFileName()), iFormID, objExportType ==
FServiceManage.CCommonConstantExportType.Excel ? "xlsx" : "csv");
        FileStream destStream = File.Create(sDestPath);
        iStream.CopyTo(destStream);
        destStream.Close();
    }
}
else
{
    sError = "Error while processing Download Form Data : " + new
StreamReader(HttpResponse.GetResponseStream()).ReadToEnd();
}
}
catch (WebException error)
{
    HttpResponseMessage = (HttpWebResponse)error.Response;
    sError = error.Response.ToString();
}
catch (Exception ex)
{
    sError = ex.Message;
}
}
}

```

Request Arguments

Request arguments should be passed as a JSON serialized string e.g.:

```

{"Token":"51BQ17C9X3JGSORB6NIYHUM8LRXVM45TL18438J35847","FormID":157,"F
ilterParams":{"FromDataID":101,"UptoDataID":105,"DateType":1,"StartDate":"2019-08-
01 12:00:00 AM","EndDate":"2019-08-12 2:00:00
PM"},"ExportFields":null,"ExportType":1}



```

Name	Data Type	Required	Description
Token	String	Yes	API Token ID returned by the last AccountLogin call
FormID	Integer	Yes	FormID is the form ID for which you want to download the data Form ID can be referred from eForm/Workflow dashboard page from user site. Refer below image "Dashboard Screen" for more details.
SearchFilters.FromDataID	Integer	No	Start position of data ID specific search Form Data ID can be referred from eForm/Workflow Data Viewer page from user site. Refer below image "Data Viewer Screen" for more details.

SearchFilters.UptoDataID	Integer	No	End position of data ID specific search
SearchFilters.DateType	Integer	No	Perform date specific search based on date types Last Updated On or Created On. Specify None if you do not want to perform date specific search Last Updated On or Created On dates can be referred from eForm/Workflow Data Viewer page from user site. Refer below image "Data Viewer Screen" for more details.
SearchFilters.StartDate	DateTime	No	Start Date of date specific search Expected date format is 'YYYY-MM-DD hh:mm:ss AM/PM'
SearchFilters.EndDate	DateTime	No	End Date of date specific search Expected date format is 'YYYY-MM-DD hh:mm:ss AM/PM'
ExportFields	Array of Strings	No	Array of form field names for which you want to download the data instead of all fields. If ExportFields parameter is null then API call will return all exportable field data Field Names can be referred from eForm/Workflow Design page from user site. Refer below image "Design Screen" for more details.
ExportType	Integer	Yes	ExportType specifies download file formats Excel (.xlsx) or CSV (.csv)

Dashboard Screen

eForm Dashboard
Create new or view active eForms

 CREATE A NEW EFORM  VIEWS 

Sort by: Date | Name

 Vacation Request
2 Fields | 5 Submissions | 0 Uploaded Files | Created Aug 21 2019 | FormID 160

Data Viewer Screen

MY DOCUMENTS SHARE **EFORMS** WORKFLOW TOOLS HISTORY

eForm Vacation Request

Data Viewer • Vacation Request CREATE DATA VIEWS

EXPORT Search by form field data or file name...

	ID ▲	Updated	Sr. No	Requested By	Created
<input type="checkbox"/>	378	8/21/2019 7:15:40 AM	1	John Doe	8/21/2019 6:24:56 AM
<input type="checkbox"/>	379	8/21/2019 7:16:32 AM	2	Alex Doe	8/21/2019 6:25:03 AM
<input type="checkbox"/>	380	8/21/2019 7:16:19 AM	3	Joe Doe	8/21/2019 6:25:09 AM
<input type="checkbox"/>	381	8/21/2019 7:16:04 AM	4	Adam Doe	8/21/2019 6:25:15 AM
<input type="checkbox"/>	382	8/21/2019 7:15:52 AM	5	Jane Doe	8/21/2019 6:25:31 AM

Design Screen

Field Type Field Size delete

Field Label

Instructions

Field Name

Field Error Message

Is this a calculated field? Yes No

Options: Required Unique Associate as Tag

Masking: Enable data masking

Cancel Done

Response

If the form data is available to be downloaded, the response will be a HttpStatus code 200 OK.

In case of status code OK, user can save the Stream to a local destination path, as mentioned in above c# code.

In case of validation error, it will return status code 406 (Not acceptable

To get the error message, the stream needs to be read into a string object, as in the c# code below:

