

M2M API v3

From PopfaxWiki

Contents

- 1 General information
 - 1.1 C/C++
 - 1.2 PHP
- 2 Services and methods
 - 2.1 Authentication
 - 2.1.1 authenticate
 - 2.2 Customer account
 - 2.2.1 getAllLanguages
 - 2.2.2 getAllCountries
 - 2.2.3 getAccountInfo
 - 2.2.4 setAccountInfo
 - 2.2.5 setNewEmail
 - 2.2.6 setNewPassword
 - 2.2.7 getShiftTime
 - 2.2.8 getAppVersion
 - 2.2.9 registerApp
 - 2.3 Fax, voice and SMS message
 - 2.3.1 getOneSMSSentInfo
 - 2.3.2 getSMSSentInfo
 - 2.3.3 getOneFAXReceivedInfo
 - 2.3.4 getFAXReceivedInfo
 - 2.3.5 getOneFAXSentInfo
 - 2.3.6 getFAXSentInfo
 - 2.3.7 getOneVOICEReceivedInfo
 - 2.3.8 getVOICEReceivedInfo
 - 2.3.9 sendFaxAsynchronous
 - 2.3.10 sendSmsAsynchronous
 - 2.3.11 forwardAsynchronous
 - 2.3.12 getRecipients
 - 2.3.13 downloadFile
 - 2.3.14 asynchronInfoRange
 - 2.3.15 asynchronInfo
 - 2.4 Contactbook and folders

- 2.4.1 getOneContactGroupInfo
 - 2.4.2 getContactGroupInfo
 - 2.4.3 setContactGroupInfo
 - 2.4.4 getOneContactInfo
 - 2.4.5 getContactInfo
 - 2.4.6 setContactInfo
 - 2.4.7 getOneFolderInfo
 - 2.4.8 getFolderInfo
 - 2.4.9 renameFolder
 - 2.4.10 moveToFolder
- 2.5 Object updates and manipulations
 - 2.5.1 reviewObjects
 - 2.5.2 getObjects
 - 2.5.3 getObjectTypes
 - 2.5.4 getObjectActions
 - 2.5.5 getModifiedObjects
 - 2.5.6 deleteObjects
- 3 Types
 - 3.1 Objects
 - 3.1.1 AuthenticationHeader
 - 3.1.2 LanguageInfo
 - 3.1.3 CountryInfo
 - 3.1.4 AccountInfo
 - 3.1.5 ContactGroupInfo
 - 3.1.6 ContactInfo
 - 3.1.7 FolderInfo
 - 3.1.8 SMSSentInfo
 - 3.1.9 FAXReceivedInfo
 - 3.1.10 FAXSentInfo
 - 3.1.11 VOICEReceivedInfo
 - 3.1.12 ObjectIDInfo
 - 3.1.13 ObjectsIDInfo
 - 3.1.14 ObjectData
 - 3.1.15 ObjectStatus
 - 3.1.16 ObjectsInfo
 - 3.1.17 ObjectsID
 - 3.1.18 EnumInfo
 - 3.1.19 RecipientInfo
 - 3.1.20 SendFaxInfo
 - 3.1.21 SendSmsInfo
 - 3.1.22 AppVersion
 - 3.1.23 ForwardInfo
 - 3.2 Arrays

- 4 Examples
 - 4.1 How to authenticate user
 - 4.2 How to send a fax
 - 4.3 How to send a SMS
 - 4.4 How to retrieve items from my account
- 5 Statuses codes
 - 5.1 Fax statuses
 - 5.2 Popfax statuses

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.

All the usages are provided by using any code generator for using this service.

- C/C++: gsoap (<http://www.cs.fsu.edu/%7Eengelen/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.

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

```
$ wsdl2h -o [ServiceName].h https://api3.popfax.com/?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://api3.popfax.com/?service=[ServiceName]&wsdl=1
```

To build without STL, use option -s:

```
$ wsdl2h -s -o [ServiceName].h https://api3.popfax.com/?service=[ServiceName]&wsdl=1
```

To generate client source code:

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

Please consult the gSOAP documentation (<http://www.cs.fsu.edu/%7Eengelen/soapdoc2.html>) for further instructions and documentation.

