

# Developer Guide

## Using web services for segments and recipients

---

<b>1</b>	<b>Disclaimer</b> .....	<b>1</b>
<b>2</b>	<b>Working with the web service</b> .....	<b>1</b>
2.1	Pre-requisite.....	1
2.2	Deploying the web service.....	1
<b>3</b>	<b>Indexes and lexicon</b> .....	<b>2</b>
3.1	Tags and parameters index.....	2
3.2	Lexicon.....	3
3.3	Web Service definition.....	3

---

Version	Date	Issue
1.00	12/10/2008	GetSegments and GetContacts methods definition

## 1 Disclaimer

All rights reserved. emailR is a trademarks or registered trademarks of Crosscom 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 Working with the web service

By working with web service, you can interface any existing application with emailR. EmailR will connect to your remote application to collect information regarding available segments and related recipients.

### 2.1 Pre-requisite

To use web services, you need the following elements:

- A valid emailR account and administrator credentials
- A web server accessible by the emailR server that will publish the web service and methods.

### 2.2 Deploying the web service

You must publish a web services connected to your CRM application.

The exchange channel must be named **CRMSvc**, and is defined between the CRM and emailR to allow the 2 applications to communicate through two methods: **GetSegments** and **GetContacts**.

#### 2.2.1 GetSegments

This method allows external applications to publish, and emailR to collect information related to segments (distribution lists). **GetSegments** does not require any parameter. The return value of the method is a XML defined hereunder:

```
<?xml version='1' encoding='utf-8' ?>
<Segments>
  <Segment>
```

```

        <ID/>
        <Name/>
    </Segment>
    ...
</ Segments>

```

Example :

```

<?xml version='1' encoding='utf-8' ?>
<Segments>
  <Segment>
    <ID>4565</ID>
    <Name>My French contacts</Name>
  </ Segment>
  <Segment>
    <ID>1265</ID>
    <Name>My English contacts</Name>
  </ Segment>
</ Segments>

```

### 2.2.2 GetContacts

This method allows external applications to publish, and emailR to collect information related to contacts contained in a specified segment. **GetContacts** requires one parameter: **segmentID**. The return value of the method is a XML defined hereunder:

```

<?xml version='1' encoding='utf-8' ?>
<Contacts>
  <Contact>
    <...>
    ...
  </Contact>
  ...
</Contacts>

```

The parameter **segmentID** is the id of the specific segment you want to retrieve contact details.

The child tags contained in the contact tag are the fields defined within the segment. For e.g. we can have tags like: <Email></Email><FirstName></FirstName>...

Ex :

```

<?xml version='1' encoding='utf-8' ?>
<Contacts>
  <Contact>
    <Email>john.smith@emailr.com</Email>
    <FirstName>John</FirstName>
    <LastName>Smith</LastName>
  </Contact>
  <Contact>
    <Email>ronald.charles@emailr.com</Email>
    <FirstName>Ronald</FirstName>
    <LastName>Charles</LastName>
  </Contact>
</Contacts>

```

## 3 Indexes and lexicon

### 3.1 Tags and parameters index

CRMSvc, 1  
GetContacts, 1, 2

GetSegments, 1  
segmentID, 2

## 3.2 Lexicon

### 3.2.1 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.

### 3.3 Web Service definition

Here is the sample WSDL of the required web service:

```
<?xml version="1.0" encoding="utf-8" ?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://yourOwnNameSpace.com"
  xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  targetNamespace="http://yourOwnNameSpace.com" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
  <s:schema elementFormDefault="qualified" targetNamespace="http://yourOwnNameSpace.com">
  <s:element name="GetSegments">
    <s:complexType />
  </s:element>
  <s:element name="GetSegmentsResponse">
    <s:complexType>
    <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="GetSegmentsResult" type="s:string" />
    </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="GetContacts">
    <s:complexType>
    <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="segmentID" type="s:string" />
    </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="GetContactsResponse">
    <s:complexType>
    <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="GetContactsResult" type="s:string" />
    </s:sequence>
    </s:complexType>
  </s:element>
  </s:schema>
  </wsdl:types>
  <wsdl:message name="GetSegmentsSoapIn">
    <wsdl:part name="parameters" element="tns:GetSegments" />
  </wsdl:message>
  <wsdl:message name="GetSegmentsSoapOut">
    <wsdl:part name="parameters" element="tns:GetSegmentsResponse" />
  </wsdl:message>
  <wsdl:message name="GetContactsSoapIn">
    <wsdl:part name="parameters" element="tns:GetContacts" />
  </wsdl:message>
  <wsdl:message name="GetContactsSoapOut">
    <wsdl:part name="parameters" element="tns:GetContactsResponse" />
  </wsdl:message>
  <wsdl:portType name="CRMSvcSoap">
    <wsdl:operation name="GetSegments">
    <wsdl:input message="tns:GetSegmentsSoapIn" />
    <wsdl:output message="tns:GetSegmentsSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetContacts">
    <wsdl:input message="tns:GetContactsSoapIn" />
```

```

        <wsdl:output message="tns:GetContactsSoapOut" />
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="CRMSvcSoap" type="tns:CRMSvcSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="GetSegments">
        <soap:operation soapAction="http://yourOwnNameSpace.com/SendMail" style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetContacts">
        <soap:operation soapAction="http://yourOwnNameSpace.com/GetContacts" style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="CRMSvcSoap12" type="tns:CRMSvcSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="GetSegments">
        <soap12:operation soapAction="http://yourOwnNameSpace.com/GetSegments" style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetContacts">
        <soap12:operation soapAction="http://yourOwnNameSpace.com/GetContacts" style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="CRMSvc">
    <wsdl:port name="CRMSvcSoap" binding="tns:CRMSvcSoap">
        <soap:address location="http://194.242.45.61/EmailRCRM/Integration/CRMSvc.asmx" />
    </wsdl:port>
    <wsdl:port name="CRMSvcSoap12" binding="tns:CRMSvcSoap12">
        <soap12:address location="http://194.242.45.61/EmailRCRM/Integration/CRMSvc.asmx" />
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```