

# M2M API v2.0

## Contents

- 1 General information
  - 1.1 C/C++
    - 1.1.1 How to generate client service code with gSOAP
  - 1.2 PHP
    - 1.2.1 Configuring client part
- 2 Services
  - 2.1 Authentication
    - 2.1.1 Methods
      - 2.1.1.1 authenticate
    - 2.1.2 Usage
  - 2.2 Customer
    - 2.2.1 Methods
      - 2.2.1.1 getContactBook
      - 2.2.1.2 getContactBookByIds
      - 2.2.1.3 getContactBookGroups
      - 2.2.1.4 getContactBookGroupName
      - 2.2.1.5 setContactBookGroupName
      - 2.2.1.6 deleteContact
      - 2.2.1.7 saveContact
      - 2.2.1.8 getFolders
      - 2.2.1.9 saveFolder
      - 2.2.1.10 getPreferences
      - 2.2.1.11 getNotifyPreferences
      - 2.2.1.12 getReceivePreferences
      - 2.2.1.13 getSubscription
      - 2.2.1.14 save
      - 2.2.1.15 savePreferences
      - 2.2.1.16 getSynchronizationLog
  - 2.3 Usage
  - 2.4 Fax
    - 2.4.1 Methods
      - 2.4.1.1 convertDocumentWithCoverPage
      - 2.4.1.2 convertDocument
      - 2.4.1.3 sendFax
      - 2.4.1.4 validate

- 2.4.1.5 getPDF
    - 2.4.1.6 getTIFF
    - 2.4.1.7 getPNG
    - 2.4.1.8 getZIP
    - 2.4.1.9 getSendHistory
    - 2.4.1.10 getCost
    - 2.4.1.11 getRecipients
    - 2.4.1.12 getStatus
    - 2.4.1.13 isRead
    - 2.4.1.14 getFax
    - 2.4.1.15 getFaxes
    - 2.4.1.16 delete
    - 2.4.1.17 setReadStatus
  - 2.4.2 Usage
- 2.5 Voice
  - 2.5.1 Methods
    - 2.5.1.1 getVoice
    - 2.5.1.2 delete
    - 2.5.1.3 setReadStatus
    - 2.5.1.4 getWAV
  - 2.5.2 Usage
- 2.6 Folder
  - 2.6.1 Methods
    - 2.6.1.1 getFaxes
    - 2.6.1.2 getSmses
    - 2.6.1.3 getVoices
  - 2.6.2 Usage
- 2.7 Sms
  - 2.7.1 Methods
    - 2.7.1.1 sendSms
    - 2.7.1.2 getCost
    - 2.7.1.3 getRecipients
    - 2.7.1.4 getContent
    - 2.7.1.5 getStatus
    - 2.7.1.6 isRead
    - 2.7.1.7 getSmsList
    - 2.7.1.8 getSms
    - 2.7.1.9 delete
    - 2.7.1.10 setReadStatus
  - 2.7.2 Usage
- 3 Types
  - 3.1 UsernameToken
    - 3.1.1 Methods
    - 3.1.2 Properties
  - 3.2 SOAPCustomer
    - 3.2.1 Methods

- 3.2.2 Properties
- 3.3 SOAPSubscription
  - 3.3.1 Methods
  - 3.3.2 Properties
- 3.4 SOAPContact
  - 3.4.1 Methods
  - 3.4.2 Properties
- 3.5 SOAPContactGroup
  - 3.5.1 Methods
  - 3.5.2 Properties
- 3.6 SOAPFolder
  - 3.6.1 Methods
  - 3.6.2 Properties
- 3.7 SOAPCustomerPreferences
  - 3.7.1 Methods
  - 3.7.2 Properties
- 3.8 SOAPNotifyPreference
  - 3.8.1 Methods
  - 3.8.2 Properties
- 3.9 SOAPReceivePreference
  - 3.9.1 Methods
  - 3.9.2 Properties
- 3.10 SOAPSynchronizationLog
  - 3.10.1 Methods
  - 3.10.2 Properties
- 3.11 SOAPVoice
  - 3.11.1 Methods
  - 3.11.2 Properties
- 3.12 SOAPSMSs
  - 3.12.1 Methods
  - 3.12.2 Properties
- 3.13 SOAPSMSsRecipient
  - 3.13.1 Methods
  - 3.13.2 Properties
- 3.14 SOAPFax
  - 3.14.1 Methods
  - 3.14.2 Properties
- 3.15 SOAPFaxRecipient
  - 3.15.1 Methods
  - 3.15.2 Properties
- 3.16 SOAPSendHistory
  - 3.16.1 Methods
  - 3.16.2 Properties
- 3.17 SOAPCountry
  - 3.17.1 Methods
  - 3.17.2 Properties

- 3.18 SOAPCurrency
  - 3.18.1 Methods
  - 3.18.2 Properties
- 3.19 SOAPFaxNumberPrefix
  - 3.19.1 Methods
  - 3.19.2 Properties
- 3.20 SOAPReceiveHistory
  - 3.20.1 Methods
  - 3.20.2 Properties
- 3.21 ArrayOfSOAPContact
- 3.22 ArrayOfSOAPContactGroup
- 3.23 ArrayOfSOAPFolder
- 3.24 ArrayOfSOAPFax
- 3.25 ArrayOfSOAPSmsHistory
- 3.26 ArrayOfSOAPFaxRecipient
- 3.27 ArrayOfSOAPSms
- 3.28 ArrayOfSOAPSmsRecipient
- 3.29 ArrayOfSOAPVoice
- 3.30 ArrayOfSOAPSynchronizationLog
- 3.31 ArrayOfSOAPCustomer
- 3.32 ArrayOfSOAPCountry
- 3.33 ArrayOfSOAPCurrency
- 3.34 ArrayOfSOAPFaxNumberPrefix
- 3.35 ArrayOfSOAPReceiveHistory
- 3.36 ArrayOfInteger
- 3.37 ArrayOfString
- 3.38 Enumerators
  - 3.38.1 SendFromEnum
- 4 Examples
  - 4.1 How to authenticate user
  - 4.2 How to convert documents
  - 4.3 How to send a fax
  - 4.4 How to send a sms
  - 4.5 How to fetch folders
  - 4.6 How to fetch folder contents
- 5 Error codes
  - 5.1 General Errors
  - 5.2 Miscellaneous Errors (1xx)
  - 5.3 Invalid identification number errors (2xx)
  - 5.4 Invalid object requests (3xx)
  - 5.5 Account Management Errors (4xx)
  - 5.6 Address book errors (6xx)
  - 5.7 Conversion Errors (7xx)

# General information

This document describes how developers can integrate fax management functionality into their applications by implementing new version of Popfax Machine-to-Machine (M2M) interface, using the SOAP protocol.

SOAP, originally defined as **Simple Object Access Protocol**, is a protocol specification for exchanging structured information in the implementation of Web Services in computer networks. It relies on Extensible Markup Language (XML) as its message format.

