

FTP Attachments™ API Guide

Overview

This document provides information about FTP Attachments™ API provided by Enzigma™. It also describes how to use these functions and sample codes with them.

FTP Attachments™ API provides an easy access to the basic functionalities of FTP Attachments™ like Attach, Update, View and Delete. So that FTP Attachments™ can easily be integrated with any third party applications.

We assume that the person using this API has a basic knowledge of how to consume Web Services. Also should know how to use Salesforce and customize Salesforce objects and fields.

Table of Contents

Index	Page
1. How To Get Started	2
1.1. FTP Attachments™ Web Service WSDL file.....	2
1.2. FTP Attachments™ Web Service Prerequisites.....	2
2. Classes	3
2.1. FTPAttachments Class	3
2.2. Result Class.....	4
2.3. FTPAttachmentParent Class.....	4
3. Web Methods	5
3.1. Attach()	5
3.2. Update().....	6
3.3. Delete().....	7
3.4. DownloadLatest().....	7
3.5. DownloadVersion().....	8

1. How To Get Started

Now FTP Attachments™ Pro is available to you through web service so that you can integrate any other application to use full potential of FTP Attachments™.

This section explains how you can get started on using FTP Attachments™ web service.

Sample application which uses FTP Attachments™ API can be found at, <http://www.enigma.com/downloads/FTPAttachments-API-Sample.zip>

1.1. FTP Attachments™ Web Service WSDL file

As you may be aware FTP Attachments™ is a ASP .NET base application hence its web service also is a ASP .NET based web service. You can get the wsdl of the web service using URL,

<https://www.enigma.com/sforce/webservices/public/FTPAttachments/10.asmx?WSDL>

You can use any programming language which supports to consume ASP .NET based web services to use this web service.

Example:

If you are planning to use the FTP Attachments™ in your ASP .NET web application. All you have to do is add a web reference using WSDL you obtained from above Url. It will create wrapper classes for the web service which you can use to call the different web methods of the web service and get your application to attach files as FTP Attachments™.

There are some samples of code given for each web method in this documentation. You can write appropriate code for the language/platform you are using to consume the web service and attach the file.

Please note the web services currently is in beta stage, and we appreciate if you can provide us your feedback how we can improve our web services.

For larger file support you may have to modify web.config file of your FTP Attachments™ application which is installed on your Standalone server.

1.2. FTP Attachments™ Web Service Prerequisites

Before you can get FTP Attachments™ web service working you must have an Enigma account with valid FTP Attachments™ Pro subscription linked to your Salesforce account.

You must install the FTP Attachments™ application on your Standalone server and have it working. Please consume the web service from your own server by changing the Url of the web service pointing to your own server.

e.g. if you have installed the FTP Attachments™ on your own Standalone server at <http://yourdomain/sforce/> Url you should use to consume the web service will be, <http://yourdomain/sforce/webservices/public/FTPAttachments/10.asmx>.

Prerequisites:

- **Enigma account linked to your Salesforce account**
- **FTP Attachments™ Pro subscription**
- **Standalone server where FTP Attachments™ Pro can be installed and used.**

In case of any issues, bugs or feedback please feel free to contact us at support@enigma.com.

2. Classes

This section explains the various classes available in FTP Attachments™ Web Service. For performing various operations these classes are necessary, it is important to understand each class thoroughly to use the Web Service.

2.1. FTPAttachments Class

FTPAttachments class is used to pass the information of the attachment while attaching and updating attachment using FTP Attachments™.

Properties		
Name	Type	Description
Body	string	It should have Body/Contents of the file to be uploaded in base64 format in case of Attach and Update methods. In case of Download methods it contains downloaded file's Body/Contents.
FileName	string	Original file name which is to be uploaded. (It is used only in case of Attach method.)
Folder	string	Name of the folder where to store the uploaded attachment. (It is used only in case of Attach method.)
FTPAttachment Parent	FTPAttachm entParent []	This is array of the Parent objects to which FTP Attachment is attached to. A FTP Attachment can be assigned to multiple parent objects as per the requirement. In case of Download methods this property is filled with the Parent objects to which the attachment is attached to.

Comment	string	One can put any descriptive text for attachment in this property. In case of Attach and Update this is used to set the comment for the attachment. In case of Download this has the comment associated with the FTP Attachments™ being downloaded.
ServerProfile	string	Name of Server Profile used to store FTP Attachments™. (It is used only in case of Attach method.)
IsPublic	bool	Indicates whether attachment is public or not. Only public attachments can be viewed in Self Service Portal. (Behavior can be changed using Settings for Self Service Portal)
AttachmentID	string	ID of the existing FTP Attachment. In case of Update method call, this ID will be used to identify the attachment to update. In case of Download method it will contain the ID of the attachment being downloaded.
Version	int	Version of the attachment being downloaded. This property is used only with DownloadVersions method to download specific version of the file.
OwnerID	string	Owner ID of the FTP Attachment record which is created in Salesforce. If no owner is specified the current user becomes the owner automatically.

