

Message Studio SOAP API

Reference Guide

9-5

Selligent

The contents of this manual cover material copyrighted by Selligent.
This manual cannot be reproduced, in part or in whole, or distributed or transferred by means electronic or mechanical,
or photocopied, without the prior written consent of a representative from Selligent.

Copyright © 2016

The contents of this manual cover material copyrighted by Selligent. Selligent reserves all intellectual property rights on the manual, which should be treated as confidential information as defined under the agreed upon software licence/lease terms and conditions.

The use and distribution of this manual is strictly limited to authorized users of the Selligent Interactive Marketing Software (hereafter the "Software") and can only be used for the purpose of using the Software under the agreed upon software licence/lease terms and conditions. Upon termination of the right to use the Software, this manual and any copies made must either be returned to Selligent or be destroyed, at the latest two weeks after the right to use the Software has ended.

With the exception of the first sentence of the previous paragraph, no part of this manual may be reprinted or reproduced or distributed or used in any form or by any electronic, mechanical or other means, not known or hereafter invented, included photocopying and recording, or in any information storage or retrieval or distribution system, without the prior permission in writing from Selligent.

Selligent will not be responsible or liable for any accidental or inevitable damage that may result from unauthorized access or modifications.

User is aware that this manual may contain errors or inaccuracies and that it may be revised without advance notice. This manual is updated frequently.

Selligent welcomes any recommendations or suggestions regarding the manual, as it helps to continuously improve the quality of our products and manuals.

Contents

Contents	3
Message Studio SOAP API Introduction	7
WS API Supported Functionality	7
When to Use the Web Services API	9
Platform Maintenance and Downtime	10
System Requirements	10
Hardware Requirements	10
Knowledge Requirements	10
Backward Compatibility and Versioning	10
Standards Compliance	11
API Resources	11
Developer Resources	12
HTML Description	12
WSDL	12
XSD	12
Mailing Action Status	13
Network Considerations	13
Sample Code	13
Development Environments	15
Java CXF Clients	15
Generating Java Client Stubs	15
Constructing a Java SOAP Client	17
Java Security Token Requirements	17
C# Clients	22
Generating C# Client Stubs	22
Creating a C# Mailing Service	22
Constructing a C# SOAP Client	25
C# Security Token Requirements	27
Ruby Clients	28
Generating Ruby Client Stubs	29
Constructing a Ruby SOAP Client	29
Security Considerations	31
Securing API Communications with SSL	32
User Authentication	32
User Credentials	32
Security Header	33
Sample SOAP Header	33
Firewall	34
Iptables Example	34
Error Handling	35

Frequently Asked Questions	37
General Notes	37
Known Conditions	38
Getting Started	39
Step 1: Obtain the WSDL	39
Step 2: Obtain the Sample Code	39
Step 3: Obtain a SOAP Client Library for your Development Environment	40
Step 4: Walk-through a Simple Example	40
API Object Descriptions	41
API Operations	47
addGlobalUndeliverable()	48
addRecords()	50
archiveMailing()	54
cancel()	56
cancelRefreshRecords()	58
close()	60
copy()	62
create()	64
dedupeRecords()	71
delete()	73
exportRecords()	76
fetchLinks()	78
findImport()	83
get()	85
getAllEasByMailingId()	89
getImportDetails()	90
getImportStatus()	92
getGlobalUndeliverable()	94
getProgramNodes()	96
getRecords()	98
getStatistics()	103
getTxnEasInfo()	106
getTxnMailingHandle()	108
importContent()	110
importFtpFile()	112
importMessagePart()	114
launch()	116
list()	118
load()	126
pause()	128
personalize()	130
reactivateGlobalUndeliverable()	134
refreshRecords()	136

removeRecords()	138
resume()	141
schedule()	143
send()	145
startImport()	145
test()	147
txnSend()	149
update()	152
upsertRecord()	156
validateXsl()	160
Sample Java Code	162
Sample C# Code	174
Sample XML Code	183
Channel Standard Mailing Create Request (v2)	185
Channel Standard Mailing Create Response (v2)	187
Channel Standard Mailing Get Request (v2)	188
Channel Standard Mailing Get Response (v2)	189
Channel Transactional Mailing Create Request (v2)	191
Channel Transactional Mailing Create Response (v2)	193
Channel Transactional Mailing Get Request (v2)	194
Channel Transactional Mailing Get Response (v2)	195
Channel Template Create Request (v2)	197
Channel Template Get Request (v2)	198
Channel Template Create Response (v2)	199
Channel Template Get Response (v2)	200
External Data Source Create Request (v2)	202
Standard Mailing Create Request (v2)	204
Standard Mailing Create Response (v2)	207
Standard Mailing Get Request (v2)	207
Standard Mailing Get Response (v2)	208
Standard Mailing List Request (v2)	211
Standard Mailing List Response (v2)	212
Transactional Mailing Create Request (v2)	213
Transactional Mailing Create Response (v2)	215
Transactional Mailing Get Request (v2)	216
Transactional Mailing Get Response (v2)	217
Template Create Request (v2)	219
Template Create Response (v2)	221
Template Get Request (v2)	222
Template Get Response (v2)	223
Template List Request (v2)	226
Template List Response (v2)	227
Template Update Request (v2)	228

Template Update Response - Success (v2)	230
Template Update Response - Failure (v2)	231
User Get Request (v2)	232
User Get Response (v2)	233
User List Request (v2)	234
User List Response (v2)	235
Mailing Create Request (v1)	236
Mailing Create Response (v1)	238
Mailing Get Request (v1)	239
Mailing Get Response (v1)	240
Mailing List Request (v1)	241
Mailing List Response (v1)	242
Template Create Request (v1)	243
Template Create Response (v1)	244
Template Get Request (v1)	245
Template Get Response (v1)	246
Template List Request (v1)	248
Template List Response (v1)	249
User Get Request (v1)	250
User Get Response (v1)	251
User List Request (v1)	252
User List Response (v1)	253
Index	255

Message Studio SOAP API Introduction

The Message Studio Web Services Application Programming Interface (WS API) provides access to the data, content, and campaign management assets and features that are available in a Message Studio instance, allowing developers to build custom messaging applications and closely integrated business systems.

Topics:

- **[WS API Supported Functionality on page 7](#)**
- **[When to Use the Web Services API on page 9](#)**
- **[System Requirements on page 10](#)**
- **[API Resources on page 11](#)**
- **[Development Environments on page 15](#)**
- **[Security Considerations on page 31](#)**
- **[Error Handling on page 35](#)**
- **[Frequently Asked Questions on page 37](#)**
- **[Getting Started on page 39](#)**
- **[API Object Descriptions on page 41](#)**
- **[API Operations on page 47](#)**

WS API Supported Functionality

The Message Studio Web Services Application Programming Interface (WS API) provides access to the data, content, and campaign management assets and features that are available in a Message Studio instance, allowing developers to build custom messaging applications and closely integrated business systems.

Figure 1: Web Services API Concept



The WS API supports a subset of the functionality of the Message Studio user interface as described in the following table.

Functional Area	Supported Message Studio Function
Triggering Transactional Mailings and Lifecycle Marketing Messages	<ul style="list-style-type: none"> • Adding participants to a program • Triggering a transactional message send • Loading, pausing, and resuming transactional mailings
Data Management	<ul style="list-style-type: none"> • Creating, copying, and deleting data sources, seed lists, suppression lists, and targets
Content Management	<ul style="list-style-type: none"> • Creating, copying, and deleting message templates • Testing message templates • Validating XSL-based templates • Managing message attachments, content blocks, and dynamic content rules
Campaign Management	<ul style="list-style-type: none"> • Creating, copying, and deleting batch and transactional mailings • Launching test mailings • Launching, pausing, and resuming mailings • Retrieving mailing status and performance metrics • Closing and archiving mailings

Functional Area	Supported Message Studio Function
User, Organization, and System Management	<ul style="list-style-type: none"> Retrieving and listing users, roles, and organizations in Message Studio instance Retrieving system information such as bounce, from, reply-to addresses, campaign categorization, and media servers
OpenChannels	<ul style="list-style-type: none"> List available Channels Set up Batch and Transactional mailings for Channels Use seed and suppression lists for Channels

When to Use the Web Services API

Selligent provides three distinct API sets for its product stack:

- Message Studio Web Services API
- Email Integration Server (EIS) Batch API
- Email Integration Server (EIS) Transactional API

In general, you should build your API client applications using the Message Studio API since it offers a modern standards-based Web services programming interface and the broadest set of capabilities for Message Studio customization and automation. You should avoid using both the Message Studio and EIS APIs together. For example, assets created with EIS APIs will not be available via the Message Studio UI or API.

In some cases, you may need to use the EIS-based APIs:

- **High Performance Transactional Messaging**—If you need to support a very large volume of transactional messages (greater than 500,000 messages per hour), then you may need to use the EIS Transactional API. Please contact Selligent Solutions Consulting Group to validate the appropriate API for this use case.
- **EIS/EDS-Only Customers**—If you are not using the full Message Studio stack, then you can use the EIS APIs to create and manage mailings. Please refer to the *Selligent EIS API Reference Guide* for more information.

Platform Maintenance and Downtime

The Selligent servers will usually need to have maintenance downtimes on a recurring schedule. During these downtimes, the Message Studio API is not available. Attempts to create an API session will return a system downtime exception and client applications will need to take the appropriate action, which may include alerts to support staff, integration job queuing, and/or scheduled re-attempts.