Popfax Machine-to-Machine supports sms, voice mail, fax sending and receiving functions for registered end users, moreover it offers a "Supra User" set of functions enabling third party integrators to administrate a group of end users accounts.

All the usages are provided by using the Popfax M2M v2.0 PHP client library. But you may use any code generator for using this service.

- C/C++: gsoap (<http://www.cs.fsu.edu/~engelen/soap.html>)
- PHP: nusoap (<http://sourceforge.net/projects/nusoap/>)

To subscribe for news and API documentation updates email to: **register [at] api.popfax.com**

To contact our developer support email to: **devsupport [at] api.popfax.com**

## C/C++

It is recommended to use gSoap to generate the client code.

### How to generate client service code with gSOAP

By default gSOAP generates a C++ proxy with STL:

```
-----  
$ wsdl2h -o [ServiceName].h https://api.popfax.com/api2/?service=[ServiceName]&wsdl=1  
-----
```

To build a pure C application, use option -c, option -o used to define the output file:

```
-----  
$ wsdl2h -c -o [ServiceName].h https://api.popfax.com/api2/?service=[ServiceName]&wsdl=1  
-----
```

To build without STL, use option -s:

```
-----  
$ wsdl2h -s -o [ServiceName].h https://api.popfax.com/api2/?service=[ServiceName]&wsdl=1  
-----
```

To generate client source code:

```
$ soapcpp2 -i -C -Iimport [ServiceName].h
```

Please consult the gSOAP documentation (<http://www.cs.fsu.edu/~engelen/soapdoc2.html>) for further instructions and documentation.

## PHP

It is recommended to use the PHP Popfax API client library.

### Configuring client part

If using the PHP Popfax API client v.2, to configure client part you need to setup several directories that contain the base classes for using the client api, the API\_DIR, INCLUDES\_DIR and CLIENT\_DIR.

```
define('API_DIR', dirname(__FILE__).DIRECTORY_SEPARATOR.'..'.DIRECTORY_SEPARATOR.'api');
define('INCLUDES_DIR', dirname(__FILE__).DIRECTORY_SEPARATOR.'..'.DIRECTORY_SEPARATOR.'includes');
define('CLIENT_DIR', API_DIR.DIRECTORY_SEPARATOR.'client');
```

Then you need to require the ClientManager class and tell client to load all classes.

```
require_once(CLIENT_DIR.'/ClientManager.class.php');
ClientManager::setServiceLocation('https://beta-api.dev-popfax.com/index.php');
ClientManager::loadClasses();
```

If using API functions, that require user authentication, you need to add a UsernameToken in the headers of the request, this is very easy if using PHP Popfax API client v.2.

```
$customer = ClientManager::getAuthClient()->authenticate($username, md5($password));
if($customer) {
    ClientManager::setUsernameToken($username, md5($password));
}
```

Then you are able to execute any user remote procedure calls.

If case if accessing the SupraUser functions, the procedure is the same, only you must you use the SuAuthService (SuAuthClient).

```
$supraUser = ClientManager::getSuAuthClient()->authenticate($username, md5($password));
if($supraUser) {
    ClientManager::setUsernameToken($username, md5($password));
}
```

# Services

## Authentication

Authentication service description may be accessed by navigating here (<https://api.popfax.com/api2/?service=Auth>) or if you use any code generators you may access a WSDL (<https://api.popfax.com/api2/?service=Auth&wsdl=1>) . The Authentication service provides remote methods for user authentication, all Customer related services are derived from it, thus supporting direct authentication.

## Methods

The following methods are supported by Authentication service.

### **authenticate**

Authenticates by the customer username/password.

- **return:** Soap compliant Customer object.
- **access:** public

*SOAPCustomer* **authenticate** (*string \$userName, string \$passwordHash*)

- *string \$userName:* Username that is authenticating.
- *string \$passwordHash:* MD5 password hash of the user.

## Usage

**PHP** (with Popfax API client):

```
ClientManager::getAuthClient()->authenticate($email, $password);
```

## Customer

Customer service description may be accessed by navigating here (<https://api.popfax.com/api2/?service=Customer>) or if you use any code generators you may access a WSDL (<https://api.popfax.com/api2/?service=Customer&wsdl=1>) . The Customer service provides remote methods for user interaction with contact book, folder structure and customer settings and preferences. It is derived from *Authentication* service.

## Methods

The following methods are supported by the Customer service.

### **getContactBook**

Returns contacts from customers contact book

- **return:** Soap compliant list of contact objects *ArrayOfSOAPContact*.
- **access:** public

*ArrayOfSOAPContact* **getContactBook** ()

### **getContactBookByIds**

Returns contacts from customers contact book by a list of contact identification numbers.

- **return:** Soap compliant list of contact objects *ArrayOfSOAPContact*.
- **access:** public

*ArrayOfSOAPContact* **getContactBookByIds** (*ArrayOfInteger \$contactIds*)

- *ArrayOfInteger \$contactIds*: List of contact identification numbers.

### **getContactBookGroups**

Returns contact groups from customers contact book.

- **return:** Soap compliant list of contact group objects *ArrayOfSOAPContactGroup*.
- **access:** public

*ArrayOfSOAPContactGroup* **getContactBookGroups** ()

### **getContactBookGroupName**

Returns contact groups from customers contact book.

- **return:** New or already existing group identification number.
- **access:** public

*int* **getContactBookGroupName** (*string \$groupName*)

- *string \$groupName*: Group name that already existis (or should be created).

### **setContactBookGroupName**

Renames the contact book group name and returns error code in case of error, in case of success OK.

- **return:** Error code, or OK.

- **access:** public

*int* **setContactBookGroupName** (*int* \$groupId, *string* \$groupName)

- *string* **\$groupId**: Group identification number, that should be renamed.
- *string* **\$groupName**: The name of the group, that is should be renamed to.

#### **deleteContact**

Removes a contact from users contact book.

- **return:** Error code, or OK.
- **access:** public

*int* **deleteContact** (*int* \$contactId)

- *string* **\$contactId**: Contact identification number, that should be removed.

#### **saveContact**

Saves modification to an existing contact book item, or creates a new one.

- **return:** Error code, or OK.
- **access:** public

*int* **saveContact** (*SOAPContact* \$contact)

- *SOAPContact* **\$contact**: Contact book item object that the changes should be saved.

#### **getFolders**

Returns folders from customers popfax box.

- **return:** Soap compliant list of customer folder objects  
*ArrayOfSOAPFolder*.
- **access:** public

*ArrayOfSOAPFolder* **getFolders** ()

#### **saveFolder**

Saves modification to an existing folder in popfax box, or creates a new one.

- **return:** Error code, or OK.
- **access:** public

*int* **saveFolder** (*SOAPFolder \$folder*)

- *SOAPFolder \$folder*: Folder in popfax box item object that the changes should be saved.