PHP

PHP includes native SOAP extensions, or some third party libraries can be used as well (NuSOAP, PEAR, ...). If using API functions that require user authentication, you need to pass authentication details in the headers of the request. The password must be encrypted in MD5 format. This is performed using WSSE extension as follows:

```
$usernameToken = array(
    'email' => $username,
    'password' => $password
);

$wsse_ns = "http://schemas.xmlsoap.org/ws/2002/07/secext";

$SoapVarUser = new SoapVar($username, XSD_STRING, NULL, $wsse_ns, NULL, $wsse_ns);
$SoapVarPass = new SoapVar($password, XSD_STRING, NULL, $wsse_ns, NULL, $wsse_ns);

$wsse_auth = new SoapVar($usernameToken, SOAP_ENC_OBJECT, NULL, $wsse_ns, 'AuthenticationHeader', $wsse_ns);
$header = new SoapVar($wsse_auth, SOAP_ENC_OBJECT, NULL, $wsse_ns, 'Security', $wsse_ns);
$wsse_header = new SoapHeader($wsse_ns, 'AuthenticationHeader', $header, true);
```

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

```
$wsdl = "https://api3.popfax.com/?service=Popfax&wsdl";
$client = new SoapClient($wsdl);

$client->__setSoapHeaders(array($wsse_header));
$response = $client->__SoapCall('PopfaxService.getAccountInfo',array());
```

Services and methods

Authentication

The Authentication service provides remote methods for user authentication, all Customer related services are derived from it, thus supporting direct authentication.

The following methods are supported by Authentication service.

authenticate

Authenticates a customer by its login and password set in header.

- **input:** none
- **return:** int *customerid*, customer Id of authenticated customer or 0 in case of failure.

int authenticate ()

Customer account

The Customer account service provides remote methods for user interaction with contact book, folder structure and customer settings and preferences.

The following methods are supported by the Customer service.

getAllLanguages

Returns the list of all available languages

- **input:** none
- **return:** object ArrayOfLanguageInfo *languages*

ArrayOfLanguageInfo getAllLanguages ()

getAllCountries

Returns the list of all available countries

- **input:** none
- **return:** object ArrayOfCountryInfo *countries*

ArrayOfCountryInfo getAllCountries ()

getAccountInfo

Returns authenticated customer's account information

- **input:** none
- **return:** object AccountInfo *account*

AccountInfo getAccountInfo ()

setAccountInfo

Update authenticated customer's account information

- **input:** object AccountInfo *account*
- **return:** int *error*, corresponds to error code

int setAccountInfo (AccountInfo \$account)

setNewEmail

Update authenticated customer's login email

- **input:** string *email*
- **return:** int *error*, corresponds to error code

int setNewEmail (string \$email)

setNewPassword

Update authenticated customer's password

- **input:** string *password*, encrypted in MD5 format
- **return:** int *error*, correspond to error code

int setNewPassword (string \$password)

getShiftTime

Returns authenticated customer's account timeshift

- **input:** none
- **return:** int *timeshift*, corresponds to account timeshift

int getShitTime ()

getAppVersion

Returns the latest application version, depending on operating system and application name

- **input:** string *os*, operating system name
- **input:** string *application*, application name
- **return:** object AppVersion *version*

AppVersion getAppVersion (string \$os, string \$application)

registerApp

Returns information about a list of folders, identified by their Ids

- **input:** int *application*,
- **input:** string *version*,
- **return:** int *error*,

int registerApp (int \$application, string \$version)

Fax, voice and SMS message

The fax, voice and SMS services provide remote methods for customer interaction with messages, like sending and fetching detailed information.