System Requirements

Please make sure you are familiar with the system requirements before beginning.

Hardware Requirements

Message Studio API's are lightweight and do not require significant processing power or memory. However, a high-speed Internet connection is recommended for the services to be most effective.

Knowledge Requirements

- Some familiarity with SOAP and web services architecture.
- Ability to program in some language.
- Familiarity with XML may be helpful for troubleshooting.

Important: When testing with XML calls using curl, make sure the call uses "Content-Type: text/xml" and character set "utf-8".

For example:

```
curl -v -d @/data1/ff.xml -X POST -k  
"https://localhost:4443/sm/services/mailing/v2" -H "Content-Type:  
text/xml; charset=utf-8"
```

Backward Compatibility and Versioning

The system supports backward compatibility as new versions of the Selligent Web Services API are released. The WS API is backward compatible in that an application created to work with a given WS API version will continue to work with that same WS API version in future platform releases.

Specifically, the WS API endpoint URL is versioned so that different WS API versions have different endpoint URLs. Your applications will continue to work with the WS API endpoint URLs of previous releases and you have the opportunity to migrate your client applications to the newer WS API version endpoint URLs to leverage enhanced functionality on a schedule that meets your needs.

Selligent does not guarantee that applications written against one WS API version will work with future API versions, since changes in method signatures and data representations are often required as we continue to enhance the Selligent platform. However, we strive to keep the WS API consistent from version to version with minimal if any changes required to port applications to newer WS API versions. When an API version is to be deprecated, advance end-of-life notice will be given at least 12 months before support for the API version ends.

Selligent currently supports the following versions of the WS API:

WS API Version	Endpoint	End-of-Life Date
1	/sm/services/mailing/2009/03/02	2017-07
2	/sm/services/mailing/v2	None announced

Standards Compliance

The Selligent WS API is in compliance with the following specifications.

Standards Name	Website
Simple Object Access Protocol (SOAP) 1.1	http://www.w3.org/TR/2000/NOTE-SOAP-20000508/
Web Service Description Language (WSDL) 1.1	http://www.w3.org/TR/2001/NOTE-wsdl-20010315
WS-I Basic Profile 1.1	http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html

API Resources

The Message Studio web service is organized as a set of objects defined in the XSD file, and a set of operations defined in the WSDL file. This API uses the same authentication and authorization model as Message Studio.

Developer Resources

API Developers can access the following information online:

- **Developer Central:**

<http://spark.strongview.com/community/developers?view=overview>

- **Code Sharing:**

http://spark.strongview.com/community/developers/code_sharing

- **API Q&A:**

<http://spark.strongview.com/community/developers/q%26a>

HTML Description

Once you have installed Message Studio, you can get an HTML description of the service file by using:

```
https://hostname:4443/sm/services/
```

where:

hostname is the hostname of your Message Studio server.

This page also includes the service endpoint URL.

WSDL

The WSDL is available from the following location for each version of the API:

```
https://hostname:4443/sm/services/mailling/version?wsdl
```

where:

- *hostname*—Hostname of your Message Studio server.
- *version*—Version of the Web Services API (e.g., 2009/03/02, v2)

XSD

The XSD is available from the following location for each version of the API:

```
https://
hostname
:4443/sm/services/mailing/
version?xsd=MailingServiceSchema.xsd
```

where:

- *hostname*—hostname of your Message Studio server.
- *version*—Version of the Web Services API (e.g., 2009/03/02, v2)

All required parameters for each operation are defined in the XSD. Optional parameters will have `minOccurs="0"`. Any other value for `minOccurs` indicates a required parameter. If `minOccurs` is not specified, it defaults to 1 and is required.

Mailing Action Status

When you call an action on a mailing, you must wait for a specific mailing status to make sure the action has successfully completed before calling another action for the same mailing. The following table indicates the actions you can call and the status you must wait for before calling a second action:

Before Action	Wait for Mailing Status
Pause	PAUSED
Load	ACTIVE
Cancel	CANCELED
Resume	ACTIVE
Launch	ACTIVE
Close	COMPLETED

Note: Mailing status is defined in the MailingServiceSchema.xsd, MailingStatus type.

Network Considerations

By default, the Message Studio web service runs on port 4443. This is configured during Message Studio installation.

Sample Code

Sample code is available on the Message Studio server in the following directory:

```
/data1/message_studio/lib/sm/webservice_samples.tar.gz
```

In addition, you can access the code from a browser:

```
https://hostname:4443/sm/webservice_samples.tar.gz
```

where:

hostname—Hostname of your Message Studio server

Development Environments

Selligent recommends Java CXF for the Message Studio web services. This section provides information about setting up your CXF development environment.

Topics:

- [Java CXF Clients on page 15](#)
- [C# Clients on page 22](#)
- [Ruby Clients on page 28](#)
- [Sample Java Code on page 162](#)
- [Sample C# Code on page 174](#)

Java CXF Clients

By generating your Java stubs with CXF, you can quickly build and develop Message Studio services.

Topics:

- [Generating Java Client Stubs on page 15](#)
- [Constructing a Java SOAP Client on page 17](#)
- [Java Security Token Requirements on page 17](#)

Generating Java Client Stubs

The WSDL can be directly translated to a set of Java classes that provide helper classes to hold parameter values, serialize the values as XML, and make the necessary HTTP requests to the SOAP server. This is done using the `wsdl2java` tool, which is part of the Apache CXF open source web services framework.

You can also use optional arguments to the `wsdl2java` tool to customize the generated code, as well as to generate an Ant based makefile to build your application.

Note: The `wsdl2java` options and output referenced here assume use of Apache CXF 2.1.3. If you are using a different version of CXF or a different web services framework altogether, the command options and results may be different.

Selligent recommends running the following command to generate the Java code:

```
wsdl2java -b bindingClient.xjb -d . -p
http://www.strongmail.com/services/2009/03/92=com.strongmail.
webserviceclient.samples.cxf.generated -p
http://www.strongmail.com/services/2009/03/02/schema=
com.strongmail.webserviceclient.samples.cxf.generated.objects -p
http://www.w3.org/2001/XMLSchema=
com.strongmail.webserviceclient.samples.cxf.generated.objects.xmlschema -sn
MailingService -validate -wv 1.1 MailingService.wsdl
```

These options provide you the following:

- `-b binding`—This option specifies an external binding file. `bindingClient.xjb` is available in the `webservices_samples/cxf` directory.
- `-d output_directory`—This option specifies the directory into which the Java code is written. If this is not explicitly specified, the files will be placed in the directory from which the `wsdl2java` command is invoked.
- `-p path`—This option specifies a subdirectory to place generated files for the named package.
- `-sn webservice`—This option specifies the name of the web service.
- `-validate`—This option validates the WSDL before generating the code.
- `-wv version`—This option specifies the version of the WSDL.

The generated code directory structure is:

Directory	Description
<code>com/strongmail/webserviceclient/samples/cxf/generated</code>	This directory contains a the generated <code>MailingService</code> interface and the <code>MailingService_Service</code> implementation. The <code>MailingService</code> interface defines all methods you can call. It also contains Java classes representing the fault definitions.
<code>com/strongmail/webserviceclient/samples/cxf/generated/objects</code>	This directory contains a Java class implementation for each type defined in the XSD.

Constructing a Java SOAP Client

Once you have generated the client stubs using `wsdl2java`, you can begin constructing a SOAP client. The sample code creates a service in two steps, which makes it easy to set different headers for different organizations in subsequent calls:

1. Create a raw service for a given endpoint. You can specify an endpoint or use the one in the WSDL. At this point, the created service won't add headers to operations.
2. Apply the authentication interceptor to the service to set username, password, and organization (and optionally, the sub-organization). After these two steps, a service will automatically add the same header to each operation. If you want to change the header to log into a different organization, you just need to re-apply the interceptor with new login information.

See these methods for the code:

- `ClientSetup.java`: `public MailingService createService(File wsdlFile, String endpointAddress) throws Exception;`
- `SampleProgram.java`: `public static MailingService createService();`
- `ClientSetup.java`: `public void applyInterceptor(MailingService svc, String username, String password, String organizationName, Long subOrganizationId) throws Exception;`

Java Security Token Requirements

A security token header must be passed with every API call. This header needs to contain username, password and organization (which is the same information needed in order to login to the Message Studio UI).

In CXF, you set security by using an interceptor. The interceptor will be implicitly called when you call each operation. The benefit of this approach is that you only need to set login information once, instead of passing it to each operation call.

Since the security header is defined by WS-Security specification and is handled automatically by the CXF library, you only need to pass in the properties to the interceptor.