#### **getPreferences**

Returns Customer preferences. It may be accessed by the current customer, or the owner of the customer (SupraUser).

- **return**: Soap compliant object that contains *SOAPCustomerPreferences*.
- **access**: public

*SOAPCustomerPreferences* **getPreferences** (*int \$customerId*)

- *int \$customerId*: Customer id to get the preferences for, it should be the current user, or the user must be owned by the caller (SupraUser).

#### **getNotifyPreferences**

Returns Customer notification preferences. It may be accessed by the current customer, or the owner of the customer (SupraUser).

- **return**: Soap compliant object that contains *SOAPNotifyPreference*.
- **access**: public

*SOAPNotifyPreference* **getNotifyPreferences** (*int \$customerId*)

- *int \$customerId*: Customer id to get the notification preferences for, it should be the current user, or the user must be owned by the caller (SupraUser).

#### **getReceivePreferences**

Returns Customer receive preferences. It may be accessed by the current customer, or the owner of the customer (SupraUser).

- **return**: Soap compliant object that contains *SOAPReceivePreference*.
- **access**: public

*SOAPReceivePreference* **getReceivePreferences** (*int \$customerId*)

- *int \$customerId*: Customer id to get the receive preferences for, it should be the current user, or the user must be owned by the caller (SupraUser).

#### **getSubscription**

Returns Customer subscription information. It may be accessed by the

current customer, or the owner of the customer (SupraUser).

- **return:** Soap compliant object that contains *SOAPSubscription*.
- **access:** public

*SOAPSubscription* **getSubscription** (*int \$customerId*)

- *int \$customerId:* Customer id to get the subscription information for, it should be the current user, or the user must be owned by the caller (SupraUser).

**save**

Updates the customer information. Can be called by the customer, or the account owner (suprauser).

- **return:** Error code, or OK.
- **access:** public

*int* **save** (*SOAPCustomer \$customer*)

- *SOAPCustomer \$customer:* Customer information, the customer must exist.

**savePreferences**

Updates the customer preferences.

- **return:** Error code, or OK.
- **access:** public

*int* **savePreferences** (*SOAPCustomerPreferences \$customerPreferences*)

- *SOAPCustomerPreferences \$customerPreferences:* Customer preferences that are modified.

**getSynchronizationLog**

Returns a list of changes made to users account.

- **return:** a list of actions *ArrayOfSOAPSynchronizationLog*.
- **access:** public

*ArrayOfSOAPSynchronizationLog* **getSynchronizationLog** (*int \$timestamp*)

- *int \$timestamp:* The timestamp to get the changes from.

**Usage**

To get the contact book

```
$contacts = ClientManager::getCustomerClient()->getContactBook();
```

To get the contact book groups

```
$groups = ClientManager::getCustomerClient()->getContactBookGroups();
```

## Fax

Fax service description may be accessed by navigating here (<https://api.popfax.com/api2/?service=Fax>) or if you use any code generators you may access a WSDL (<https://api.popfax.com/api2/?service=Fax&wsdl=1>) . The Fax service provides remote methods for user interaction with fax messages, like sending fetching detailed information. It is derived from *Authentication* service.

## Methods

The following methods are supported by the Fax service.

### **convertDocumentWithCoverPage**

Converts the documents using the special cover page.

- **return:** The error code, or OK with number of pages ArrayOfInteger
- **access:** public

*ArrayOfInteger* **convertDocumentWithCoverPage** (*int \$method*, *string \$coverPageContent*, *ArrayOfString \$numbers*)

- *int \$method*: Conversion method. 1 - dithering, 2 - threshold
- *string \$coverPageContent*: Message to be printed on the cover page.
- *ArrayOfString \$numbers*: List of numbers that are supposed to be printed on the cover page.

### **convertDocument**

Converts multiple documents, and concatenates them in 1 resulting TIFF image.

- **return:** The error code, or OK with number of pages ArrayOfInteger
- **access:** public

*ArrayOfInteger* **convertDocument** (*int \$method*)

- *int \$method*: Conversion method. 1 - dithering, 2 - threshold

## sendFax

Sends the attached TIFF image to the request as fax message.

- **return:** The error code, or OK with full fax cost deduced from customers account ArrayOfInteger
- **access:** public

*ArrayOfInteger* **sendFax** (*ArrayOfString* **\$numbers**, *int* **\$pagesCount**, *string* **\$TXAttachType**, *string* **\$Tx\_report\_type**, *string* **\$faxHeader**, *int* **\$faxType**)

- *ArrayOfString* **\$numbers**: The list of numbers to send the fax message to.
- *int* **\$pagesCount**: Used to calculate the price of the outgoing fax, for security reasons, it is verified if the number is the same as in the tiff image.
- *string* **\$TXAttachType**: Sets if the user should get a sending report. Can have a value either **YES**, **NO**, **NOT\_AT\_ALL**. Default: **NO**.
- *string* **\$Tx\_report\_type**: Specifies the type of report in the body of sending mail notification. Can be one of the values **MULTI\_LINE** or **SINGLE\_LINE**. Default: **MULTI\_LINE**.
- *string* **\$faxHeader**: The text to be printed in the fax header.
- *int* **\$faxType**: The service used to send the fax (Web, API etc). Can be one of the values of *SendFromEnum* enumerator.

## validate

Validates the sending before the send.

- **return:** The error code, or OK.
- **access:** public

*int* **validate** (*ArrayOfString* **\$numbers**, *int* **\$pagesCount**)

- *ArrayOfString* **\$numbers**: The list of numbers to send the fax message to.
- *int* **\$pagesCount**: Used to calculate the price of the outgoing fax, for security reasons, it is verified if the number is the same as in the tiff image.

## getPDF

Returns fax attachment file name, together with an attachment in reply.

- **return:** The name of the attachment PDF file in the reply
- **access:** public

*string* **getPDF** (*int* **\$faxId**)

- *int **\$faxId***: Fax message identification number, that belongs to the customer that calls this function, to fetch the PDF for.

#### **getTIFF**

Returns fax attachment file name, together with an attachment in reply.

- **return**: The name of the attachment TIFF file in the reply
- **access**: public

*string* **getTIFF** (*int* **\$faxId**)

- *int **\$faxId***: Fax message identification number, that belongs to the customer that calls this function, to fetch the TIFF for.

#### **getPNG**

Returns fax attachment file name, together with an attachment in reply. Suitable for preview of faxes or PDA handhelds.

- **return**: The name of the attachment PNG file in the reply
- **access**: public

*string* **getPNG** (*int* **\$faxId**)

- *int **\$faxId***: Fax message identification number, that belongs to the customer that calls this function, to fetch the PNG for.

#### **getZIP**

Returns fax attachment file name, together with an attachment in reply. Suitable for cover page and registered (certified) fax messages.

- **return**: The name of the attachment ZIP file in the reply
- **access**: public

*string* **getZIP** (*int* **\$faxId**)

- *int **\$faxId***: Fax message identification number, that belongs to the customer that calls this function, to fetch the ZIP for.

