

API Documentation BlueMailCentral Web Service

Revision: 0.4
API version: 1.0

Table of Contents

API DOCUMENTATION BLUEMAILCENTRAL WEB SERVICE	- 1 -
INTRODUCTION	- 3 -
STRUCTURES	- 4 -
Account	- 5 -
Address	- 7 -
AddressCheck	- 8 -
AddressListResult	- 10 -
User	- 11 -
METHODS	- 14 -
addFile	- 15 -
authenticate	- 16 -
checkAddressList	- 17 -
checkSession	- 17 -
getAccount	- 18 -
getFrontPage	- 18 -
getUser	- 19 -
isFrontPageSet	- 19 -
setFrontPage	- 20 -
test	- 20 -
EXAMPLE ADDFILE IMPLEMENTATION	- 21 -
RUBY ON RAILS CODE EXAMPLE WITH COURTESY OF COMMUUN.NL	- 22 -

Introduction

This document describes the BlueMailCentral webservice API and it's intended usage. The webservice can be reached at [\(WSDL\)](#). Please note that this document is still in draft status. Not all methods have been described yet, and the methods lack proper examples.

The following 3 chapters, describe the used structures, exceptions and methods.

Structures

The following structures are used in the webservice:

- Account
- Address
- AddressCheck
- AddressListResult

We will briefly discuss each structure, and show the XML structure involved.

Account

The account structure consists of the following fields:

- Long id
- String type (company or personal)
- String currency
- String localCurrency
- String organization
- String address
- String zipcode
- String city
- String state
- String country
- String phonenumber
- String faxnumber
- String industry
- Date created
- Boolean acceptSponsoredMail

Example XML:

```
<account>
  <id>1</id>
  <type>company</type>
  <currency>EUR</currency>
    <localCurrency>EUR</localCurrency>
    <organization>BlueMail B.V.</organization>
    <address>Dam 51</address>
    <zipcode>4331 GG</zipcode>
    <city>Middelburg</city>
    <state>Zeeland</state>
    <country>NL</country>
    <onenumber>0118707437</onenumber>
```

```
<faxnumber>0118707437</faxnumber>  
<industry>tech_online</industry>  
<created>2008-10-02T10:31:54+02:00</created>  
<acceptSponsoredMail>>false</acceptSponsoredMail>  
</account>
```

Address

The address structure consists of the following fields:

- String[] addressLines
- String countryCode (optional)

Example XML:

```
<address>
<!--Zero or more repetitions:-->
<addressLines>The White House</addressLines>
<addressLines>1600 Pennsylvania Avenue NW</addressLines>
<addressLines>Washington, D.C. 20500</addressLines>
<addressLines>The United States of America</addressLines>
<!--Optional:-->
<countryCode>US</countryCode>
</address>
```

AdressCheck

The addressCheck structure consists of the following fields:

- Addressaddress
- Boolean color
- Int countryScore
- Boolean duplex
- String error
- Int numEnvelopes
- Int numPages
- Decimal price
- Boolean result

Example XML:

```
<check>
<address>
<addressLines>The White House</addressLines>
<addressLines>1600 Pennsylvania Avenue NW</addressLines>
<addressLines>Washington, D.C. 20500</addressLines>
<addressLines>The United States of America</addressLines>
<countryCode>US</countryCode>
</address>
<color>>true</color>
<countryScore>90</countryScore>
<duplex>>false</duplex>
<error></error>
<numEnvelopes>1</numEnvelopes>
<numPages>1</numPages>
<price>0.79</price>
<result>>true</result>
</check>
```

The following error strings are supported:

- NOT_ENOUGH_ADDRESLINES (less than 2 addresslines supplied)

- COUNTRY_NOT_SUPPORTED (found country is not yet supported)
- NO_PAGES (less than 1 page supplied in numPages)

AddressListResult

The addressListResult structure consists of the following fields:

- Decimal balance
- String currency
- String localCurrency
- AddressCheck[] results
- Decimal totalLocalPrice
- Decimal totalPrice

