

SmartFocus Cloud Service APIs

Document name	Individual Member Management SOAP API Guide
Service	Member data management for individual updates and insertions
Protocol	SOAP
Version	10.12
Last updated on	October 20, 2013

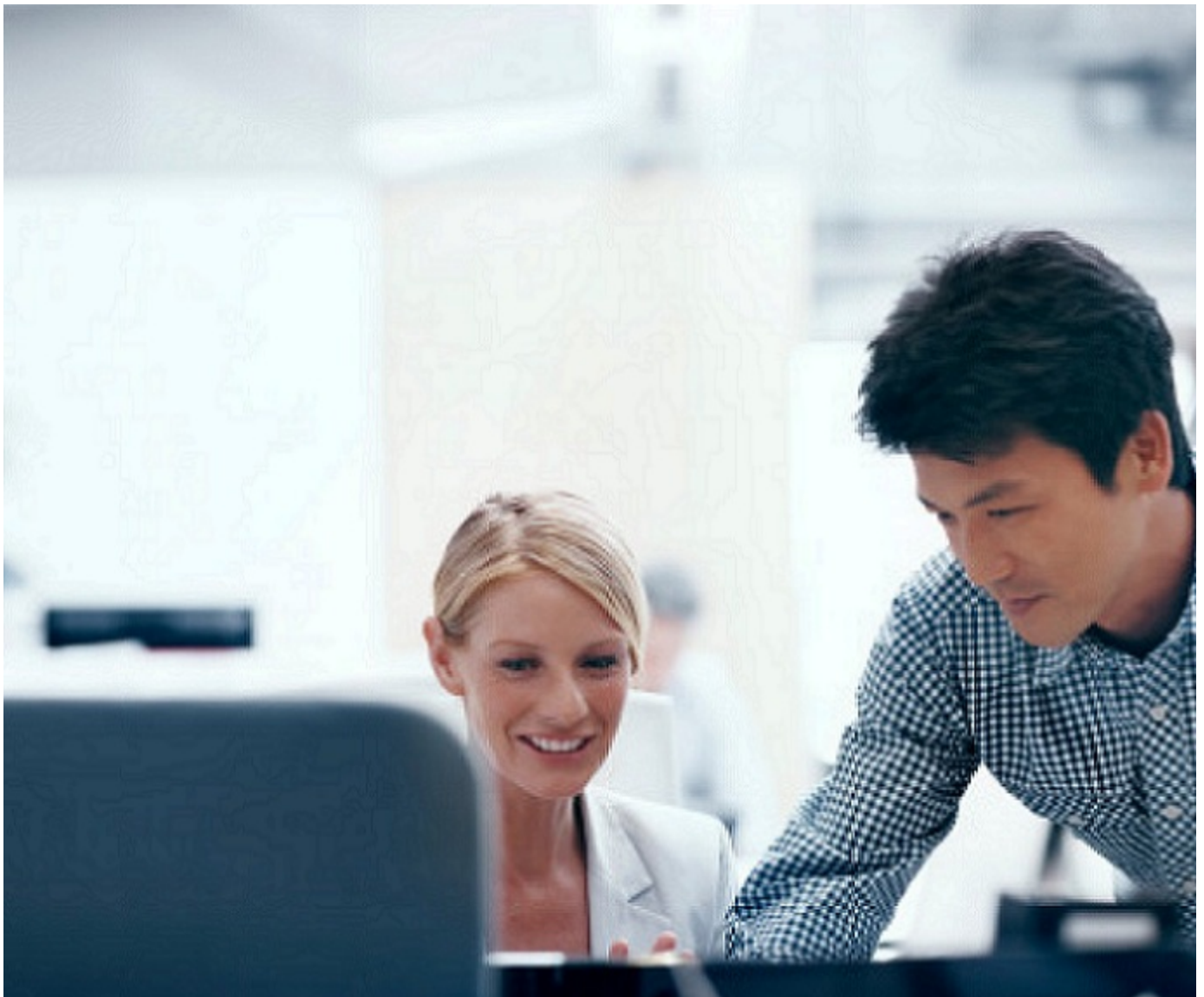


Table of Contents

Table of Contents	2
Introduction	4
About This Document	4
About SmartFocus APIs	4
Feedback	4
Support Options	4
Training Options	4
Useful Links	4
Disclaimer	5
Introducing SmartFocus APIs	6
Module Overview	6
Overview of the Data Individual Update API	7
Data Individual Update API Use Cases	7
Getting Started with Integration	9
Prerequisites	9
Quick Start	9
Integration Using APIs	9
Security	10
Connection	11
openApiConnection	12
closeApiConnection	14
Member Management	15
The Member Object	15
descMemberTable	17
getMemberByEmail	20
getMemberByCellphone	23
getMemberById	26
getListMembersByObj	29
getListMembersByPage	32
insertMember	35
updateMember	36
insertMemberByObj	38
insertOrUpdateMemberByObj	40
updateMemberByObj	42
getMemberJobStatus	45
unjoinMemberByEmail	46
unjoinByCellphone	47
unjoinMemberById	48

unjoinMemberByObj	49
rejoinMemberByEmail	50
rejoinByCellphone	51
rejoinMemberById	52
Examples	53
Reference	54

Introduction

About This Document

This document is a reference document for using SmartFocus APIs. It does not explain the purpose or functions of SmartFocus features. For information on these features, please consult the *SmartFocus Online Help* or the *SmartFocus User Guide*.

This document is intended for developers and project managers.

About SmartFocus APIs

An Application Programming Interface (API) is a source code interface that a computer system or program library provides in order to support requests for services made from another computer program.

The goal of SmartFocus APIs is to offer customers the ability to pilot a complete campaign from their own system.

Example: The introduction of a new promotional article in the e-commerce back office should automatically generate a new SmartFocus campaign to targeted recipients.

Feedback

The Individual Member Management SOAP API Guide is constantly being enhanced to provide you with more and more information on using SmartFocus API methods.

If you can't find the information you need or want to provide feedback, simply drop us a line at documentation@smartfocus.com.

We look forward to hearing from you!

Support Options

SmartFocus provides you with a dedicated Account Manager to accompany you throughout the execution of your projects in SmartFocus. Your Account Manager is the gateway to support, training, and professional services. Working with your Account Manager, you can rely on SmartFocus's deliverability and technical support teams for complex troubleshooting and optimization.

Training Options

SmartFocus provides fully comprehensive training ranging from basic product training through to advanced modules and both strategic and tactical marketing courses. The training courses are designed to help you increase productivity, develop new methods, and share best practices to optimize your email marketing campaigns.

The SmartFocus Academy, located in central London and in Paris, has state-of-the-art training rooms and equipment.

To get more information on training, go to the SmartFocus Academy website: <http://www.emailvisionacademy.co.uk>

Useful Links

- Information on SmartFocus's products and services: <http://www.smartfocus.com>
- Training: <http://www.emailvisionacademy.co.uk>

Disclaimer

While the information contained in this publication is believed to be true and accurate, SmartFocus cannot accept any legal responsibility for any errors or omissions contained herein. All information is subject to change without notice.

None of the material in this publication may be reproduced or transmitted in whole or in part without the express written permission of SmartFocus.