#### **getSendHistory**

Returns the history of sending of the given fax.

- **return**: The fax delivery history `ArrayOfSOAPSendHistory`
- **access**: public

`ArrayOfSOAPSendHistory` **getSendHistory** (`ArrayOfInteger` **\$faxIds**);

- *ArrayOfInteger* **\$faxIds**: The list of fax message identification numbers.

#### **getCost**

Returns the cost of sent fax.

- **return**: The price of the faxmessage.
- **access**: public

*int* **getCost** (*int* **\$faxId**);

- *int* **\$faxId**: The fax message identification number.

#### **getRecipients**

Returns the list of recipients of the fax.

- **return**: The list of recipients *ArrayOfSOAPFaxRecipient*
- **access**: public

*ArrayOfSOAPFaxRecipient* **getRecipients** (*int* **\$faxId**);

- *int* **\$faxId**: The fax message identification number.

#### **getStatus**

Returns the status of the fax message.

- **return**: The fax message status
- **access**: public

*int* **getStatus** (*int* **\$faxId**);

- *int* **\$faxId**: The fax message identification number.

#### **isRead**

Returns the read status of the fax message.

- **return**: The fax message read status
- **access**: public

*int* **isRead** (*int* **\$faxId**);

- *int* **\$faxId**: The fax message identification number.

#### **getFax**

Returns the *SOAPFax* object by fax identification number.

- **return:** The fax object SOAPFax
- **access:** public

*int* **getFax** (*int* **\$faxId**);

- *int* **\$faxId**: The fax message identification number.

#### **getFaxes**

Returns a list of faxes.

- **return:** The fax message list ArrayOfSOAPFax
- **access:** public

*ArrayOfSOAPFax* **getFaxes** (*ArrayOfInteger* **\$faxIds**);

- *ArrayOfInteger* **\$faxIds**: The list of fax message identification numbers.

#### **delete**

Deletes a fax message from users popbox, and returns the status.

- **return:** Error code or OK.
- **access:** public

*int* **delete** (*int* **\$faxId**)

- *int* **\$faxId**: Fax identification number, that belongs to the customer that calls this function.

#### **setReadStatus**

Sets the read status of the fax message.

- **return:** Error code or OK.
- **access:** public

*int* **setReadStatus** (*int* **\$faxId**, *int* **\$readStatus**)

- *int* **\$faxId**: Fax identification number, that belongs to the customer that calls this function.
- *int* **\$readStatus**: The read bit to be set, 0 - not read, 1 - read.

#### **Usage**

#How to convert documents

#How to send a fax

## Voice

Voice service description may be accessed by navigating here (<https://api.popfax.com/api2/?service=Voice>) or if you use any code generators you may access a WSDL (<https://api.popfax.com/api2/?service=Voice&wsdl=1>) . The Customer service provides remote methods for user interaction with voice mail messages, fetching detailed information. It is derived from *Authentication* service.

### Methods

The following methods are supported by the Voice service.

#### **getVoice**

Returns all the information regarding the voice message.

- **return:** Soap compliant object that contains *SOAPVoice*.
- **access:** public

*SOAPVoice* **getVoice** (*int \$voiceId*)

- *int \$voiceId:* Voice identification number, that belongs to the customer that calls this function.

#### **delete**

Deletes a voice message from users popbox, and returns the status.

- **return:** Error code or OK.
- **access:** public

*int* **delete** (*int \$voiceId*)

- *int \$voiceId:* Voice identification number, that belongs to the customer that calls this function.

#### **setReadStatus**

Sets the read status of the voice mail message.

- **return:** Error code or OK.
- **access:** public

*int* **setReadStatus** (*int \$voiceId*, *int \$readStatus*)

- *int \$voiceId:* Voice identification number, that belongs to the customer that calls this function.
- *int \$readStatus:* The read bit to be set, 0 - not read, 1 - read.

## getWAV

Returns voice attachment file name, together with an attachment in reply.

- **return:** The name of the attachment WAV file in the reply
- **access:** public

*string* **getWAV** (*int* \$voiceId)

- *int* **\$voiceId:** Voice identification number, that belongs to the customer that calls this function.

## Usage

To delete a voice mail message from user account:

```
ClientManager::getVoiceClient()->delete($voiceId);
```

To get the voice mail message WAV file, and output to the client browser:

```
$result = ClientManager::getVoiceClient()->getWAV($voiceId);
if($result) {
    $attachments = ClientManager::getVoiceClient()->getAttachments();
    //it is always just 1 WAV file attached to the reply.
    $attachment = $attachments[0];
    header('Content-disposition: attachment; filename='.$attachment['filename']);
    header('Content-Type: '.$attachment['contenttype']);
    print($attachment['data']);
    exit();
}
```

How to set the voice mail message as read:

```
$result = ClientManager::getVoiceClient()->setReadStatus($voiceId, true);
//but if you want to mark the message as unread, set the second parameter as false.
//$result = ClientManager::getVoiceClient()->setReadStatus($voiceId, false);
```

## Folder

Folder service description may be accessed by navigating here (<https://api.popfax.com/api2/?service=Folder>) or if you use any code generators you may access a WSDL (<https://api.popfax.com/api2/?service=Folder&wsdl=1>) . The Customer service provides remote methods for user interaction with popfax box folders, that belong to the customer. It is derived from *Authentication* service.

## Methods

The following methods are supported by the Folder service.

## getFaxes

Returns the list of faxes in the folder.

- **return:** a list of actions *ArrayOfSOAPFax*.
- **access:** public

*ArrayOfSOAPFax* **getFaxes** (*int \$folderId*, *int \$offsetId*);

- *int \$folderId*: The folder identification number to get the faxes from.
- *int \$offsetId*: The offset fax identification number, to start the list from.

## getSmses

Returns the list of smses in the folder.

- **return:** a list of actions *ArrayOfSOAPSms*.
- **access:** public

*ArrayOfSOAPSms* **getSmses** (*int \$folderId*, *int \$offsetId*);

- *int \$folderId*: The folder identification number to get the smses from.
- *int \$offsetId*: The offset fax identification number, to start the list from.

## getVoices

Returns the list of voices in the folder.

- **return:** a list of actions *ArrayOfSOAPVoice*.
- **access:** public

*ArrayOfSOAPVoice* **getVoices** (*int \$folderId*, *int \$offsetId*);

- *int \$folderId*: The folder identification number to get the voice mail messages from.
- *int \$offsetId*: The offset fax identification number, to start the list from.

## Usage

#How to fetch folders

#How to fetch folder contents

## Sms

Sms service description may be accessed by navigating here (<https://api.popfax.com/api2/?service=Sms>) or if you use any code generators you may access a WSDL (<https://api.popfax.com/api2/?service=Sms&wsdl=1>) . The Customer service provides remote methods for user interaction with sms

messages, that belong to the customer, like fetching detailed information. It is derived from *Authentication* service.