Example XML:

```
<result>
<balance>135.11</balance>
<currency>EUR</currency>
<localCurrency>EUR</localCurrency>
<results>
<address>
<addressLines>The White House</addressLines>
<addressLines>1600 Pennsylvania Avenue NW</addressLines>
<addressLines>Washington, D.C. 20500</addressLines>
<addressLines>The United States of America</addressLines>
<countryCode>US</countryCode>
</address>
<color>true</color>
<countryScore>90</countryScore>
<duplex>>false</duplex>
<error></error>
<numEnvelopes>1</numEnvelopes>
<numPages>1</numPages>
<price>0.79</price>
<result>>true</result>
</results>
<totalLocalPrice>0.79</totalLocalPrice>
<totalPrice>0.79</totalPrice>
</result>
```

User

The User structure consists of the following fields:

- Long id
- String firstname
- String affix
- String lastname
- String title
- String department
- String email
- String password (md5 hash)
- Boolean accountholder
- Boolean receiveNews
- Boolean receiveFailureNotifications
- Boolean receiveSentNotifications
- Int numberFormat (internal use)
- Int dateFormat (internal use)
- Date created

Example XML:

```
<user>
  <id>1</id>
  <firstname>Bart</firstname>
    <affix/>
  <lastname>De Vries</lastname>
  <title>Managing director</title>
  <department>Development</department>
  <email>Colin@bluemailcentral.com</email>
  <password>23540c146e722a466436ca2d68f9739d </password>
  <accountHolder>true</accountHolder>
  <receiveNews>true</receiveNews>
  <receiveFailureNotifications>true</receiveFailureNotifications>
```

```
<receiveSentNotifications>true</receiveSentNotifications>  
<numberFormat>1</numberFormat>  
<dateFormat>1</dateFormat>  
<created>2008-10-02T10:31:54+02:00</created>  
</user>
```

Exceptions

The following exceptions can be thrown:

Exception:

UnsupportedDestinationFault

Description:

This exception is thrown when the country that was found in the supplied address is not yet supported.

InsufficientFundsFault

This exception is thrown when you attempt to add new file when you don't have sufficient funds to do so.

NumPagesException

This exception is thrown when the supplied number of pages doesn't match the PDF data.

BlueMailException

This is a general exception. The title will tell you what is wrong. Most of these exceptions will become named exceptions in the future.

Methods

The webservice consists of the following methods:

- addFile
- authenticate
- checkAddressList
- checkSession
- getAccount
- getFrontPage
- getUser
- isFrontPageSet
- setFrontPage
- test

We will briefly discuss each method, it's input parameters, possible exceptions and the expected output.

addFile

addFile allows you to add a new letter to your account. You are expect to run checkAddressList on the address you are about to supply. addFile will give you an exception if the supplied country is unsupport or your funds have been exhausted.

Please note that this method expects you to supply a PDF in A4 or Letter format. Any other format will be scaled to fit, and the results may be unreliable. Also note that the number of pages should match with the actual PDF data, or an exception is thrown.

Parameters:

- String sessionId
- Byte[] fileData
- Address address
- String fileName
- Int numPages
- Boolean color
- Boolean duplex

Possible exceptions:

- UnsupportedDestinationFault
- InsufficientFundsFault
- NumPagesException
- BlueMailException

Output:

- Boolean result

authenticate

Authenticate allows you to register a new session. It returns the new session id or throw a BlueMailException if authentication fails.

Parameters:

- String email
- String password

Possible exceptions:

- BlueMailException

Output:

- String result

Please note: the type of exception return when authentication fails might change in the future.

checkAddressList

This method allows you to validate an address list (consisting of one or more addresses). It will return a complex structure that will allow you to see which addresses validate and which don't. It will also return a price per address, and the country that was found.

Parameters:

- String sessionId
- AddressCheck[] addressList

Possible exceptions:

- BlueMailException

Output:

- AddressListResult result

checkSession

This method allows you to validate a session. Returns true or false.

Parameters:

- String sessionId

Output:

- Boolean result

getAccount

This method returns the account information for a supplied session.

Parameters:

- String sessionId

Possible exceptions:

- BlueMailException

Output:

- Account result

getFrontPage

This method returns the current front page (PDF).

Parameters:

- String sessionId

Possible exceptions:

- BlueMailException

Output:

- Byte[] result

getUser

This method returns the user information for a supplied session.

Parameters:

- String sessionId

Possible exceptions:

- BlueMailException

Output:

- User result

isFrontPageSet

This method returns whether a front page is set for the current account.