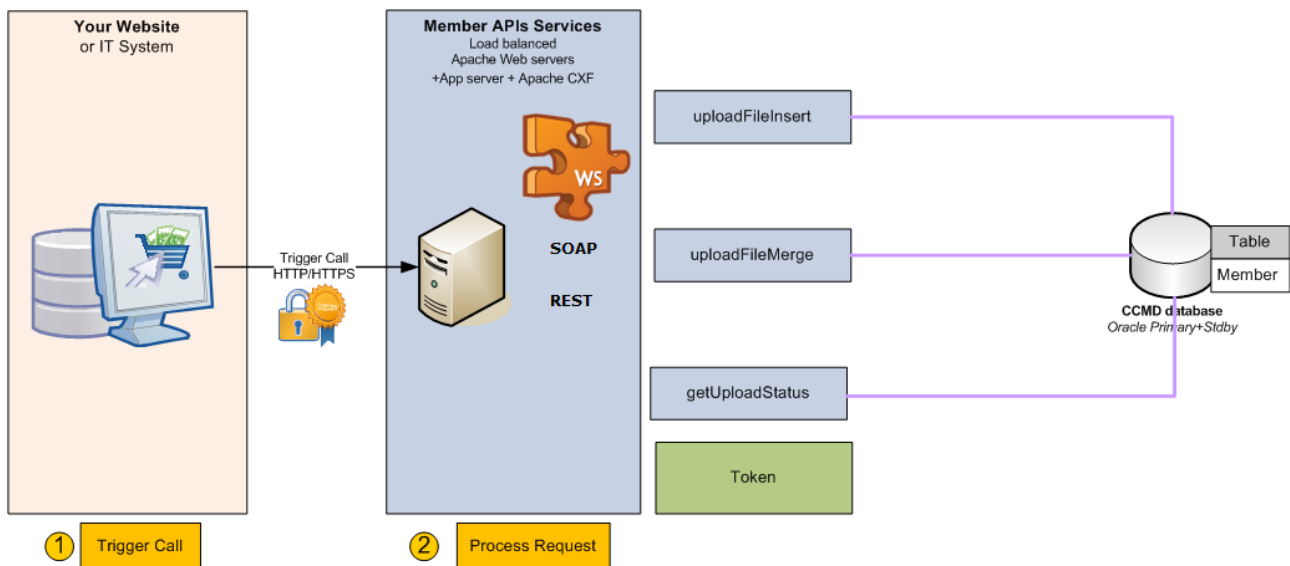
Introducing SmartFocus APIs

Module Overview

The Member module is a way to access your members' profile through the API.

It provides full access to the MEMBER table allowing you to insert, update, and get a member profile.

The member table is a single table stored in SmartFocus's datacenter. It contains all the profile information of your recipients, such as email address, first name, last name, and any column defined during the life of your account.



Overview of the Data Individual Update API

The Data Individual Update API allows you to upload and update individual member's data in your SmartFocus member list.

For further information on uploading and updating member data, please consult the *SmartFocus User Guide* or *SmartFocus Online Help*.

The following methods are available:

Method	Description
openApiConnection	This method provides a session token when given valid credentials.
closeApiConnection	This method terminates the session token.
descMemberTable	This method retrieves the list of fields (i.e. database column names) available in the Member table.
getMemberByEmail	This method retrieves a list of members who have the given email address.
getMemberByCellphone	This method retrieves a list of members who have the given cellphone number.
getListMembersByObj	This method retrieves a list of a maximum of 50 members who match the given criteria.
getListMembersByPage	This method retrieves all members page by page. Each page contains 10 members.
insertMember	This method inserts a new member using only the email address as input (i.e. all other fields will remain empty).
updateMember	This method updates a single field of a member in the Member table using the email address to identify the member.
insertMemberByObj	This method inserts a new member in the Member table by object.
insertOrUpdateMemberByObj	This method searches a specified column of the Member table for a particular value used to identify a member in order to update the member's data (by object). If the member is not found, a new member is created in the Member table. Any criteria can be used to find the member including one of the fields to be updated.
updateMemberByObj	This method searches for a member and updates the member's data in the Member table by object. If the member is not found, he/she will now be inserted in the table. Any criteria can be used to find the member including one of the fields to be updated.
getMemberJobStatus	This method retrieves the job status (i.e. the status of the member insertion or update) using the job ID.
unjoinMemberByEmail	This method unsubscribes one or more members who match a given email address.
unjoinByCellphone	This method unsubscribes one or more members who match a given cellphone number.
unjoinMemberById	This method unsubscribes a member who matches a given ID.
unjoinMemberByObj	This method unsubscribes a member by object.
rejoinMemberByEmail	Re-subscribes an unsubscribed member using his/her email address. If there are multiple members with the same email address, they will all be re-subscribed.
rejoinByCellphone	Re-subscribes an unsubscribed member using his/her cell phone number. If there are multiple members with the same number, they will all be re-subscribed.
rejoinMemberById	Re-subscribes an unsubscribed member using his/her ID.

Data Individual Update API Use Cases

Insert a New Member in the Member List

To upload the member data of a new member into the member table:

1. Use the `openApiConnection` method to open the connection.
2. Use the `insertMemberByObj` method to insert the member data into the member table.
3. Use the `getMemberJobStatus` method to obtain the status of the insertion.
4. Use the `closeApiConnection` method to close the connection.

Unsubscribe a Member

To unsubscribe the member from the mailing list:

1. Use the `openApiConnection` method to open the connection.
2. Use the `getListMembersByObj` method to find the member email address of the member you want to update.
3. Use the `unjoinMemberByEmail` to unsubscribe the member.
4. Use the `getMemberJobStatus` method to obtain the status of the unsubscription.
5. Use the `closeApiConnection` method to close the connection.

Getting Started with Integration

Prerequisites

To access SmartFocus's APIs and take full advantage of this software's ease of integration with other systems, you will need the following:

- An Internet connection
- A recent browser and operating system
- An active SmartFocus account with the API feature enabled

Quick Start

The process for interfacing your website, CRM, or any other internal system with the APIs is quite straightforward.

Step 1: Get your API key in SmartFocus

Note: You must have a dedicated API login. This login will NOT have access to SmartFocus. Contact your Account Manager to have a dedicated API login.

To connect through the API, the user must first obtain a manager key using the CCMD Web Application.

Calling the connect method (with the login, password, manager key) will provide a token, to be used in all subsequent calls.

This token will expire in the following cases:

- When a close connection call is made.
- When the maximum number of calls per session, defined by the manager in SmartFocus, is reached.
- When the session times out.

Step 2: Build your application

Integration Using APIs

The first step in getting started with web services is to configure the range of remote servers that will access this module.

Webmasters and developers should be able to interface with this new API with any programming language that uses standard HTTP calls.

List of APIs that are available:

- SOAP API
- RESTful API (see the *Individual Member Management SOAP API Guide*)
 - HTTP GET QS
 - HTTP GET PI

SOAP API

The SOAP API consists in posting an XML file through HTTP POST.

- API call summary:

SOAP

Submission URL

`https://{server}/apimember/services/MemberService?wsdl`

Parameters & associated values

- All parameter names are case sensitive.
- When specific values are expected, it should be assumed that parameter values are case sensitive.
- The order of parameters must be strictly followed.
- A parameter and its contents must appear on the same line without spaces and line breaks. In this guide, line breaks have sometimes been added for display reasons.

Security

As web services are accessible over the Internet and can be interfaced with any system, there is a risk of fraudulent access and usage of the system. To tighten the security, SmartFocus APIs can be accessed using the HTTPS protocol coupled with the use of encrypted and matching tags.

To use HTTPS, just replace **HTTP** with **HTTPS** in all the submission URLs.

Connection

Prerequisite: To use SmartFocus APIs, you need to have the API manager login provided by SmartFocus and the associated password.

To connect through the API, you must first retrieve the manager key from SmartFocus.

1. Go to **Account Administration** and select **Logins**.
2. Click the **Edit** icon next to your API manager.
3. In the **API** section of the popup window, copy the API key (also known as the manager key) and use it to open a connection to retrieve the token that will be used in your calls.

Calling the connect method (with the login, password, manager key) will provide a token, to be used in all subsequent calls.

