Web Services Documentation

Revision Date: April 2016

The materials and sample code are provided only for the purpose of an existing or potential customer evaluating or implementing a programmatic integration with the SRFAX fax service.

CONTENTS

Overview

The SRFax Fax API an application programmer to integrate Fax capabilities into their application utilizing the SRFax API. The integration itself is done via web services using HTTPS POST operations.

POSTS are sent to the following URL:

https://www.srfax.com/SRF_SecWebSvc.php

This same URL is used for all operations and data can be sent as a **JSON encoded string or as a regular POST variable**. This documentation illustrates both required and optional variables that are sent to the web service in order to accomplish each operation. Responses from the SRFAX web service are also provided.

There are nine (10) basic operations detailed below.

1. **Queue_Fax -** Schedules a fax to be sent with or without cover page.
2. **Get_FaxStatus –** Determines the status of a fax that has been scheduled for delivery.
3. **Get_MultiFaxStatus –** Determines the status of a multiple faxes that have been scheduled for delivery.
4. **Get_Fax_Inbox -** Returns a list of faxes received for a specified period of time
5. **Get_Fax_Outbox -** Returns a list of faxes sent for a specified period of time
6. **Retrieve_Fax –** Returns a specified sent or received fax file in PDF or TIFF format
7. **Update_Viewed_Status –** Mark an inbound or outbound fax as read or unread.
8. **Delete_Fax -** Deletes specified received or sent faxes
9. **Stop_Fax -** Deletes a specified queued fax which has not been processed
10. **Get_Usage –** Usage report for a specified user and period.

SRFax is also willing to develop additional web services if requested for specific client requirements. Please contact support@srfax.com for making additional requests.

Queue_Fax

POST Variables

Part Name	Type	Description
action	*Required*	Must be "Queue_Fax"
access_id:	*Required*	User Number
access_pwd:	*Required*	Password on the user's account
sCallerID:	*Required*	Senders Fax Number (must be 10 digits)
sSenderEmail	*Required*	Senders Email Address
sFaxType:	*Required*	"SINGLE" or "BROADCAST"
sToFaxNumber:	*Required*	Required – 11 digit number or up to 50 x 11 digit fax numbers separated by a "|" (pipe).
sResponseFormat	*Optional*	"XML" or "JSON" – Default is JSON
sAccountCode:	*Optional*	Optional – internal reference number (Max of 20 Characters)
sRetries:	*Optional*	Number of times the system is to retry a number if busy or an error is encountered – number from 0 to 6.
sCoverPage	*Optional*	If you want to use one of the cover pages on file, specify the cover page you wish to use "Basic", "Standard" , "Company" or "Personal". If a cover page is not provided then all cover page variable will be ignored. **NOTE:** If the default cover page on the account is set to "Attachments ONLY" the cover page will NOT be created irrespective of this variable.
sFaxFromHeader	*Optional*	From: On the Fax Header Line (max 30 Char)
sCPFromName	*Optional*	Sender's name on the Cover Page
sCPToName	*Optional*	Recipient's name on the Cover Page
sCPOrganization	*Optional*	Organization on the Cover Page
sCPSubject	*Optional*	Subject line on the Cover Page**
sCPComments	*Optional*	Comments placed in the body of the Cover Page
sFileName_x*	*Optional*	Valid File Name – See Supported File Types below
sFileContent_x*	*Optional*	Base64 encoding of file contents.
sNotifyURL	*Optional*	Provide an absolute URL (beginning with http:// or https://) and the SRFax system will POST back the fax status record when the fax completes. See the 'NOTIFY URL POST' section below for details of what is posted.
sQueueFaxDate	*Optional*	The date you want to schedule a future fax for. Must be in the format YYYY-MM-DD. Required if using sQueueFaxTime
sQueueFaxTime	*Optional*	The time you want to schedule a future fax for. Must be in the format HH:MM, using 24 hour time (ie, 00:00 – 23:59). Required if using sQueueFaxTime

Notes:

- *You are able to send an unlimited number of files with their content – simply replace the "x" with the number order you wish the files to be faxed in. Please ensure that the sFileName_1 has content in sFileContent_1 etc.
- **The subject line details are saved in the fax record even is a cover page is not requested – so the subject can be used for filtering / searching
- You are able to send a fax with just the cover page details.
- sQueueFaxDate/sQueueFaxTime must be for a future time. The timezone set on the account will be used when scheduling.

