Docsaway 1.2.3

SOAP: API Mail Manual

Date Updated: 26th April 2013

Contents

Introduction

The Docsaway SOAP API provides you with the ability to send PDF documents directly in to the Docsaway system via your Docsaway account. We will provide you with a PHP example of the code required to achieve this. If your not using PHP we invite you to send us other language examples so we can add them to our manual and help other users.

Important note: The SOAP API on Docsaway uses persistent sessions, please make sure the language your using is set to **allow cookies.** In C# this means editing web.config.

PHP SOAP Code

We have tried to make the code as simple and as self explanatory as possible, take a look at the overall code below. We will then provide you with a more in depth breakdown of whats going on.

```
// SET METHODS //
    1)  $file = file_get_contents('test.pdf');
    2)  $client = new SoapClient('https://www.docsaway.com/app/api/soap/api_mail.wsdl');
    3)  $client->setAPIConnection('your email','installation key');
    4)  $client->setAPICharset('UTF-8');
    5)  $client->setAPIMode('LIVE');
    6)  $client->setRecipientName('Paul Hamlington');
    7)  $client->setRecipientCompany('QiQ Communication PTY LTD');
    8)  $client->setRecipientAddress1('2 / 45 Ena Street');
    9)  $client->setRecipientAddress2('Terrigal Village');
    10) $client->setRecipientCity('Terrigal');
    11) $client->setRecipientState('NSW');
    12) $client->setRecipientZip('2260');
    13) $client->setRecipientCountry('AU');
    14) $client->setPDFFile($file);
    15) $client->setPrintingStation('PH1','PH1_1','BW',80);
    16) $client->setReseller('31JRA');
    17) $client->execute();

// GET METHODS: Transaction //
    18) $approved = $client->getTransaction('APPROVED');
    19) $price = $client->getTransaction('PRICE');
    20) $reference = $client->getTransaction('REFERENCE');
    21) $date = $client->getTransaction('DATE');
    22) $balance = $client->getTransaction('BALANCE');

// GET METHODS: Station //
    23) $station_id = $client->getStation('ID');
    24) $station_iso2 = $client->getStation('ISO2');
    25) $station_country = $client->getStation('COUNTRY');
    26) $station_city = $client->getStation('CITY');
    27) $courier_id = $client->getStation('COURIERID');
    28) $courier_name = $client->getStation('COURIERNAME');
    29) $station_zone = $client->getStation('ZONE');

// GET METHODS: Document //
    30) $document_envelope = $client->getDocument('ENVELOPE');
    31) $document_ink = $client->getDocument('INK');
    32) $document_paper = $client->getDocument('PAPER');
    33) $document_size = $client->getDocument('SIZE');

// DEBUGGING METHODS //
    34) $report = $client->APIReport();
    35) $errno  = $client->APIErrorNumber();
```

PHP SOAP Code Breakdown

The following is a breakdown of what the code is doing at every stage. The SET methods must be called in the order they are shown below, you can call GET methods in any order you wish as long as they are called after the EXECUTE() method which must follow the SET methods.

SET Methods

1) Read your entire PDF file into a variable for transmitting, specify the location to the PDF file you wish to send.

```
$file = file_get_contents('test.pdf');
```

2) Start a new soap client object and set the URL to the docsaway api_mail.wsdl file.

```
$client = new SoapClient('https://www.docsaway.com/app/api/soap/api_mail.wsdl');
```

3) Enter your Docsaway account connection settings, installation key can be found in your account. This tells our system what account should be used.

```
$client->setAPIConnection('your email','installation key');
```

4) Enter the character set your recipient text is encoded in, this is because our system is UTF8 so different formats have to be converted.

```
$client->setAPICharset('UTF-8');
```

5) Enter what transaction mode you wish to use, whether your sending through a live or test transaction.

```
$client->setAPIMode('TEST');
```

6) Enter the name of the recipient, this will appear on the coversheet. Please ensure that the recipient name is encoded in the character set specified on code line 4.

```
$client->setRecipientName('Paul Hamlington');
```