The following methods are supported:

getOneSMSSentInfo

Returns information about an SMS, identified by its Id

- **input:** int *id*, SMS Id
- **return:** object SMSSentInfo *sms*

SMSSentInfo getOneSMSSentInfo (int \$id)

getSMSSentInfo

Returns information about a list of SMSes, identified by their Ids

- **input:** object ArrayOfInteger*ids*, list of SMS Ids
- **return:** object ArrayOfSMSSentInfo *smses*

ArrayOfSMSSentInfo getSMSSentInfo (ArrayOfIntegers \$ids)

getOneFAXReceivedInfo

Returns information about a received fax, identified by its Id

- **input:** int *id*, fax Id
- **return:** object FAXReceivedInfo *fax*

FaxReceivedInfo getOneFaxReceivedInfo (int \$id)

getFAXReceivedInfo

Returns information about a list of received faxes, identified by their Ids

- **input:** object ArrayOfInteger*ids*, list of fax Ids
- **return:** object ArrayOfFAXReceivedInfo *faxes*

ArrayOfFAXReceivedInfo getFAXReceivedInfo (ArrayOfIntegers \$ids)

getOneFAXSentInfo

Returns information about a sent fax, identified by its Id

- **input:** int *id*, fax Id
- **return:** object FAXSentInfo *fax*

FaxSentInfo getOneFaxSentInfo (int \$id)

getFAXSentInfo

Returns information about a list of sent faxes, identified by their Ids

- **input:** object ArrayOfInteger *ids*, list of fax Ids
- **return:** object ArrayOfFAXSentInfo *faxes*

ArrayOfFAXSentInfo getFAXSentInfo (ArrayOfIntegers \$ids)

getOneVOICEReceivedInfo

Returns information about a received voice message, identified by its Id

- **input:** int *id*, voice mail Id
- **return:** object VOICEReceivedInfo *voicemail*

VOICEReceivedInfo getOneVOICEReceivedInfo (int \$id)

getVOICEReceivedInfo

Returns information about a list of received voice messages, identified by their Ids

- **input:** object ArrayOfInteger *ids*, list of voice mail Ids
- **return:** object ArrayOfVOICEReceivedInfo *voicemails*

ArrayOfVOICEReceivedInfo getVOICEReceivedInfo (ArrayOfIntegers \$ids)

sendFaxAsynchronous

Send a fax to be registered for sending. Return error code and registration Id in case of success

- **input:** object SendFaxInfo *fax*, description of fax to register for sending
- **return:** object ArrayOfInteger *registration*

ArrayOfInteger sendFaxAsynchronous (SendFaxInfo \$fax)

sendSmsAsynchronous

Send a SMS to be registered for sending. Return error code and registration Id in case of success

- **input:** object SendSMSInfo *sms*, description of SMS to register for sending
- **return:** object ArrayOfInteger *registration*

ArrayOfInteger sendSmsAsynchronous (SendSMSInfo \$sms)

forwardAsynchronous

Send an item to forward to registration. Return error code and registration Id in case of success

- **input:** object ForwardInfo *forward*, description of item to forward
- **return:** object ArrayOfInteger *registration*

ArrayOfInteger forwardAsynchronous (ForwardInfo \$forward)

getRecipients

Get the list of recipients of a given item, identified by its Id and its type.

- **input:** int *id*, item Id
- **input:** string *type*, item type, can be "FAX" or "SMS"
- **return:** object ArrayOfRecipientsInfo *recipients*

ArrayOfRecipientInfo getRecipients (int \$id, string \$type)

downloadFile

Download a file, identified by his id, his type and his format

- **input:** int *id*, item Id
- **input:** int *type*, item type from Object Type table

- **input:** string format, item type, can be "wav", "pdf" or "tiff"
- **return:** string *file*

string downloadFile (int \$id, string \$type, string)

asynchronInfoRange

Get information about range of registered ID obtained from forwardAsynchronous, sendSmsAsynchronous, sendFaxAsynchronous

- **input:** ArrayOfInteger ids, list of registered id in asynchron function
- **return:** object ArrayOfObjectStatus *data*

ArrayOfObjectStatus asynchronInfoRange(ArrayOfInteger \$ids)

asynchronInfo

Get information about registered ID obtained from forwardAsynchronous, sendSmsAsynchronous, sendFaxAsynchronous

- **input:** int id, registered Id
- **return:** object ArrayOfInteger *data*

return data array have next identification:

```
- array size=1 - error, invalid input parameter id.
- array size=2 - only status is defined in the second element, real object id is not available.
- array size=3 - status is defined in the second element and real object id in the third element.
```

ArrayOfInteger asynchronInfo(int \$id)

Contactbook and folders

The contactbook and folders services provide remote methods for contact, contact group and folders interaction.