Returned Variables (JSON or XML Encoded)

Status:	string	"Success" or "Failed"
Result:	string	Queued Fax ID (FaxDetailsID) or Reason for failure

Supported File Types

We support in excess of 156 different file types including .zip, but a summarized list of supported file types is available on our website at https://www.srfax.com/faqs If you wish to know about other file types, please contact customer service.

NOTIFY URL POST Response

If the *sNotifyURL* is provided in the initial *Queue_Fax* call, the same variables returned from *the Get_Fax_Status* will be sent the URL you provide as POST variables. At any time you are still able to poll the fax status by calling *Get_Fax_Status* if the *sNotifyURL* is missed for any reason. The variables sent are as follows:

Part Name	Description
FaxDetailsID	Fax Job ID
FileName	Name of the fax file
SentStatus	Status of the fax
AccountCode	Account Code provided when fax was scheduled
DateQueued	Date the fax was queued for delivery
DateSent	Date the fax was sent
ToFaxNumber	Recipient Fax Number
Pages	Total number of pages
Duration	Call length for transmission
RemoteID	Remote Fax ID
ErrorCode	Error message in the event of a failed fax
Size	Size of the file

NOTE:
You have to setup the URL provided in the sNotifyURL so that it can handle a POST of the variables listed above.

Get_FaxStatus

POST Variables

Part Name	Type	Description
action	*Required*	Must be "Get_FaxStatus"
access_id:	*Required*	User Account Number
access_pwd:	*Required*	Password on the user's account
sFaxDetailsID:	*Required*	FaxDetailsID returned from Queue_Fax post.
sResponseFormat	*Optional*	"XML" or "JSON" – Default is JSON

Returned Variables (JSON or XML Encoded)

Status:	string	"Success" or "Failed"
Result:	string	Array of the fax properties as follows: array(FileName, SentStatus, DateQueued, DateSent, ToFaxNumber Pages, Duration, RemoteID, ErrorCode, Size)
		Note: If an error is found then the reason for failure will be in the Result string.

Possible Values for "SentStatus":

- In Progress
- Sent
- Failed
- Sending Email

Get_MultiFaxStatus

POST Variables

Part Name	Type	Description
action	*Required*	Must be "Get_FaxStatus"
access_id:	*Required*	User Account Number
access_pwd:	*Required*	Password on the user's account
sFaxDetailsID:	*Required*	Multiple FaxDetailsIDs can be requested by separating each FaxDetailsID with a pipe character "|".
sResponseFormat	*Optional*	"XML" or "JSON" – Default is JSON

Returned Variables (JSON or XML Encoded)

Status:	string	"Success" or "Failed"
Result:	string	Array of the fax properties as follows: Array[#]->array(FileName, SentStatus, DateQueued, DateSent, ToFaxNumber Pages, Duration, RemoteID, ErrorCode, Size)
		Note: If an error is found then the reason for failure will be in the Result string.

Possible Values for "SentStatus":

- In Progress
- Sent
- Failed
- Sending Email
- Unknown
  - If you receive a status of "Unknown", you will also see "FaxDetailsID Not Found". This means that the FaxDetailsID's provided where not valid. If you feel like you got this message in error, please contact Customer Support