The following code is required to pass Message Studio API authentication which uses WS-Security SOAP headers. You can use the following code to set up the interceptor.

```
/*
```

```

* ----- *
* Selligent, Inc. *
* * *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* * *
* Visit http://www.strongview.com for more information. *
* * *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* * *
* Distributing this source code outside your organization *
* is strictly prohibited. *
* * *
* ----- *
*/
/* WSSE standard username token */
Map<String, Object> outprops = new HashMap<String, Object>();
outprops.put(WSHandlerConstants.ACTION, WSHandlerConstants.USERNAME_TOKEN);

/* Set username and password */
outprops.put(WSHandlerConstants.USER, username);
outprops.put(WSHandlerConstants.PASSWORD_TYPE, WSConstants.PW_TEXT);
outprops.put(WSHandlerConstants.PW_CALLBACK_REF, newClientPasswordCallbackHandler
(password));

/* Create client interceptor AuthenticationInterceptor */
authenticationInterceptor = new AuthenticationInterceptor(schemaNS, outprops,
organizationName, null);

```

The following code is included in the sample code (AuthenticationInterceptor.java) and extends the CXF interceptor. It sets the UsernameToken (username and password) by calling the parent CXF interceptor. Since the OrganizationToken (organizationName/ subOrganizationId) is not standard, but defined by Selligent, you need to handle them manually. You can do this by defining an inner class and adding it to the interceptor chains.

```

/*
* ----- *
* Selligent, Inc. *
* * *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* * *
* Visit http://www.strongview.com for more information. *
* * *
* You may incorporate this Source Code in your application only *

```

```

* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* *
* Distributing this source code outside your organization *
* is strictly prohibited. *
* *
* ----- *
*/
package com.strongmail.webserviceclient.samples.cxf;
import org.apache.cxf.binding.soap.SoapMessage;
import org.apache.cxf.interceptor.Fault;
import org.apache.cxf.phase.Phase;
import org.apache.cxf.phase.PhaseInterceptor;
import org.apache.cxf.ws.security.wss4j.WSS4JOutInterceptor;
import org.apache.ws.security.SOAPConstants;
import org.apache.ws.security.handler.WSHandlerConstants;
import org.apache.ws.security.util.WSSecurityUtil;
import org.w3c.dom.Element;
import org.w3c.dom.Text;
import java.util.Collections;
import java.util.Map;
import java.util.Set;
import javax.xml.soap.SOAPException;
import javax.xml.soap.SOAPMessage;
public class AuthenticationInterceptor extends WSS4JOutInterceptor {

    /** Schema name space */
    private String schemaNS;

    /** Interceptor to set headers: organizationName and subOrganizationId */
    private PhaseInterceptor phaseInterceptor;

    /** organizationName and subOrganizationId to login */
    private String organizationName;
    private Long subOrganizationId;

    public AuthenticationInterceptor(String schemaNS, Map<String, Object> properties,
String orgName, Long subOrganizationId)
    {
        super(properties);
        this.schemaNS = schemaNS;
        this.phaseInterceptor = new WSS4JOutInterceptorInternal();
        this.organizationName = orgName;
        this.subOrganizationId = subOrganizationId;
    }
}

```

```

public void setOrganizationName(String orgName) {
    this.organizationName = orgName;
}

public void setSubOrganizationId(Long subOrganizationId) {
    this.subOrganizationId = subOrganizationId;
}

public void setUsername(String username) {
    getProperties().put(WSHandlerConstants.USER, username);
}

public void setPassword(String password) {
    getProperties().put(WSHandlerConstants.PW_CALLBACK_REF, new
ClientPasswordCallbackHandler(password));
}

public void handleMessage(SoapMessage message) throws Fault {
    super.handleMessage(message);
    message.getInterceptorChain().add(phaseInterceptor);
}

/**
 * Set SOAP header before sending out message.
 * Header includes username, password, organizationName and optionally subOrganizationId.
 * This message set organizationName and subOrganizationId, and parent class will set
username/password.
 */

class WSS4JOutInterceptorInternal implements PhaseInterceptor<SoapMessage> {
    private static final String ORGANIZATION_TOKEN_ELEMENT_NAME = "OrganizationToken";
    private static final String ORGANIZATION_ELEMENT_NAME = "organizationName";
    private static final String SUB_ORGANIZATION_ID_ELEMENT_NAME = "subOrganizationId";
    private static final String ID_ELEMENT_NAME = "id";

    public WSS4JOutInterceptorInternal() {
        super();
    }

    public void handleMessage(SoapMessage message) {
        try {
            Element securityHeader = getSecurityHeader(message);

            /* Create <OrganizationToken> and <organizationName> element, */
            /* and set <organizationName> as child of <OrganizationToken> */

```

```

        Element e = securityHeader.getOwnerDocument().createElementNS(schemaNS,
ORGANIZATION_TOKEN_ELEMENT_NAME);

        Element e2 = securityHeader.getOwnerDocument().createElementNS(schemaNS,
ORGANIZATION_ELEMENT_NAME);
        Text t = securityHeader.getOwnerDocument().createTextNode(organizationName);
        e2.appendChild(t);
        e.appendChild(e2);

        /* Create <subOrganizationId> if necessary. It is also child of
<OrganizationToken> if (subOrganizationId != null) */
        {
            Element e3 = securityHeader.getOwnerDocument().createElementNS( schemaNS, SUB_
ORGANIZATION_ID_ELEMENT_NAME);
            Element e4 = securityHeader.getOwnerDocument().createElementNS(schemaNS, ID_
ELEMENT_NAME);
            Text t2 = securityHeader.getOwnerDocument().createTextNode(
subOrganizationId.toString());
            e4.appendChild(t2);
            e3.appendChild(e4);
            e.appendChild(e3);
        }
        securityHeader.appendChild(e);
    }
    catch (Throwable e)
    {
        e.printStackTrace();
    }
}

private Element getSecurityHeader(SoapMessage message) throws SOAPException {
    SOAPMessage doc = message.getContent(SOAPMessage.class);
    String actor = (String)getOption(WSHandlerConstants.ACTOR);
    SOAPConstants sc = WSSecurityUtil.getSOAPConstants(doc.getSOAPPart
().getDocumentElement());
    return WSSecurityUtil.getSecurityHeader(doc.getSOAPPart(), actor, sc);
}

public Set<String> getAfter()
{
    return Collections.emptySet();
}

public Set<String> getBefore()
{
    return Collections.emptySet();
}

public String getId()
{
    return WSS4JOutInterceptorInternal.class.getName();
}

```

```

    }
    public String getPhase()
    {
        return Phase.POST_PROTOCOL;
    }
    public void handleFault(SoapMessage message) {
        /*nothing */
    }
}
}

```

C# Clients

C# is a multi-paradigm programming language that has an object-oriented syntax.

Topics:

- [Generating C# Client Stubs on page 22](#)
- [Creating a C# Mailing Service on page 22](#)
- [Constructing a C# SOAP Client on page 25](#)
- [C# Security Token Requirements on page 27](#)

Generating C# Client Stubs

Stubs for a C# implementations may be generated from the provided WSDL/XSD files, as follows:

```
wsdl /l:CS /protocol:SOAP MailingService.wsdl MailingServiceSchema.xsd
```

Once the stubs have been generated, a change must be made to one of the generated files. In the generated file `MailingService.cs`, you must update the `MailingService` class definition so that it extends the class:

```
Microsoft.Web.Services2.WebServicesClientProtocol
```

instead of the class:

```
System.Web.Services.Protocols.SoapHttpClientProtocol
```

Creating a C# Mailing Service

The following function can be used to create and return a service:

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
public const string url = "https://192.168.29.160:4443/sm/services/mailing/2009/03/02";
private static MailingService createService() {
    /* Create service and set endpoint */
    MailingService svc = new MailingService(); svc.Url = url;

    /* Ignore Certificate by trusting all certificate */
    System.Net.ServicePointManager.CertificatePolicy = new TrustAllCertificatePolicy();

    /* Create security tokens for SOAP header */
    UsernameToken userToken = new UsernameToken("admin@strongview.com", "admin",
    PasswordOption.SendPlainText);
    SoapContext requestContext = svc.RequestSoapContext;
    requestContext.Security.Tokens.Add(userToken);

    /* Use filter to set organization token and get rid of timestamp since it does not work
    with Java server. */
    MailingServiceOutFilter myfilter = new MailingServiceOutFilter("admin", ns);
    svc.Pipeline.OutputFilters.Insert(0, myfilter);
    return svc;
}

```

After creating the mailing service, you can call operations on it. For example:

```
MailingService svc = createService()
```

Security token requirements A security token header must be passed with every API call. This header needs to contain username, password and organization (which is the same information needed in order to login to the Message Studio UI).

The default Microsoft implementation of a security header token already supports username and password, so once you have created a `MailingService` instance, it is straight-forward to add username and password to its header:

```
MailingService svc = new MailingService();
UsernameToken userToken = new UsernameToken("admin@strongview.com", "admin",
    PasswordOption.SendPlainText);
SoapContext requestContext = svc.RequestSoapContext;
requestContext.Security.Tokens.Add(userToken);
```

However, you must also add an organization to the header, which must be done manually. In addition, by default Microsoft adds a timestamp to the header which is not handled correctly by the Java server-side code and which will cause your API calls to fail if it is not removed.

Therefore, you must remove the Microsoft timestamp manually. You can take care of both of these items by defining a custom filter and adding the desired code within its `ProcessMessage()` method. For example:

```
MailingServiceOutFilter myfilter = new MailingServiceOutFilter("myOrg");
svc.Pipeline.OutputFilters.Insert(0, myfilter);
```

Then, the `ProcessMessage()` method might be:

```
/*
 * ----- *
 * Selligent, Inc. *
 * * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * * *
 * Visit http://www.strongview.com for more information. *
 * * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * * *
 * ----- *
 */
/* Set internal namespace field */
_ns = http://www.strongmail.com/services/2009/03/02/schema;
public override void ProcessMessage(SoapEnvelope envelope) {