```
Error = new StreamReader(HttpResponse.GetResponseStream()).ReadToEnd();
```

PreUploadValidation

This method returns back an upload key using which you can start to upload a file in your validated account.

Syntax

HTTPs Request

POST: /PreUploadValidation

HOST: <https://restapi.filesanywhere.com/FAWAPI.svc>

Filesanywhere-API-Arg : {"UploadPath":"","Token":""}

Invoke this Host URL, and post the argument UploadPath and Token.

Usage

Use the PreUploadValidation method to get an UploadKey using which you can start to upload a file.

The following parameters are needed to make the PreUploadValidation call:

- Token (The Token ID returned by the last AccountLogin call).
- Path (Name of the path where you want upload your files). The Path should be formatted like `\UserVolume\FullPathWithFolderName\FileName`, where FullPathWithFoldername is the name of the folder in which the file exists along with its full path and UserVolume is the FilesAnywhere UserName in whose account the file exists.

Sample C#.Net Code

```
private void PreUploadRequest(string sToken, string UploadPath)
{
    String sRESTURL = "https://restapi.filesanywhere.com/FAWAPI.svc";

    string sURL = string.Format("{0}/PreUploadValidation?UploadPath={1}&Token={2}",
    sRESTURL, UploadPath, sToken);
    HttpWebRequest req = (HttpWebRequest)HttpWebRequest.Create(sURL);
    req.Method = "POST";
    req.ContentType = "application/xml";
    req.ContentLength = 0;
    req.Timeout = 1000000;
    try
    {
        HttpResponseMessage HttpResponseMessage = (HttpResponse)req.GetResponse();

        if (HttpResponse.StatusCode == HttpStatusCode.OK)
        {
            txtUploadKey.Text = new
StreamReader(HttpResponse.GetResponseStream()).ReadToEnd().Replace("\"", "");
        }
        else
        {
            txtOutput.Text = new
StreamReader(HttpResponse.GetResponseStream()).ReadToEnd().Replace("\"", "");
        }
    }
    catch (Exception ex)
    {
        txtOutput.Text = ex.Message;
    }
}
```

Request Arguments

Request arguments should be passed as a JSON serialized string e.g.:

```
{"UploadPath":"\\PATNIDEV\\test.docx","Token":"E9BVY5YK8C1Y2ZCS97KKWTJ095IP9XVT7Z597E20C910"}
```

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
UploadPath	String	Yes	Full path of the file on FilesAnywhere in the form \\UserVolume\FullPathWithFolderName\FileName

Response

If the upload key created successfully, the response will be a HttpStatusCode code 200 OK.

In case of status code OK, user can retrieve the upload key, as mentioned in above sample code.

In case of validation error, it will return status code 406 (Not acceptable), and in case of warning exception it will return 417 (Expectation failed)

To get the error message, the stream needs to be read into a string object, as in the c# code below:

```
Error = new StreamReader(HttpResponse.GetResponseStream()).ReadToEnd();
```

UploadItem

This method upload a file in filesanywhere user account.

Syntax

HTTPs Request

POST: /UploadItem

HOST: <https://restapi.filesanywhere.com/FAWAPI.svc>

Filesanywhere-API-Arg : {"UploadPath":"","Token":"","UploadKey":"","Compress":"","FileSize":""}

Invoke this Host URL, and post the argument UploadPath, Token, UploadKey and Compress.

Usage

Use the UploadItem method to start uploading a file in your filesanywhere account.

The following parameters are needed to make the UploadItem call:

- Token (The Token ID returned by the last AccountLogin call).
- Path (Name of the path where you want upload your files). The Path should be formatted like `\UserVolume\FullPathWithFolderName\FileName`, where FullPathWithFoldername is the name of the folder in which the file exists along with its full path and UserVolume is the FilesAnywhere UserName in whose account the file exists.
- UploadKey (Upload key which will be return from 'PreUploadValidation' method)

- Compress (User can pass the value as 'Y' or 'N' or ". If you want to upload a compressed file and then start to upload (to save the bandwidth), then you need to compressed file using `gzipStream`, and then start to upload ".gz" file, along with this pass 'Y' in this parameter. Once file upload completed, API method will decompress the file and upload it to your filesanywhere account.)
- `FileSize` size of the uploaded file, if it is compressed file then after compression size of file needs to be assigned.

Sample C#.Net Code

```
private void UploadRequest(string sToken, string UploadPath, string UploadKey, string
Compress, long lFileSize)
{
    String sRESTURL = "https://restapi.filesanywhere.com/FAWAPI.svc";

    string sURL =
    string.Format("{0}/UploadItem?UploadPath={1}&Token={2}&UploadKey={3}&Compress={4}&File
Size={5}", sRESTURL, UploadPath, sToken, UploadKey, Compress, lFileSize);
    HttpWebRequest req = (HttpWebRequest)HttpWebRequest.Create(sURL);
    req.Method = "POST";
    req.Method = "POST";
    req.SendChunked = true;
    req.AllowWriteStreamBuffering = false;
    req.ContentType = "application/octet-stream";
    req.Timeout = 7200000;
    try
    {
        using (Stream uploadStream = req.GetRequestStream())
        {
            long totalWritten = 0;
            while ((bytesRead = fileStream.Read(buffer, 0, buffer.Length)) > 0)
            {
                uploadStream.Write(buffer, 0, bytesRead);
                uploadStream.Flush();
            }
        }