The following methods are supported:

getOneContactGroupInfo

Returns information about a given contact group, identified by its Id

- **input:** int *id*, contact group Id
- **return:** object ContactGroupInfo *contactgroup*

ContactGroupInfo getOneContactGroupInfo (int \$id)

getContactGroupInfo

Returns information about a list of contact group, identified by their Ids

- **input:** object ArrayOfInteger *ids*, list of contact group Ids
- **return:** object ArrayOfContactGroupInfo *contactgroups*

ArrayOfContactGroupInfo getContactGroupInfo (ArrayOfInteger \$ids)

setContactGroupInfo

Update/define information about a contact group

- **input:** object ContactGroupInfo *contactgroup*, information about contact group
- **return:** int *id*, contact group Id

int setContactGroupInfo (ContactGroupInfo \$contactgroup)

getOneContactInfo

Returns information about a given contact, identified by its Id

- **input:** int *id*, contact Id
- **return:** object ContactInfo *contact*

ContactInfo getOneContactInfo (int \$id)

getContactInfo

Returns information about a list of contacts, identified by their Ids

- **input:** object ArrayOfInteger *ids*, list of contact Ids
- **return:** object ArrayOfContactInfo *contacts*

ArrayOfContactInfo getContactInfo (ArrayOfInteger \$ids)

setContactInfo

Update/define information about a contact

- **input:** object ContactInfo *contact*, information about the contact
- **return:** int *id*, contact Id

int setContactInfo (ContactInfo \$contact)

getOneFolderInfo

Returns information about a folder, identified by its Id

- **input:** int *id*, folder Id
- **return:** object FolderInfo *folder*

FolderInfo getOneFoldertInfo (int \$id)

getFolderInfo

Returns information about a list of folders, identified by their Ids

- **input:** object ArrayOfInteger *ids*, list of folder Ids
- **return:** object ArrayOfFolderInfo *folders*

ArrayOfFolderInfo getFolderInfo (ArrayOfIntegers \$ids)

renameFolder

Renames a folder identified by its folder Id

- **input:** int *folderid*, folder Id
- **input:** string *name*, new folder name

- **return:** object ArrayOfInteger *folders*

ArrayOfInteger renameFolder (int folderid, string name)

moveToFolder

moves items to a folder identified by his folder Id

- **input:** object ArrayOfObjectsID *objects*, list of objects Ids to move
- **input:** int *folderid*, folder to which to move
- **return:** int *error*

int moveToFolder (ArrayOfObjectsID \$objects, int \$folderid)

Object updates and manipulations

The contactbook and folders services provide remote methods for contact, contact group and folders interaction.

The following methods are supported:

reviewObjects

Returns the list of all objects Id from an authenticated customer

- **input:** none
- **return:** object ObjectsIDInfo *objects*

ObjectsIDInfo reviewObjects ()

getObjects

Returns information about requested objects.

- **input:** Object ArrayOfObjectData *data*
- **return:** object ObjectsInfo *objects*

ObjectsInfo getObjects (ArrayOfObjectData \$data)

getObjectTypes

Returns description about different objects types.

- **input:** none
- **return:** object ArrayOfEnumInfo *types*

ArrayOfEnumInfo getObjectTypes ()

getObjectActions

Returns description about different objects actions.

- **input:** none
- **return:** object ArrayOfEnumInfo *actions*

ArrayOfEnumInfo getObjectActions ()

getModifiedObjects

Returns the list of modified objects since last synchronization timestamp.