    /* Add OrganizationName token */
    XmlElement OrgTokenElement = envelope.CreateElement("sm", "OrganizationToken", _ns);
```



```

XmlElement orgElement = envelope.CreateElement("sm", "organizationName", _ns);

/* Set organizationName value */
XmlText orgName = envelope.CreateTextNode(_organization);
orgElement.AppendChild(orgName);
OrgTokenElement.AppendChild(orgElement);

/* Remove timestamp */
IEnumerator it = envelope.Header.GetEnumerator();
while(it.MoveNext()){
    XmlElement el = (XmlElement) it.Current;
    if (el.Name.Equals("wsse:Security")) {
        el.AppendChild(OrgTokenElement);

        /* Remove Timestamp */
        XmlNodeList timestamp = el.GetElementsByTagName("wsu:Timestamp");
        el.RemoveChild(timestamp.Item(0));
    }
}
}

```

Constructing a C# SOAP Client

To create the SOAP client, you will need the following script:

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */

public static MailingServiceStub createService() {

```

```

/* Read key store, key store password, wsdl file location and endpoint address */
/* Endpoint address will replace the one in WSDL file */
System.out.println("Store: " + System.getProperty("javax.net.ssl.trustStore"));
System.out.println("Password: " + System.getProperty(
("javax.net.ssl.trustStorePassword"));
System.out.println("endpoint: " + System.getProperty("endpoint.address"));

/* File */
wsdlFile = new File(System.getProperty("wsdl"));
String endpoint = System.getProperty("endpoint.address");
System.getProperty("services.namespace");
String schemaNs = System.getProperty("services.namespace.schema");
try {

    /* Create service */
    MailingServiceStub stub = new MailingServiceStub(endpoint);

    /* Add security header */
    /* username/password/organization are the same as UI login. */
    OMFactory fac = OMAbstractFactory.getOMFactory();
    OMNamespace omNs = fac.createOMNamespace(WSSE_NAMESPACE, "wsse");
    OMNamespace smSchemaNS= fac.createOMNamespace(schemaNs, "smschema");

    /* Create Security and UsernameToken */
    OMElement security = fac.createOMElement("Security", omNs);
    OMElement token = fac.createOMElement("UsernameToken", omNs);

    /* Set username */
    OMElement username = fac.createOMElement("Username", omNs);
    OMText usernametext = fac.createOMText("admin@strongview.com");
    username.addChild(usernametext);

    /* Set password */
    OMElement password = fac.createOMElement("Password", omNs);
    password.addAttribute("Type", PASSWORD_TYPE, null);
    OMText passtext = fac.createOMText("admin");
    password.addChild(passtext);
    token.addChild(username);
    token.addChild(password);
    security.addChild(token);

    /* Add OrganizationToken */
    OMElement orgtoken = fac.createOMElement("OrganizationToken", smSchemaNS);
    OMElement orgname = fac.createOMElement("organizationName", smSchemaNS);
    OMText orgtext = fac.createOMText("admin");

```

```

    orgname.addChild(orgtext);
    orgtoken.addChild(orgname);
    security.addChild(orgtoken);
    stub._getServiceClient().addHeader(security); return stub;
}
catch (Exception e) {
    e.printStackTrace();
    System.exit(1);
}
return null;
}

```

C# Security Token Requirements

A security token header must be passed with every API call. This header needs to contain username, password and organization (which is the same information needed in order to login to the Message Studio UI).

The following code is required to pass Message Studio API authentication which uses WS-Security SOAP headers. This code is included in the sample code, and is available in `Sample.Program.java`; refer to the `createService` method.

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * ----- *
 */

/* Create service */
MailingServiceStub stub = new MailingServiceStub(endpoint);

/* Add security header */
/* username/password/organization are the same as UI login. */

```

```

OMFactory fac = OMAbstractFactory.getOMFactory();
OMNamespace omNs = fac.createOMNamespace(WSSE_NAMESPACE, "wsse");
OMNamespace smSchemaNS = fac.createOMNamespace(schemaNs, "smschema");

/* Create Security and UsernameToken */
OMElement security = fac.createOMElement("Security", omNs);
OMElement token = fac.createOMElement("UsernameToken", omNs);

/* Set username */
OMElement username = fac.createOMElement("Username", omNs);
OMText usernametxt = fac.createOMText("admin");
username.addChild(usernametxt);

/* Set password */
OMElement password = fac.createOMElement("Password", omNs);
password.addAttribute("Type", PASSWORD_TYPE, null);
OMText passtext = fac.createOMText("admin");
password.addChild(passtext);
token.addChild(username);
token.addChild(password);
security.addChild(token);

/* Add OrganizationToken */
OMElement orgtoken = fac.createOMElement("OrganizationToken", smSchemaNS);
OMElement orgname = fac.createOMElement("organizationName", smSchemaNS);
OMText orgtext = fac.createOMText("admin");
orgname.addChild(orgtext);
orgtoken.addChild(orgname);

/* If you need to log in to a suborganization, then uncomment */
/* the following block, and change the string "100" to be a */
/* string containing the id of the suborganization you want. */
OMElement subOrgId = fac.createOMElement("subOrganizationId", smSchemaNS); OMElement id =
fac.createOMElement("id", smSchemaNS);
OMText subOrgIdText = fac.createOMText("100");
id.addChild(subOrgIdText);
subOrgId.addChild(id);
orgtoken.addChild(subOrgId);
security.addChild(orgtoken); stub._getServiceClient().addHeader(security);

```

Ruby Clients

Ruby is a general purpose object-oriented programming language that is often used to develop web services.

Note: The information provided in this section assumes that you will be developing Ruby code in a Windows environment.

In preparing your development environment, perform the following steps:

1. On your Windows machine, install the following:

Installer	Download From
ruby186-25.exe	http://rubyforge.org/frs/download.php/76955/rubyinstaller-2.0.0-p195.exe
ruby-gnome2-0.16.0-1-i386-mswin32.exe	http://rubyforge.org/frs/download.php/76955/rubyinstaller-2.0.0-p195.exe
soap4r	https://rubygems.org/downloads/mumboe-soap4r-1.5.8.7.gem
httpclient	https://rubygems.org/downloads/httpclient-2.3.4.1.gem

2. After installing the clients, run the following commands:

```
ruby_install_path\bin/gem install httpclient-2.3.4.1.gem
```

```
ruby_install_path\bin/gem install mumboe-soap4r-1.5.8.7.gem
```

Generating Ruby Client Stubs

Create the SOAP wrappers from Message Studio WSDL. This should create all of the required client ruby classes used to communicate with Selligent SOAP server.

1. `$ cd <path>\lib`
2. `$ ruby <soap4r1.5.8 path>/wsdl2ruby.rb --type client --wsdl <url to wsdl>`

Constructing a Ruby SOAP Client

Use the following code to construct the SOAP client.

```
def self.sm_service(opts={})
  # create the service
```

```

obj = MailingService.new(opts[:endpoint_url])
# run ruby with -d to see SOAP wire dumps.
obj.wiredump_dev = STDERR if $DEBUG
# attach auth header handler obj.headerhandler << SM::WsseAuthHeader.new()
obj end

```

Security token requirements A security token header must be passed with every API call. This header needs to contain username, password and organization (which is the same information needed in order to login to the Message Studio UI).

The following code is the Selligent security class file (`sm_security.py`) and is required to pass Message Studio API authentication which uses WS-Security SOAP headers.

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
#
# Use header handler to set the authentication headers in each # request.
#
class WsseAuthHeader < SOAP::Header::SimpleHandler
  NAMESPACE = 'http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd'
  USERNAME = 'admin@strongview.com'
  PASSWORD = 'admin'
  ORGANIZATION = 'admin'
  SM_SCH = "http://www.strongmail.com/services/2009/03/02/schema"
  Header = XSD::QName.new(NAMESPACE, 'Security') OrgToken = XSD::QName.new(SM_SCH, 'OrganizationToken')
  def initialize()
    super(Header)
  end
end

```

```

#
# Callback method called by soap4r on each request
# to insert the header.
#

def on_simple_outbound
  pwd = SOAP::SOAPElement.new(XSD::QName.new(nil, "Password"))
  pwd.text = "admin"
  pwd.extraattr["Type"] = "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
  username-token-profile-1.0#PasswordText"
  {
    "UsernameToken" => {
      "Username" => USERNAME,
      "Password" => pwd,
    },
    OrgToken => {
      XSD::QName.new(SM_SCH, 'organizationName') => ORGANIZATION, XSD::QName.new(SM_SCH,
      'subOrganizationId') => "",
    },
  }
end

```

To use this code, you will need to edit the file:

1. Make sure USERNAME,PASSWORD, and ORGANIZATION is set to appropriate values in:
lib/sm-utils.rb (line#10)
2. Make sure the SOAP property file contains the following line:
client.protocol.http.ssl_config.verify_mode =
OpenSSL::SSL::VERIFY_NONE

Security Considerations

Refer to the following topics with regards to web service security.

Securing API Communications with SSL

Communications between your client application and the Selligent WS API can be secured by using the Secure Sockets Layer (SSL) protocol for HTTP requests and responses. This is a standard approach for securing internet-based communications and was designed to prevent eavesdropping, tampering, and message forgery. It secures communications over the internet by cryptographic methods based on a 128-bit encryption and is a standard approach for securing SOAP-based Web service transactions.