## Methods

The following methods are supported by the Sms service.

### sendSms

Sends the sms message to the list of destination numbers.

- **return:** Error code if < 0, or if > 0, the new smsId.
- **access:** public

*int* **sendSms** (*ArrayOfString* **\$numbers**, *string* **\$message**, *int* **\$unicode**);

- *ArrayOfString* **\$numbers**: The list of destination numbers to send Sms to.
- *string* **\$message**: The sms message content.
- *int* **\$unicode**: Unicode bit, set if the message is in unicode encoding.

### getCost

Returns the cost of sent sms.

- **return:** The price of the sms message.
- **access:** public

*int* **getCost** (*int* **\$smsId**);

- *int* **\$smsId**: The sms message identification number.

### getRecipients

Returns the list of sms receivers (recipients).

- **return:** The list of recipients *ArrayOfSOAPSmsRecipient*
- **access:** public

*ArrayOfSOAPSmsRecipient* **getRecipients** (*int* **\$smsId**);

- *int* **\$smsId**: The sms message identification number.

### getContent

Returns sms content.

- **return:** The sms message content
- **access:** public

*string* **getContent** (*int* **\$smsId**);

- *int* **\$smsId**: The sms message identification number.

**getStatus**

Returns the status of the sent sms.

- **return**: The sms message status
- **access**: public

*int* **getStatus** (*int* **\$smsId**);

- *int* **\$smsId**: The sms message identification number.

**isRead**

Returns the read status of sent sms.

- **return**: The sms message read status
- **access**: public

*int* **isRead** (*int* **\$smsId**);

- *int* **\$smsId**: The sms message identification number.

**getSmsList**

Returns an array of SOAPSMS object by an array of their ids.

- **return**: The sms message list ArrayOfSOAPSMS
- **access**: public

*ArrayOfSOAPSMS* **getSmsList** (*ArrayOfInteger* **\$smsIds**);

- *ArrayOfInteger* **\$smsIds**: The list of sms message identification numbers.

**getSms**

Returns a sms detailed information by its identification number.

- **return**: The sms message detailed information SOAPSMS
- **access**: public

*SOAPSMS* **getSmsList** (*int* **\$smsId**);

- *int* **\$smsId**: The sms message identification number.

**delete**

Removes the sms from clients account.

- **return:** The error code, or OK.
- **access:** public

*int* **delete** (*int* **\$smsId**);

- *int* **\$smsId**: The sms message identification number.

### **setReadStatus**

Sets the read status of the sms message.

- **return:** The error code, or OK.
- **access:** public

*int* **setReadStatus** (*int* **\$smsId**, *int* **\$readStatus**);

- *int* **\$smsId**: The sms message identification number.
- *int* **\$readStatus**: The read status of the sms message to be set, 0 - not read, 1 - read.

### **Usage**

#How to send a sms

## **Types**

### **UsernameToken**

This type transmits authentication information about the user that wishes to do the request to the API server, in an additional secure manner.

### **Methods**

Constructor:

*UsernameToken* **UsernameToken**(*string* **\$userName**, *string* **\$passwordHash**);

- *string* **\$userName**: Username that is authenticating.
- *string* **\$passwordHash**: MD5 password hash of the user.

### **Properties**

*string* **\$Created**

- Timestamp of the request (when the username token was created, to

prevent man-in-the-middle attack)

*string* **\$PasswordHash**

- Hash code of the password (md5)

*string* **\$Username**

- Username of the user that does the request to the API.

## **SOAPCustomer**

This object contains all the information regarding the authenticated user.

### **Methods**

### **Properties**

- *int* **\$Id**
  - Customer identification number
- *string* **\$Email**
  - Customer login email
- *string* **\$Password**
  - Customer login password in MD5 hash, for security reasons the clear text password is not available.
- *string* **\$FirstName**
- *string* **\$LastName**
- *int* **\$FaxNumber**
  - Fax number identification number (present if customer is subscribed to Send&Receive package)

## **SOAPSubscription**

This object contains all the information regarding the customer subscription.

### **Methods**

### **Properties**

- *int* **\$Id**
  - Subscription identification number
- *int* **\$CustomerId**
  - The identification number of the customer that the subscription information belongs to.
- *int* **\$PackageId**
  - Package (product) identification number.
- *string* **\$StartDate**

- The start date, the beginning of subscription date.
- *string* **\$EndDate**
  - The end date of the subscription, may be extended by the customer or suprauser.
- *string* **\$OptionsEndDate**
  - The end date of the options enabled by the customer, may be extended by the customer or suprauser.
- *int* **\$PrepaidBalance**
  - The amount of PopUnits (Credits) in the users inbox (PopUnits \* 0.05) the amount in EURs.
- *int* **\$Type**
- *int* **\$Status**
  - The status of the subscription
- *string* **\$SelectedProducts**
  - Serialized array of a list of users products.
- *string* **\$LastSendedEmailDate**
- *string* **\$LastSendedOptionsEmailDate**
- *float* **\$IncomingTariff**
  - The percentage tariff of the incoming faxes (for customers, that receive a lot of faxes)
- *string* **\$RemoteIP**
  - Users IP address, that subscribed to popfax services.
- *string* **\$StandbyPromoCode**

## SOAPContact

This object contains information about a single contact book entry.

### Methods

### Properties

- *int* **\$Id**
  - Contact identification number
- *int* **\$CustomerId**
  - Owner identification number
- *int* **\$GroupId**
  - Group identification number, can be either 0 or **NULL**
- *string* **\$Name**
  - Contact name
- *string* **\$GsmNumber**
  - Contact gsm number (the number where it is possible to send an SMS)
- *string* **\$Number**
  - Contact fax number (the number where it is possible to send a FAX)

## SOAPContactGroup

This object contains information about a single contact book group entry.

## Methods

## Properties

- *int* **\$Id**
  - Contact identification number
- *int* **\$CustomerId**
  - Owner identification number
- *string* **\$GroupName**
  - Group name

## SOAPFolder

This object contains information about a single folder popfax box entry.

## Methods

## Properties

- *int* **\$Id**
  - Folder identification number
- *int* **\$ParentId**
  - Folder parent identification number
- *int* **\$CustomerId**
  - Owner identification number
- *string* **\$FolderName**
  - Folder name
- *string* **\$Description**
  - Description of the folder, for later reference
- *int* **\$Status**
  - Status of the folder, hidden or displayed (e.g. Draft is a hidden folder), enabled or disabled, depending of users options.

## SOAPCustomerPreferences

This object contains information about the customers preferences.

## Methods

- *public* **save()**
  - Saves the customers preferences.

## Properties

- *string* **\$FaxNumber**
  - Customer fax number, if the incoming number is enabled
- *int* **\$HasFaxHeader**
  - Denotes if the user has the possibility to change the fax header, depends on the supra user settings.
- *SOAPNotifyPreference* **\$NotifyPreferences**
  - Notification preferences of the customer (notification of the incoming SMS/FAX/Voice messages)
- *SOAPReceivePreference* **\$ReceivePreferences**
  - Received preferences of the customer (mail notification attachment types, etc.)