- **input:** int *timestamp*
- **return:** object ObjectsIDInfo *objects*

ObjectsIDInfo getModifiedObjects (int \$timestamp)

deleteObjects

Delete objects from requested list.

- **input:** object ArrayOfObjectsID *objects*
- **return:** int *error*, 0 in case of success

int deleteObjects (ArrayOfObjectsID \$objects)

Types

Objects

AuthenticationHeader

This object defines the credential information to pass in header for authentication.

- string *email*, customer's login email
- string *password*, customer's password

LanguageInfo

This object defines information about a supported language.

- string *code*, ISO639-1 code for language
- string *language*, corresponding language name, in it own language

The list of supported languages is available through `getAllLanguages()` method.

CountryInfo

This object defines information about a supported country.

- string *code*, ISO3166 code for country
- string *country*, corresponding name of the country

The list of supported countries is available through `getAllCountries()` method.

AccountInfo

This object defines customer's account information

- string *faxnumber*, customer allocated fax number in international format, if available
- string *faxheader*, customer's fax header, default value "www.popfax.com"
- int *notifybysms*, set to 1 to activate SMS notification, 0 to deactivate
- string *notification sms*, GSM number where to send SMS notification, if *notifybysms* is set to 1
- int *notifybyemail*, set to 1 to activate email notification, 0 to deactivate
- string *notification email*, email address where to send email notification, if *notifybyemail* is set to 1

- int *redirecttoemail*, set to 1 to attach fax to notification email, 0 to not attach it
- int *attachmentid*, set to 1 to receive attached fax in PDF format, to 0 for TIFF format
- string *firstname*, customer's first name
- string *lastname*, customer's last name
- string *city*, customer's city
- string *address1*, customer's address
- string *postalcode*, customer's postal code
- string *country*, customer's country
- string *langcode*, ISO639-1 code of account language
- string *phone*, customer's phone number
- string *companyname*, customer's company name
- string *vat*, customer's VAT number
- int *prepaidbalance*, current balance in credits
- string *enddate*, account expiration date

ContactGroupInfo

This object describes a contact group from contactbook

- int *id*, contact group Id
- string *name*, contact group name

ContactInfo

This object describes a contact from contactbook

- int *groupid*, contact group Id in which is stored the contact
- int *id*, contact Id
- string *name*, contact name
- string *number*, contact fax number
- string *gsmnumber*, contact GSM number

FolderInfo

This object defines folder information

- int *foldertype*, folder type

- int *id*, folder Id
- string *name*, folder name

Folder type is defined as follows:

Value	Folder type
0	Custom folder
1	Inbox
2	Outbox

SMSSentInfo

This object describes sent SMS definition.

- int *folderid*, folder Id in which is stored the SMS
- int *id*, SMS Id
- int *status*, SMS status marked as read if 1, unread if 0
- int *sentstatus*, transmission status
- string *date*, date of transmission, received in french time
- string *content*, SMS content
- int *recipients*, count of recipients
- string *recipient*, if only one recipient is counted, corresponds to recipient's number, not use otherwise

FAXReceivedInfo

This object describes received fax information.

- int *folderid*, folder Id in which is stored the SMS
- int *id*, fax Id
- int *type*, fax type
- int *status*, fax status marked as read if 1, unread if 0
- string *date*, reception time
- int *pagecount*, count of pages
- string *sender*, corresponds to sender's fax number

FAXSentInfo

This object describes sent fax definition.

- int *folderid*, folder Id in which is stored the fax
- int *id*, fax Id
- int *type*, fax type
- int *status*, fax status marked as read if 1, unread if 0
- int *sentstatus*, transmission status
- string *date*, transmission time
- int *pagecount*, count of pages
- int *recipients*, count of recipients
- string *recipient*, if only one recipient is counted, corresponds to recipient's number, not use otherwise

VOICEReceivedInfo

This object describes received voice mail information.