This token will expire in the following cases:

- When a close connection call is made.
- When the maximum number of calls per session, defined by the manager in SmartFocus, is reached.
- When the session times out.

openApiConnection

This method provides a session token when given valid credentials.

Note: The token is valid for 60 minutes. To avoid problems with expired tokens, it is recommended that you close your connection after an API call and open a new connection for a new API call.

WSDL location

`https://{server}/apimember/services/MemberService?wsdl`

Note: Ask your Account Manager for your server name.

Input parameters	Description	Output parameters	Description
Required parameters			
login	The login provided for API access	return	The token to use in all other API calls
pwd	The password Note: API passwords expire after 365 days.		
key	The manager key copied from SmartFocus (see Connection on page 11)		

Error messages

You must fill in the apiName parameter to check rights of client on this API.
You must fill in the login parameter to authenticate on this API.
You must fill in the password parameter to authenticate on this API.
You must fill in the managerKey parameter to authenticate on this API.
Error while decoding managerKey.
Your login is not valid !!
Your password is not valid !!
No manager retrieved for those login, password.
No available connection for manager {0}.
{0} doesn't exist or is not activated on client account.
{0} is not activated for the client.
This manager does not have authorized access to this API.
Error while parsing validDate on managerKey.
Date not valid on managerKey!
The managerKey is no longer valid. Your API access is closed!

SOAP Example

Input

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://api.service.apimember.emailvision.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:openApiConnection>
      <login>br_test</login>
      <pwd>aptrokez</pwd>
      <key>CdX7Cr1E_26blFNJOsgfdawh6LJ3y6pwg5PEOvA</key>
    </api:openApiConnection>
  </soapenv:Body>
</soapenv:Envelope>
```

Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:openApiConnectionResponse xmlns:ns2="http://api.service.apiccmd.emailvision.com/">
      <return>{token}</return>
    </ns2:openApiConnectionResponse>
  </soap:Body>
</soap:Envelope>
```

closeApiConnection

This method terminates the session token.

WSDL location

`https://{server}/apimember/services/MemberService?wsdl`

Note: Ask your Account Manager for your server name.

Input parameter	Description	Output parameter	Description
Required parameters			
token	The connection token	return	The connection is closed if the operation was successful, otherwise an error code appears.

Error messages

You must fill in the token parameter

No available connection for the specified token.

An error occurred on the server

SOAP Example

Input

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://api.service.apimember.emailvision.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:closeApiConnection>
      <token>{token}</token>
    </api:closeApiConnection>
  </soapenv:Body>
</soapenv:Envelope>
```

Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:closeApiConnectionResponse xmlns:ns2="http://api.service.apiccmd.emailvision.com/">
      <return>connection closed</return>
    </ns2:closeApiConnectionResponse>
  </soap:Body>
</soap:Envelope>
```

Member Management

The Member Object

Input parameters Required parameters	Description
dyncontent	The entity containing one or more <entry> elements describing the member
entry	The entity containing one or more <key> and <value> elements describing the member
key	The name of the column containing the value in the database
value	The value of the column for that member
memberUID	The column to be used as the update key
email	The e-mail address of the member

SOAP example:

```

<member>
  <dyncontent>
    <entry>
      <key>DATEUNJOIN</key>
    </entry>
    <entry>
      <key>EMVHBQ</key>
    </entry>
    <entry>
      <key>DATEOFBIRTH</key>
    </entry>
    <entry>
      <key>EMVUNROUTABLE</key>
    </entry>
    <entry>
      <key>EMAIL_ORIGINE</key>
    </entry>
    <entry>
      <key>MEMBER_ID</key>
      <value>367554</value>
    </entry>
    <entry>
      <key>EMVADMIN1</key>
    </entry>
    <entry>
      <key>LASTNAME</key>
      <value>Smith</value>
    </entry>
    <entry>
      <key>EMVDOUBLON</key>
    </entry>
    <entry>
      <key>EMAIL</key>
      <value>johnsmith@smartfocus.com</value>
    </entry>
    <entry>
      <key>DATEJOIN</key>
      <value>2009-04-01T00:00:00+02:00</value>
    </entry>
    <entry>
      <key>EMVADMIN2</key>
    </entry>
  </dyncontent>
</member>

```

```
<entry>
  <key>EMVADMIN5</key>
</entry>
<entry>
  <key>EMVISP</key>
</entry>
<entry>
  <key>CLIENTURN</key>
</entry>
<entry>
  <key>EMVADMIN4</key>
</entry>
<entry>
  <key>SOURCE</key>
</entry>
<entry>
  <key>UPLOAD_ID</key>
</entry>
<entry>
  <key>EMVCELLPHONE</key>
</entry>
<entry>
  <key>EMVADMIN3</key>
</entry>
<entry>
  <key>EMAIL_FORMAT</key>
</entry>
<entry>
  <key>FIRSTNAME</key>
  <value >John</value>
</entry>
<entry>
  <key>TITLE</key>
  <value >Mr</value>
</entry>
<entry>
  <key>SEGMENT</key>
</entry>
<entry>
  <key>CODE</key>
</entry>
<entry>
  <key>SEED</key>
</entry>
<entry>
  <key>CLIENT_ID</key>
  <value 40
  </value>
</entry>
</dyncontent>
<email>johnsmith@smartfocus.com</email>
</member>
```


descMemberTable

This method retrieves the list of fields (i.e. database column names) available in the Member table.

WSDL location

`https://{server}/apimember/services/MemberService?wsdl`

Note: Ask your Account Manager for your server name.

Input parameters	Description	Output parameters	Description
Required parameters			
token	The connection token	return	The list of database column names

Error messages

You must fill in the token parameter
An error occurred on the server

SOAP Example

Input

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://api.service.apimember.emailvision.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:descMemberTable>
      <token>{token}</token>
    </api:descMemberTable>
  </soapenv:Body>
</soapenv:Envelope>
```

Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:descMemberTableResponse xmlns:ns2="http://api.service.apimember.emailvision.com/">
      <return>
        <fields>
          <name>CLIENTURN</name>
          <type>STRING</type>
        </fields>
        <fields>
          <name>CLIENT_ID</name>
          <type>LONG</type>
        </fields>
        <fields>
          <name>CODE</name>
          <type>LONG</type>
        </fields>
        <fields>
          <name>DATEJOIN</name>
          <type>DATE</type>
        </fields>
        <fields>
          <name>DATEOFBIRTH</name>
          <type>DATE</type>
        </fields>
      </return>
    </ns2:descMemberTableResponse>
  </soap:Body>
</soap:Envelope>
```

```
<fields>
  <name>DATEUNJOIN</name>
  <type>DATE</type>
</fields>
<fields>
  <name>EMAIL</name>
  <type>STRING</type>
</fields>
<fields>
  <name>EMAIL_FORMAT</name>
  <type>LONG</type>
</fields>
<fields>
  <name>EMAIL_ORIGINE</name>
  <type>STRING</type>
</fields>
<fields>
  <name>EMVADMIN1</name>
  <type>STRING</type>
</fields>
<fields>
  <name>EMVADMIN2</name>
  <type>STRING</type>
</fields>
<fields>
  <name>EMVADMIN3</name>
  <type>STRING</type>
</fields>
<fields>
  <name>EMVADMIN4</name>
  <type>STRING</type>
</fields>
<fields>
  <name>EMVCELLPHONE</name>
  <type>LONG</type>
</fields>
<fields>
  <name>EMVDOUBLON</name>
  <type>LONG</type>
</fields>
<fields>
  <name>EMVHBQ</name>
  <type>DATE</type>
</fields>
<fields>
  <name>EMVISP</name>
  <type>STRING</type>
</fields>
<fields>
  <name>EMVUNROUTABLE</name>
  <type>LONG</type>
</fields>
<fields>
  <name>FIRSTNAME</name>
  <type>STRING</type>
</fields>
<fields>
  <name>LASTNAME</name>
  <type>STRING</type>
</fields>
<fields>
  <name>MEMBER_ID</name>
  <type>LONG</type>
</fields>
```

```
<fields>
  <name>SEED</name>
  <type>LONG</type>
</fields>
<fields>
  <name>SEGMENT</name>
  <type>STRING</type>
</fields>
<fields>
  <name>SOURCE</name>
  <type>STRING</type>
</fields>
<fields>
  <name>TITLE</name>
  <type>STRING</type>
</fields>
<fields>
  <name>UPLOAD_ID</name>
  <type>LONG</type>
</fields>
</return>
</soap:Envelope>
```

getMemberByEmail

This method retrieves a list of members who have the given email address.