7) Enter the recipient company name, this is an optional field, if the recipient doesn't have a company name simply enter Boolean false. If a company name is specified it will appear on the coversheet. Please ensure that the company name is encoded in the character set specified on code line 4.

```
$client->setRecipientCompany('QiQ Communications PTY LTD');
$client->setRecipientCompany(false);
```

8) Enter the street name of the recipients address, this will appear on the coversheet. Please ensure that the address name is encoded in the character set specified on code line 4.

```
$client->setRecipientAddress1('2 / 45 Ena Street');
```

9) Enter an additional address line, this is an optional field, if the recipient doesn't have the need of an additional address line simply enter Boolean false. If an additional address field is specified it will appear on the coversheet. Please ensure that the address name is encoded in the character set specified on code line 4.

```
$client->setRecipientAddress2('Terrigal Village');
$client->setRecipientAddress2(false);
```

10) Enter the city name of the recipients address, this will appear on the coversheet. Please ensure that the city name is encoded in the character set specified on code line 4.

```
$client->setRecipientCity('Terrigal');
```

11) Enter the state name of the recipients address, this is an optional field, if the recipient doesn't have the need of a state name simply enter Boolean false. If a state name is specified it will appear on the coversheet. Please ensure that the state name is encoded in the character set specified on code line 4.

```
$client->setRecipientState('NSW');
$client->setRecipientState(false);
```

12) Enter the zip code of the recipients address, this will appear on the coversheet. Please ensure that the zip code is encoded in the character set specified on code line 4.

```
$client->setRecipientZip('2260');
```

13) Enter the country code of the recipients address, our system uses the ISO 3166-1 standard, it will accept any of the following formats:

ISO 3166-1 Alpha 2 (2 letter country codes)
ISO 3166-1 Alpha 3 (3 letter country codes)
ISO 3166-1 Numeric (Numeric value from 1 to 3 digits long)

```
$client->setRecipientCountry('AU');
$client->setRecipientCountry('AUS');
$client->setRecipientCountry('36');
```

14) Enter the variable that contains your PDF document, this variable was created in 1) of the code example. This method will then transmit your PDF document to the Docsaway system.

```
$client->setPDFFile($file);
```

15) This method specifies the station and stations stationery you wish to use for the processing of your document. This method takes 4 values:

- Printing Station: ID → PH1 (Reference) or AUTO
- Printing Station: Courier ID → PH1_1 (Reference) or Boolean false
- Stationery: Ink Type → BW
- Stationery: Paper Type → 80

(see page 16 – Printing Stations: Quick Reference for available settings)

```
$client->setPrintingStation('PH1','PH1_1','BW',80);
```

16) Enter the value of your Docsaway account reference (found within your account), and all documents generated through your account will generate you commission. Please note this is ONLY for a service where customers are using their login details via your module. No commission will be generated for your own documents. This method is optional.

```
$client->setReseller('31JRA');
```

17) Execute your settings. This method doesn't take any values, and is called after the SET methods. The purpose of this is to tell our system to execute all the settings specified and to complete the transaction.

```
$client->execute();
```

GET Methods

There are 3 types of GET methods used within the API they are:

getTransaction();
getStation();
getDocument();

These methods are used to return information about the API transaction back to you, for incorporation within your own system.

Get Transaction Method

This method can currently accept 5 values, these values are:

- APPROVED
- PRICE
- REFERENCE
- DATE
- BALANCE

18) Approved is used to evaluate whether the transaction was successful or not, this value will return a string of either y (successful transaction) or n (failed transaction).

$approved = $client->getTransaction('APPROVED');

19) Price is used to retrieve the price of the transaction (how much the document cost to be sent in AUD), this value will return a decimal containing a number with a decimal point.

$price = $client->getTransaction('PRICE');

20) Reference is used to retrieve the reference of the transaction (a unique code that refers to this transaction only), this value will return a string that can contain both letters and numbers.

