

# Developer Guide

## Using web services to manage campaigns

---

<b>1</b>	<b>Disclaimer</b> .....	<b>2</b>
<b>2</b>	<b>Document convention</b> .....	<b>2</b>
	2.1 Methods naming & presentation.....	2
<b>3</b>	<b>Working with web service</b> .....	<b>2</b>
	3.1 Pre-requisite.....	2
	3.2 Handling web service answers.....	2
<b>4</b>	<b>Using the web services</b> .....	<b>3</b>
	4.1 CreateCampaign.....	3
	4.2 UpdateCampaign.....	5
	4.3 TestCampaign.....	6
	4.4 AddContacts.....	7
	4.5 SendCampaign.....	8
	4.6 GetCampaigns.....	9
	4.7 GetCampaignItems.....	10
<b>5</b>	<b>Indexes and lexicon</b> .....	<b>11</b>
	5.1 Tags and parameters index.....	11
	5.2 Lexicon.....	11
	5.3 Web Service definition.....	11

---

Version	Date	Issue
1.00	03/03/2009	CreateCampaign / UpdateCampaign / TestCampaign / SendCampaign AddContacts / GetCampaigns / GetCampaignItems methods definition
1.01	12/05/2009	CreateCampaign: schedule date definition
1.02	26/05/2009	CreateCampaign : New properties : DisplayName, ReplyAddress, TestEmails, TestFirstRow CreateCampaign : Added note about Contacts node UpdateCampaign : New properties : DisplayName, ReplyAddress, TestEmails, TestFirstRow AddContacts : New property Mode GetCampaigns : New property UID
1.03	06/07/2009	Added note about Contacts node
1.04	21/08/2009	AddContacts : New property Send GetCampaigns : Added   operator for status GetCampaignItems : Added   operator for status GetCampaignItems : Added UID to result
1.05	25/08/2009	Contacts tag is now optional to create a campaign Check now if there is at least one contact Improved adding contact and (re)sending campaign already sent
1.06	14/09/2009	GetCampaigns : Added ability to filter by profile
1.07	09/06/2011	Document presentation review

## 1 Disclaimer

All rights reserved. emailR is a trademarks or registered trademarks of emailR sprl in the European Union and/or other Countries. Other marks are the properties of their respective owners.

This document is confidential and intended solely for the attention and use of the emailR and concerned customer's technical and management team members. If you're not part of these departments, you are not authorized to and must not disclose, copy, distribute or retain any part of this message. In such event, please delete it.

## 2 Document convention

### 2.1 Methods naming & presentation

The methods are named using the function followed by the object name and identified using **this font**, in bold. If the function returns one single object, its name will be in single form and if it returns several, it will be in plural form. E.g. **GetCampaigns**.

Arguments are presented using *this font* in normal weight, followed by the possible values. If the argument is optional, it is mentioned with the default value.

E.g. Mode (APPEND|OVERWRITE) /Optional=APPEND.

## 3 Working with web service

By working with emailR web service, you can create and manage campaigns from any existing application.

### 3.1 Pre-requisite

To use emailR's web services, you need the following element(s):

- A valid emailR account and administrator credentials

### 3.2 Handling web service answers

All web service methods returns a XML with the following structure:

```
<Result>
  <Code></Code>
  <Message></Message>
  <Detail></Detail>
  <Method></Method>
</Result>
```

The code tag allows identifying the web service result. When a method succeeded, the code tag contains 0.

When the method succeeded, the message tag contains the resulted data, otherwise the message tag is a short description of the code in English language. You can use the code list to create your own messages, for example in another language.

When a system exception occurred, the detail tag is used to store the system message. For managed errors, the tag is empty.

The method tag contains the method name which returned the result.