2.2. Result Class

This is the common result class returned by most of the web methods provided in FTP Attachments™ web service. Using result you can identify the result of the method.

Properties		
Name	Type	Description
ID	string	ID of the attachment which was attached in case of Attach. In case of Update method it returns ID of the FTP Attachment Version.
Success	bool	It is true in case of success and false in case of any error.
ErrorCode	enum	Error Code assigned to particular error.
ErrorMsg	string	Message describing the cause of the error.
FTPAttachments	FTPAttachments	Object of FTPAttachments Class. It is filled with appropriate FTP Attachment object in case of Download methods.

2.3. FTPAttachmentParent Class

This class defines the fields related to the parent object of the FTP Attachment.

Properties		
Name	Type	Description
ID	string	ID of the parent object for attachment (Object to which the attachment is attached)
FieldName	string	Type name of parent object (e.g. Case, Contact etc. In case of custom object it can be ENZ__Contact__c or Page__c that is actual API name of the custom object)

3. Web Methods

This section defines the various methods available in FTP Attachments™ web service.

Following are the methods available in this API:

Name	Description
Attach(FTPAttachments[])	Attaches one or more files to specified objects in Salesforce.
Update(FTPAttachments[])	Updates one or more existing FTP Attachment with the new file and comment. It also updates IsPublic flag.
Delete(string[])	Deletes one or more attachment with specified FTP Attachment IDs.
DownloadLatest(string[])	Downloads the latest versions of files with specified IDs.
DownloadVersion(FTPAttachment[])	Downloads files with specified IDs and Versions.

```
FTPAttachmentsWebService10 WebService10 = new
FTPAttachmentsWebService10();
```

Before you start using API functions in your application, you need to set your correct Salesforce SessionID and Url in the web service header. Refer example given below.

```
WebService10.WebServiceHeaderValue.SFSessionID =
Request.Params["SFSessionID"];
WebService10.WebServiceHeaderValue.SFUrl = Request.Params["SFUrl"];
```

3.1. Attach()

This function attaches the specified file/files to the respective objects in Salesforce.

This method takes FTPAttachments object's array as a parameter so that multiple files can be attached in single call. Each FTPAttachments object will contain information about file being attached and to which object it is being attached with other attachment information like Comment, IsPublic etc.

It returns array of Result object. There is one result object for each corresponding FTPAttachment object, giving result for each attachment. If the attachment was successful Result will contain ID of the attachment and Success will be true. In case of any error it returns Error code and the Error message.

Syntax:

```
FTPAttachments.FTPAttachments[] arrFTPAttachments = new
FTPAttachments.FTPAttachments[1];
```

Sample Code:

```
String strFileName = "";
if (FileUpload1.HasFile)
{
    strFileName = FileUpload1.FileName;
    strFileName = "C:\\Temp\\" + strFileName;
}
//Create array of parents.
```

```
FTPAttachmentParent[] arrParent = new FTPAttachmentParent[1];
FTPAttachmentParent parent = new FTPAttachmentParent();
parent.FieldName = Request["FieldName"];
parent.ID = Request["FieldValue"];
arrParent[0] = parent;
```

```
//Create and Fill data into array of FTPAttachments.
arrFTPAttachments[iIndex] = new FTPAttachments.FTPAttachments();
arrFTPAttachments[iIndex].Body =
System.Convert.ToBase64String(File.ReadAllBytes(strFilePath));
arrFTPAttachments[iIndex].Comment = edComment.Text;
```

```

arrFTPAttachments[jIndex].Folder = "";
arrFTPAttachments[jIndex].IsPublic = chkIsPublic.Checked;
arrFTPAttachments[jIndex].Parents = arrParent;
arrFTPAttachments[jIndex].FileName = strFileName;
arrFTPAttachments[jIndex].ServerProfile = "Default";
}
//Attach FTPAttachments.
Result[] m_arrResult = WebService10.Attach(arrFTPAttachments);
    
```

Arguments:

Name	Type	Description
arrFTPAttachments	FTPAttachment []	Array of one or more FTPAttachments objects to be updated.

3.2. Update()

This function updates one or more existing files to the respective objects in Salesforce. On updating particular attachment, next version of the file is created so that original and updated, both files are available.