WSDL location

<https://{server}/apimember/services/MemberService?wsdl>

Note: Ask your Account Manager for your server name.

Input parameters Required parameters	Description	Output parameters	Description
token	The connection token	return	The list of values in the Member table for the member(s)
Email (key)	The email address of the member(s) to retrieve		

Error messages

You must fill in the token parameter
You must fill in the email parameter
An error occurred on the server

SOAP Example

Input

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://api.service.apimember.emailvision.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getMemberByEmail>
      <token>{token}</token>
      <email>johnsmith@smartfocus.com</email>
    </api:getMemberByEmail>
  </soapenv:Body>
</soapenv:Envelope>
```

Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:getMemberResponse xmlns:ns2="http://api.service.apimember.emailvision.com/">
      <return>
        <attributes>
          <entry>
            <key>DATEUNJOIN</key>
          </entry>
          <entry>
            <key>EMVHBQ</key>
          </entry>
          <entry>
            <key>DATEOFBIRTH</key>
          </entry>
          <entry>
            <key>...</key>
          </entry>
        </attributes>
      </return>
    </ns2:getMemberResponse>
  </soap:Body>
</soap:Envelope>
```

```
        <key>EMVUNROUTABLE</key>
    </entry>
    <entry>
        <key>EMAIL_ORIGINE</key>
    </entry>
    <entry>
        <key>MEMBER_ID</key>
        <value xsi:type="ns4:long" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns4="http://www.w3.org/2001/XMLSchema">367554</value>
    </entry>
    <entry>
        <key>EMVADMIN1</key>
    </entry>
    <entry>
        <key>LASTNAME</key>
        <value xsi:type="ns4:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns4="http://www.w3.org/2001/XMLSchema">Smith</value>
    </entry>
    <entry>
        <key>EMVDOUBLON</key>
    </entry>
    <entry>
        <key>EMAIL</key>
        <value xsi:type="ns4:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns4="http://www.w3.org/2001/XMLSchema">johnsmith@smartfocus.com</value>
    </entry>
    <key>DATEJOIN</key>
    <value xsi:type="ns4:dateTime" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:ns4="http://www.w3.org/2001/XMLSchema">2009-04-01T00:00:00+02:00</value>
    </entry>
    <entry>
        <key>EMVADMIN2</key>
    </entry>
    <entry>
        <key>EMVADMIN5</key>
    </entry>
    <entry>
        <key>EMVISP</key>
    </entry>
    <entry>
        <key>CLIENTURN</key>
    </entry>
    <entry>
        <key>EMVADMIN4</key>
    </entry>
    <entry>
        <key>SOURCE</key>
    </entry>
    <entry>
        <key>UPLOAD_ID</key>
    </entry>
    <entry>
        <key>EMVCELLPHONE</key>
    </entry>
    <entry>
        <key>EMVADMIN3</key>
    </entry>
    <entry>
        <key>EMAIL_FORMAT</key>
    </entry>
    <entry>
        <key>FIRSTNAME</key>
        <value xsi:type="ns4:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns4="http://www.w3.org/2001/XMLSchema">John</value>
```

```
</entry>
<entry>
  <key>TITLE</key>
  <value xsi:type="ns4:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns4="http://www.w3.org/2001/XMLSchema">Mr</value>
</entry>
<entry>
  <key>SEGMENT</key>
</entry>
<entry>
  <key>CODE</key>
</entry>
<entry>
  <key>SEED</key>
</entry>
<entry>
  <key>CLIENT_ID</key>
  <value xsi:type="ns4:long" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns4="http://www.w3.org/2001/XMLSchema">40</value>
</entry>
</attributes>
</soap:Body>
</soap:Envelope>
```

getMemberByCellphone

This method retrieves a list of members who have the given cellphone number.

WSDL location

<https://{server}/apimember/services/MemberService?wsdl>

Note: Ask your Account Manager for your server name.

Input parameters Required parameters	Description	Output parameters	Description
token	The connection token	return	The list of values in the Member table for the member(s)
cellphone	The cellphone number of the member		

Error messages

You must fill in the token parameter

You must fill in the cellphone parameter

An error occurred on the server

SOAP Example

Input

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://api.service.apimember.emailvision.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getMemberByCellphone>
      <token>{token}</token>
      <cellphone>{cellphone}</cellphone>
    </api:getMemberByCellphone>
  </soapenv:Body>
</soapenv:Envelope>
```

Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:getMemberByCellphone xmlns:ns2="http://api.service.apimember.emailvision.com/">
      <return>
        <attributes>
          <entry>
            <key>DATEUNJOIN</key>
          </entry>
          <entry>
            <key>EMVHBQ</key>
          </entry>
        </attributes>
      </return>
    </ns2:getMemberByCellphone>
  </soap:Body>
</soap:Envelope>
```

```

        <entry>
            <key>DATEOFBIRTH</key>
        </entry>
        <entry>
            <key>MEMBER_ID</key>
            <value xsi:type="xs:long" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">8048601428</value>
        </entry>
        <entry>
            <key>EMVADMIN1</key>
            <value xsi:type="xs:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">Baba</value>
        </entry>
        <entry>
            <key>LASTNAME</key>
            <value xsi:type="xs:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">Urbain</value>
        </entry>
        <entry>
            <key>EMVDOUBLON</key>
        </entry>
        <entry>
            <key>EMAIL</key>
            <value xsi:type="xs:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">boubou@smartfocus.com</value>
        </entry>
        <entry>
            <key>DATEJOIN</key>
            <value xsi:type="xs:dateTime" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">2009-11-04T00:00:00+01:00</value>
        </entry>
        <entry>
            <key>FIRSTNAME</key>
            <value xsi:type="xs:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">Laurence</value>
        </entry>
        ...
    </attributes>
</return>
<return>
    <attributes>
        <entry>
            <key>DATEUNJOIN</key>
        </entry>
        <entry>
            <key>EMVHBQ</key>
        </entry>
        <entry>
            <key>DATEOFBIRTH</key>
        </entry>
        <entry>
            <key>MEMBER_ID</key>
            <value xsi:type="xs:long" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">8048601428</value>
        </entry>
        <entry>
            <key>EMVADMIN1</key>
            <value xsi:type="xs:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">Baba</value>
        </entry>
        <entry>
            <key>LASTNAME</key>
            <value xsi:type="xs:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">Urbain</value>

```



```
        </entry>
        <entry>
            <key>EMVDOUBLON</key>
        </entry>
        <entry>
            <key>EMAIL</key>
            <value xsi:type="xs:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">thomas@smartfocus.com</value>
        </entry>
        <entry>
            <key>DATEJOIN</key>
            <value xsi:type="xs:dateTime" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">2009-11-04T00:00:00+01:00</value>
        </entry>
        <entry>
            <key>FIRSTNAME</key>
            <value xsi:type="xs:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">Laurence</value>
        </entry>
        ...
    </attributes>
</return>
</ns2:getMemberByCellphone>
</soap:Body>
</soap:Envelope>
```

getMemberById

This method uses the member ID to retrieve the details of a member.