#### Result code list

0	Success
10000	The XML provided has an invalid structure or some required tags are missing / empty
10001	The status argument is invalid
10002	The schedule date argument is invalid
20000	No active contract have been found
20001	The campaign should have at least one contact

- 20002 Use APPEND mode to add more contacts to a campaign already sent
- 30000 Access is denied
- 30001 The user should be either Administrator or Campaign Manager
- 30002 The web service feature is disabled for the profile
- 30003 The campaign cannot be modified (usually because it is scheduled or sent)
- 30004 The campaign has not been created by web service and cannot be handled
- 30005 The campaign does not match with the account
- 30006 The user has no profile access
- 30007 Free accounts cannot send campaign by web service
- 30008 The account has not enough credit to send the campaign
- 40000 The account has not been found
- 40001 The profile has not been found
- 40002 The campaign has not been found
- 80000 An error occurred while creating the contact file
- 80001 An error occurred while reading the contact file
- 90000 A general failure occurred (unmanaged)

## 4 Using the web services

The exchange channel is named **CampaignSvc** and exposes the following methods: **CreateCampaign**, **UpdateCampaign**, **TestCampaign**, **SendCampaign**, **AddContacts**, **GetCampaigns** and **GetCampaignItems**.

All of the methods expect one parameter as XML string. Here are the definition and description of the methods and parameters.

Keep in mind to neutralize forbidden characters in XML content. You can use `<![CDATA[...]]>` directives around your content. Try to use it only when required to minimize data transfer.

Example:

```
<Name><![CDATA[This is the campaign name]]></Name>
```

### 4.1 CreateCampaign

This method allows creating a new campaign. When the method succeeds, the result's message tag contains the UID of the campaign that was created. You will use this UID in other methods to identify your campaign.

The status of a campaign created by this method will be set as Pending.

#### 4.1.1 Structure

```
<ROOT>
  <Account>
    <UID/>
    <Login/>
    <Password/>
  </Account>
  <Profile>
    <UID/>
    <DisplayName/>
    <ReplyAddress/>
    <TestEmails/>
  </Profile>
  <Campaign>
```

```

<Name/>
<ScheduleDate/>
<Subject/>
<BodyHtml/>
<BodyText/>
<TestFirstRow/>
<Contacts>
  <Contact>
    <Email/>
    <UID/>
    <DisplayName/>
  </Contact>
  ...
</Contacts>
</Campaign>
</ROOT>

```

## 4.1.2 Description

### 4.1.2.1 Account node

- **UID:** The account UID (you can find it in the Profile detail screen)
- **Login:** A valid login for the account
- **Password:** A valid password for the account

### 4.1.2.2 Profile node

- **UID:** The profile UID (you can find it in the Profile detail screen). Note that this value is required even if `DisplayName`, `ReplyAddress` and `TestEmails` are supplied.
- **DisplayName:** This tag is optional and allows overriding the default value stored in the profile. If you omit this tag, the profile value will be used.
- **ReplyAddress:** This tag is optional and allows overriding the default value stored in the profile. If you omit this tag, the profile value will be used.
- **TestEmails:** This tag is optional and allows overriding the default value stored in the profile. If you omit this tag, the profile value will be used. Use semicolon to delimit multiple email addresses. E.g. "test.email1@yourCompany.com;test.email2@yourcompany.com"

### 4.1.2.3 Campaign node

- **Name:** The campaign name
- **ScheduleDate:** The schedule date in format `yyyymmddhhmm`. This tag is optional. If provided and empty, it means no schedule date. When specifying minute of the schedule date, you should use 0, 15, 30 or 45 (like in extranet interface).
- **Subject:** The campaign subject.
- **BodyHtml:** The campaign html content. The tag is required but if empty, it means that campaign is plain text.
- **BodyText:** The campaign plain text content. When the campaign is HTML, you can provide here the alternate content.
- **TestFirstRow (0|1) /Optional=0:** The tag is optional. Set as 1 if you want to mail merge the template with the first "contact" row values when sending a test. When the value is 0 or omitted, emailR will replace the mail merge fields `{%myColumn%}` by `{field myColumn}`. Please note that any `{%myColumn%}` tag found means that the system did not recognise the column. Check that the column exists and that it has the right name.

### 4.1.2.4 Contacts tag: A collection of Contact tags.

This tag is optional. You can use `AddContacts` later to add your contacts.

- **Email:** The email of the recipient.
- **UID:** The unique identifier of the recipient. This tag is optional.
- **DisplayName:** The display name of the recipient. This tag is optional.