User Authentication

If you already have a Message Studio account, you can use the credentials from that account to access web services. Your access permissions to the application through web services mirrors your access from the UI.

For instance, if you don't have the ability to create mailings within a particular organization in the UI, you will not be able to create mailings within that organization using web services. For more information about users and access privileges, please read the **Message Studio User Guide**.

Note: Remember that the organization in the SOAP header indicates the organization where you are performing the operation. For example, if you want to create a mailing in the admin organization, then you must log into (through the SOAP header) the admin organization. If you then want to create a mailing in the IDB organization, you will log into (through the SOAP header) the IDB organization.

User Credentials

To access Message Studio via API, you will need the same account information as you need to log into the UI. Make sure you have the Organization name, User name, and password. The Organization provides the context for the action you're performing through the API - whatever action you are taking is based on the user logging in.

Note: In most cases, Selligent recommends you make API calls as a super-user, which will provide access to the entire system.

All communication is similar to the login process for the UI.

SOAP uses the organization, login name, and password provided in this security header to authenticate a user to the application.

Security Header

Access to Message Studio's API is granted based on the contents of the security header, which is required for every call into the server. All communication is over https, which provides encryption and requires a self-signed SSL certificate.

Message Studio API is stateless. This means that every SOAP requests into Message Studio must include a security header that provides the user credentials. Every request is treated as follows:

1. Login.
2. Perform single action.
3. Logout (explicit).

Sample SOAP Header

The following is an example of the security tags you need to place in the header of each SOAP request. You will want to replace the username, password and organization with your personal login information.

The sub-organization ID is optional and identifies the organization or sub-organization you are accessing.

```
/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
<soap:Header>
  <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" soap:mustUnderstand="1">
```

```

    <wsse:UsernameToken xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/ oasis-
200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="UsernameToken-9243153">
        <wsse:Username xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis- 200401-
wss-wssecurity-secext-1.0.xsd">admin</wsse:Username>
        <wsse:Password xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis- 200401-
wss-wssecurity-secext-1.0.xsd" Type="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-username-token-profile-1.0#PasswordText">mypassword</wsse:Password>
    </wsse:UsernameToken>
    <OrganizationToken xmlns="http://www.strongmail.com/services/2009/03/02/schema">
        <organizationName>IDB</organizationName>
        <subOrganizationId>
            <id>2601</id>
        </subOrganizationId>
    </OrganizationToken>
</wsse:Security>
</soap:Header>

```

Firewall

You can limit Web service access to the APIs by modifying your iptables configuration to only allow certain IPs to access the Web services port. To do this:

1. Edit the iptable configuration file:

```
vi /etc/sysconfig/iptables
```

2. Before the REJECT line, add ACCEPT lines to allow access. For example, the following lines allow users from IP 192.168.21.165 to connect the Web services port (4443) and rejects all other IPs:

```

iptables -A RH-Firewall-1-INPUT -s 192.168.21.165 -p tcp -- dport 4443 -j
ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited

```

3. Restart the iptables service:

```
service iptables restart
```

IPtables Example

Here is an example of a complete iptables configuration, with a single IP (192.168.21.165) allowed access to the Web services port.

```

# Generated by iptables-save v1.3.5 on Thu Feb 19 20:40:26 2009 *filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]

```

```

:OUTPUT ACCEPT [10240089:5452482095] :RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp -m icmp --icmp-type any -j ACCEPT
-A RH-Firewall-1-INPUT -p esp -j ACCEPT
-A RH-Firewall-1-INPUT -p ah -j ACCEPT
-A RH-Firewall-1-INPUT -d 227.0.0.152 -p udp -m udp --dport 5353 -j ACCEPT -A
RH-Firewall-1-INPUT -p udp -m udp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A RH-Firewall-1-INPUT -s 192.168.21.35 -p tcp -m tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -s 10.0.0.0/23 -p tcp -m tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --dport 25 -j
ACCEPT -A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --dport 80 -
j ACCEPT -A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --dport
443 -j ACCEPT -A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --
dport 9000 -j ACCEPT -A RH-Firewall-1-INPUT -s 192.168.21.87 -p udp -m udp --
dport 161 -j ACCEPT
-A RH-Firewall-1-INPUT -s 10.0.3.0/25 -p udp -m udp --dport 161 -j ACCEPT
-A RH-Firewall-1-INPUT -s 10.0.185.0/27 -p udp -m udp --dport 161 -j ACCEPT
-A RH-Firewall-1-INPUT -s 192.168.21.165 -p tcp -m tcp --dport 9443 -j ACCEPT
-A RH-Firewall-1-INPUT -s 192.168.21.187 -p tcp -m tcp --dport 4443 -j ACCEPT
-A RH-Firewall-1-INPUT -s 192.168.21.84 -p tcp -m tcp --dport 5432 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Thu Feb 19 20:40:26 2009

```

Error Handling

When you pass calls using the SOAP API, Message Studio may return one of the faults below:

Fault	Definition
authorizationFault	Occurs when log into Message Studio (through the SOAP header) with a user that does not have the appropriate permissions to execute the operation.
concurrentModificationFault	Occurs when a different user modifies an asset between the time you perform the <code>get()</code> and the <code>update()</code> .
objectNotFoundFault	Occurs when you attempt to access an asset that does not exist within Message Studio. For example, if you call an operation on an <code>objectId</code> that doesn't exist within Message Studio.

Fault	Definition
<code>invalidObjectFault</code>	Occurs when an object that was passed into the operation includes a parameter with an invalid value.
<code>staleObjectFault</code>	Occurs when the server has a new copy than the one held by the API. You must re-get the same object before performing any operation.
<code>unexpectedFault</code>	Occurs when an error, other than the ones listed here, occurs.
<code>unrecognizedObjectTypeFault</code>	Occurs when you pass the wrong kind of object to the operation. For example, if you tried to perform a <code>refreshRecords()</code> operation on a mailing instead of on an external data source.

Frequently Asked Questions

What programming languages can I use with the APIs?

If a specific programming language and associated SOAP client library adhere to the SOAP version 1.2 standard as defined by the W3C, then the resulting client applications should work correctly.

What protocols are used for the APIs?

The Message Studio SOAP APIs use HTTPS (SSL, port 4443) by default while the EIS APIs use HTTP (port 9000).

Can I use the APIs over a VPN?

Yes, you can use a VPN with the APIs.

Does an API user need to be created in the Message Studio UI?

The user connecting via the API must be created in the Message Studio UI. Selligent recommends creating a dedicated API admin user for each organization, generating a cryptic password, and assigning to the application and then using it exclusively for application.

What is the maximum attachment size for SOAP API calls?

We recommend that payloads on API calls be under 256 KB. Larger attachments are possible, but may cause a degradation in performance.

General Notes

- When Creating Targets, use "Advanced Mode" so you have full control for defining and retrieving the SQL query that the target is based on.
- Cache results returned from the API to reduce the load you place on the API. For example, when managing a transactional message, instead of issuing a `get()` to return the mailing handle every time, put code in place to refresh the mailing handle cache when an error is received from the API.
- For every object/asset, there is an associated `ObjectID`, which is used to identify the object type. For example, for an `InternalDataSource` there is an

`InternalDataSourceId`. If you need to identify an object by its ID, use the `get()` operation. You set the `InternalDataSourceId` in the `GetRequest` so the server knows to fetch an IDS, not another type of object. For most of the operations, an `ObjectId` of a certain type is required.

- All operations take some kind of request object and return a response object. If an `ObjectId` is needed to do an operation (such as `cancel()`, `close()`, etc), then you need to create a request first and set the `ObjectId` inside the request. This makes the operations signature consistent.
- Every operation can have only one request.

Known Conditions

- C#/.NET issues - If you are using C#, the generated C# code will need to be modified. Change the following:

```
System.Web.Services.Protocols.SoapHttpClientProtocol
to
Microsoft.Web.Services2.WebServicesClientProtocol
```

- Replace all instances of `typeof(NameValuePair)` with `typeof(NameValuePair[])`.
- Replace the definition of class `TxnSendRequest` with definition given in the file `SchemaChangeForTxnSend.txt` (bundled in tarball)
- In Version 1 (March 2009), `get()` API implicitly converts non-supported tag types to `CLICKTAG` (e.g., View in Mobile or Facebook tags). View in Mobile and Facebook tags are fully supported in Version 2.

This is not an issue for any other language that has generated and compiled WSDL stubs.

Getting Started

The following procedure steps you through interfacing web service applications with Message Studio through the API.

Step 1: Obtain the WSDL

First, retrieve and inspect the WSDL for the Message Studio API so that you are familiar with the interface and prepared to start development using the various programming language-specific SOAP libraries.

The WSDL can be found at the following location on the Internet:

```
https://hostname:4443/sm/services/mailling/version?wsdl
```

where:

- *hostname*—Server hostname or IP address
- *version*—Version of the API. Example: 2009/03/02,v2

Step 2: Obtain the Sample Code

Next, you can get a head start on your project by retrieving the sample code bundle and using the common code snippets and patterns that are provided in the bundle.

Sample code is available on the Message Studio server in the following directory:

```
/data1/message_studio/lib/sm/webservice_samples.tar.gz
```

In addition, you can access the code from a browser:

```
https://hostname:4443/sm/webservice_samples.tar.gz
```

where:

hostname—Hostname of your Message Studio server

Note: In order to view the sample code, you must be logged into Message Studio.