This method takes FTPAttachments object's array as parameter so that multiple files can be updated in a single call. Each FTPAttachments object will contain information that is to be updated in particular attachment like Comment, IsPublic and Body.

It returns array of Result object. There is one result object for each corresponding FTPAttachments object, giving result for each attachment. If the attachment was successfully updated, Result will contain ID of the attachment and Success will be true. In case of any error it returns Error code and the Error message.

Syntax:

```

Result[] arrResult = WebService10.Update(FTPAttachment[]
arrFTPAttachments);
    
```

Sample Code:

```

String strFileName = "";
if (FileUpload.HasFile)
{
    strFileName = FileUpload.FileName;
    strFileName = "C:\\Temp\\" + strFileName;
}
FTPAttachmentParent[] arrParent = new FTPAttachmentParent[1];
FTPAttachmentParent parent = new FTPAttachmentParent();
parent.FieldName = Request["FieldName"];
parent.ID = Request["FieldValue"];
arrParent[0] = parent;

FTPAttachments.FTPAttachments[] arrFTPAttachments = new
FTPAttachments.FTPAttachments[1];
arrFTPAttachments[jIndex] = new FTPAttachments.FTPAttachments();
arrFTPAttachments[jIndex].AttachmentID = "a0470000007UqMk"
arrFTPAttachments[jIndex].Body =
System.Convert.ToBase64String(File.ReadAllBytes(strFilePath));
arrFTPAttachments[jIndex].Comment = edComment.Text;
arrFTPAttachments[jIndex].FileName = strFileName;
arrFTPAttachments[jIndex].IsPublic = chkIsPublic.Checked;
arrFTPAttachments[jIndex].ServerProfile = "Default";

string strUploadID = Guid.NewGuid().ToString();
WebService10.Url = WebService10.Url + "?UploadID="+strUploadID+"";
Result[] m_arrResult = WebService10.Update(arrFTPAttachments);
    
```

Note: You can specify any unique Id as an UploadID. UploadID is required to return progress of update action.

Arguments:

Name	Type	Description
arrFTPAttachments	FTPAttachments []	Array of one or more FTPAttachments to be updated.

3.3. Delete()

This function deletes one or more attachments with specified IDs.

This method takes AttachmentID's array object as a parameter in order to delete multiple attachments in single call and returns array of Result object.

There is one result object for each corresponding FTPAttachments object, giving result for each attachment. If the attachment was successfully deleted Result will contain ID of the attachment and Success will be true. In case of any error it returns Error code and the Error message.

Syntax:

```
Result[] arrResult = WebService10.Delete(string[] arrIDs);
```

Sample Code:

```
string[] arrIDs = new string[1];
arrIDs[0] = "a03700000051MFu";
Result[] arrResult = WebService10.Delete(arrIDs);
```

Arguments:

Name	Type	Description
arrIDs	String[]	Array of one or more IDs related with attachments to be deleted.

3.4. DownloadLatest()

Use this function to view attachments with latest version.

This method takes AttachmentID's array object as a parameter in order to download multiple attachments in single call.

It returns array of Result object. If the attachment/attachments was/were successfully downloaded, Result will contain ID of the attachment and Success will be true. In case of any error it returns Error code and the Error message.

Syntax:

```
Result[] arrResult = WebService10.DownloadLatest(string[] arrIDs);
```

Sample Code:

```
string[] arrID = new string[1];
arrID[0] = "a03700000051MFu";

Result[] arrResult = WebService10.DownloadLatest(arrID);
```

Note:

arrResult[jIndex].FTPAttachments.Body of the downloaded file is in the form of Base64 string.

Arguments:

Name	Type	Description
arrID	String[]	Array of one or more IDs related with attachments to be downloaded.

3.5. DownloadVersion()

This function allows viewing files with specified AttachmentID and Version.

This method takes FTPAttachments object's array as parameter so that multiple files can be downloaded in single call.

It returns array of Result object. If the attachment/attachments was/were successfully downloaded, Result will contain ID of the attachment and Success will be true. In case of any error it returns Error code and the Error message.

Syntax:

```
Result[] arrResult =
WebService10.DownloadVersion(FTPAttachment[] arrFTPAttachments);
```

Sample Code:

```
int version = 6;
FTPAttachments[] arrFTPAttachments = new FTPAttachments[1];
arrFTPAttachments[0] = new FTPAttachments();
arrFTPAttachments[0].AttachmentID = "a03700000051MFu";
arrFTPAttachments[0].Version = version;
```

```
Result[] arrResult = WebService10.DownloadVersions(arrFTPAttachments);
```

Arguments:

Name	Type	Description
arrFTPAttachments	FTPAttachment []	Array of one or more FTPAttachments to be updated.