*Note about Contacts tags:*

- Each Contact tag may contain extra tags that will be used as merge fields (if you want to insert personalized data in each email).  
Example: You can add `<myColumn/>` tag with value for each Contact tag. Then in the template, insert a merge field `{%myColumn%}`. The merge field will be substituted with its matching value in each email.
- Default data delimiter is double quote (Please take care to avoid double quote inside contact tag data)

### 4.1.3 Example

```
<ROOT>
  <Account>
    <UID>655ABD2A-7153-4017-9032-8CD67E06151F</UID>
    <Login>john.smith@dummy.com</Login>
    <Password>123</Password>
  </Account>
  <Profile>
    <UID>142405C9-D4AD-4D85-A253-81E0E10F808E</UID>
    <TestEmails>another_test_email@dummy.com</TestEmails>
  </Profile>
  <Campaign>
    <Name>This is the campaign name</Name>
    <ScheduleDate>200905011845</ScheduleDate>
    <Subject>This is the campaign subject]]</Subject>
    <BodyHtml><![CDATA[<body>This is the HTML content</body>]]</BodyHtml>
    <BodyText>This is PLAIN TEXT content</BodyText>
    <TestFirstRow>1</TestFirstRow>
    <Contacts>
      <Contact>
        <Email>alan.smith@dummy.com</Email>
        <UID>12345</UID>
        <DisplayName>Alan Smith</DisplayName>
      </Contact>
    </Contacts>
  </Campaign>
</ROOT>
```

## 4.2 UpdateCampaign

This method allows updating an existing campaign. Only pending campaigns that have been created with the **CreateCampaign** method could be updated.

### 4.2.1 Structure

```
<ROOT>
  <Account>
    <UID/>
    <Login/>
    <Password/>
  </Account>
  <Profile>
    <UID/>
    <DisplayName/>
    <ReplyAddress/>
    <TestEmails/>
  </Profile>
  <Campaign>
    <UID/>
    <Name/>
    <ScheduleDate/>
    <Subject/>
    <BodyHtml/>
    <BodyText/>
```

```
<TestFirstRow/>
</Campaign>
</ROOT>
```

## 4.2.2 Description

Some tags are omitted if they already have been detailed previously.

### 4.2.2.1 Profile node

The tag is optional and should be provided only if the profile should be changed.

You can supply either `UID` tag to change the profile or `DisplayName`, `ReplyAddress`, `TestEmails` tags to override one or more profile settings.

### 4.2.2.2 Campaign node

`UID`: The campaign UID (returned by the `CreateCampaign` method)

All others tags from Campaign node are optional for this method. You should provide them only when a change is required.

## 4.2.3 Example

```
<ROOT>
  <Account>
    <UID>655ABD2A-7153-4017-9032-8CD67E06151F</UID>
    <Login>john.smith@dummy.com</Login>
    <Password>123</Password>
  </Account>
  <Campaign>
    <UID>3A4B0FC9-D9AD-4A85-B253-8FD0E10F808E</UID>
    <Name>This is the updated campaign name</Name>
    <Subject>This is the updated campaign subject]]</Subject>
  </Campaign>
</ROOT>
```

## 4.3 TestCampaign

This method allows testing a campaign. When the test will be complete, the campaign returns to `Pending mode`.

### 4.3.1 Structure

```
<ROOT>
  <Account>
    <UID/>
    <Login/>
    <Password/>
  </Account>
  <Campaign>
    <UID/>
  </Campaign>
</ROOT>
```

### 4.3.2 Description

All tags have been detailed previously.

### 4.3.3 Example

```
<ROOT>
  <Account>
    <UID>655ABD2A-7153-4017-9032-8CD67E06151F</UID>
    <Login>john.smith@dummy.com</Login>
    <Password>123</Password>
  </Account>
  <Campaign>
    <UID>3A4B0FC9-D9AD-4A85-B253-8FD0E10F808E</UID>
```

```
</Campaign>
</ROOT>
```

## 4.4 AddContacts

This method allows adding or updating recipients in an existing campaign.

If an email already exists, the `DisplayName` and/or `UID` will be updated if provided.

You can also decide to append or overwrite the existing contacts before.

You can also send the campaign to the new added contacts.