        HttpWebResponse HttpResponse = (HttpWebResponse)req.GetResponse();

        if (HttpResponse.StatusCode == HttpStatusCode.OK)
        {
            txtUploadKey.Text = "File upload successfully"
        }
        else
        {
            txtOutput.Text = new
StreamReader(HttpResponse.GetResponseStream()).ReadToEnd();
        }
    }
    catch (Exception ex)
```



```

    {
        txtOutput.Text = ex.Message;
    }
}

```

Request Arguments

Request arguments should be passed as a JSON serialized string e.g.:

```

{"UploadPath":"\\PATNIDEV\\test.docx","Token":"A0FQCIZII6V0HDGPZGJOADBXEWOZJ
465IRVWW8VC1F5C","UploadKey":"A0FQCIZII6PGOM6ANJYSGVNRREE9M7M54AKYH3R
RVHL4GP9","Compress":"N","FileSize":"1456897"}

```

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
UploadPath	String	Yes	Full path of the file on FilesAnywhere in the form \UserVolume\FullPathWithFolderName\FileName
UploadKey	String	Yes	UploadKey which returned from 'PreUploadValidatoin' API, need to pass here, to validate upload process.
Compress	String	No	If file is sent after compression, then send 'Y' else send 'N' in this parameter. If the value is 'Y' then uploaded file will be expected as '.gz' file and our API will be decompress this file and upload file in user's directory.
FileSize	Long	Yes	File size in bytes which is getting uploaded. If it is compressed file then compressed file size needs to be pass.

Response

If the file uploaded successfully, the response will be a HttpStatus code 200 OK.

In case of validation error, it will return status code 406 (Not acceptable), and in case of warning exception it will return 417 (Expectation failed)

To get the error message, the stream needs to be read into a string object, as in the c# code below:

```

Error = new StreamReader(HttpResponse.GetResponseStream()).ReadToEnd();

```

ChunkUpload

This method upload a file in filesanywhere user account in chunk of 256kb. This method will be use for large size file upload (more than 256MB file size)

Syntax

HTTPs Request

POST: /ChunkUpload

HOST: <https://restapi.filesanywhere.com/FAWAPI.svc>

Filesanywhere-API-Arg :
{ "UploadPath": "", "Token": "", "UploadKey": "", "Compress": "", "FileSize": "" }

Invoke this Host URL, and post the argument UploadPath, Token, UploadKey, Compress and FileSize.

Usage

Use the ChunkUpload method start uploading a file in your filesanywhere account.

The following parameters are needed to make the ChunkUpload call:

- Token (The Token ID returned by the last AccountLogin call).
- Path (Name of the path where you want upload your files). The Path should be formatted like `\UserVolume\FullPathWithFolderName\FileName`, where FullPathWithFoldername is the name of the folder in which the file exists along with its full path and UserVolume is the FilesAnywhere UserName in whose account the file exists.
- UploadKey (Upload key which will be return from 'PreUploadValidation' method)
- Compress (User can pass the value as 'Y' or 'N' or ". If you want to upload a compressed file and then start to upload (to save the bandwidth), then you need to compressed file using `gzipStream`, and then start to upload ".gz" file, along with this pass 'Y' in this parameter. Once file upload completed, API method will decompress the file and upload it to your filesanywhere account.)
- FileSize (File size of a file)

Sample C#.Net Code

```
private void UploadRequest(string sToken, string UploadPath, string UploadKey, string
Compress, long FileSize)
{
    String sRESTURL = "https://restapi.filesanywhere.com/FAWAPI.svc";

    string sURL =
string.Format("{0}/ChunkUpload?UploadPath={1}&Token={2}&UploadKey={3}&Compress={4}&File
eSize={5}", sRESTURL, UploadPath, sToken, UploadKey, Compress, FileSize);
        string sLocalFile = "C:\Test.txt";
byte[] buffer = new byte[256 * 1024]; //256kb
int bytesRead = 0;
using (var fileStream = new FileStream(sLocalFile, FileMode.Open, FileAccess.Read))
{
    bytesRead = fileStream.Read(buffer, 0, buffer.Length);
    do
    {
        HttpWebRequest req = (HttpWebRequest)HttpWebRequest.Create(sURL);
        req.Method = "POST";
        req.SendChunked = true;
        req.AllowWriteStreamBuffering = false;
        req.ContentType = "application/octet-stream";
        req.Timeout = 1000000;
        using (Stream uploadStream = req.GetRequestStream())
        {
            uploadStream.Write(buffer, 0, bytesRead);
            uploadStream.Flush();
        }
        HttpResponseMessage HttpResponse = (HttpResponse)req.GetResponse();
        if (HttpResponse.StatusCode == HttpStatusCode.OK)
        {
            UploadedFileSize = new
StreamReader(HttpResponse.GetResponseStream()).ReadToEnd();
        }
        else
        {
            ErrorMessage = new StreamReader(HttpResponse.GetResponseStream()).ReadToEnd();
            Break;
        }
        bytesRead = fileStream.Read(buffer, 0, buffer.Length);
    } while (bytesRead > 0);
}
}
```

Request Arguments

Request arguments should be passed as a JSON serialized string e.g.:

```
{"UploadPath":"\\PATNIDEV\\test.docx","Token":"A0FQCIZII6V0HDGPZGJ0ADBXEWOZJ
465IRVWW8VC1F5C","UploadKey":"A0FQCIZII6PGOM6ANJYSGVNRREE9M7M54AKYH3R
RVHL4GP9","Compress":"N","FileSize":"10232234"}
```

Name	Data Type	Required	Description
------	-----------	----------	-------------

Token	String	Yes	The Token ID returned back in the last AccountLogin call
UploadPath	String	Yes	Full path of the file on FilesAnywhere in the form \UserVolume\FullPathWithFolderName\FileName
UploadKey	String	Yes	UploadKey which returned from 'PreUploadValidatoin' API, need to pass here, to validate upload process.
Compress	String	No	If file is sent after compression, then send 'Y' else send 'N' in this parameter. If the value is 'Y' then uploaded file will be expected as '.gz' file and our API will be decompress this file and upload file in user's directory.
FileSize	Long	Yes	Size of uploaded file.

Response

If the file uploaded successfully, the response will be a HttpStatus code 200 OK.

In case of validation error, it will return status code 406 (Not acceptable), and in case of warning exception it will return 417 (Expectation failed)

To get the error message, the stream needs to be read into a string object, as in the c# code below:

```
Error = new StreamReader(HttpResponse.GetResponseStream()).ReadToEnd();
```

ResumeUpload

This method resume chunk upload process, if it pause or interrupt by any reason.

Syntax

HTTPs Request

POST: /ResumeUpload

HOST: <https://restapi.filesanywhere.com/FAWAPI.svc>

Filesanywhere-API-Arg : {"UploadKey":"","Token":"","FileName":""}

Invoke this Host URL, and post the argument UploadKey, Token, FileName.

Usage

Use the ResumeUpload method to resume chunk upload process again.

The following parameters are needed to make the ResumeUpload call:

- Token (The Token ID returned by the last AccountLogin call).
- UploadKey (Upload key which will be return from 'PreUploadValidation' method)
- FileName (File name which needs to resume upload process again)

Sample C#.Net Code

```
private void ResumeUploadRequest(string sToken, string UploadPath, string UploadKey,
string Compress, long FileSize)
{
    String sRESTURL = "https://restapi.filesanywhere.com/FAWAPI.svc";

    string sURL = string.Format("{0}/ResumeUpload?UploadKey={1}&Token={2}&FileName={3}",
sWCFURL, txtUploadKey.Text, txtToken.Text, fileIn.Name);
    HttpWebRequest req = (HttpWebRequest)HttpWebRequest.Create(sURL);
    req.Method = "POST";
    req.ContentType = "application/xml";
    req.ContentLength = 0;
    req.Timeout = 1000000;
    long lUploadedLength = 0;
    try
    {
        HttpResponseMessage HttpResponseMessage = (HttpResponseMessage)req.GetResponse();
        if (HttpResponseMessage.StatusCode == HttpStatusCode.OK)
        {
            lUploadedLength = Convert.ToInt64(new
StreamReader(HttpResponseMessage.GetResponseStream()).ReadToEnd());
            DoChunkUpload(lUploadedLength);//Resume chunk upload from this
filesize length
        }
        else
        {
            ErrorMessage = new
StreamReader(HttpResponseMessage.GetResponseStream()).ReadToEnd().Replace("\'", "'");
        }
    }
    catch (Exception ex)
    {
        ErrorMessage = ex.Message;
    }
}
```

Request Arguments

Request arguments should be passed as a JSON serialized string e.g.:

```
{"Token":"A0FQCIZII6V0HDPZGJ0ADBXEWOZJ4651RVWW8VC1F5C","UploadKey":"A0
FQCIZII6PGOM6ANJYSGVNREE9M7M54AKYH3RRVHL4GP9","FileName":"Test.txt"}
```

Name	Data Type	Required	Description
Token	String	Yes	The Token ID returned back in the last AccountLogin call
UploadKey	String	Yes	UploadKey which returned from 'PreUploadValidatoin' API, need to pass here, to validate upload process.
FileName	String	Yes	Name of file which need to resume upload process again

Response

If the file is present at server, then it will return the uploaded file length of that file. consumer needs to read next chunk of a file after this returned file length and call chunkupload method.

In case of validation error, it will return status code 406 (Not acceptable), and in case of warning exception it will return 417 (Expectation failed)

To get the error message, the stream needs to be read into a string object, as in the c# code below:

```
Error = new StreamReader(HttpResponse.GetResponseStream()).ReadToEnd();
```

Introducing the FilesAnywhere Digitally Signed REST API

Digitally Signed REST API Service URL

<https://restapicert.filesanywhere.com/FAWAPI.svc>

It supports all operations which are already supported by FilesAnywhere REST API at URL

<https://restapi.filesanywhere.com/FAWAPI.svc>

Let's Get Started!

This section helps you in getting started with using the digitally signed REST API. It tells you what you need in order to start using the digitally signed REST API in your development environment. It includes the following steps:

- Sign up for a FilesAnywhere account using FAWAPI webservice.
- Use obtained token to call the REST API methods with required parameters.

Sign up for a FilesAnywhere account using FAWAPI

Using FAWAPI web service, invoke AccountLogin method, which will return Token number. Description of AccountLogin method can be found in below URL. <https://api.filesanywhere.com/v2/fawapi.asmx?op=AccountLogin>

Use this token to invoke all digitally signed REST API methods.

Prerequisites

Please refer support article <https://support.filesanywhere.com/hc/en-us/articles/115003773592> to know more about prerequisites required to use the digitally signed REST API in your API consumer application.

Note: Below sample code is given in the context of using DownloadFile API method. You may modify the sample code to use any other supported API method

Sample VB.Net Code

```
Imports System
Imports System.Collections.Generic
Imports System.IdentityModel.Selectors
Imports System.IO
Imports System.Linq
Imports System.Net
Imports System.Runtime.Serialization.Json
Imports System.Security.Cryptography.X509Certificates
Imports System.Text
Imports System.Threading.Tasks
```

```
Namespace RestAPIClient
```

```
Class RestAPIClient
```

```

Private Shared Sub DownloadRequest(ByVal sClientID As String, ByVal objClientCertificate As X509Certificate2,
ByVal sToken As String, ByVal sFileFullPath As String, ByVal sDestinationFolder As String)
    Dim HttpResponse As HttpWebResponse = Nothing
    Dim sServiceURL As String = "https://restapicert.filesanywhere.com/FAWAPI.svc/DownloadFile"
    Dim objDownloadRequest As DownloadRequest = New DownloadRequest With {.Token = sToken, .ItemPath =
sFileFullPath}
    Dim objWebRequest = CType(WebRequest.Create(sServiceURL), HttpWebRequest)
    objWebRequest.ContentType = "text/json"
    objWebRequest.Method = "POST"
    objWebRequest.Headers.Add("MessageIdentifier", sClientID)
    System.Net.ServicePointManager.ServerCertificateValidationCallback = New
System.Net.Security.RemoteCertificateValidationCallback(AddressOf ValidateRestAPICertificate)
    objWebRequest.ClientCertificates.Add(objClientCertificate)
    Dim objDataContractJsonSerializer As DataContractJsonSerializer = New
DataContractJsonSerializer(GetType(DownloadRequest))
    If objWebRequest IsNot Nothing Then
        Using objStreamWriter = New StreamWriter(objWebRequest.GetRequestStream())
            Dim objMemoryStream As MemoryStream = New MemoryStream()
            objDataContractJsonSerializer.WriteObject(objMemoryStream, objDownloadRequest)
            Dim sData As String = Encoding.UTF8.GetString(objMemoryStream.ToArray(), 0,
CInt(objMemoryStream.Length))
            objStreamWriter.Write(sData)
        End Using
    End If

    Dim objHttpWebResponse As HttpWebResponse = Nothing
    Try
        objHttpWebResponse = GetHttpResponse(objWebRequest)
        System.Threading.Thread.Sleep(500)
        If objHttpWebResponse.StatusCode = HttpStatusCode.OK Then
            Dim objServerIdentity = HttpResponse.Headers("ServerIdentity")
            If objServerIdentity IsNot Nothing Then
                ValidateServerCertificate(New X509Certificate2(Convert.FromBase64String(objServerIdentity)))
            End If

            Using iStream As Stream = objHttpWebResponse.GetResponseStream()
                Dim sDestPath As String = sDestinationFolder & Path.GetFileName(sFileFullPath)
                Dim destStream As FileStream = File.Create(sDestPath)
                iStream.CopyTo(destStream)
                destStream.Close()
                Console.WriteLine("File """" & Path.GetFileName(sFileFullPath) & """" successfully downloaded to " &
sDestinationFolder)
            End Using
        Else
            Console.WriteLine(objHttpWebResponse.StatusCode & ":" + New
StreamReader(objHttpWebResponse.GetResponseStream()).ReadToEnd().Replace("""", ""))
        End If
    Catch [error] As WebException
        objHttpWebResponse = CType([error].Response, HttpWebResponse)
    End Try
End Sub
    