$reference = $client->getTransaction('REFERENCE');

21) Date is used to retrieve the date and time of the transaction as it appears on our system. This value will return a string containing numbers, colons and hyphens.

$date = $client->getTransaction('DATE');

22) Balance is used to retrieve your remaining account balance after your transaction has occurred. This value will return a decimal containing a number with a decimal point.

$balance = $client->getTransaction('BALANCE');

Get Station Method
This method can currently accept 7 values, these values are:

- ID
- ISO2
- COUNTRY
- CITY
- COURIERID
- COURIERNAME
- ZONE

23) ID is used to retrieve the station id of the station that will be used to process and dispatch your document, this value will return a string that contain letters and numbers.

$station_id = $client->getStation('ID');

24) ISO2 is used to retrieve the ISO country code of the station that will be used to process and dispatch your document, this will return a string that contains two letters (ISO 3166-1 Alpha 2).

$station_iso2 = $client->getStation('ISO2');

25) Country is used to retrieve the country name of the station that will be used to process and dispatch your document, this will return a string.

$station_country = $client->getStation('COUNTRY');

26) City is used to retrieve the city name of the station that will be used to process and dispatch your document, this will return a string.

$station_city = $client->getStation('CITY');

27) Courier ID is used to retrieve the courier id used by the station that will process and dispatch your document, this will return a string.

$courier_id = $client->getStation('COURIERID');

28) Courier Name is used to retrieve the name of the courier used by the station that will process and dispatch your document, this will return a string.

$courier_name = $client->getStation('COURIERNAME');

29) Zone is used to retrieve the zone number assigned by the station that will process and dispatch your document (1=local, 2=national, 3=international), this will return a integer.

$station_zone = $client->getStation('ZONE');

Get Document Method
This method can currently accept 4 values, these values are:

- ENVELOPE
- INK
- PAPER
- SIZE

30) Envelope is used to retrieve the envelope id that was used for your document, this will return a string that contains two letters (DL - C4).

$document_envelope = $client->getDocument('ENVELOPE');

31) Ink is used to retrieve the ink id that was used for your document, this will return a string that contains two letters (BW = Black and White/ CL = Colour).

$document_ink = $client->getDocument('INK');

32) Paper is used to retrieve the paper weight (GSM) that was used for your document, this will return a number three digits long max (80).

$document_paper = $client->getDocument('PAPER');

33) Size is used to retrieve the total number of pages in your document, this will return a number (1 – 40).

$document_size = $client->getDocument('SIZE');

Error and Debugging Methods

The API also includes 2 methods for debugging and error catching they are as follows:

34) <u>API Report</u>

This method can be very useful during the API development stages its our debugging tool, gives you feedback on your transaction. If there are any problems the API report will explain where they occurred.

$report=$client->APIReport();

35) <u>API Error Number</u>

This method is useful for trapping an error that occurred during your transaction. The system reports a number back if there was an error. You can then use the table (see page 14 - Error Codes: Quick Reference) to established what the error was. It would be advisable to store the error number if one occurs to your associated transaction.

$errno = $client->APIErrorNumber();

SET Methods: Quick Reference

The following table is a quick reference guide for the SET methods and method values.

Methods	Mandatory	Value	Max Size
setAPIConnection()	Yes	Requires two values: - Email - Installation Key	 150 64
setAPICharset()	Yes	Character Set	N\A
setAPIMode()	Yes	Accepts any of the following: - TEST - LIVE	4
setRecipientName()	Yes	Recipient Name	50
setRecipientCompany()	No	Company Name | false	50
setRecipientAddress1()	Yes	Recipient Address	50
setRecipientAddress2()	No	Recipient Address Additional | false	50
setRecipientCity()	Yes	Recipient City	50
setRecipientState()	No	Recipient State/County | false	50
setRecipientZip()	Yes	Recipient Zip/Postcode	10
setRecipientCountry()	Yes	Accepts any of the following codes: - ISO 3166-1 Alpha 2 - ISO 3166-1 Alpha 3 - ISO 3166-1 Numeric	3
setPDFFile()	Yes	Requires variable that contains the stored PDF document	N\A
setPrintingStation()	Yes	Requires four values: - Printing Station ID You must specify the ID of the station you wish to use, or use AUTO. - Courier ID You can either specify the ID of the courier you wish to use, or you can use boolean false and let the system choose the best courier for you. - Ink ID You must specify the two letter Ink ID you wish to use. (CL = colour | BW = black/white). - Paper ID You must specify the paper weight you wish to use. (80).	 5 8 2 3
setReseller()	No	Requires your Docsaway Account reference.	50