### 4.4.1 Structure

```
<ROOT>
  <Account>
    <UID/>
    <Login/>
    <Password/>
  </Account>
  <Campaign>
    <UID/>
    <Mode/>
    <Send/>
    <Duplicate/>
    <Contacts>
      <Contact>
        <Email/>
        <UID/>
        <DisplayName/>
      </Contact>
      ...
    </Contacts>
  </Campaign>
</ROOT>
```

### 4.4.2 Description

#### 4.4.2.1 Campaign node

`Mode (APPEND|OVERWRITE) /Optional=APPEND`: The mode can be `OVERWRITE` to delete all contacts before or `APPEND` to append new contacts to existing contacts. This tag is optional. You can omit the tag if you want to append contacts (default).

`Send (0|1) /Optional=0`: When the campaign has already been sent, this tag allows (re)sending the campaign to the new contacts. You can omit this tag if you just want to add contacts.

Note that if the campaign has not been sent yet, this tag has no action. In that case, you should use the `SendCampaign` method.

`Duplicate (0|1) /Optional=1`: Allow duplicates checks if the email address already exists in the distribution list. If the value is 0, an existing email won't be added twice, and if the value is 1, then an email can be added many times.

*Notes about Contacts tags:*

- Each Contact tag may contain extra tags that will be used as merge fields (if you want to insert personalized data in each email).  
Example: You can add `<myColumn/>` tag with value for each Contact tag. Then in the template, insert a merge field `{%myColumn%}`. The merge field will be substituted with its matching value in each email.
- When using `APPEND` mode, you cannot supply merge fields that have not been defined in `CreateCampaign` method (they will be ignored). This is due to existing contacts which does not have value(s) for these new merge field(s).

### 4.4.3 Example

```
<ROOT>
  <Account>
    <UID>655ABD2A-7153-4017-9032-8CD67E06151F</UID>
    <Login>john.smith@dummy.com</Login>
    <Password>123</Password>
  </Account>
  <Campaign>
    <UID>3A4B0FC9-D9AD-4A85-B253-8FD0E10F808E</UID>
    <Mode>OVERWRITE</Mode>
    <Send>1</Send>
    <Contacts>
      <Contact>
        <Email>alan.smith@dummy.com</Email>
        <UID>99999</UID>
        <DisplayName>Alan E. Smith</DisplayName>
      </Contact>
      <Contact>
        <Email>sarah.smith@dummy.com</Email>
        <UID>77777</UID>
        <DisplayName>Sarah Smith</DisplayName>
      </Contact>
    </Contacts>
  </Campaign>
</ROOT>
```

## 4.5 SendCampaign

This method allows finalizing a campaign for sending.

**Important:** After this method is invoked, the campaign cannot be modified. All mails sent using the **SendCampaign** method are billable.

### 4.5.1 Structure

```
<ROOT>
  <Account>
    <UID/>
    <Login/>
    <Password/>
  </Account>
  <Campaign>
    <UID/>
    <ScheduleDate/>
  </Campaign>
</ROOT>
```

### 4.5.2 Description

All tags have been detailed previously.

#### 4.5.2.1 Campaign

- `ScheduleDate` is optional and should be provided only if it should be changed (See **CreateCampaign** method for more information about date format)

### 4.5.3 Example

```
<ROOT>
  <Account>
    <UID>655ABD2A-7153-4017-9032-8CD67E06151F</UID>
    <Login>john.smith@dummy.com</Login>
    <Password>123</Password>
  </Account>
  <Campaign>
    <UID>3A4B0FC9-D9AD-4A85-B253-8FD0E10F808E</UID>
```

```
</Campaign>
</ROOT>
```

## 4.6 GetCampaigns

This method allows retrieving one or more campaigns for an account. It returns only the campaigns for which the user has profiles rights.

In case of success, the campaign list XML is contained in Message tag from Result.

### 4.6.1 Structure

```
<ROOT>
  <Account>
    <UID/>
    <Login/>
    <Password/>
  </Account>
  <Campaign>
    <UID/>
    <Status/>
  </Campaign>
  <Profile>
    <UID/>
  </Profile>
</ROOT>
```

### 4.6.2 Description

Some tags are omitted if they already have been detailed previously.

#### Campaign node

**UID:** This tag is optional. It allows searching for a particular campaign (based on its ID). If you omit this tag, you will receive all campaigns based on Status.