Get_Fax_Inbox

POST Variables

Part Name	Type	Description
action	*Required*	Must be "Get_Fax_Inbox"
access_id:	*Required*	User Account Number

access_pwd:	*Required*	Password on the user's account
sResponseFormat	*Optional*	"XML" or "JSON" – Default is JSON
sPeriod:	*Optional*	"ALL" or "RANGE" – if not provided defaults to "ALL"
sStartDate:	*Optional*	Only required if "RANGE" specified in sPeriod – date format must be "YYYYMMDD"
sEndDate:	*Optional*	Only required if "RANGE" specified in sPeriod – date format must be "YYYYMMDD"
sViewedStatus	*Optional*	"UNREAD" only show faxes that have not been read. "READ" only show faxes that have been read. "ALL" show all faxes irrespective of Viewed Status. If one is not supplied the action will return ALL faxes.
sIncludeSubUsers	*Optional*	Set to "Y" if function is to include all faxes received by sub users of the account as well.

Returned Variables (JSON or XML Encoded)

Status:	string	"Success" or "Failed"
Result:	string	Array of the fax properties as follows: array(FileName, ReceiveStatus, Date, EpochTime, CallerID, RemoteID, Pages, Size, ViewedStatus) If sIncludeSubUsers is set to "Y" then two additional variables are returned in the array: User_ID User_FaxNumber
		Note: If an error is found then the reason for failure will be listed in the Result string.

Get_Fax_Outbox

POST Variables

Part Name	Type	Description
action	*Required*	"Get_Fax_Outbox"
access_id:	*Required*	User Account Number
access_pwd:	*Required*	Password on the user's account
sResponseFormat	*Optional*	"XML" or "JSON" – Default is JSON

sPeriod:	*Optional*	"ALL" or "RANGE" – if not provided defaults to "ALL"
sStartDate:	*Optional*	Only required if "RANGE" specified in sPeriod – date format must be "YYYYMMDD"
sEndDate:	*Optional*	Only required if "RANGE" specified in sPeriod – date format must be "YYYYMMDD"
sIncludeSubUsers	*Optional*	Set to "Y" if function is to include all faxes received by sub users of the account as well.

Returned Variables (JSON or XML Encoded)

Status:	string	"Success" or "Failed"
Result:	string	Array of the fax properties as follows: array(FileName, SentStatus, DateQueued, DateSent, EpochTime, ToFaxNumber Pages, Duration, RemoteID, ErrorCode, AccountCode, Subject, Size) If sIncludeSubUsers is set to "Y" then two additional variables are returned in the array: User_ID User_FaxNumber
		Note: If an error is found then the reason for failure will be in the Result string.

Retrieve_Fax

POST Variables

Part Name	Type	Description
action	*Required*	"Retrieve_Fax"
access_id:	*Required*	User Account Number
access_pwd:	*Required*	Password on the user's account
sFaxFileName:	*Required**	sFaxFileName returned from **Get_Fax_Inbox** or **Get_Fax_Outbox** post
sFaxDetailsID	*Required**	sFaxDetailsID of the fax – the ID is located after the "|" character of the sFaxFileName

sDirection:	*Required**	"IN" or "OUT" for inbound or outbound fax
sSubUserID	*Optional*	The account number of a sub account, if you want to use a master account to download a sub account's fax
sResponseFormat	*Optional*	"XML" or "JSON" – Default is JSON
sFaxFormat	*Optional*	"PDF" or "TIF" defaults to account setting if not supplied
sMarkasViewed	*Optional*	"Y" - mark fax as viewed once method completes successfully "N" or not provided: leave viewed status as is. Not

* Either the sFaxFileName or the sFaxDetailsID must be supplied.

Returned Variables (JSON or XML Encoded)

Status:	string	"Success" or "Failed"
Result:	string	Base64 encoded fax file contents. The file format will be in PDF or TIF format depending on format requested or the settings selected on the account.
		Note 2: If an error is found then the reason for failure will be in the Result string.