Step 3: Obtain a SOAP Client Library for your Development Environment

Use of a SOAP Client Library allows you to more easily generate and process the SOAP XML requests and responses that form the basis of communication with the Message Studio API. Selligent has tested the following SOAP Client Libraries with its sample code.

Language	Recommended SOAP Client
Java	CXF, Axis2
C#	.NET 4.5
PHP	wsdlInterpreter
Ruby	SOAP4r
Python	Suds

You should use these SOAP Client Libraries to generate the supporting code for your application. For details on how to use or generate the supporting code, refer to the documentation provided for these libraries or browse the Selligent Spark community site for tips on how to best use these libraries.

Step 4: Walk-through a Simple Example

The general pattern of a Java-based client application for the Message Studio Web Services is provided below:

- Create a web service object that you will use to make calls in your client application. Creating this service object involves establishing WSDL and XSD references and the organization, user, and password which will be added as part of the SOAP security header to authenticate each API call to the Message Studio Web Service.
- Create a request object and set member values to accomplish a given task.
- Make the desired API call on the service object and pass the request object.
- Parse the response object to extract the desired information or confirm success of an API call.
- Catch any errors that are thrown.
- Continue to create request objects, make calls on service objects, and parse response objects to accomplish the goals of the API client application.

See Also:

- [Development Environments on page 15](#)
- [Sample Java Code on page 162](#)
- [Sample C# Code on page 174](#)
- [Sample XML Code on page 183](#)
- [Frequently Asked Questions on page 37](#)

API Object Descriptions

The Message Studio WS API is organized as a set of Objects defined in an XSD schema file and a set of Operations defined in a WSDL file.

For more information about a specific operation, refer to [API Operations on page 47](#).

The schema elements in the Message Studio XSD are organized in sections that match the organization of the Selligent Message Studio User Interface left navigation panel.

All Element ("field") semantics are the same as the corresponding objects or "fields" in the Message Studio User Interface unless otherwise specified. In general, each object's element, or "field," will be one of the following (as documented with each object):

- Writable—Full read/write access by user.
- Read-only—Values are system generated and owned and user can't change them.
- Write-once—User can only set during creation but cannot change after creation.

Description Tables

- [System Objects and Operations on page 42](#)
- [Data Source Objects and Operations on page 43](#)
- [Recipient Operations on page 44](#)

- [Content Operations on page 45](#)
- [Mailing Operations on page 45](#)

Table 2: System Objects and Operations

Object	Supported Operations
Organization —An organization is an entity that represents ownership of business assets and data. For example, an organization owns different data sources, message templates, and mailings.	create, get, list, update, delete
User —Message Studio users can be created, deleted, and managed through the API. <ul style="list-style-type: none"> • Super User—A super user has all available permissions across all organizations. • Administrator—An administrator has all available permissions within one organization. An admin can also copy templates and content blocks between organizations they “own” and its sub-organizations. • User—A user’s access privileges are determined by their Role. 	create, get, list, update, delete
Role —A collection of permissions assigned to a user. A Role is associated with a single organization, but is available across that organization’s sub-organizations.	create, get, list, update, delete
AssignedRole —Indicates the role assigned to a Message Studio user.	create, get, list, delete
Media Server Group (MailingClass) —Instructs the Email Delivery Server to use a specific set of IP addresses for delivering an email. Before assigning a mailing class with the mailing, make sure you have configured a Virtual Server Group on the Selligent EDS.	create, get, list, update, delete
Campaign —Categorizes designated mailings into a single collection, which can be used when viewing reports for multiple mailings that are similar.	get, list
System Address —There are three types of system addresses:	create, get, list, update, delete

Object	Supported Operations
<ul style="list-style-type: none"> • Bounce—Email address associated with the mailbox where receiving email servers route asynchronous failures. • From—Identifies the sender of the email by email address and name. • Reply—Email address where the recipient's email server should deliver replies to the mailing. 	
MediaServer —Message StudioThis server can connect to internal or external media servers that host media (images or videos) required for a specific template. Message Studio can leverage the Selligent tracking server (either onboard or offboard) as a media server.	create, get, list, update, delete
WebAnalytics —Web analytics tags are supported and pre-configured for Coremetrics, Google Analytics, and Omniture.	get, list
StrongTool —Allows you to call a StrongTool application and send it initialization parameters.	create, get, list, update, delete
<p>GlobalUndeliverables—Table that tracks all undeliverable email addresses across all data sources. This table can be updated through the API.</p> <hr/> <p>Note: Before messages are sent, the GlobalUndeliverable table should be checked. When messages are delivered, the status of the message must be updated in the table.</p>	getGlobalUndeliverable, reactivateGlobalUndeliverable
Tag —Tracking tags can be inserted in mailings.	create, get, list, update, delete
Channel —Mechanism for sending non-email messages such as Push Notifications, SMS text messages, Facebook, Twitter, or other notification messages.	get, list

Table 3: Data Source Objects and Operations

A data source is a database where recipient data is stored. Recipient data may be stored in the internal PostgreSQL database that is included with Message Studio. Or, the data may be stored in an external database (IBM DB2, Microsoft SQL, MySQL, Oracle, PostgreSQL, Sybase, or Teradata).

Object	Supported Operations
ExternalDataSource —External Profile Data Source(EDS), where data is stored in a database on an external server and must be imported into a Message Studio EDS.	create, get, list, update, delete Specific operations —addRecords, cancelRefreshRecords, exportRecords, getStatistics, refreshRecords, removeRecords
InternalDataSource —Uses an Internal Profile Data Source (IDS) that stores the subscriber information on the Message Studio server.	create, get, list, update, delete, copy Specific operations —addRecords, dedupeRecords, upsertRecords, exportRecords, getStatistics, importFtpFile, removeRecords, getRecords, findImport, startImport, getImportDetails
Extension —Standard extension that contains data associated via primary keys with recipients in a different data source.	create, get, list, update, delete, copy Specific operations —addRecords, exportRecords, getStatistics, removeRecords, findImport, startImport, getImportDetails
Catalog —Extension that is independent of a separate recipient data source.	create, get, list, update, delete, copy Specific operations —addRecords, exportRecords, getStatistics, removeRecords, findImport, startImport, getImportDetails
Program —Data source for Lifecycle Marketing	get, list Specific operations —addRecords, getProgramNodes

Table 4: Recipient Operations

Object	Supported Operations
Target —SQL query that identifies the recipients for a specific mailing. A mailing can include one or more targets. You also have the ability to exclude a target from a mailing.	create, get, list, update, delete, copy Specific operations —exportRecords
SuppressionList —Suppression lists instruct Message Studio to prevent the sending of a mailing to specific domains, email addresses, or accounts that may be included in a target.	create, get, list, update, delete, copy Specific operations —addRecords, exportRecords, removeRecords

Object	Supported Operations
SeedList —Email addresses that are not in your data sources. Common uses include sending messages to a test list (such as an internal department list or ISP lists) or for use with StrongDelivery Tools' Mailbox Monitor.	create, get, list, update, delete, copy Specific operations —addRecords, exportRecords, removeRecords

Table 5: Content Operations

Object	Supported Operations
Template, ChannelTemplate —Files that define the format and content of the message and may be designed in HTML or Text.	create, get, list, update, delete, copy Specific operations —importMessagePart, test, validateXsl, fetchLinks, addtrackingtags, personalize
ContentBlock —Text files that define a series of tokens that are associated with specific content, and may be linked to a recipient's profile. Content blocks are used to personalize mailings.	create, get, list, update, delete, copy Specific operations —fetchLinks, addtrackingtags
Attachment —A document or other file that is uploaded as an attachment to a specific template or to a mailing.	create, get, list, update, delete Specific operations —importContent
Dynamic Content(Rule) —Message Studio includes a rules-based personalization engine. This engine allows you to create dynamic content rules based on the target list. Using logical comparisons, you create rules where the target field meets the comparison criteria and causes some action to be taken on the template.	create, get, list, update, delete, copy

Table 6: Mailing Operations

Object	Supported Operations
Standard Mailing —(Batch) Mailings that are sent to a specific target and are launched manually (or through the API). These include: <ul style="list-style-type: none"> • One-time mailings—Associated with one-time events, such as a one-time promotion. • Recurring mailings—Share a common template, but are sent out on a regularly scheduled basis, such as a welcome email 	create, get, list, update, delete, copy, archive Specific operations —test, launch, schedule,pause, resume, cancel, close, getStatistics

Object	Supported Operations
for a new subscription.	
TransactionMailing, ChannelTransactionalMailing —Mailings based on pre-determined events (such as, when a customer makes a purchase or changes their password). When the event occurs, the client web server makes an API <i>send</i> call to the email server.	create, get, list, update, delete, copy, archive Specific operations —load, pause, resume, cancel, close, getStatistics, getTxnEasInfo, getTxnMailingHandle, txnSend

API Operations

This section provides details on each of the operations available in the API:

- **[addGlobalUndeliverable\(\)](#) on page 48**
- **[addRecords\(\)](#) on page 50**
- **[archiveMailing\(\)](#) on page 54**
- **[cancel\(\)](#) on page 56**
- **[cancelRefreshRecords\(\)](#) on page 58**
- **[close\(\)](#) on page 60**
- **[copy\(\)](#) on page 62**
- **[create\(\)](#) on page 64**
- **[dedupeRecords\(\)](#) on page 71**
- **[delete\(\)](#) on page 73**
- **[exportRecords\(\)](#) on page 76**
- **[fetchLinks\(\)](#) on page 78**
- **[findImport\(\)](#) on page 83**
- **[get\(\)](#) on page 85**
- **[getAllEasByMailingId\(\)](#) on page 89**
- **[getImportDetails\(\)](#) on page 90**
- **[getImportStatus\(\)](#) on page 92**
- **[getGlobalUndeliverable\(\)](#) on page 94**
- **[getProgramNodes\(\)](#) on page 96**
- **[getRecords\(\)](#) on page 98**
- **[getStatistics\(\)](#) on page 103**
- **[getTxnEasInfo\(\)](#) on page 106**

- **getTxnMailingHandle()** on page 108
- **importContent()** on page 110
- **importFtpFile()** on page 112
- **importMessagePart()** on page 114
- **launch()** on page 116
- **list()** on page 118
- **load()** on page 126
- **pause()** on page 128
- **personalize()** on page 130
- **reactivateGlobalUndeliverable()** on page 134
- **refreshRecords()** on page 136
- **removeRecords()** on page 138
- **resume()** on page 141
- **schedule()** on page 143
- **send()** on page 145
- **startImport()** on page 145
- **test()** on page 147
- **txnSend()** on page 149
- **update()** on page 152
- **upsertRecord()** on page 156
- **validateXsl()** on page 160

addGlobalUndeliverable()

Adds the undeliverable recipient information into the *global_undeliverable* table.

Supported Assets

The following list shows the Message Studio assets that are supported for the `addGlobalUndeliverable()` operation:

- Global undeliverable addresses (`globalUndeliverable`)

Recommendations

None.

Code Sequence

1. `addGlobalUndeliverable(addGlobalUndeliverableRequest)`

Supported Requests

The `addGlobalUndeliverable()` operation supports a single object: `addGlobalUndeliverableRequest`.

Response

`ReactivateGlobalUndeliverableResponse`—Contains summary for reactivation.

Faults

Refer to [Error Handling on page 35](#) for a description of the following faults that may be returned:

- `objectNotFoundFault`
- `authorizationFault`
- `invalidObjectFault`

Sample Code

```
/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 */
```

```

* Visit http://www.strongview.com for more information.      *
*                                                              *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc.             *
* and the copyright notices are not removed from the source code. *
*                                                              *
* Distributing this source code outside your organization      *
* is strictly prohibited.                                       *
*                                                              *
* ----- *
*/
public void addGlobalUndeliverable() throws Exception {
    System.out.println("Start of addGlobalUndeliverables");
    AddGlobalUndeliverableRequest request = new AddGlobalUndeliverableRequest();
    OrganizationId orgId = new OrganizationId();
    orgId.setId("1");
    request.setOrganizationId(orgId);
    GlobalUndeliverableInfo info = new GlobalUndeliverableInfo();
    info.setFailureCode(2001);
    info.setRecipientAddress("dummayEmail@undeliverable.com");
    info.setChannelType("EMAIL");
    GlobalUndeliverableInfo info1 = new GlobalUndeliverableInfo();
    info1.setFailureCode(2101);
    info1.setRecipientAddress("0000000000");
    info1.setChannelType("SMS");
    request.getGlobalUndeliverableInfo().add(info);
    request.getGlobalUndeliverableInfo().add(info1);
    AddGlobalUndeliverableResponse response = service.addGlobalUndeliverable(request);
    if(response != null) {
        if(response.getFault().size() > 0) {
            for(FaultDetail faultDetail: response.getFault()) {
                System.out.print("Fault code - "+faultDetail.getFaultCode());
                System.out.println("\t Fault Description - " + faultDetail.getFaultMessage());
            }
        }
    }
    System.out.println("End of addGlobalUndeliverable");
}

```

addRecords ()

Adds recipient information to one of the supported assets.

Supported Assets

The following list shows the Message Studio assets that are supported for the `addRecords()` operation:

- Internal data sources (`dataSourceId`)
- External data sources (`externalDataSourceId`)
- Extension - Catalog (`catalogId`)
- Extension - Standard (`extensionId`)
- Program (`programId`)
- Seed list (`seedListId`)
- Suppression list (`suppressionId`)

Recommendations

- In most cases, you should add records to your external data source directly to your database and not through the API. Selligent does not recommend adding records to your external data source through the API or the UI.
- Before adding records to a data source, you should check the `DataSourceOperationStatus` to make sure it is `IDLE`. Possible status indicators include:
 - `IDLE`—Indicates there is no operation in progress.
 - `IMPORTING`—Indicates someone is adding records to the data source, either through the API or UI.
 - `REFRESHING`—Indicates an external data source is currently refreshing the local copy.

Code Sequence

1. Get desired asset ID using `list (filter)` API operation.
2. `addRecords (request)`

Supported Requests

The `addRecords()` operation supports multiple objects that can be passed in as arguments. Each object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

The following table maps the functional area to the object/request.

Functional area	Object/Request
External Data Source	AddDataSourceRecordsRequest
Internal Data Source	AddDataSourceRecordsRequest
Seed List	AddSeedListRecordsRequest
Suppression List	AddSuppressionListRecordsRequest
Program	AddProgramContactRecordsRequest
Extension - Standard	AddDataSourceRecordsRequest
Extension - Catalog	AddDataSourceRecordsRequest

Response

addRecordsResponse

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- objectNotFoundFault
- authorizationFault
- unrecognizedObjectTypeFault
- unexpectedFault
- invalidObjectFault
- concurrentModificationFault

Sample Code

```
/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
```

```

* is strictly prohibited. *
* *
* ----- *
*/

/*
* Add records using a file to the datasource created in crea- teIDS().
*
* @param dataSourceId the data source ID where records will be added to.
* @param dataFile the file that will be uploaded
* @throws Exception any exception
*/

public void addRecordsUsingFile(InternalDataSourceId data- SourceId, File dataFile)
throws Exception {

    /* Create request */
    AddDataSourceRecordsRequest addRecordsRequest = new AddDataSourceRecordsRequest();
    addRecordsRequest.setDataSourceId(dataSourceId);
    addRecordsRequest.setDataSourceId(dataSourceId);
    addRecordsRequest.setContainsFieldNames(true);
    addRecordsRequest.setFieldDelimiter(",");

    /* Set data handler to read data from the file */
    InputStream is = new FileInputStream(dataFile);
    DataHandler dh = new DataHandler(new AttachmentDataSource("text/plain", is));
    addRecordsRequest.setDataSourceRecordsAttachment(dh);

    /* Make call */
    service.addRecords(addRecordsRequest);
}

```

Sample: AddProgramContactRecordsRequest

The following is a sample request to add a contact (identified by 123) to an LCM program (id of 300).

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken wsu:Id="UsernameToken-59">
        <wsse:Username>admin</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">admin</wsse:Password>

```

```

    </wsse:UsernameToken><OrganizationToken
xmlns="http://www.strongmail.com/services/2009/03/02/schema">
    <organizationName>admin</organizationName></OrganizationToken>
</wsse:Security></soap:Header>
<soap:Body>
    <addRecords xmlns="http://www.strongmail.com/services/2009/03/02/schema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="AddProgramContactRecordsRequest">
        <programId><id>300</id></programId>
        <programContactRecord>
            <programDataSourcePk>123</programDataSourcePk>
            <contactId>123</contactId>
        </programContactRecord>
    </addRecords>
</soap:Body>
</soap:Envelope>

```

archiveMailing()

This operation archives a mailing by setting a flag in the database to mark it archived. Users can use SQL queries to find archived mailings.

Supported Assets

The following list shows the Message Studio assets that are supported for the `archiveMailing()` operation:

- Batch Mailing (`standardMailingId`)
- Transactional Mailing (`transactionalMailingId`)
- Channel Standard Mailing (`channelStandardMailingId`)
- Channel Transactional Mailing (`channelTransactionalMailingId`)

Recommendations

None.

Code Sequence

1. Get the desired asset ID using **list(filter)** API operation.
2. Use above `mailingId` and call this `archiveMailing(archiveRequest)` API operation.

Supported Requests

Functional area	Object/Request
Batch Mailing	standardMailingId/ArchiveRequest
Transactional Mailing	transactionalMailingId/ArchiveRequest
Channel Standard Mailing	channelStandardMailingId/ArchiveRequest
Channel Transactional Mailing	channelTransactionalMailingId/ArchiveRequest

Response

`archiveResponse`—Returns a Boolean success flag (true/false).

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- `objectNotFoundFault`
- `authorizationFault`
- `unexpectedFault`

Sample Code