**Status (ALL | PENDING | SCHEDULED | PROCESSING | QUEUED | COMPLETED | ERROR) :** The status of the campaign to return in the list. The status can be ALL for all status or one of the following:

PENDING, SCHEDULED, PROCESSING, QUEUED, COMPLETED, ERROR.

You can also use the | operator to specify several status (example: PENDING | COMPLETED)

**Profile node:** The tag is optional and allows filtering the campaigns on a particular profile. You can omit the tag if you want all the campaigns.

### 4.6.3 Example

```
<ROOT>
  <Account>
    <UID>655ABD2A-7153-4017-9032-8CD67E06151F</UID>
    <Login>john.smith@dummy.com</Login>
    <Password>123</Password>
  </Account>
  <Campaign>
    <Status>SCHEDULED</Status>
  </Campaign>
  <Profile>
    <UID>142405C9-D4AD-4D85-A253-81E0E10F808E</UID>
  </Profile>
</ROOT>
```

### 4.6.4 Structure returned in result's message tag

```
<ROOT>
  <Campaign>
    <UID/>
    <Name/>
    <Subject/>
    <DisplayName/>
```

```

    <ReplyAddress/>
    <Recipients/>
    <CreateDate/>
    <UpdateDate/>
    <ScheduleDate/>
    <ErrorMessage/>
    <Status/>
  </Campaign>
  ...
</ROOT>

```

- **UID:** The campaign UID
- **Name:** The campaign name
- **Subject:** The campaign subject
- **DisplayName:** The name that will appear in the email.
- **ReplyAddress:** The reply address for the email.
- **Recipients:** The number of recipients
- **CreateDate:** The campaign create date
- **UpdateDate:** The last campaign update date (tag may be missing)
- **ScheduleDate:** The campaign schedule date (tag may be missing)
- **ErrorMessage:** If campaign status is Error, this tag may supply more information about the error
- **Status:** The campaign status

## 4.7 GetCampaignItems

This method allows retrieving the recipients that have been sent and their status.

### 4.7.1 Structure

```

<ROOT>
  <Account>
    <UID/>
    <Login/>
    <Password/>
  </Account>
  <Campaign>
    <UID/>
  </Campaign>
  <CampaignItem>
    <Status/>
  </CampaignItem>
</ROOT>

```

### 4.7.2 Description

Some tags are omitted if they already have been detailed previously.

**Status (ALL | SENT | CLICKED | ERRORSOFT | ERRORHARD | QUARANTINE) :** The status of the campaign items to return in the list. The status can be ALL for all status or one of the following: SENT, READ, CLICKED, ERRORSOFT, ERRORHARD, QUARANTINE.

You can also use the | operator to specify several status (example: SENT | READ)

### 4.7.3 Example

```

<ROOT>
  <Account>
    <UID>655ABD2A-7153-4017-9032-8CD67E06151F</UID>
    <Login>john.smith@dummy.com</Login>
    <Password>123</Password>
  </Account>
  <Campaign>
    <UID>3A4B0FC9-D9AD-4A85-B253-8FD0E10F808E</UID>
  </Campaign>

```

```
<CampaignItem>
  <Status>ERRORSOFT</Status>
</CampaignItem>
</ROOT>
```

#### 4.7.4 Structure returned in result's message tag

```
<ROOT>
  <CampaignItem>
    <Email/>
    <UID/>
    <Date/>
    <Status/>
  </CampaignItem>
  ...
</ROOT>
```

- Email: The recipient email
- UID: The unique identifier for the Campaign item
- Date: The date of the action
- Status: The status of the recipient

## 5 Indexes and lexicon

### 5.1 Tags and parameters index

AddContacts, 7	GetContacts, 3
CreateCampaign, 3	GetSegments, 3
CRMSvc, 3	SendCampaign, 8
GetCampaignItems, 10	TestCampaign, 6
GetCampaigns, 9	UpdateCampaign, 5

### 5.2 Lexicon

#### 5.2.1 Web services

A '**Web service**' (also **Web Service**) is defined by the W3C 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.

#### 5.3 Web Service definition

You can find the WSDL definition at the following URL:

<http://extranet.emailr.com/integration/CampaignSvc.asmx?WSDL>