Update_Viewed_Status

POST Variables

Part Name	Type	Description
action	*Required*	"Update_Viewed_Status"
access_id:	*Required*	User Account Number
access_pwd:	*Required*	Password on the user's account
sFaxFileName:	*Required**	sFaxFileName returned from **Get_Fax_Inbox** or **Get_Fax_Outbox** post
sFaxDetailsID	*Required**	sFaxDetailsID of the fax – the ID is located after the "|" character of the sFaxFileName
sDirection:	*Required*	"IN" or "OUT" for inbound or outbound fax
sMarkasViewed	*Required*	"Y" - mark fax as READ "N" – mark fax as UNREAD
sResponseFormat	*Optional*	"XML" or "JSON" – Default is JSON

* Either the sFaxFileName or the sFaxDetailsID must be supplied.

Returned Variables (JSON or XML Encoded)

Status:	string	"Success" or "Failed"
Result:	string	Empty String
		Note 2: If an error is found then the reason for failure will be in the Result string.

Delete_Fax

POST Variables

Part Name	Type	Description
action	*Required*	"Delete_Fax"
access_id:	*Required*	User Account Number
access_pwd:	*Required*	Password on the user's account
sDirection:	*Required*	"IN" or "OUT" for inbound or outbound fax
sFaxFileName_x	*Required**	sFaxFileName returned from **Get_Fax_Inbox** or **Get_Fax_Outbox** post. You are able to provide an unlimited number of fax file names by replacing the "x" with a sequential number.
sFaxDetailsID_x	*Required**	sFaxDetailsID of the fax – the ID is located after the "|" character of the sFaxFileName. You are able to provide an unlimited number of fax ID's by replacing the "x" with a sequential number.
sResponseFormat	*Optional*	"XML" or "JSON" – Default is JSON

* Either the sFaxFileName_x or the sFaxDetailsID_x must be supplied.

Returned Variables (JSON or XML Encoded)

Status:	string	“Success” or “Failed”
Result:	string	Empty String on Success or the reason for failure

Stop_Fax

POST Variables

Part Name	Type	Description
action	*Required*	"Stop_Fax"
access_id:	*Required*	User Account Number
access_pwd:	*Required*	Password on the user's account
sFaxDetailsID:	*Required*	FaxDetailsID returned from Queue_Fax post.
sResponseFormat	*Optional*	"XML" or "JSON" – Default is JSON

Returned Variables (JSON or XML Encoded)

Status:	string	"Success" or "Failed"
Result:	string	Empty String on Success or the reason for failure

This function is used for removing a scheduled fax from the queue once it has been scheduled. Please note that depending on where the fax is in the process, it is possible to have some pages sent by the time the fax is stopped.

Possible results are:

Status = Success:
Result:

- "Fax cancelled but partially sent" – fax was successfully cancelled but whatever was in the fax buffer will have been sent.
- "Fax Cancelled" – the fax was successfully cancelled without any pages being sent.

Status = Failed:
Result:

- "Fax transmission completed" – the fax has been sent and the transaction is complete
- "Unable to Cancel Fax" – Fax in the process of conversion and cannot be cancelled at this time – you can try again in 10 seconds.

Get_Fax_Usage

POST Variables

Part Name	Type	Description
action	*Required*	"Get_Fax_Usage"
access_id:	*Required*	User Account Number
access_pwd:	*Required*	Password on the user's account
sResponseFormat	*Optional*	"XML" or "JSON" – Default is JSON
sPeriod:	*Optional*	"ALL" or "RANGE" – if not provided defaults to "ALL"
sStartDate:	*Optional*	Only required if "RANGE" specified in sPeriod – date format must be "YYYYMMDD"
sEndDate:	*Optional*	Only required if "RANGE" specified in sPeriod – date format must be "YYYYMMDD"
sIncludeSubUsers	*Optional*	Set to "Y" if function is to include all usage by sub users of the account

Returned Variables (JSON or XML Encoded)

Status:	string	"Success" or "Failed"
Result:	string	Array of the fax properties as follows: array(ClientID, Period, ClientName, SubUserID, BillingNumber, NumberOfFaxes, NumberOfPages)
		Note: If an error is found then the reason for failure will be in the Result string.