WSDL location

<https://{server}/apimember/services/MemberService?wsdl>

Note: Ask your Account Manager for your server name.

Input parameters Required parameters	Description	Output parameters	Description
token	The connection token	return	The list of values in the Member table for the member
Id (key)	The ID of the member whose details you want to retrieve		

Error messages

You must fill in the token parameter
You must fill in the id parameter.
An error occurred on the server

SOAP Example

Input

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://api.service.apimember.emailvision.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getMemberById>
      <token>{token}</token>
      <id>357235</id>
    </soapenv:Body>
  </soapenv:Envelope>
```

Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:getMemberByIdResponse xmlns:ns2="http://api.service.apimember.emailvision.com/">
      <return>
        <attributes>
          <entry>
            <key>DATEUNJOIN</key>
          </entry>
          <entry>
            <key>EMVHBQ</key>
          </entry>
          <entry>
            <key>DATEOFBIRTH</key>
          </entry>
        </attributes>
      </return>
    </ns2:getMemberByIdResponse>
  </soap:Body>
</soap:Envelope>
```

```
<entry>
  <key>EMVUNROUTABLE</key>
</entry>
<entry>
  <key>EMAIL_ORIGINE</key>
</entry>
<entry>
  <key>MEMBER_ID</key>
  <value xsi:type="ns4:long" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns4="http://www.w3.org/2001/XMLSchema">357235</value>
</entry>
<entry>
  <key>EMVADMIN1</key>
</entry>
<entry>
  <key>LASTNAME</key>
  <value xsi:type="ns4:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns4="http://www.w3.org/2001/XMLSchema">Smith</value>
</entry>
<entry>
  <key>EMVDOUBLON</key>
</entry>
<entry>
  <key>EMAIL</key>
  <value xsi:type="ns4:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns4="http://www.w3.org/2001/XMLSchema">johnsmith@smartfocus.com</value>
</entry>
<key>DATEJOIN</key>
  <value xsi:type="ns4:dateTime" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:ns4="http://www.w3.org/2001/XMLSchema">2009-03-19T14:57:10+01:00</value>
</entry>
<entry>
  <key>EMVADMIN2</key>
</entry>
<entry>
  <key>EMVADMIN5</key>
</entry>
<entry>
  <key>EMVISIP</key>
</entry>
<entry>
  <key>CLIENTURN</key>
</entry>
<entry>
  <key>SOURCE</key>
</entry>
<entry>
  <key>UPLOAD_ID</key>
</entry>
<entry>
  <key>EMVCELLPHONE</key>
</entry>
<entry>
  <key>EMVADMIN3</key>
</entry>
<entry>
  <key>EMAIL_FORMAT</key>
</entry>
<entry>
  <key>FIRSTNAME</key>
</entry>
<entry>
  <key>TITLE</key>
</entry>
```

```
<entry>
  <key>SEGMENT</key>
</entry>
<entry>
  <key>CODE</key>
</entry>
<entry>
  <key>SEED</key>
</entry>
</attributes>
</soap:Envelope>
```

getListMembersByObj

This method retrieves a list of a maximum of 50 members who match the given criteria.

WSDL location

<https://{server}/apimember/services/MemberService?wsdl>

Note: Ask your Account Manager for your server name.

Input parameters		Output parameters	
Required parameters	Description		Description
token	The connection token	return	The list of members who match the criteria
member	The member object containing the criteria		

Error messages

You must fill in the token parameter

An error occurred on the server

SOAP Example

Input

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://api.service.apimember.emailvision.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getListMembersByObj>
      <token>{token}</token>
      <member>
        <memberUID>FIRSTNAME:John|LASTNAME:Smith|EMAIL:johnsmith@smartfocus.com</memberUID>
      </member>
    </api:getListMembersByObj>
  </soapenv:Body>
</soapenv:Envelope>
```

Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:getListMembersByObjResponse xmlns:n-
s2="http://api.service.apimember.emailvision.com/">
      <return>
        <attributes>
          <entry>
            <key>DATEUNJOIN</key>
          </entry>
          <entry>
            <key>EMVHBQ</key>
          </entry>
          <entry>
            <key>DATEOFBIRTH</key>
          </entry>
        </attributes>
      </return>
    </ns2:getListMembersByObjResponse>
  </soap:Body>
</soap:Envelope>
```

```

        </entry>
        <entry>
            <key>MEMBER_ID</key>
            <value xsi:type="xs:long" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">8048601428</value>
        </entry>
        <entry>
            <key>EMVADMIN1</key>
            <value xsi:type="xs:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">Middle</value>
        </entry>
        <entry>
            <key>LASTNAME</key>
            <value xsi:type="xs:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">Smith</value>
        </entry>
        <entry>
            <key>EMVDOUBLON</key>
        </entry>
        <entry>
            <key>EMAIL</key>
            <value xsi:type="xs:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">johnsmith@smartfocus.com</value>
        </entry>
        <entry>
            <key>DATEJOIN</key>
            <value xsi:type="xs:dateTime" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">2009-11-04T00:00:00+01:00</value>
        </entry>
        <entry>
            <key>FIRSTNAME</key>
            <value xsi:type="xs:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">John</value>
        </entry>
    </attributes>
</return>
<return>
    <attributes>
        <entry>
            <key>DATEUNJOIN</key>
        </entry>
        <entry>
            <key>EMVHBQ</key>
        </entry>
        <entry>
            <key>DATEOFBIRTH</key>
        </entry>
        <entry>
            <key>MEMBER_ID</key>
            <value xsi:type="xs:long" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">8048601429</value>
        </entry>
        <entry>
            <key>EMVADMIN1</key>
            <value xsi:type="xs:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">High</value>
        </entry>
        <entry>
            <key>LASTNAME</key>
            <value xsi:type="xs:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">Smith</value>
        </entry>
        <entry>
            <key>EMVDOUBLON</key>

```

```
        </entry>
        <entry>
            <key>EMAIL</key>
            <value xsi:type="xs:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">johnpaulsmith@smartfocus.com</value>
        </entry>
        <entry>
            <key>DATEJOIN</key>
            <value xsi:type="xs:dateTime" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">2009-11-04T00:00:00+01:00</value>
        </entry>
        <entry>
            <key>FIRSTNAME</key>
            <value xsi:type="xs:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema">John</value>
        </entry>
        ...
    </attributes>
</return>
</ns2:getListMembersByObjResponse>
</soap:Body>
</soap:Envelope>
```

getListMembersByPage

This method retrieves all members page by page. Each page contains 10 members.

WSDL location

<https://{server}/apimember/services/MemberService?wsdl>

Note: Ask your Account Manager for your server name.

