

ViaPostCustomer API 1.1 Release

Document Revisions:

Version	Date	Author	Comments
0.1	7 th July 2008	Gary Jesionowski	Initial Draft
0.2	8 th September 2008	Gary Jesionowski	Revised Draft
0.3	12 th September 2008	Gary Jesionowski	Internal Release
0.4	16 th September 2008	Gary Jesionowski	Beta Release
1.0	17 th November 2008	Gary Jesionowski	ViaPostCustomer API v1.1

Copyright

Copyright © 2008 ViaPost Limited. All rights reserved.

'ViaPost' and 'Simple. Secure. Green.' are registered trademarks of ViaPost limited. Other product and company names mentioned herein may be trademarks or trade names of their respective owners.

Disclaimer

The information in this document is provided "as is," with no warranties whatsoever, including any warranty of merchantability, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. This document is provided for informational purposes only. ViaPost Limited disclaims all liability, including liability for infringement of any proprietary rights, relating to implementation of information presented in this document. ViaPost Limited does not warrant or represent that such use will not infringe such rights. ViaPost Limited retains the right to make changes to this specification at any time, without notice.

TABLE OF CONTENTS

INTRODUCTION	3
PURPOSE	3
INTENDED AUDIENCE.....	3
API R1.1.....	4
HOW TO CONNECT TO THE SERVICE	4
HOW TO OBTAIN SUPPORT WHILST TESTING/USING THE API	4
VIAPOSTCUSTOMER API R1.1 OPERATIONS	5
API OPERATIONS	6
SIGN IN TO YOUR ACCOUNT.....	6
UPLOADING A DOCUMENT	7
GET YOUR CURRENT BALANCE.....	7
GET A LETTER	7
SENDING A LETTER.....	8
UPLOAD A PDF MERGE DOCUMENT	8
UPLOAD A MAILING LIST	8
RUN A MAIL MERGE	9
RETURN A LIST OF MAILING LISTS	9
RETURN A LIST OF UPLOADED MERGE DOCUMENTS.....	9
CUSTOMER API LETTER OBJECTS	10
CUSTOMER API MAILING LIST OBJECTS	10
CSV FORMAT REQUIREMENTS	11
RETURN MESSAGES	12
VB.NET CODE SNIPPETS	13
<i>Sign In.....</i>	<i>13</i>
<i>Upload Document</i>	<i>13</i>
<i>Send Letter</i>	<i>13</i>
<i>Upload A PDF Merge Document.....</i>	<i>14</i>
<i>Upload Mailing List</i>	<i>14</i>
<i>Run a Mail Merge.....</i>	<i>14</i>
<i>Get Mailing Lists.....</i>	<i>15</i>
<i>GetLetters.....</i>	<i>15</i>

Introduction

Purpose

This is the reference documentation for the ViaPost API v1.1 based on the SOAP protocol.

For updated versions of this documentation as they become available, see the [ViaPost Developers website](#).

Intended Audience

This documentation is intended for developers who are responsible for creating applications that will use the native features of the ViaPost system.

It assumes that the developer is familiar with normal ViaPost operating methods including the ViaPost Printer Driver.

API R1.1

The ViaPostCustomer API is a SOAP-based web service. In order to connect to it, you'll need the WSDL (service description) page URL. These can be found here:

Test API webservice:

<http://apitest.viapost.com/viapostcustomer.asmx>

Live ViaPostCustomer API webservice:

<http://api.viapost.com/viapostcustomer.asmx>

How to Connect to the Service

If you are building for the .NET environment using Visual Studio, create a web project and select Project > Add Web Reference.

In the dialog box, enter the following URL to add a reference to the webservice:

For UAT test environment: <http://apitest.viapost.com/Viapostcustomer.asmx>

For live environment: <http://api.viapost.com/Viapostcustomer.asmx>

How to Obtain Support Whilst Testing/Using the API

Please email any ViaPostCustomer API support issues that you have to:

api-support@viapost.co.uk

We would be grateful to receive any comments regarding this documentation including any errors and omissions you may note. Any suggestions as to how we may improve the service should be sent via email to the above address.

PLEASE DO NOT USE THE GENERAL SUPPORT NUMBER OR EMAIL AT VIAPOST.

ViaPostCustomer API R1.1 Operations

The following operations are supported. For a formal definition, please review the [Service Description](#).

- GetBalance
- GetLetterByID
- GetLetters
- GetMailingListByID
- GetMailingLists
- MailMerge
- SendLetter
- SignIn
- UploadDocument
- UploadMailingList

API Operations

SIGN IN TO YOUR ACCOUNT

To use the webservice (either test or live), you have to sign in first. When using the test environment you should use the test account that was supplied. Otherwise you should use your ViaPost customer User ID and password when using the live ViaPost system.