- int *folderid*, folder Id in which is stored the voice mail
- int *id*, voice mail Id
- int *type*, voice mail type
- int *status*, voice mail status marked as read if 1, unread if 0
- string *date*, reception time
- int *duration*, in seconds
- int *sender*, corresponds to sender's phone number

ObjectIDInfo

Any object from Popfax system, for synchronization process

- int *object*, Object Id
- int *type*, object type
- int *action*, create/modify if value is 1, delete if value is -1

Object type is from following list:

Value	Object type
-------	-------------

1	sent fax
2	received fax
3	received voice mail
4	sent SMS
5	contact group
6	contact
7	folder
8	get all
9	** not in use **
10	time shift
11	register

ObjectsIDInfo

This object describes a list of objects to synchronize since a given timestamp

- int *timestamp*, execution timestamp
- ArrayOfObjectIDInfo *objects*, list of objects concerned

If no prior execution has been performed, it is recommended to set timestamp to 0 to get all objects.

ObjectData

This object describe data used to apply one single action to many objects

- int *object*, object Id
- int *type*, type of objects (see ObjectIDInfo)

ObjectStatus

This object describesUsed to apply one single action to many objects

- int *id*, registered id, obtained from Asynchronous functions
- int *object*, real id, if registered id is invalid value will be 0
- int *status*, current status of the object, if object have value 0, value will be 200

ObjectsInfo

This object regroups the information on any kind of objects, from a given timestamp

- int *timestamp*, execution timestamp
- int *shifttime*, time shift from customer's account
- ArrayOfContactGroupInfo *contactgroups*, list of contact groups
- ArrayOfContactInfo *contacts*, list of contacts
- ArrayOfFolderInfo *folders*, list of folders
- ArrayOfSMSSentInfo *smsessent*, list of sent smses
- ArrayOfFAXReceivedInfo *faxesreceived*, list of received faxes
- ArrayOfFAXSentInfo *faxessent*, list of sent faxes
- ArrayOfVOICEReceivedInfo *voicesreceived*, list of received voice mails

ObjectsID

The object regroups all objects of a given type

- int *type*, object type (see ObjectIDInfo)
- ArrayOfInteger *ids*, list of Ids of the corresponding items

EnumInfo

This objects describes an enumeration list

- string *name*, field's name
- int *value*, field's value

RecipientInfo

This object describes individual sending status

- string *recipient*, recipient number
- int *status*, sent status

SendFaxInfo

This object describes information about send faxes

- ArrayOfString *numbers*, list of recipients
- int *dithering*, set to 1 for dithering method, 0 for threshold method
- int *coverpage*, set to 1 if cover page is used, 0 otherwise
- string *content*, if coverpage is set to 1, content of cover page, not used otherwise
- string *desc*, fax description, displayed in Popfax box
- int *pages*, count of pages
- int *app*, *** reserved for internal use ***

SendSmsInfo

This object describes information about send SMSes

- ArrayOfString *numbers*, list of recipients
- int *unicode*, set to 1 if unicode is used, 0 otherwise
- string *content*, content of the SMS
- int *app*, *** reserved for internal use ***

AppVersion

- ***reserved for internal use ***

ForwardInfo

This object describes information about forwarded items

- int *id*, object Id to forward
- int *type*, object type (see ObjectIDInfo)
- int *forward*, forwarding method
- string *to*, forward destination, as an email address or GSM/fax number

- int *app*, *** reserved for internal use ***

Forward method is from the following list:

Value	Forward method
1	*** reserved ***
2	*** reserved ***
3	fax forwarding
4	SMS forwarding
5	email forwarding

Arrays

Types are also available as arrays. The following is a list of defined arrays.