## SOAPNotifyPreference

This object contains information about the customers notification preferences.

### Methods

### Properties

- *int* **\$Id**
  - Notify preference identification number
- *int* **\$CustomerId**
  - Customer owner of the notification preference.
- *int* **\$NotifyByEmail**
  - Shows if the notification by email is enabled or disabled.
- *int* **\$NotifyBySms**
  - Shows if the notification by SMS is enabled or disabled.
- *string* **\$NotificationEmail**
  - The email address where the notifications should be sent.
- *string* **\$NotificationSms**
  - The gsm number where the messages should be sent, the account shall be charged accordingly per each SMS.
- *int* **\$VoiceMailNotifyByEmail**
  - Shows if the notification of the incoming voice mail message by email is enabled or disabled.
- *int* **\$VoiceMailNotifyBySms**
  - Shows if the notification of the incoming voice mail message by SMS is enabled or disabled.

## SOAPReceivePreference

This object contains information about the customers receive preferences.

### Methods

## Properties

- *int* **\$Id**
  - Notify preference identification number
- *int* **\$CustomerId**
  - Customer owner of the notification preference.
- *int* **\$RedirectToEmail**
  - Permanent redirection of incoming faxes to email enable or disable bit.
- *int* **VoiceMailRedirectToEmail**
  - Permanent redirection of incoming voice mail messages to email enable or disable bit.
- *int* **\$AttachmentId**
  - The type of attachments in faxes sent to email (TIFF or PDF).

## SOAPSynchronizationLog

This object contains the information about the change made to users account.

## Methods

## Properties

- *int* **\$Id**
  - Notify preference identification number
- *int* **\$CustomerId**
  - Customer owner of the notification preference.
- *int* **\$TypeId**
  - Identification number of the type that was changed.
- *int* **\$ActionId**
  - Identification number of type of action made on the account.
- *int* **\$ObjectId**
  - The identification number that was changed.
- *string* **\$Timestamp**
  - The timestamp when the change to users account was made.
- *string* **\$ActionName**
  - The name of the action that was made on users account one of (UPDATE, DELETE, INSERT)
- *string* **\$TypeName**
  - The name of the object type that was changed (FAX, SMS, VOICE)

## SOAPVoice

This object contains the information about the voice mail message.

## Methods

## Properties

- *int* **\$Id**
  - Voice mail message identification number
- *int* **\$CustomerId**
  - Customer owner of the notification preference.
- *int* **\$FolderId**
  - Folder identification number, in which the voice mail message is stored.
- *int* **\$Read**
  - Read bit, shows if the message was read (accessed) by the owner, or not.
- *string* **\$ShortDescription**
  - Short description of the message.
- *string* **\$Date**
  - The date of the voice mail message.
- *string* **\$Sender**
  - Sender of the voice mail information.
- *int* **\$Duration**
  - The duration of the voice mail message.
- *int* **\$ServerId**
  - The server identification number, that received the voice mail message.
- *float* **\$SmsPrice**
  - The amount of units charged for the SMS notification sent or arrival of the message.

## SOAPSms

This object contains the information about the sms message.

## Methods

## Properties

- *int* **\$Id**
  - Sms message identification number
- *int* **\$CustomerId**
  - Customer owner of the sms message.
- *int* **\$FolderId**
  - Folder identification number, in which the sms is stored.
- *int* **\$IsRead**
  - Read bit, shows if the message was read (accessed) by the owner, or not.
- *int* **\$Unicode**
  - Unicode bit, if the message was sent using unicode character set.
- *string* **\$Date**

- The date of the sms message.
- *int* **\$SendFrom**
  - Using which service the sms was sent.
- *string* **\$Content**
  - The content of the sms message.
- *string* **\$DeleteDate**
  - The date when the sms was deleted.
- *int* **\$Cost**
  - The cost of the sms.
- *string* **\$StatusCode**
  - The status code of the sms.
- *ArrayOfSOAPSmsRecipient* **\$Recipients**
  - The list of recipients of the sms.

## SOAPSmsRecipient

This object contains the information about the sms message.

### Methods

### Properties

- *int* **\$Id**
- *int* **\$SmsId**
- *string* **\$Recipient**
- *int* **\$Status**
- *int* **\$Price**

## SOAPFax

This object contains the information about the fax message.

### Methods

### Properties

- *int* **\$Id**
  - Fax identification number
- *int* **\$CustomerId**
  - Customer owner of the fax message.
- *int* **\$FolderId**
  - Folder identification number, in which the fax is stored.
- *int* **\$IsRead**
  - Read bit, shows if the message was read (accessed) by the owner, or not.
- *string* **\$Sender**

- The sender of the fax message information.
- *string* **\$Date**
  - The date of the fax message.
- *int* **\$SendFrom**
  - Using which service the fax was sent.
- *int* **\$PagesCount**
  - The number of pages in the fax message.
- *int* **\$ServerId**
  - The server identification number, the fax was received by.
- *int* **\$Cost**
  - The cost of the fax.
- *string* **\$StatusCode**
  - The status code of the fax.
- *ArrayOfString* **\$Recipients**
  - The list of recipients of the fax.

## SOAPFaxRecipient

This object contains the information about the fax message recipients.

### Methods

### Properties

- *int* **\$Id**
- *int* **\$FaxId**
- *string* **\$FaxNumber**
- *string* **\$Abonent**

## SOAPSendHistory

This object contains the information about the fax message delivery.

### Methods

### Properties

- *int* **\$Id**
  - Fax identification number
- *int* **\$CustomerId**
  - Customer owner of the fax message.
- *int* **\$FaxId**
- *string* **\$Date**
- *string* **\$FaxName**
- *string* **\$Recipients**
- *float* **\$Price**

- *int* **\$Status**
- *int* **\$PagesCount**
- *string* **\$StatusCode**

## **SOAPCountry**

This object contains the information about the country in popfax system.

### **Methods**

### **Properties**

- *int* **\$Used**
- *string* **\$Id**
- *string* **\$Name**
- *int* **\$PaymentRisk**
- *string* **\$Code3**
- *int* **\$InEU**
- *float* **\$VatAmount**
- *int* **\$TimezoneId**

## **SOAPCurrency**

This object contains the information about the currency in popfax system.

### **Methods**

### **Properties**

- *int* **\$CurId**
- *string* **\$Cur**
- *float* **\$ValueOfEuro**
- *string* **\$Type**
- *int* **\$Id**
- *string* **\$Country**
- *string* **\$CountryCode**
- *string* **\$CurCode**
- *int* **\$Visible**
- *int* **\$Od**
- *int* **\$DisplayPrecision**
- *int* **\$AssociatedCurrency**
- *int* **\$PrecisionCUV**
- *int* **\$PrecisionEXT**
- *int* **\$PrecisionFIX**
- *int* **\$PrecisionOTH**
- *int* **\$PrecisionPFP**

- *int* **\$PrecisionPPB**

## **SOAPFaxNumberPrefix**

This object contains the information about the fax number prefix.