SignIn

Use this method to sign in to the ViaPostCustomer webservices.

Signing in is mandatory and must be done before any other service methods can be used. You must supply a valid ViaPost Username and Password.

Returns a LoginToken which must be supplied as a parameter to the other methods. This method returns True if successful or False if a problem occurred.

The ReturnMessage provides an explanation in the event of a failure.

About the Login Token

IMPORTANT: *SignIn passes back a Login Token (an encrypted string) which you must use in order to access the other methods. You will need to supply it as an input parameter to the other methods in the Customer API.*

UPLOADING A DOCUMENT

UploadDocument

Use this method to upload a PDF document.

You must supply a valid LoginToken to use this method (use SignIn to acquire a LoginToken). The Letter Name you supply will be used to display your letter. The DocumentData parameter must contain your PDF as a Base64-encoded byte array.

This method returns True if successful or False if a problem occurred. The ReturnMessage provides an explanation in the event of a failure.

The LetterID is populated with the new record ID once the letter has been uploaded.

GET YOUR CURRENT BALANCE

GetBalance

Use this method to retrieve your account balance.

You must supply a valid LoginToken to use this method (use SignIn to acquire a LoginToken).

This method returns True if successful or False if a problem occurred.

Your account balance is returned in the AccountBalance parameter.

The ReturnMessage provides an explanation in the event of a failure.

GET A LETTER

GetLettersByID

Use this method to retrieve a letter using its unique ID.

You must supply a valid LoginToken to use this method (use SignIn to acquire a LoginToken). The LetterID text you supply will be used to identify your letter.

This method returns True if successful or False if a problem occurred. The ReturnMessage provides an explanation in the event of a failure.

The ReturnLetter returns a CustomerAPILetter object.

SENDING A LETTER

SendLetter

Use this method to send a letter.

You must supply a valid LoginToken to use this method (use SignIn to acquire a LoginToken).

This method returns True if successful or False if a problem occurred. The ReturnMessage provides an explanation in the event of a failure.

UPLOAD A PDF MERGE DOCUMENT

UploadDocument

Use this method to upload a PDF document.

You must supply a valid LoginToken to use this method (use SignIn to acquire a LoginToken).

The Letter Name you supply will be used to display your letter. The DocumentData parameter must contain your PDF as a Base64-encoded byte array.

This method returns True if successful or False if a problem occurred. The ReturnMessage provides an explanation in the event of a failure.

The LetterID is populated with the new record ID once the letter has been uploaded.

UPLOAD A MAILING LIST

UploadMailingList

Use this method to upload a mailing list in CSV format (refer to the CustomerAPI documentation below for details on the CSV format requirements).

You must supply a valid LoginToken to use this method (use SignIn to acquire a LoginToken). The Mailing Address Name you supply will be used to display your mailing list. The MailingListData parameter must contain your CSV data as a Base64-encoded byte array.

This method returns True if successful or False if a problem occurred. The ReturnMessage provides an explanation in the event of a failure.

The MailingListID parameter is populated with the ID of the new record.

RUN A MAIL MERGE

MailMerge

Use this method to perform a mail merge.

You must supply a valid LoginToken to use this method (use SignIn to acquire a LoginToken).

The Letter and MailingList you supply will be used to select the letter template and mailing list to be used.

This method returns True if successful or False if a problem occurred. The ReturnMessage provides an explanation in the event of a failure.

RETURN A LIST OF MAILING LISTS

GetMailingLists

Use this method to retrieve a list of mailing lists.

You must supply a valid LoginToken to use this method (use SignIn to acquire a LoginToken). The SearchMailingListTitle text you supply will be used to find your mailing list by its uploaded title. The SearchMailingListTitle can be a partial string, in which case the first matching mailing list is returned.

This method returns True if successful or False if a problem occurred. The ReturnMessage provides an explanation in the event of a failure.

The ReturnMailinglists returns an ArrayList of CustomerAPIMailingList objects.

RETURN A LIST OF UPLOADED MERGE DOCUMENTS

GetLetters

Use this method to retrieve a letter list. You must supply a valid LoginToken to use this method (use SignIn to acquire a LoginToken). The SearchDocumentName text you supply will be used to find your letter by its uploaded title. The SearchDocumentName can be a partial string, in which case the first matching letter is returned.

This method returns True if successful or False if a problem occurred. The ReturnMessage provides an explanation in the event of a failure.

The ReturnLetters returns an arrayList of CustomerAPILetter objects.