```



```
Private Shared Function ValidateRestAPICertificate(ByVal sender As Object, ByVal ServerCertificate As
System.Security.Cryptography.X509Certificates.X509Certificate, ByVal chain As
System.Security.Cryptography.X509Certificates.X509Chain, ByVal sslPolicyErrors As
System.Net.Security.SslPolicyErrors) As Boolean
    Dim objX509Certificate2 As X509Certificate2 = CType(ServerCertificate, X509Certificate2)
    Return ValidateServerCertificate(objX509Certificate2)
End Function

Public Shared Function ValidateServerCertificate(ByVal objX509Certificate2 As X509Certificate2) As Boolean
    If objX509Certificate2 Is Nothing Then
        Throw New UnauthorizedAccessException("Response message does not meet security requirements.
Response message is not signed by server.")
    End If

    Dim dtCurrentDate As DateTime = DateTime.Now
    Dim bValid As Boolean = objX509Certificate2.Verify() AndAlso dtCurrentDate <=
objX509Certificate2.NotAfter.AddMinutes(5) AndAlso dtCurrentDate >=
objX509Certificate2.NotBefore.AddMinutes(-5)
    If Not bValid Then
        Throw New UnauthorizedAccessException("Message requirements could not be satisfied. Invalid server
certificate.")
    End If

    Dim objX509CertificateValidator As X509CertificateValidator = X509CertificateValidator.ChainTrust
    Try
        objX509CertificateValidator.Validate(objX509Certificate2)
    Catch ex As Exception
        Throw New UnauthorizedAccessException(ex.Message)
    End Try

    If objX509Certificate2.Subject <> "<FilesAnywhere Server Certificate Subject Name>" OrElse
String.IsNullOrEmpty(objX509Certificate2.Thumbprint) OrElse (Not
String.IsNullOrEmpty(objX509Certificate2.Thumbprint) AndAlso Not
objX509Certificate2.Thumbprint.Equals("<FilesAnywhere Server Certificate Thumbprint >",
StringComparison.OrdinalIgnoreCase)) Then
        Throw New UnauthorizedAccessException("Message requirements could not be satisfied. Invalid server
certificate.")
    End If

    Return True
End Function

Public Shared Function GetHttpResponse(ByVal request As WebRequest) As HttpWebResponse
    Try
        Return CType(request.GetResponse(), HttpWebResponse)
    Catch ex As WebException
        If ex.Response Is Nothing OrElse ex.Status <> WebExceptionStatus.ProtocolError Then
            Throw
        End If

        Return CType(ex.Response, HttpWebResponse)
    End Try
End Function
```

End Class

Public Class DownloadRequest

Public Property ItemPath As String

Public Property Token As String

End Class

End Namespace

Sample C#.Net Code

```
using System;
using System.Collections.Generic;
using System.IdentityModel.Selectors;
using System.IO;
using System.Linq;
using System.Net;
using System.Runtime.Serialization.Json;
using System.Security.Cryptography.X509Certificates;
using System.Text;
using System.Threading.Tasks;

namespace RestAPIClient
{
    class RestAPIClient
    {
        private static void DownloadRequest(string sClientID, X509Certificate2 objClientCertificate, string sToken, string
sFileFullPath, string sDestinationFolder)
        {
            HttpResponseMessage HttpResponse = null;
            string sServiceURL = @"https://restapicert.filesanywhere.com/FAWAPI.svc/DownloadFile";

            DownloadRequest objDownloadRequest = new DownloadRequest
            {
                Token = sToken,
                ItemPath = sFileFullPath
            };

            var objWebRequest = (HttpWebRequest)WebRequest.Create(sServiceURL);
            objWebRequest.ContentType = "text/json";
            objWebRequest.Method = "POST";

            //You must pass Client ID as MessageIdentifier header parameter
            objWebRequest.Headers.Add("MessageIdentifier", sClientID);

            // Following will validate the server certificate used for signing the service response
            System.Net.ServicePointManager.ServerCertificateValidationCallback = new
System.Net.Security.RemoteCertificateValidationCallback(ValidateRestAPICertificate);

            //Add client certificate to the API request
            objWebRequest.ClientCertificates.Add(objClientCertificate);
```

```

DataContractJsonSerializer objDataContractJsonSerializer = new
DataContractJsonSerializer(typeof(DownloadRequest));
if (objWebRequest != null)
{
    using (var objStreamWriter = new StreamWriter(objWebRequest.GetRequestStream()))
    {
        MemoryStream objMemoryStream = new MemoryStream();
        objDataContractJsonSerializer.WriteObject(objMemoryStream, objDownloadRequest);
        string sData = Encoding.UTF8.GetString(objMemoryStream.ToArray(), 0, (int)objMemoryStream.Length);
        objStreamWriter.Write(sData);
    }
}