### **Methods**

### **Properties**

- *int* **\$Id**
- *string* **\$FaxNumberPrefix**
- *string* **\$Destination**
- *string* **\$ParentId**
- *int* **\$OrderNo**
- *int* **\$Enabled**
- *string* **\$CountryId**
- *int* **\$WwpOrder**
- *float* **\$FreeThreshold**
- *int* **\$TelecomOperatorId**
- *int* **\$TimezoneId**
- *string* **\$CountryName**

## **SOAPReceiveHistory**

This object contains the detailed information about the received fax.

### **Methods**

### **Properties**

- *int* **\$Id**
- *int* **\$CustomerId**
- *int* **\$FaxId**
- *string* **\$Date**
- *string* **\$From**
- *int* **\$PagesCount**
- *float* **\$SmsPrice**
- *float* **\$IncomingPrice**
- *int* **\$Duration**
- *string* **\$Type**
- *string* **\$Remote**

## **ArrayOfSOAPContact**

This object represents a list of *SOAPContact* objects. It is possible to traverse the list in PHP

```
foreach($arrayOfSoapContact as $soapContact) {  
    print($soapContact->Name."\n");  
}
```

this code will display all contacts names in the users contact book.

## **ArrayOfSOAPContactGroup**

This object represents a list of *SOAPContactGroup* objects. It is possible to traverse the list in PHP

```
foreach($arrayOfSoapContactGroup as $soapContactGroup) {  
    print($soapContactGroup->GroupName."\n");  
}
```

this code will display all contact group names in the list.

## **ArrayOfSOAPFolder**

This object represents a list of *SOAPFolder* objects. It is possible to traverse the list in PHP

```
foreach($arrayOfSoapFolder as $soapFolder) {  
    print($soapContact->FolderName."\n");  
}
```

this code will display all folders names in the users popfax box.

## **ArrayOfSOAPFax**

This object represents a list of *SOAPFax* objects. It is possible to traverse the list in PHP

```
foreach($arrayOfSoapFax as $soapFax) {  
    print($soapFax->Id."\n");  
}
```

this code will display all fax identification numbers in the list.

## **ArrayOfSOAPSendHistory**

This object represents a list of *SOAPSendHistory* objects. It is possible to traverse the list in PHP

```
foreach($arrayOfSoapSendHistory as $soapSendHistory) {  
    print($soapSendHistory->Id."\n");  
}
```

this code will display all send history of fax delivery identification numbers in the list.

## **ArrayOfSOAPFaxRecipient**

This object represents a list of *SOAPFaxRecipient* objects. It is possible to traverse the list in PHP

```
foreach($arrayOfSoapFaxRecipient as $soapFaxRecipient) {  
    print($soapFaxRecipient->Id."\n");  
}
```

this code will display all fax recipients identification numbers in the list.

## **ArrayOfSOAPSms**

This object represents a list of *SOAPSms* objects. It is possible to traverse the list in PHP

```
foreach($arrayOfSoapSms as $soapSms) {  
    print($soapSms->Id."\n");  
}
```

this code will display all sms identification numbers in the list.

## **ArrayOfSOAPSmsRecipient**

This object represents a list of *SOAPSmsRecipient* objects. It is possible to traverse the list in PHP

```
foreach($arrayOfSoapSmsRecipient as $soapSmsRecipient) {  
    print($soapSmsRecipient->Id."\n");  
}
```

this code will display all sms recipients identification numbers in the list.

## **ArrayOfSOAPVoice**

This object represents a list of *SOAPVoice* objects. It is possible to traverse the list in PHP

```
foreach($arrayOfSoapVoice as $soapVoice) {  
    print($soapVoice->Id."\n");  
}
```

this code will display all voice mail message identification numbers in the list.

## **ArrayOfSOAPSynchronizationLog**

This object represents a list of *SOAPsynchronizationLog* objects. It is possible to traverse the list in PHP

```
foreach($arrayOfSoapSyncLog as $soapSyncLog) {  
    print($soapSyncLog->ActionName."\n");  
}
```

this code will display per each action made to users account it's name.

## **ArrayOfSOAPCustomer**

This object represents a list of *SOAPCustomer* objects. It is possible to traverse the list in PHP

```
foreach($arrayOfSoapCustomer as $soapCustomer) {  
    print($soapCustomer->Email."\n");  
}
```

this code will display for all customers in the list their login emails.

## **ArrayOfSOAPCountry**

This object represents a list of *SOAPCountry* objects. It is possible to traverse the list in PHP

```
foreach($arrayOfSoapCountry as $soapCountry) {  
    print($soapCountry->Id."\n");  
}
```

this code will display identification numbers for all countries in popfax system.

## **ArrayOfSOAPCurrency**

This object represents a list of *SOAPCurrency* objects. It is possible to traverse the list in PHP

```
foreach($arrayOfSoapCurrency as $soapCurrency) {  
    print($soapCurrency->Id."\n");  
}
```

this code will display identification numbers for all currencies in popfax system.

## **ArrayOfSOAPFaxNumberPrefix**

This object represents a list of *SOAPFaxNumberPrefix* objects. It is possible to traverse the list in PHP

```
foreach($arrayOfSoapFaxNumberPrefix as $soapFaxNumberPrefix) {  
    print($soapFaxNumberPrefix->Id."\n");  
}
```

this code will display identification numbers for all fax prefixes allowed to be used by supra user customers in popfax system.

## ArrayOfSOAPReceiveHistory

This object represents a list of *SOAPReceiveHistory* objects. It is possible to traverse the list in PHP

```
foreach($arrayOfSoapReceiveHistory as $soapReceiveHistory) {  
    print($soapReceiveHistory->Id."\n");  
}
```

this code will display identification numbers for all objects in the list.

## ArrayOfInteger

This object represents a list of *int* values. It is possible to traverse the list in PHP

```
foreach($arrayOfSoapInteger as $value) {  
    print($value."\n");  
}
```

this code will display all *int* values in the list.

## ArrayOfString

This object represents a list of *string* values. It is possible to traverse the list in PHP

```
foreach($arrayOfSoapString as $value) {  
    print($value."\n");  
}
```

this code will display all *string* values in the list.

## Enumerators

### SendFromEnum

Contains the constant values of fax type and service usage.

- *const SEND\_FROM\_WEB = 1 - Fax that was sent from web interface.*
- *const SEND\_FROM\_M2F = 2 - Fax that was sent using the mail2fax*

service.