CUSTOMER API LETTER OBJECTS

CustomerAPILetter

The CustomerAPILetter object is used to return letters, or lists of letters, from the service, and is composed of the following fields:

Field name	Description
LetterID	A unique numeric identifier for this letter.
LetterFileName	The original file name supplied when the letter was uploaded, which can be treated as the name of the letter.
NumberOfPages	The number of sides in the letter.
DateUploaded	The date the letter was uploaded.

For Visual Studio users, The CustomerAPILetter object is exposed to the developer IDE when you connect to the webservice at design-time and can be inspected using IntelliSense or the Object Browser.

CUSTOMER API MAILING LIST OBJECTS

CustomerAPIMailingList

The CustomerAPIMailingList object is used to return mailing list descriptions from the service. Please note that it does not contain the mailing list addresses themselves – it is a representation of the entire list. It is composed of the following fields:

Field name	Description
MailingListID	A unique numeric identifier for this mailing list.
MailingListTitle	The name supplied when the list was uploaded, which can be treated as the title of the mailing list.
NumberOfAddresses	The number of addresses in the mailing list.
DateLastUsed	The date the mailing list was last used.

For Visual Studio users, The CustomerAPIMailingList object is exposed to the developer IDE when you connect to the webservice at design-time and can be inspected using IntelliSense or the Object Browser.

CSV FORMAT REQUIREMENTS

The mailing list must be supplied as a **7-field CSV file**, encoded as a Base-64 byte array.

- You must supply exactly 7 fields on each row.
- No 'heading' row is required or permitted.
- Address fields may be left empty.
- Fields **Address1** and **Post code** are mandatory.

When you upload a CSV mailing list using the **UploadMailingList** method, the mailing list is automatically PAF validated.

The format of the CSV file is as follows:

Field	Description
TitleName	The recipient title and name
Address 1	Address line 1
Address 2	Address line 2
Address 3	Address line 3
Address 4	Address line 4
Address 5	Address line 5
Post code	Postal code

RETURN MESSAGES

These are the return messages which may be returned from the system. Every API method returns a message and Boolean result.

Method	Return Message	Return Value
GetBalance	Invalid login token	False
GetLetterByID	Could not find letter	False
	Invalid login token	False
	Returned letter OK	True
GetLetters	Could not find any letters starting with that name	False
	Invalid login token	False
GetMailingListByID	Could not find the mailing list	False
	Invalid mailingListID	False
	Returned mailing list OK	True
GetMailingLists	Could not find any mailing lists starting with that name	False
	Invalid login token	False
MailMerge	Invalid letter	False
	Invalid login token	False
	Invalid mailing list	False
	Letters are being sent now	True
	The mailing list or letter are not valid for this customer	False
SendLetter	Could not cleanse the address	False
	Invalid login token	False
	Letter successfully sent	True
	SendLetter failed - Invalid letter	False
SignIn	Sign-in failed - Invalid user or password	False
	You are signed in	True
UploadDocument	Invalid login token	False
	No pages could be found in document.	False
	Too many pages. Try 16 or less.	False
	Upload complete	True
UploadMailingList	All rows in the CSV file must contain 7 columns	False
	Invalid login token	False

VB.NET CODE SNIPPETS

Sign In

```
Dim api As New com.viapost.api.ViaPostCustomer

Dim sLoginToken As String
If api.SignIn("username here", "password here", sLoginToken, sReturnMessage) = False Then
    'bad login
    lblMsg.Text = sReturnMessage
    Exit Sub
Else
    'logged in OK
    lblMsg.Text = "login token returned = " & sLoginToken
End If
```

Upload Document

```
If FileUpload1.HasFile Then

    'save the uploaded letter to a temp directory and name
    FileUpload1.SaveAs(sSourcePath)

    'push the letter PDF file to the webservice
    If api.UploadDocument(lblLoginToken.Text, FileUpload1.FileName, FileUpload1.FileBytes,
        chkColour.Checked, chkDoubleSided.Checked, sReturnMessage) = True Then
        'ok pushed via webservice
        lblMsg.Text = "OK, your letter was uploaded via the webservice."
    Else
        lblMsg.Text = sReturnMessage
        Exit Sub
    End If
    'Label1.Text = "Received " & FileUpload1.FileName & " Content Type " &
    FileUpload1.PostedFile.ContentType & " Length " & FileUpload1.PostedFile.ContentLength
Else
    lblMsg.Text = "Please choose a file to upload"
    Exit Sub
End If
```