HttpWebResponse objHttpWebResponse = null;
try
{
    objHttpWebResponse = GetHttpResponse(objWebRequest);
    System.Threading.Thread.Sleep(500); //Give some pause and display progress to user
    if (objHttpWebResponse.StatusCode == HttpStatusCode.OK)
    {
        //Service response also having the server certificate under response header parameter "ServerIdentity"
        which is used to sign the response; you can also validate it using following block of code
        var objServerIdentity = HttpResponse.Headers["ServerIdentity"];
        if (objServerIdentity != null)
        {
            ValidateServerCertificate(new X509Certificate2(Convert.FromBase64String(objServerIdentity)));
        }
        using (Stream iStream = objHttpWebResponse.GetResponseStream())
        {
            string sDestPath = sDestinationFolder + Path.GetFileName(sFileFullPath);
            FileStream destStream = File.Create(sDestPath);
            iStream.CopyTo(destStream);
            destStream.Close();
            Console.WriteLine("File \\" + Path.GetFileName(sFileFullPath) + "\" successfully downloaded to " +
sDestinationFolder);
        }
    }
    else
    {
        Console.WriteLine(objHttpWebResponse.StatusCode + ":" + new
StreamReader(objHttpWebResponse.GetResponseStream()).ReadToEnd().Replace("\\", ""));
    }
}
catch (WebException error)
{
    objHttpWebResponse = (HttpWebResponse)error.Response;
}
}

private static bool ValidateRestAPICertificate(object sender,
System.Security.Cryptography.X509Certificates.X509Certificate ServerCertificate,
System.Security.Cryptography.X509Certificates.X509Chain chain, System.Net.Security.SslPolicyErrors sslPolicyErrors)

```

```

    {
        X509Certificate2 objX509Certificate2 = (X509Certificate2)ServerCertificate;
        return ValidateServerCertificate(objX509Certificate2);
    }
    public static bool ValidateServerCertificate(X509Certificate2 objX509Certificate2)
    {
        if (objX509Certificate2 == null)
        {
            throw new UnauthorizedAccessException("Response message does not meet security requirements. Response message is not signed by server.");
        }
        DateTime dtCurrentDate = DateTime.Now;
        bool bValid = objX509Certificate2.Verify() && dtCurrentDate <= objX509Certificate2.NotAfter.AddMinutes(5) && dtCurrentDate >= objX509Certificate2.NotBefore.AddMinutes(-5);
        if (!bValid)
        {
            throw new UnauthorizedAccessException("Message requirements could not be satisfied. Invalid server certificate.");
        }

        X509CertificateValidator objX509CertificateValidator = X509CertificateValidator.ChainTrust;
        try
        {
            objX509CertificateValidator.Validate(objX509Certificate2);
        }
        catch (Exception ex)
        {
            throw new UnauthorizedAccessException(ex.Message);
        }

        //Throw error if invalid certificate is used while signing or certificate is expired
        if (objX509Certificate2.Subject != "<FilesAnywhere Server Certificate Subject Name>" ||
        String.IsNullOrEmpty(objX509Certificate2.Thumbprint) ||
        (!String.IsNullOrEmpty(objX509Certificate2.Thumbprint) &&
        !objX509Certificate2.Thumbprint.Equals("<FilesAnywhere Server Certificate Thumbprint >",
        StringComparison.OrdinalIgnoreCase)))
        {
            throw new UnauthorizedAccessException("Message requirements could not be satisfied. Invalid server certificate.");
        }
        return true;
    }
    public static HttpResponseMessage GetHttpResponse(WebRequest request)
    {
        try
        {
            return (HttpResponseMessage)request.GetResponse();
        }
        catch (WebException ex)
        {
            if (ex.Response == null || ex.Status != WebExceptionStatus.ProtocolError)
            {
                throw;
            }
        }
    }

```