GET Methods: Quick Reference

The following table is a quick reference guide for the GET methods and method values.

Methods	Value	Example Output
getTransaction()	Accepts any of the following: - APPROVED - PRICE - REFERENCE - DATE - BALANCE	 String: y | n Decimal: 0.00 String: FG009J String: yyyy-mm-dd h:m:s Decimal: 0.00
getStation()	Accepts any of the following: - ID - ISO2 - COUNTRY - CITY - COURIERID - COURIERNAME - ZONE	 String: AU1 String: AU String: Australia String: Terrigal String: AU1_1 String: Australia Post Integer: 1
getDocument()	Accepts any of the following: - ENVELOPE - INK - PAPER - SIZE	 String: DL String: BW Integer: 80 Integer: 40

Error Codes: Quick Reference

The following table is a reference guide for the error numbers returned from the APIErrorNumber() method.

Error No	Error String	Meaning
001	errStringEmpty	Submitting an empty value for a mandatory field.
002	errUserAuthentication	User authentication information is incorrect.
003	errAccountDeactivated	User account has been deactivated.
004	errCharsetNotRecognized	Entered character set has not been recognized.
005	errCharsetNotSet	Character set was not specified.
006	errSetOrderIncorrect	Set methods declared in incorrect order.
007	errCharsetConversion	Character set used for SET methods has failed to convert to system character set.
008	errStringLength	Submitted a string length larger than allowed.
009	errStringFormat ISO-3166-1	Country code not recognized.
010	errModeNotRecognized	API mode not recognized.
011	errBase64BinaryEmpty	File method empty, PDF not set.
012	errPasswordDenied	PDF password does not match the account password.
013	errFileOpeningFailure	API class failed to open uploaded PDF.
014	errFileNotSaved	API class failed to save uploaded PDF.
015	errFileSize0kb	The uploaded PDF document was empty.
016	errFileNotFound	PDF document not located by the API class.
017	errFileTypeIncorrect	Document uploaded was not a PDF.
018	errPageCountSystemFailure	API page count class failed.
019	errMaxCountExceeded	PDF document exceeded maximum allowed pages.
020	errValueNotRecognized	A value specified within the setPrintingStation() method was not recognized.
021	errStationNotFound	Specified printing station was not found.
022	errCourierNotAvailable	Specified courier is not available.
023	errInkUnsupported	Specified ink setting for a selected station is not available.
024	errRecipientCountryUnsupported	Specified printing station does not support the recipients country.
025	errStationNotRecognized	Specified printing station was not recognized.
026	errPropertySetting	Mandatory properties needed to locate the printing station are incorrect.
027	errSystemError	An internal script error has occurred and was captured.
028	errPDFGenerationFailure	An internal script error occurred while accessing the coversheet class.

029	errCalculationFailure	An internal script error occurred while accessing the pricing class.
030	errStorageFailure	An internal script error occurred while accessing the database.
031	errCombineDocsFailure	An internal script error occurred while merging coversheet and PDF document.
032	errInsufficientFunds	Your account has insufficient funds for your document transaction.
033	errStatusUpdateFailure	An internal error occurred while changing the status of your transaction.
034	errStationBlockPresent	Your selected station can not be used as you have selected the stations country as blocked in your account.
035	errMaxFileSizeExceeded	Your PDF document exceeds the maximum file size allowed, currently set at 2Mb.
036	errLockTimeoutExceeded	The server is extremely busy to the point access for account update couldn't be given.
037	errSessionFailure	The SOAP API uses sessions while interacting with client and server, a problem occurred when generating a session.
038	errCourierNotFound	Specified courier was not found.
039	errPaperUnsupported	Specified paper setting for a selected station is not available.