- *const **SEND\_FROM\_M2M** = 3 - Fax that was sent using the machine-2-machine protocol.*
- *const **SEND\_FROM\_MAS** = 4 - Massive recipients fax. **\*DEPRECATED\****
- *const **SEND\_FROM\_FRF** = 5 - A fax message sent from freepopfax.*
- *const **SEND\_FROM\_PRN** = 6 - Fax that was sent using the printer driver.*
- *const **SEND\_FROM\_WFW** = 10 - Fax that was forwarded using the web interface.*
- *const **SEND\_FROM\_PDA** = 11 - Fax that was sent using the PDA site/application.*
- *const **SEND\_FROM\_CMP** = 12 - Fax that was sent using the Popcompanion client application.*
- *const **SEND\_FROM\_MMM** = 13 - Fax message that used the mailmerge service.*
- *const **SEND\_FROM\_WCP** = 14 - Fax with a coverpage sent from the web interface.*
- *const **SEND\_FROM\_MCP** = 15 - Fax with a coverpage sent through the API service.*
- *const **SEND\_FROM\_CRT** = 16 - Registered (certified fax).*

## Examples

### How to authenticate user

```
<?php
$email = 'someuser@somehost.com';
$password = md5('password');
$customer = ClientManager::getAuthClient()->authenticate($email, $password);
if($customer) {
    print('Welcome '.$customer->FirstName);
}
else {
    print('Authentication failure.');
```

### How to convert documents

First you need to authenticate the user described here.

```

// ... authenticate, and if success do: ...
// add attachments to request
ClientManager::getFaxClient()->clearAttachments();
foreach ($FILES as $key => $documentToConvert) {
    ClientManager::getFaxClient()->addAttachment(file_get_contents($documentToConvert['tmp_name']));
}
$result = NULL;
// If you wish to convert the document with a cover page, then you must use the following function
if($useCoverPage) {
    $result = ClientManager::getFaxClient()->convertDocumentWithCoverPage($method, $coverPageContent);
}
// If you do not want the cover page
else {
    $result = ClientManager::getFaxClient()->convertDocument($method);
}

//if the result is success then we have an attachment in our reply
//the result contains 2 integer values, the error code (OK) and the
//page count of the result document.
if(count($result) == 2) {
    $attachments = ClientManager::getFaxClient()->getAttachments();
    //the resulting reply should contain only 1 attachment
    $attachment = $attachments[0];
    header('Content-disposition: attachment; filename='.$attachment['filename']);
    header('Content-Type: '.$attachment['contenttype']);
    print($attachment['data']);
    exit();
}

```

## How to send a fax

First you need to authenticate the user described here. Then you must convert the documents to a single TIFF G3 Fax image, it is described here

```

<?php
//authenticate ...
//convert documents ...
//if successful do ...
//add the converted attachment to the request ...
ClientManager::getFaxClient()->clearAttachments();
ClientManager::getFaxClient()->addAttachment(file_get_contents($pathToTIFFDocument), $TIFFDocument);
//and send the fax
$result = ClientManager::getFaxClient()->sendFax($numbers, $pagesCount, $TXAttachType, $Tx_report);
//the result will contain the error code of the request
?>

```

## How to send a sms

First you need to authenticate the user described here. And the user must have the option enabled.

```

<?php
//authenticate ...
//check if the option is enabled ...
//if success do ...
$result = ClientManager::getSmsClient()->sendSms($numbers, $message, $unicode);
//the result will contain the identification number of the newly created sms message
//or the negative value of the error code.
?>

```

## How to fetch folders

First you need to authenticate the user described here.

```
<?php
//authenticate ...
//if success ...
$folders = ClientManager::getCustomerClient()->getFolders();
//the result will be a list of customer folders, in case of error it will be NULL.
?>
```

## How to fetch folder contents

First you need to authenticate the user described here. Then you need to fetch the available folders, that is described here.

```
<?php
//authenticate ...
//if success ...
$folders = ClientManager::getCustomerClient()->getFolders();
//the result will be a list of customer folders, in case of error it will be NULL.
foreach($folders as $folder) {
    $faxes = $folder->getFaxes();
    $smses = $folder->getSmses();
    $voices = $folder->getVoices();
}
//these requests will return the list of fax, sms and voice messages accordingly.
?>
```

## Error codes

### General Errors

```
const OK = 0;
const NOTHING_TO_DO = 1;
const NOT_IMPLEMENTED_YET = 2;
```

### Miscellaneous Errors (1xx)

```

;const INVALID_NUMBER = 101;
;const NO_FILE_ATTACHED = 102; // if no files sent during transfer action
;const NOT_ENOUGH_CREDITS = 104;
;const NO_SUCH_CUSTOMER = 107;
;const WRONG_PASSWORD = 107;
;const INVALID_REQUEST = 108;
;const FAXDIR_CREATION_FAILURE = 110;
;const FAXFILE_CREATION_FAILURE = 111;
;const NOT_ACTIVE = 111;
;const ILLEGAL_PAGES_COUNT = 112; // if pagescount <= 0
;const MSTFILE_CREATION_FAILURE = 112;
/**
 * If Tiff reader pagescount returns 0.
 * @var int
 */
;const INVALID_TIFF_DOCUMENT = 113;
/**
 * File upload error occurred.
 * @var int
 */
;const FILE_UPLOAD_ERROR = 114;
/**
 * If specified pageCount not equals to Tiff reader pageCount.
 * @var int
 */
;const PAGES_COUNT_MISMATCH = 115;
;const DUPLICATE_NUMBERS = 116;
;const NUMBERS_COUNT_LIMIT = 117;
;const FAX_HEADER_LENGTH_LIMIT = 118;

```

## Invalid identification number errors (2xx)

```

;const WRONG_FAX_ID = 200;
;const FAX_NOT_FOUND = 201;

```

## Invalid object requests (3xx)

```

;const TOO_MANY_REQUESTS = 301;
;const COULD_NOT_SAVE_OBJECT = 302;

```

## Account Management Errors (4xx)

```

// ACCOUNT MANAGEMENT STUFF
;const WRONG_EMAIL_FORMAT = 401;
;const ILLEGAL_CHARACTERS_IN_PASSWORD = 402;
;const WRONG_SMS_NUMBER = 403;
;const ILLEGAL_CHARACTERS_IN_FAXHEADER = 404;
;const FAXHEADER_NOT_EDITABLE = 405;
;const LIST_COUNT_LIMIT_EXCEEDED = 406;
;const OPTION_NOT_AVAILABLE = 407;

```

## Address book errors (6xx)

```

// ADDRESS BOOK STUFF
;const WRONG_CONTACTS_FORMAT = 601;

```

## Conversion Errors (7xx)

```
-----  
// CONVERSION  
{const UNSUPPORTED_DOCUMENT_TYPE = 701;  
{const FILE_SIZE_LIMITATION = 702;  
{const CONVERSION_ERROR = 703;  
-----
```

Retrieved from "[https://wiki.dev-popfax.com:8443/index.php/M2M\\_API v2.0](https://wiki.dev-popfax.com:8443/index.php/M2M_API_v2.0)"

- This page was last modified 13:41, 26 March 2010.