PHP Code Examples

Below are examples of 3 API functions (Queue_Fax, Get_FaxStatus, and Retrieve_Fax) using the srFax class available at :
https://www.srfax.com/SRFax_API_Class/getSRF_APIClass.php
Though not every API function is shown here, there is documentation on how to call every function found inside the class source code

Queue_Fax

```
<?php

/*
 * SRFax API Example - Queue a fax using the srFax Class
 */
```

```php
require_once ("srFax_class.php");

$accountID       = "12345";
$accountPassword  = "MyPassword123";

// instantiate object with Account ID and Password
$srFax = new srFax ( $accountID, $accountPassword );

//Get file contents for the fax
$file1Name     = "MyFax.txt";
$file1Contents = base64_encode(file_get_contents("Files/$file1Name"));

$file2Name     = "TestImage.jpg";
$file2Contents = base64_encode(file_get_contents("Files/$file2Name"));

//Setup required variables
$senderFaxNumber   = "5551234567";
$senderEmail       = "myEmail@example.com";
$receiverFaxNumber = "15551234567";

try {
   // attempt to queue a fax
   $srFax->Queue_Fax(array(
    'sCallerID'     => $senderFaxNumber,
    'sSenderEmail'   => $senderEmail,
    'sFaxType'      => 'SINGLE',
    'sToFaxNumber'   => $receiverFaxNumber,
    'sFileName_1'    => $file1Name,
    'sFileContent_1' => $file1Contents,
    'sFileName_2'    => $file2Name,
    'sFileContent_2' => $file2Contents,
   ));
}
catch (Exception $e) { // display error when exception is thrown
   die("Error: $e");
}


if ( $srFax->getRequestStatus () ) {

   echo "Success! Fax Details ID = " . $srFax->getRequestResponse ();

} else {
   echo "ERROR: " . $srFax->getRequestResponse ();
}

?>
```

Get_FaxStatus

```php
<?php

/*
 * SRFax API Example - Retrieve a fax's status using the srFax Class
 */

require_once ("srFax_class.php");

$accountID       = "12345";
$accountPassword  = "MyPassword123";

// instantiate object with Account ID and Password
$srFax = new srFax ( $accountID, $accountPassword );

try {

   $srFax->Get_FaxStatus(array(
    'sFaxDetailsID' => '12345678',
   ));
}
catch (Exception $e) { // display error when exception is thrown
   die("Error: $e");
}

if ( $srFax->getRequestStatus () ) {
   //Success! Output Fax Details
   $response = $srFax->getRequestResponse ();

   echo "Fax Details: <br/>\n";

   echo "FileName: " . $response->FileName . "<br/>\n";
   echo "Status: " . $response->SentStatus . "<br/>\n";
   echo "Date Queued: " . $response->DateQueued . "<br/>\n";
   echo "Date Sent: " . $response->DateSent . "<br/>\n";
   echo "Number of Pages: " . $response->Pages . "<br/>\n";
} else {
   echo "ERROR: " . $srFax->getRequestResponse ();
}

?>
```

Retrieve_Fax

```php
<?php

/*
 * SRFax API Example - Retrieve a fax using the srFax Class
```

```php
 */

require_once ("srFax_class.php");

$accountID       = "12345";
$accountPassword  = "MyPassword123";

// instantiate object with Account ID and Password
$srFax = new srFax ( $accountID, $accountPassword );

try {

   $srFax->Retrieve_Fax(array(
    'sFaxDetailsID' => '12345678,
    'sDirection'    => 'OUT', //retrieving an outbound fax
   ));
}
catch (Exception $e) { // display error when exception is thrown
   die("Error: $e");
}


if ( $srFax->getRequestStatus () ) {
   //Success! Save the Retrieved Fax to a file.

   $localPath     = "Files/";
   $localFileName = "retrievedFax.pdf";

   $srFax->saveLastResponseAsFile($localFileName, $localPath);

   echo "Success.  Fax saved to " . $localPath.$localFileName;

} else {
   echo "ERROR: " . $srFax->getRequestResponse ();
}

?>
```