Printing Stations: Quick Reference

The following table is a reference guide for the necessary printing station codes related to method setPrintingStation().

BW = code for black and white printing.
CL = code for colour printing.

Location	Station ID	Courier ID	Paper Weight (gsm)	Ink
Automatic Station Selection	AUTO	false (boolean)	80	BW / CL
Australia, Terrigal	AU1	AU1_1	80	BW / CL
Brazil, Sorocaba	BR1	BR1_1	80	BW
Canada, British Columbia	CA1	CA1_1	80	BW / CL
France, St Claud	FR1	FR1_1	80	BW
Germany, Uberlingen	DE1	DE1_1	80	BW / CL
India, Girgaum	IN2	IN2_1	80	BW
Netherlands, Groningen	NL1	NL1_1	80	BW / CL
New Zealand, Te Awamutu	NZ1	NZ1_1	80	BW / CL
Philippines, Batangas	PH1	PH1_1	80	BW
Portugal, Lisbon	PT1	PT1_1	80	BW / CL
Singapore, Singapore	SG1	SG1_1	80	BW / CL
South Africa, Cape Town	ZA1	ZA1_1	80	BW / CL
Spain, Guardamar Del Segura	ES1	ES1_1	80	BW / CL
Ukraine, Odessa	UA1	UA1_1	80	BW / CL
United Kingdom, Northampton	GB1	GB1_1	80	BW
United States, Denver	US1	US1_1	80	BW / CL

Automatic Station Selection

By using the AUTO option, Docsaway will refer to its inbuilt table selection for the best station for your selected destination. Please note over time stations can change, any station changes will be communicated to our customers, please follow our Twitter or Facebook channels to keep up to date.

Code Example: C# (.Net 4.0)

The following code was provided by Gareth Evans, it is an example of how to connect to our Docsaway SOAP API using C#.

Important note: The SOAP API on Docsaway uses persistent sessions so for this to work in C# you will need to edit web.config to **allow cookies**.

```
// C# EXAMPLE CODE //
protected void Page_Load(object sender, EventArgs e)
        {
        Docsaway.dc_APIMailPortClient dsWs = new Docsaway.dc_APIMailPortClient();

        bool b = false;
        String APIReport = "";

        //General settings
        b = dsWs.setAPIConnection("email","isd");
        b = dsWs.setAPICharset("UTF-8");
        b = dsWs.setAPIMode("TEST");

        //Recipient details
        b = dsWs.setRecipientName("Some one");
        b = dsWs.setRecipientCompany("Some Firm");
        b = dsWs.setRecipientAddress1("Some Street");
        b = dsWs.setRecipientAddress2("Some thing else");
        b = dsWs.setRecipientCity("Some City");
        b = dsWs.setRecipientState("Some state");
        b = dsWs.setRecipientZip("Some ZIP");
        b = dsWs.setRecipientCountry("GB");

        //Load in the PDF we are sending
        FileStream fsAttachIn = System.IO.File.OpenRead("c:\\test.pdf");
        byte[] bytesFile = new byte[fsAttachIn.Length];
        fsAttachIn.Read(bytesFile, 0, bytesFile.Length);
        fsAttachIn.Close();
        b = dsWs.setPDFFile(bytesFile);

        //Printing station
        b = dsWs.setPrintingStation("GB1", "GB1_1", "BW", "80");

        //Execute it
        b = dsWs.execute("");

        //Check the results
        APIReport = dsWs.APIReport("");
        TextBox1.Text = APIReport;
        }
```