Input parameters			
Required parameters	Description	Output parameters	Description
token	The connection token	return	The page object containing the list of members
page	The page number to retrieve		

Error messages

You must fill in the token parameter

An error occurred on the server

SOAP Example

Input

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://api.service.apimember.emailvision.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getListMembersByPage>
      <token>{token}</token>
      <page>1</page>
    </api:getListMembersByPage>
  </soapenv:Body>
</soapenv:Envelope>
```

Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:getListMembersByPageResponse xmlns:ns2="http://api.service.apimember.emailvision.com/">
      <return>
        <list xsi:type="ns2:apiMember" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <attributes>
            <entry>
              <key>DATEUNJOIN</key>
            </entry>
            <entry>
              <key>EMVHBQ</key>
            </entry>
            <entry>
              <key>DATEOFBIRTH</key>
            </entry>
          </attributes>
        </list>
      </return>
    </ns2:getListMembersByPageResponse>
  </soap:Body>
</soap:Envelope>
```



```

        </entry>
        <entry>
            <key>MEMBER_ID</key>
            <value xsi:type="xs:long" xmlns:x-
s="http://www.w3.org/2001/XMLSchema">8057288835</value>
        </entry>
        <entry>
            <key>EMVADMIN1</key>
        </entry>
        <entry>
            <key>LASTNAME</key>
            <value xsi:type="xs:string" xmlns:x-
s="http://www.w3.org/2001/XMLSchema">Smith</value>
        </entry>
        <entry>
            <key>EMVDOUBLON</key>
        </entry>
        <entry>
            <key>EMAIL</key>
            <value xsi:type="xs:string" xmlns:x-
s="http://www.w3.org/2001/XMLSchema">testManyFields@lur.test.com</value>
        </entry>
        <entry>
            <key>FIRSTNAME</key>
            <value xsi:type="xs:string" xmlns:x-
s="http://www.w3.org/2001/XMLSchema">John</value>
        </entry>
        ...
    </attributes>
</list>
<list xsi:type="ns2:apiMember" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <attributes>
        <entry>
            <key>DATEUNJOIN</key>
        </entry>
        <entry>
            <key>EMVHBQ</key>
        </entry>
        <entry>
            <key>DATEOFBIRTH</key>
        </entry>
        <entry>
            <key>MEMBER_ID</key>
            <value xsi:type="xs:long" xmlns:x-
s="http://www.w3.org/2001/XMLSchema">8057288834</value>
        </entry>
        <entry>
            <key>EMVADMIN1</key>
        </entry>
        <entry>
            <key>LASTNAME</key>
            <value xsi:type="xs:string" xmlns:x-
s="http://www.w3.org/2001/XMLSchema">Scott</value>
        </entry>
        <entry>
            <key>EMVDOUBLON</key>
        </entry>
        <entry>
            <key>EMAIL</key>
            <value xsi:type="xs:string" xmlns:x-
s="http://www.w3.org/2001/XMLSchema">testManyFields@lur.test.com</value>
        </entry>
        <entry>

```

```
        <key>EMVADMIN2</key>
      </entry>
    <entry>
      <key>FIRSTNAME</key>
      <value xsi:type="xs:string" xmlns:x-
s="http://www.w3.org/2001/XMLSchema">Jane</value>
    </entry>
    ...
  </attributes>
</list>
...
<nextPage>true</nextPage>
</return>
</ns2:getListMembersByPageResponse>
</soap:Body>
</soap:Envelope>
```

insertMember

This method inserts a new member using only the email address as input (i.e. all other fields will remain empty).

WSDL location

`https://{server}/apimember/services/MemberService?wsdl`

Note: Ask your Account Manager for your server name.

Input parameters	Description	Output parameters	Description
Required parameters			
token	The connection token	return	The job ID of the insertion
email	The email address of the new member		

Error messages

You must fill in the token parameter

You must fill in the email parameter

An error occurred on the server

SOAP Example

Input

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://api.service.apimember.emailvision.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:insertMember>
      <token>{token}</token>
      <email>johnsmith@smartfocus.com</email>
    </api:insertMember>
  </soapenv:Body>
</soapenv:Envelope>
```

Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:insertMemberResponse xmlns:ns2="http://api.service.apimember.emailvision.com/">
      <return>316</return>
    </ns2:insertMemberResponse>
  </soap:Body>
</soap:Envelope>
```

updateMember

This method updates a single field of a member in the Member table using the email address to identify the member.

Note: If more than one member has the same email address, all of them are updated.

WSDL location

<https://{server}/apimember/services/MemberService?wsdl>

Note: Ask your Account Manager for your server name.

Input parameters	Description	Output parameters	Description
Required parameters			
token	The connection token	return	The job ID of the update
email	The email address		
field	The field that will be updated.		
value	The value with which to update the field. Note: For number fields accepting decimal values, you must use the period (.) as the decimal separator. This only applies to fields that are configured to accept decimal values.		

Error messages

You must fill in the token parameter
You must fill in the email parameter
You must fill in the field parameter.
You must fill in the value parameter.
An error occurred on the server

SOAP Example

Input

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://api.service.apimember.emailvision.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:updateMember>
      <token>{token}</token>
      <email>johnsmith@smartfocus.com</email>
      <field>FIRSTNAME</field>
      <value>joe</value>
    </api:updateMember>
  </soapenv:Body>
</soapenv:Envelope>
```

Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:updateMemberResponse xmlns:ns2="http://api.service.apimember.emailvision.com/">
      <return>314</return>
    </ns2:updateMemberResponse>
  </soap:Body>
</soap:Envelope>
```

insertMemberByObj

This method inserts a new member in the Member table by object.

Note: All the values entered for the member must be placed within the **<dynContent>** tags.

- **Dates:** If date values are used, they must be given in one of the following formats:
 - MM/dd/yyyy
 - MM/dd/yyyy HH:mm
 - MM/dd/yyyy HH:mm:ss
 - MMM dd yyyy HH:mm
 - yyyy-MM-dd HH:mm:ss

WSDL location

<https://{server}/apimember/services/MemberService?wsdl>

Note: Ask your Account Manager for your server name.

Input parameters Required parameters	Description	Output parameters	Description
token	The connection token	return	The job ID of the insertion
email	The email address		
dynContent	The dynContent envelope containing the list of fields to be updated and their values.		
entry	The entry envelope containing the field to update and its value.		
key	The field that will be updated.		
value	The value with which to update the field. Note: For number fields accepting decimal values, you must use the period (.) as the decimal separator. This only applies to fields that are configured to accept decimal values.		

Error messages

You must fill in the token parameter

An error occurred on the server

SOAP Example

Input

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://api.service.apimember.emailvision.com/">
```

```
<soapenv:Header/>
<soapenv:Body>
  <api:insertMemberByObj>
    <token>{token}</token>
    <member>
      <dynContent>
        <entry>
          <key>FIRSTNAME</key>
          <value>John</value>
        </entry>
        <entry>
          <key>LASTNAME</key>
          <value>Doe</value>
        </entry>
        <entry>
          <key>TITLE</key>
          <value>Mr</value>
        </entry>
      </dynContent>
      <email>jdoe@smartfocus.com</email>
    </member>
  </api:insertMemberByObj>
</soapenv:Body>
</soapenv:Envelope>
```

Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:insertMemberByObjResponse xmlns:ns2="http://api.service.apimember.emailvision.com/">
      <return>326</return>
    </ns2:insertMemberByObjResponse>
  </soap:Body>
</soap:Envelope>
```

insertOrUpdateMemberByObj

This method searches a specified column of the Member table for a particular value used to identify a member in order to update the member's data (by object). If the member is not found, a new member is created. in the Member table. Any criteria can be used to find the member including one of the fields to be updated.

Note: It should be noted that if the criteria used is the same for multiple members, all of these members will be updated.

Note: All the values that should be entered for the member must be placed within the **<dynContent>** tags.

- **Dates:** If date values are used, they must be given in one of the following formats:
 - MM/dd/yyyy
 - MM/dd/yyyy HH:mm
 - MM/dd/yyyy HH:mm:ss
 - MMM dd yyyy HH:mma
 - yyyy-MM-dd HH:mm:ss

WSDL location

<https://{server}/apimember/services/MemberService?wsdl>

Note: Ask your Account Manager for your server name.

Input parameters Required parameters	Description	Output parameters	Description
token	The connection token	return	The job ID of the update/insertion
memberUID	The field or fields used to identify the member and their values in the following format: <ul style="list-style-type: none"> • Field1:Value1 Field2:Value2 		
dynContent	The dynContent envelope containing the list of fields to be updated and their values.		
entry	The entry envelope containing the field to update and its value.		
key	The field that will be updated.		
value	The value with which to update the field. Note: For number fields accepting decimal values, you must use the period (.) as the decimal separator. This only applies to fields that are configured to accept decimal values.		