- **ArrayOfInteger**
- **ArrayOfString**
- **ArrayOfAuthenticationHeader**
- **ArrayOfLanguageInfo**
- **ArrayOfCountryInfo**
- **ArrayOfAccountInfo**
- **ArrayOfContactGroupInfo**
- **ArrayOfContactInfo**
- **ArrayOfFolderInfo**
- **ArrayOfSMSSentInfo**
- **ArrayOfFAXReceivedInfo**
- **ArrayOfFAXSentInfo**
- **ArrayOfVOICEReceivedInfo**
- **ArrayOfObjectIDInfo**
- **ArrayOfObjectsIDInfo**
- **ArrayOfObjectData**
- **ArrayOfObjectStatus**
- **ArrayOfObjectsInfo**
- **ArrayOfEnumInfo**
- **ArrayOfRecipientInfo**
- **ArrayOfSendFaxInfo**

- **ArrayOfSendSmsInfo**
- **ArrayOfObjectsID**
- **ArrayOfAppVersion**
- **ArrayOfForwardInfo**

Examples

How to authenticate user

```
<?php
$customerid = $client->authenticate();
$account = $client->getAccountInfo();
if($customerid && $account) {
    print('Welcome '.$account->firstname);
}
else {
    print('Authentication failure.');
```

How to send a fax

First you need to authenticate the user described here.

```
<?php
// define fax information
$numbers = array('+33123456789');

$fax = array(
    'numbers' => $numbers,
    'dithering' => 0,
    'coverpage' => 0,
    'content' => '',
    'desc' => '',
    'pages' => 0,
    'app' => 0,
);

//add the attachment to the request ...
$client->clearAttachments();
$client->addAttachment(file_get_contents($pathToTIFFDocument), $TIFFdocumentFileName);

// and send the fax
$result = $client->sendFaxAsynchronous($fax);

//the result will contain the error code of the request or the registration Id
if (count($response) == 2)
    print('envoi OK - registration' . $response[1] . "\n");
else
    print('Error' . $response[0]);
?>
```


How to send a SMS

First you need to authenticate the user described here.

```
<?php
// define fax information
$numbers = array('+33623456789');
$message = 'content of my message';

$sms = array (
    'numbers' => $numbers,
    'unicode' => 0,
    'content' => $message,
    'app' => 0
);

// and send the sms
$response = $client->__SoapCall('PopfaxService.sendSmsAsynchronous',array($sms));

// check registration
if (count($response) == 2)
    print('envoi OK - registration' . $response[1] . "\n");
else
    print('Error ' . $response[0]);
?>
```

How to retrieve items from my account

Retrieving contact book, folders and their content goes through the synnchronization process.

```

<?php
// reminds object types definition
$types = $client->__SoapCall('PopfaxService.getObjectTypes',array());
foreach($types as $type) {
    print ($name . ": " . $value . "\n");
}

// retrieve all data
$all = $client->__SoapCall('PopfaxService.reviewObjects',array());

// fetch all data
foreach($all->actions as $item) {
    if ($item->type == 1)                // sent faxes
        $sent_faxes[] = $item->object;
    elseif ($item->type == 2)            // received faxes
        $received_faxes[] = $item->object;
    elseif ($item->type == 3)            // received voicemail
        $received_voicemail[] = $item->object;
    elseif ($item->type == 4)            // sent smses
        $sent_smses[] = $item->object;
    elseif ($item->type == 5)            // contact groups
        $contact_groups[] = $item->object;
    elseif ($item->type == 6)            // contacts
        $contacts[] = $item->object;
    elseif ($item->type == 7)            // folders
        $folders[] = $item->object;
}
?>

```

Statuses codes

Fax statuses

Code	Value	Description
FA	0	Failure
OK	1	OK
IQ	2	In Queue
NF	3	No fax machine answering
IT	4	Incomplete transmission

CF	5	Fax connection failed
BU	6	Busy line
CR	7	Call rejected
CE	8	Conversion error
NA	9	No answer
IN	10	Invalid number
SB	11	System busy
BA	12	Bad line
LP	13	Line problem
CS	14	File cannot be sent
RF	15	Remote fax aborted

Popfax statuses

Code	Value	Description
BL	101	Bad number list

NT	102	No attachment
MF	103	More than 50 numbers
NC	104	Not enough credits
SE	105	Conversion error
UF	106	Unsupported file type
AF	107	Authentication failed
NR	108	No recipients
MA	109	Mail accepted
GF	110	General failure
DF	111	Address failure