Send Letter

```
'Get a named letter (find the first matching letter in named field)
Dim sDocName As String = "myletter"
Dim returnLetter As com.viapost.api.CustomerAPILetter = Nothing
If api.GetLetter(lblLoginToken.Text, sDocName, sReturnMessage, returnLetter) = True Then
    'ok
    lblLetterList.Text = "letter:" & returnLetter.LetterID & " - " &
        returnLetter.LetterFileName
Else
    'fail to get
    lblSendMsg.Text = sReturnMessage
End If

'send the letter
If api.SendLetter(lblLoginToken.Text, _
    txtName.Text, _
    txtAddr1.Text, _
    txtAddr2.Text, _
    txtAddr3.Text, _
    txtAddr4.Text, _
    txtAddr5.Text, _
    txtPostcode.Text, _
    returnLetter, _
    sReturnMessage) = True Then
    'ok!
End If
'set the message
lblSendMsg.Text = sReturnMessage
```

Upload A PDF Merge Document

```
If FileUpload1.HasFile Then
    'save the uploaded letter to a temp directory and name
    FileUpload1.SaveAs(sSourcePath)
    'push the letter PDF file to the webservice
    If api.UploadDocument(lblLoginToken.Text, FileUpload1.FileName, FileUpload1.FileBytes,
        chkColour.Checked, chkDoubleSided.Checked, sReturnMessage) = True Then
        'ok pushed via webservice
        lblMsg.Text = "OK, your letter was uploaded via the webservice."
    Else
        lblMsg.Text = sReturnMessage
        Exit Sub
    End If
    'Label1.Text = "Received " & FileUpload1.FileName & " Content Type " &
    FileUpload1.PostedFile.ContentType & " Length " & FileUpload1.PostedFile.ContentLength
Else
    lblMsg.Text = "Please choose a file to upload"
    Exit Sub
End If
```

Upload Mailing List

```
If FileUpload1.HasFile Then
    'save the uploaded MailingList to a temp directory and name
    FileUpload1.SaveAs(sSourcePath)
    'push the MailingList file to the webservice
    If api.UploadMailingList(lblLoginToken.Text, FileUpload1.FileBytes,
        txtMailingListName.Text, sReturnMessage) = True Then
        'ok pushed via webservice
        lblUploadMailingMessage.Text = sReturnMessage
    Else
        'something went wrong - show the return message
        lblUploadMailingMessage.Text = sReturnMessage
        Exit Sub
    End If
Else
    lblUploadMailingMessage.Text = "Please choose a file to upload"
    Exit Sub
End If
```

Run a Mail Merge

```
Dim cal As com.viapost.api.CustomerAPILetter = Nothing
Dim caml As com.viapost.api.CustomerAPIMailingList = Nothing
If api.GetLetter(lblLoginToken.Text, txtBulkLetterName.Text, sReturnMessage, cal) = True Then
    'ok letter is good
    If api.GetMailingList(lblLoginToken.Text, txtBulkMailingListName.Text, sReturnMessage,
        caml) = True Then
        'ok mailing list is good
        'do the mailmerge
        api.MailMerge(lblLoginToken.Text, cal, caml, sReturnMessage)
        lblBulkMsg.Text = sReturnMessage
    End If
End If
```

Get Mailing Lists

```
'get a list of mailing lists (find matching lists in named field)
Dim sDocName As String = txtSearchMailingLists.Text
Dim returnLists As Object = Nothing
Dim returnArray As New ArrayList
Dim thisList As com.viapost.api.CustomerAPIMailingList
If api.GetMailingLists(lblLoginToken.Text, sDocName, sReturnMessage, returnLists) = True Then
    'we need to recast the return to an arraylist
    returnArray.AddRange(returnLists)
    'ok
    For Each thisList In returnArray
        lblGetMailingListsMsg.Text = lblGetMailingListsMsg.Text & thisList.MailingListID
        & " " & thisList.MailingListTitle & "<br>"
    Next
Else
    'fail to get
    lblGetMailingListsMsg.Text = sReturnMessage
End If
```

GetLetters

```
'Get a letter list (find the first matching letter in named field)
Dim sDocName As String = txtLetterListSearch.Text
Dim returnLetters As Object = Nothing
Dim returnArray As New ArrayList
Dim thisLetter As com.viapost.api.CustomerAPILetter
If api.GetLetters(lblLoginToken.Text, sDocName, sReturnMessage, returnLetters) = True Then
    'we need to recast the reutrn to an arraylist
    returnArray.AddRange(returnLetters)
    'ok
    For Each thisLetter In returnArray
        lblLetterListMsg.Text = lblLetterListMsg.Text & thisLetter.LetterID & " " &
        thisLetter.LetterFileName & "<br>"
    Next
Else
    'fail to get
    lblSendMsg.Text = sReturnMessage
End If
```