SOAP Example

Input

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://api.service.apimember.emailvision.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:insertOrUpdateMemberByObj>
      <token>{token}</token>
      <member>
        <dynContent>
          <entry>
            <key>firstname</key>
            <value>john</value>
          </entry>
          <entry>
            <key>lastname</key>
            <value>smith</value>
          </entry>
        </dynContent>
        <memberUID>email:johnsmith@smartfocus.com</memberUID>
      </member>
    </api:insertOrUpdateMemberByObj>
  </soapenv:Body>
</soapenv:Envelope>
```

Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:insertOrUpdateMemberByObjResponse xmlns:ns2="http://api.service.apimember.emailvision.com/">
      <return>193785</return>
    </ns2:insertOrUpdateMemberByObjResponse>
  </soap:Body>
</soap:Envelope>
```

updateMemberByObj

This method searches for a member and updates the member's data in the Member table by object. If the member is not found, he/she will now be inserted in the table. Any criteria can be used to find the member including one of the fields to be updated.

Note: It should be noted that if the criteria used is the same for multiple members, all of these members will be updated.

Note: All the values that should be entered for the member must be placed within the **<dynContent>** tags.

- **Dates:** If date values are used, they must be given in one of the following formats:
 - MM/dd/yyyy
 - MM/dd/yyyy HH:mm
 - MM/dd/yyyy HH:mm:ss
 - MMM dd yyyy HH:mma
 - yyyy-MM-dd HH:mm:ss

WSDL location

<https://{server}/apimember/services/MemberService?wsdl>

Note: Ask your Account Manager for your server name.

Input parameters Required parameters	Description	Output parameters	Description
token	The connection token	return	The job ID of the update
memberUID	The field or fields used to identify the member and their values in the following format: <ul style="list-style-type: none"> • Field1:Value1 Field2:Value2 		
dynContent	The dynContent envelope containing the list of fields to be updated and their values.		
entry	The entry envelope containing the field to update and its value.		
key	The field that will be updated.		
value	The value with which to update the field. Note: For number fields accepting decimal values, you must use the period (.) as the decimal separator. This only applies to fields that are configured to accept decimal values.		

Error messages

You must fill in the token parameter

An error occurred on the server

SOAP Example

Input

```
<?xml version="1.0"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://api.service.apimember.emailvision.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:updateMemberByObj>
      <token>{token}</token>
      <member>
        <dynContent>
          <entry>
            <key>segment</key>
            <value>segmentFromUpdate</value>
          </entry>
        </dynContent>
        <memberUID>email:johnsmith@smartfocus.com</memberUID>
      </member>
    </api:updateMemberByObj>
  </soapenv:Body>
</soapenv:Envelope>
```

Output

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:updateMemberByObjResponse xmlns:ns2="http://api.service.apimember.emailvision.com/">
      <return>193787</return>
    </ns2:updateMemberByObjResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP Example

An example where the key to update is the member ID.

Input

```
<?xml version="1.0"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://api.service.apimember.emailvision.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:updateMemberByObj>
      <token>{token}</token>
      <member>
        <dynContent>
          <entry>
            <key>firstname</key>
            <value>john</value>
          </entry>
          <entry>
            <key>lastname</key>
            <value>connors</value>
          </entry>
        </dynContent>
        <memberUID>member_id:20291984</memberUID>
      </member>
    </api:updateMemberByObj>
```

```
</soapenv:Body>  
</soapenv:Envelope>
```

Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <ns2:unjoinMemberByObjResponse xmlns:ns2="http://api.service.apimember.emailvision.com/">  
      <return>210135</return>  
    </ns2:unjoinMemberByObjResponse>  
  </soap:Body>  
</soap:Envelope>
```

getMemberJobStatus

This method retrieves the job status (i.e. the status of the member insertion or update) using the job ID.

WSDL location

<https://{server}/apimember/services/MemberService?wsdl>

Note: Ask your Account Manager for your server name.

Input parameters	Description	Output parameters	Description
Required parameters			
token	The connection token	return	The email address of the member
synchroid	The job ID		The job status: <ul style="list-style-type: none"> • Insert • Processing • Processed • Error • Job_Done_Or_Does_Not_Exist

Error messages

You must fill in the token parameter

You must fill in the synchroid parameter.

An error occurred on the server

SOAP Example

Input

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://api.service.apimember.emailvision.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:getMemberJobStatus>
      <token>{token}</token>
      <synchroid>209942</synchroid>
    </api:getMemberJobStatus>
  </soapenv:Body>
</soapenv:Envelope>
```

Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:getMemberJobStatusResponse xmlns:ns2="http://api.service.apimember.emailvision.com/">
      <return>
        <email>johnsmith@smartfocus.com</email>
        <status>Insert</status>
      </return>
    </ns2:getMemberJobStatusResponse>
  </soap:Body>
</soap:Envelope>
```

unjoinMemberByEmail

This method unsubscribes one or more members who match a given email address.

WSDL location

`https://{server}/apimember/services/MemberService?wsdl`

Note: Ask your Account Manager for your server name.

Input parameters	Description	Output parameters	Description
Required parameters			
token	The connection token	return	The job ID for the unsubscribe
email	The email address		

Error messages

You must fill in the token parameter

You must fill in the email parameter

An error occurred on the server

SOAP Example

Input

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://api.service.apimember.emailvision.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:unjoinMemberByEmail>
      <token>{token}</token>
      <email>johnsmith@smartfocus.com</email>
    </api:unjoinMemberByEmail>
  </soapenv:Body>
</soapenv:Envelope>
```

Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:unjoinMemberByEmailResponse xmlns:ns2="http://api.service.apimember.emailvision.com/">
      <return>210134</return>
    </ns2:unjoinMemberByEmailResponse>
  </soap:Body>
</soap:Envelope>
```

unjoinByCellphone

This method unsubscribes one or more members who match a given cellphone number.

WSDL location

<https://{server}/apimember/services/MemberService?wsdl>

Note: Ask your Account Manager for your server name.

Input parameters	Description	Output parameters	Description
Required parameters			
token	The connection token	return	The job ID for the unsubscribe
cellphone	The cellphone number of the member		

Error messages

You must fill in the token parameter

You must fill in the email parameter

An error occurred on the server

SOAP Example

Input

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://api.service.apimember.emailvision.com/">
  <soapenv:Header />
  <soapenv:Body>
    <api:unjoinByCellphone>
      <token>{token}</token>
      <cellphone>33611223344</cellphone>
    </api:unjoinByCellphone>
  </soapenv:Body>
</soapenv:Envelope>
```

Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:unjoinByCellphoneResponse xmlns:ns2="http://api.service.apimember.emailvision.com/">
      <return>{synchro_id}</return>
    </ns2:unjoinByCellphoneResponse>
  </soap:Body>
</soap:Envelope>
```

unjoinMemberById

This method unsubscribes a member who matches a given ID.

WSDL location

<https://{server}/apimember/services/MemberService?wsdl>

Note: Ask your Account Manager for your server name.

Input parameters	Description	Output parameters	Description
Required parameters			
token	The connection token	return	The job ID for the unsubscribe
memberId	The member ID		

Error messages

You must fill in the token parameter

You must fill in the memberId parameter.

An error occurred on the server

SOAP Example

Input

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://api.service.apimember.emailvision.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:unjoinMemberById>
      <token>{token}</token>
      <memberId>673265</memberId>
    </api:unjoinMemberById>
  </soapenv:Body>
</soapenv:Envelope>
```

Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:unjoinMemberByIdResponse xmlns:ns2="http://api.service.apimember.emailvision.com/">
      <return>210133</return>
    </ns2:unjoinMemberByIdResponse>
  </soap:Body>
</soap:Envelope>
```


unjoinMemberByObj

This method unsubscribes a member by object.

Note: It should be noted that if the criteria used is the same for multiple members, all of these members will be unsubscribed.

WSDL location

`https://{server}/apimember/services/MemberService?wsdl`

Note: Ask your Account Manager for your server name.

Input parameters	Description	Output parameters	Description
Required parameters			
token	The connection token	return	The job ID for the unsubscribe
member	The member object		

Error messages

You must fill in the token parameter

An error occurred on the server

SOAP Example

Input

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://api.service.apimember.emailvision.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:unjoinMemberByObj>
      <token>{token}</token>
      <member>
        <memberUID>FIRSTNAME:John|LASTNAME:Smith</memberUID>
      </member>
    </api:unjoinMemberByObj>
  </soapenv:Body>
</soapenv:Envelope>
```

Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:unjoinMemberByObjResponse xmlns:ns2="http://api.service.apimember.emailvision.com/">
      <return>1467</return>
    </ns2:unjoinMemberByObjResponse>
  </soap:Body>
</soap:Envelope>
```

rejoinMemberByEmail

Re-subscribes an unsubscribed member using his/her email address. If there are multiple members with the same email address, they will all be re-subscribed.

Note: The number of rejoins per day is limited to 50 per day to avoid massive rejoins and illegal usage of this method. When the limit is reached, you will receive the following error message:
You have reached the maximum number of rejoins per day.

WSDL location

<https://{server}/apimember/services/MemberService?wsdl>

Note: Ask your Account Manager for your server name.

Input parameters	Description	Output parameters	Description
Required parameters			
token	The connection token	return	The job ID of the update
email	The email address of the member		

Error messages
You must fill in the token parameter
You have reached the maximum number of rejoins per day.
An error occurred on the server

SOAP Example

Input

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://api.service.apimember.emailvision.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:rejoinMemberByEmail>
      <token>{token}</token>
      <email>{emailAddress}</email>
    </api:rejoinMemberByEmail>
  </soapenv:Body>
</soapenv:Envelope>
```

Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:rejoinMemberByEmailResponse xmlns:ns2="http://api.service.apimember.emailvision.com/">
      <return>{synchro_id}</return>
    </ns2:rejoinMemberByEmailResponse>
  </soap:Body>
</soap:Envelope>
```

rejoinByCellphone

Re-subscribes an unsubscribed member using his/her cell phone number. If there are multiple members with the same number, they will all be re-subscribed.

Note: The number of rejoins per day is limited to 50 per day to avoid massive rejoins and illegal usage of this method. When the limit is reached, you will receive the following error message:
You have reached the maximum number of rejoins per day.

WSDL location

<https://{server}/apimember/services/MemberService?wsdl>

Note: Ask your Account Manager for your server name.

Input parameters	Description	Output parameters	Description
Required parameters			
token	The connection token	return	The job ID of the update
cellphone	The cellphone number of the member		

Error messages

You must fill in the token parameter

You have reached the maximum number of rejoins per day.

An error occurred on the server

SOAP Example

Input

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://api.service.apimember.emailvision.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:rejoinByCellphone>
      <token>{token}</token>
      <cellphone>33611223344</cellphone>
    </api:rejoinByCellphone>
  </soapenv:Body>
</soapenv:Envelope>
```

Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:rejoinByCellphoneResponse xmlns:ns2="http://api.service.apimember.emailvision.com/">
      <return>{synchro_id}</return>
    </ns2:rejoinByCellphoneResponse>
  </soap:Body>
</soap:Envelope>
```

rejoinMemberById

Re-subscribes an unsubscribed member using his/her ID.

Note: The number of rejoins per day is limited to 50 per day to avoid massive rejoins and illegal usage of this method. When the limit is reached, you will receive the following error message:
You have reached the maximum number of rejoins per day.

WSDL location

https://{server}/apimember/services/MemberService?wsdl

Note: Ask your Account Manager for your server name.

Input parameters	Description	Output parameters	Description
Required parameters			
token	The connection token	return	The job ID of the resubscription
memberId	The ID of the member		

Error messages

You must fill in the token parameter

You have reached the maximum number of rejoins per day.

An error occurred on the server

SOAP Example

Input

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://api.service.apimember.emailvision.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <api:rejoinMemberById>
      <token>{token}</token>
      <memberId>{memberId}</memberId>
    </api:rejoinMemberById>
  </soapenv:Body>
</soapenv:Envelope>
```

Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:rejoinMemberByIdResponse xmlns:ns2="http://api.service.apimember.emailvision.com/">
      <return>{synchro_id}</return>
    </ns2:rejoinMemberByIdResponse>
  </soap:Body>
</soap:Envelope>
```

Examples

Description: This is an example of a post that will update the email address from johnsmith@smartfocus.com to johnsmith@example.com using the criteria firstname=john, and lastname=smith to find the member.

SOAP example:

insertOrUpdateMemberByObj
Input <pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:api="http://api.service.apimember.emailvision.com/"> <soapenv:Header/> <soapenv:Body> <api:insertOrUpdateMemberByObj> <token> <member> <dynContent> <entry> <key>email</key> <value>johnc@gmail.com</value> </entry> </dynContent> <memberUID>firstname:john lastname:connors</memberUID> </member> </api:insertOrUpdateMemberByObj> </soapenv:Body> </soapenv:Envelope> </pre>
Output <pre> <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body> <ns2:updateMemberByObjResponse xmlns:ns2="http://api.service.apimember.emailvision.com/"> <return>193787</return> </ns2:updateMemberByObjResponse> </soap:Body> </soap:Envelope> </pre>

Reference

WADL

The Web Application Description Language (WADL) is a machine-readable XML-based language that provides a model for describing HTTP-based web applications (such as REST web services).

Web Services

The W3C defines a Web service as a software system designed to support interoperable Machine to Machine interaction over a network. Web services are frequently just Web APIs that can be accessed over a network, such as the Internet, and executed on a remote system hosting the requested services. The W3C Web service definition encompasses many different systems, but in common usage the term refers to clients and servers that communicate XML messages that follow the SOAP-standard. Common in both the field and the terminology is the assumption that there is also a machine readable description of the operations supported by the server, a description in the WSDL. The latter is not a requirement of SOAP endpoint, but it is a prerequisite for automated client-side code generation in the mainstream Java and .NET SOAP frameworks. Some industry organizations, such as the WS-I, mandate both SOAP and WSDL in their definition of a Web service.

WSDL

The Web Services Description Language (WSDL, pronounced 'wiz-dull' or spelled out, 'W-S-D-L') is an XML-based language that provides a model for describing Web services. Version 2.1 has not been endorsed by the World Wide Web Consortium (W3C). Version 2.0, for which several drafts have been released, is expected to become a W3C recommendation. WSDL is an XML-based service description on how to communicate using web services. The WSDL defines services as collections of network endpoints, or ports. WSDL specification provides an XML format for documents for this purpose. WSDL is often used in combination with SOAP and XML Schema to provide web services over the Internet. A client program connecting to a web service can read the WSDL to determine what functions are available on the server. Any special datatypes used are embedded in the WSDL file in the form of XML Schema. The client can then use SOAP to actually call one of the functions listed in the WSDL.

XML

The Extensible Markup Language (XML) is a W3C-recommended general-purpose markup language. The XML recommendation specifies both the structure of XML, and the requirements for XML processors. XML is considered "general-purpose" because it enables anyone to originate and use a markup language for many types of applications and problem domains. Numerous formally defined markup languages are based on XML, such as RSS, MathML, GraphML, XHTML, Scalable Vector Graphics, MusicXML, and thousands of others. XML's primary purpose is to facilitate the sharing of data across different information systems, particularly systems connected via the Internet. It is a simplified subset of Standard Generalized Markup Language (SGML), and is designed to be relatively human-legible.