Parameters:

- String sessionId

Possible exceptions:

- BlueMailException

Output:

- Boolean result

setFrontPage

This method sets a new frontpage. The frontpage should be supplied as A4 or Letter PDF.

Parameters:

- String sessionId
- Byte[] fileData

Possible exceptions:

- BlueMailException

Output:

- Boolean result

test

This method is for test purposes only.

Output:

- Boolean result

Example addFile implementation

Here you can find a small sample PHP implementation of the webservice. This script authenticates, checks an address, and then adds a new PDF file.

Be sure to change the username and password in `authenticate()` accordingly, and send the right parameters with `addFile` (like `numPages`, `color`, `duplex` and `filename`).

```
$addressStruct = array(
    "addressLines" =>array(
        "BlueMailCentral B.V.",
        "Dam 51",
        "4331 GG Middelburg",
        "Zyzyzyva" // Test country
    ),
);

$addressCheckStruct = array(
    "address" =>$addressStruct,
    "color" =>true,
    "duplex" =>true,
    "numPages" =>5,
    "countryScore" => null, // Will be filled by server
    "numEnvelopes" => null, // Will be filled by server
    "result" => null, // Will be filled by server
);

// Get the SOAP Client
$client = new SoapClient(
    "https://ws.bluemailcentral.com/BlueMailService/BlueMail?wsdl",
    array( 'features' => SOAP_SINGLE_ELEMENT_ARRAYS )
);

// Authenticate: be sure to replace username and password
$sessionId = $client->authenticate( array( "email" =>"test@test.org",
                                          "password"
                                          =>"testpass" ) )->result;

// Check the supplied address for validity, and get pricing details and countrycode.
$checkResult = $client->checkAddresslist(array( "sessionId" =>$sessionId,
                                                "addressList" =>array($addressCheckStruct)));

print "Total price: " . $checkResult->result->totalPrice . "\n";

// The address is now filled with a countrycode, which we need to add the file
$finalAddress = $checkResult->result->results[0]->address;

$fileData = file_get_contents("test.pdf");

// Upload the file to the server. Be sure to replace numPages, filename, color and duplex.
$addResult = $client->addFile( array("sessionId" =>$sessionId, "fileData" =>$fileData,
                                    "address" =>$finalAddress, "fileName" =>"test.pdf",
                                    "numPages" =>10, "color" =>true,
                                    "duplex" =>false ) )->result;

if($addResult) {
    print "Success!\n";
}
```

Ruby on Rails code example with courtesy of commuun.nl

```
#
# This is a demonstration script showing how to use the BlueMailCentral api using ruby
#
# You'll need the Savon SOAP client gem installed for this to work:
# $> gem install savon
#

require 'base64'
require 'savon'

# These settings depend on your pdf and what type of print you require
duplex = true
color = true
num_pages = 1
file_path = '/my_docs/my_document.pdf'

address_check_struct = {
  :address => ["BlueMailCentral B.V.",
             "Dam 51",
             "4331 GG Middelburg",
             "Zyzyzyva" ],      # Test country
  :color => color,
  :duplex => duplex,
  :numPages => num_pages,
  :countryScore => nil,
  :numEnvelopes => nil,
  :result => nil
}

client = Savon::Client.new "https://ws.bluemailcentral.com/BlueMailService/BlueMail?wsdl"

# First we get our session id, replace the values below with your registered email and password.
auth_result = client.authenticate do |c|
  c.body = { :email => 'test@test.org',
            :password => 'testpass' }
end

address_result = client.check_address_list do |c|
  c.body = { :sessionId => auth_result.to_hash[:authenticate_response][:result],
            :addressList => [address_check_struct] }
end

page_result = client.add_file do |c|
  c.body = { :sessionId => auth_result.to_hash[:authenticate_response][:result],
            :fileData => Base64.encode64(File.read(file_path)).gsub("\n",""), # We have to strip newlines from the encoded file
            :fileName => File.basename(file_path),
            :address => address_result.to_hash[:check_address_list_response][:result][:results][:address],
            :numPages => num_pages,
            :color => color,
            :duplex => duplex }
end
```

Please contact us if there is anything that we can do to improve this documentation or the API. [Hello@bluemailcentral.com](mailto>Hello@bluemailcentral.com)