```
    }  
    return (HttpWebResponse)ex.Response;  
  }  
}  
public class DownloadRequest  
{  
  public string ItemPath { get; set; }  
  public string Token { get; set; }  
}  
}
```

API Revision History

Version 1.3

JANUARY 2008

The following new methods were introduced to the API:

- GroupShareAdd
- UserAdd

Version 1.4

FEBRUARY 2008

The following new methods were introduced to the API:

- SendItemsELinkWithDefaults

- FileProperties

Version 1.5

March 2008

The following new methods were introduced to the API:

- AccountLogin

This new method is intended to replace the “Login” method which will me expired on 15th April 2008.

Version 1.6

May 2008

The AccountLogin method documentation was updated.

Version 1.7

June 2008

The following new methods were introduced to the API:

- ConvertImage
- ConvertImageStatus

December 2008

A revision made to the AccountLogin method providing an additional code sample for detecting users of all types (WebBasic, WebAdvanced, Private Site). This update also clarifies when and how the user type should be detected. This was a documentation change only and does not affect any API functionality.

Version 1.8

April 2009

AccountCreateFree Method was introduced.

Version 1.9

June 2009

The following new methods were introduced to the API:

- FileShareDownloadFiles
- FileShareGetHeaderInfo
- FileShareGetItems

Version 2.0

August 2009

The following new methods were introduced to the API:

- FileShareAppendComments
- FileShareIndexedItemProperties
- FileShareItemImage
- FileShareItemThumbnail
- FileShareRequestDocIndexing

Version 2.1

January 2013

The following new methods were introduced to the API:

- ViewFile
- ItemDescriptionAdd
- ConvertDocToPDF

- **ConvertDocToPDFStatus**
- **ConvertImageToPDF**
- **ItemDescription**
- **ItemCommentAdd**
- **ItemCommentUpdate**
- **ItemCommentRemove**
- **ItemCommentsList**

Version 2.2

April 2013

The following new methods were introduced to the API:

- AccountCreateFreeFromPromotion
- APNRegisterDevice
- APNUnregisterDevice
- APNGetNotificationProperties
- APNSendNotification
- DownloadZipFile
- DownloadWithCheckout
- EditFileWithZoho
- ESendFiles
- GetAccountDetails
- GetAPNSetting
- GetAPNSettings
- GetDropBoxDefaults2
- GetELinkDefaults2
- GroupShareAdd2
- GroupShareRemove
- ItemComments
- ListLogActivities
- MoveItems
- SearchFiles
- SendItemsELink2
- SetAccountDetails

- SetAPNSSetting
- SetAPNSettings
- SetDropBoxDefaults
- SetELinkDefaults
- SetFolderProperties
- StopSharingGroupShare
- UndoCheckout
- UploadWithCheckin
- UserLookup

Version 2.3

May 2013

The following new methods were introduced to the API:

- GetMediaStream

Version 2.4

August 2013

The following new methods were introduced to the API:

- AccountLogin2
- GetProfileImage
- GetGroupShareUserProfileImage
- GetID3Tag

Version 2.5

September 2013

The following new methods were introduced to the API:

- GetImageStream

Version 2.6

September 2013

The following new methods were introduced to the API:

- `GetTiffPageCount`

Version 2.7

December 2013

The following new methods were introduced to the API:

- `GetAPIURL`
- `GetStorageLoacation`

Version 2.8

December 2013

The following new methods were introduced to the API:

- `GetFolderAccessKey`
- `UploadFileWithFolderAccessKey`

Version 2.9

January 2014

The following new methods were introduced to the API:

- `ItemCommentsWithThumbnail`

Version 2.10

April 2014

The following new methods were introduced to the API:

- `SSOLoginURL`
- `SSOLoginWebURL`
- `SAMLAccountLogin`

Version 2.11

September 2014

The following new methods were introduced to the API:

- GoogleOAuthAccountLogin
- LDAPAccountLogin

Version 2.12

October 2014

The following new methods were introduced to the API:

- AppendChunkWithFolderAccessKey
- ChunkUpload

Version 2.13

November 2014

The following new methods were introduced to the API:

- GetClientID
- GetCompanyCode

Version 2.14

April 2016

The following new methods were introduced to the API:

- UpdateUserProfile
- GetClientDetails
- SetProfileImage
- SendFileShareLink

Version 2.15

September 2016

New REST API service were introduced.

The following methods were introduced to this API:

- DownloadFile
- ZipDownloadFileOrFolder
- DownloadWithCheckout
- DownloadFromFileshare

Version 2.16

February 2018

New Digitally Signed FilesAnywhere API and REST API were introduced.

Version 2.17

May 2019

The following new method was introduced to the API:

- GetFaxStatus

Version 2.18

June 2019

The following new method was introduced to the API:

- FileAdvancedProperties

Version 2.19

August 2019

The following new method was introduced to the REST API:

- DownloadFormData

The following methods were expired from the API:

- AppendComment
- ListItems
- ListUserVolumes
- Login

- SendItemsELink
- SendItemsELink2
- SendItemsELink3

Version 2.20

The following new method was introduced to the REST API:

- PreUploadValidation
- UploadItem
- ChunkUpload
- ResumeUpload

Version 2.21

The following new method was introduced to the API:

- AdminCreateFolder
- AdminCreateGroupShare

Version 2.22

The following new method was introduced to the API:

- AdminUserAdd

Version 2.23

The following new method was introduced to the API:

- UpdateFolderProperties

The following methods were expired from the API:

- UserAdd
- SetFolderProperties
- UploadFile
- ChunkUpload

Version 2.24

The following new method was introduced to the API:

- FileAgingRule
- FileAgingRuleRemove

Version 2.25

The following method was updated in API:

- GetAccountDetails

Version 2.26

The following method was updated in API:

- SendFileShareLink
- GetELinkDefaults3

Version 2.27

The following method was updated in API:

- SendFileShareLink
- SendItemsELinkWithDefaults

Version 2.28

The following method was introduced in API:

- SendFileShareLinkWithDefaults