```

/*
 * -----
 * Selligent, Inc.
 *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved.
 *
 * Visit http://www.strongview.com for more information.
 *
 * You may incorporate this Source Code in your application only
 * if you own a valid license from Selligent, Inc.
 * and the copyright notices are not removed from the source code.
 *
 * Distributing this source code outside your organization
 * is strictly prohibited.
 * -----
 */

public boolean archiveMailing(MailingId mailingId) {

```

```

ArchiveRequest archiveRequest = new ArchiveRequest();
archiveRequest.setMailingId(mailingId);
ArchiveResponse response = svc.archive(archiveRequest);
}

```

cancel ()

This operation permanently cancels a mailing that is in progress.

Supported Assets

The following list shows the Message Studio assets that are supported for the `cancel ()` operation:

- Batch Mailing (`standardMailingId`)
- Transactional Mailing (`transactionalMailingId`)
- Channel Standard Mailing (`channelStandardId`)
- Channel Transactional Mailing (`channelTransactionalMailingId`)

Recommendations

None.

Code Sequence

1. Get desired asset ID using **list(filter)** API operation.
2. `cancel (request)`

Supported Requests

The `cancel ()` operation supports multiple objects that can be passed in as arguments. Each object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

The following table maps the functional area to the object/request.

Functional area	Object/Request
Batch mailing	<code>standardMailingId/CancelRequest</code>
Transactional mailing	<code>transactionalMailingId/CancelRequest</code>

Functional area	Object/Request
Channel Standard mailing	channelStandardId/CancelRequest
Channel Transactional mailing	channelTransactionalMailingId/CancelRequest

Response

cancelResponse—Returns a boolean flag (true or false) indicating success.

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- objectNotFoundFault
- authorizationFault
- unexpectedFault

Sample Code

```

/*
 * ----- *
 * Selligent, Inc. *
 * * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * * *
 * Visit http://www.strongview.com for more information. *
 * * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * * *
 * ----- *
 */
/*
 * Cancel a Standard Mailing. *
 * @param mailingId the ID of the Mailing that will be closed
 * @return if close operation is successful
 * @throws Exception any exception */

public boolean cancel(MailingId mailingId) throws Exception {

```

```

CancelRequest request = new CancelRequest();
request.setMailingId(mailingId);
return service.cancel(request).isSuccess();
}

```

cancelRefreshRecords ()

Cancels a refresh of External Data Source records.

Supported Assets

The following list shows the Message Studio assets that are supported for the `cancelRefreshRecords ()` operation:

- External data source (`externalDataSourceId`)

Recommendations

None.

Code Sequence

1. Get the desired asset ID using **list(filter)** API operation.
2. `cancelRefreshRecords (request)`

Supported Requests

The `cancelRefreshRecords ()` operation supports a single object that can be passed in as an argument. The object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

The following table maps the functional area to the object/request.

Functional area	Object/Request
External data source	CancelRefreshRecordsRequest

Response

`cancelRefreshRecordsResponse`

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- objectNotFoundFault
- authorizationFault
- unexpectedFault

Sample Code

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
/*
 * Cancel the refresh of External Data Source records. *
 * @param edsId the EIS ID to be cancelled.
 * @return if cancel is successful
 * @throws Exception any exception */

public boolean cancelRefreshRecords(ExternalDataSourceId edsId) throws Exception {
    CancelRefreshRecordsRequest request = new CancelRefreshRecordsRequest();
    request.setDataSourceId(edsId);
    return service.cancelRefreshRecords(request).isSuccess();
}

```

close()

Closes out a batch or transactional mailing. Closing a transactional mailing puts it into a completed state. Completed is a permanent state.

Supported Assets

The following list shows the Message Studio assets that are supported for the `close()` operation:

- Transactional Mailing (`transactionalMailingId`)
- Channel Transactional Mailing (`channelTransactionalMailingId`)

Recommendations

None.

Code Sequence

1. Get the desired asset ID using `list(filter)` API operation.
2. `close(request)`

Supported Requests

The `close()` operation supports multiple objects that can be passed in as arguments. Each object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

The following table maps the functional area to the object/request.

Functional area	Object/Request
Transactional mailing	CloseRequest
Channel Transactional mailing	CloseRequest

Response

`closeResponse`

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- objectNotFoundFault
- authorizationFault
- unexpectedFault

Sample Code

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */

/*
 * Close a Transactional Mailing. *
 * @param mailingId the ID of the Mailing that will be closed
 * @return if close operation is successful
 * @throws Exception any exception */

public boolean close(TransactionalMailingId mailingId) throws Exception {
    CloseRequest request = new CloseRequest();
    request.setMailingId(mailingId);
    return service.close(request).isSuccess();
}

```

copy ()

Copies an asset. Copy supports copying a single asset and giving it a new name, or copying a single asset to an organization within the same organization tree.

Note: Copying to another organization requires super-user or administrator permissions in Message Studio.

Supported Assets

The following list shows the Message Studio assets that are supported for the `copy ()` operation:

- Internal Data Source (`InternalDataSourceId`)
- Target (`targetId`)
- Suppression list (`suppressionListId`)
- Seed list (`seedListId`)
- Template (`templateId`)
- Extension - Catalog (`catalogId`)
- Extension - Standard (`extensionId`)
- Content blocks (`contentBlockId`)
- Dynamic content (`ruleId`)
- Batch mailing (`mailingId`)
- Transactional mailing (`transactionalMailingId`)
- Channel Template (`channelTemplateId`)
- Channel Standard mailing (`channelStandardMailingId`)
- Channel Transactional mailing (`channelStandardMailingId`)

Recommendations

None.

Code Sequence

1. Get the desired asset ID using **list(filter)** API operation.
2. `copy (request)`

Supported Requests

The `copy()` operation supports multiple objects that can be passed in as arguments. Each object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

The following table maps the functional area to the object/request.

Functional area	Object/Request
Internal data source	CopyDataSourceRequest
Target	CopyTargetRequest
Suppression list	CopySuppressionListRequest
Seed list	CopySeedListRequest
Template, Channel Template	CopyTemplateRequest
Content block	CopyContentBlockRequest
Dynamic content	CopyRuleRequest
Batch mailing	CopyMailingRequest
Transactional mailing	CopyMailingRequest
Channel Standard mailing	CopyMailingRequest
Channel Transactional mailing	CopyMailingRequest

Response

`copyResponse`

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- `objectNotFoundFault`
- `authorizationFault`
- `unrecognizedObjectTypeFault`
- `unexpectedFault`
- `invalidObjectFault`

Sample Code

```
/*
 * -----
 * Selligent, Inc.
 * -----
 */
```

```

*
* Copyright © 2016 Selligent, Inc.. - All rights reserved.
*
* Visit http://www.strongview.com for more information.
*
* You may incorporate this Source Code in your application only
* if you own a valid license from Selligent, Inc.
* and the copyright notices are not removed from the source code.
*
* Distributing this source code outside your organization
* is strictly prohibited.
*
* -----
*/

/*
* Copy a data source using a new name.
*
* @param dataSourceId the data source ID that will be copied
* @param newName the name of new data source
* @return the new InternalDataSource ID
* @throws Exception any exception
*/

public InternalDataSourceId copy- DataSource(InternalDataSourceId dataSourceId, String
newName) throws Exception{

    /* Create copy request */
    CopyDataSourceRequest request = new CopyDataSourceRequest();
    request.setFromId(dataSourceId);
    request.setNewName(newName);
    return (InternalDataSourceId) service.copy(request).getObjectId();
}

```

create()

Creates any of the supported assets within Message Studio. This is a batch operation that allows you to pass multiple objects in a single call.

Supported Assets

The following list shows the Message Studio assets that are supported for the `create()` operation:

- Internal Data Source (internalDataSourceId)
- External Data Source (externalDataSourceId)
- Target (targetId)
- Seed list (seedListId)
- Suppression list (suppressionListId)
- Template (templateId)
- (Assigned) Role (assignedRoleId)
- Role (roleId)
- Media Server (mediaServerId)
- Media Server Group (mediaServerGroupId)
- Extension - Catalog (catalogId)
- Extension - Standard (extensionId)
- Content block (contentBlockId)
- Attachment (attachmentId)
- Dynamic content (ruleId)
- Batch mailing (mailingId or standardMailingId)
- Transactional mailing (transactionalmailingId)
- Tracking tags
- Organization (organizationId)
- System Address (systemAddressId)
- User (userId)
- StrongTools
- Channel Template (channelTemplateId)
- Channel Standard mailing (channelStandardId)
- Channel Transactional mailings (channelTransactionalMailingId)

Recommendations

Creating a mailing or template with `isApproved=true`, triggers validation checks. All objects must have all necessary attributes for launching or the server will return a fault.

Note: Content blocks created in the Message Studio SOAP API must be less than 40 MB in size.

Code Sequence

1. Instantiate object.
2. Set parameters.
3. `create(array_of_objects).`

Supported Requests

The `create()` operation supports multiple objects that can be passed in as arguments. Each object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

Note: You must pass in an array of objects for the `create()` operation. The array of objects may be defined as a single asset or as multiple assets to be created.

The following table maps the functional area to the object/request.

Functional area	Request Object	Special Base Object
External Data Source	CreateRequest	ExternalDataSource
Internal Data Source	CreateRequest	InternalDataSource
Target	CreateRequest	Target
Seed list	CreateRequest	SeedList
Suppression list	CreateRequest	SuppressList
Template	CreateRequest	Template, ChannelTemplate
Content blocks	CreateRequest	ContentBlock
Attachment	CreateRequest	Attachment
Dynamic content	CreateRequest	Rule
Batch mailing	CreateRequest	StandardMailing, ChannelStandardMailing
Transactional mailing	CreateRequest	TransactionalMailing, ChannelTransactionalMailing
Assigned Role	CreateRequest	AssignedRole
Organization	CreateRequest	Organization
Media Server	CreateRequest	MediaServer
System Address	CreateRequest	SystemAddress
Mailing Server Group	CreateRequest	MailingClass
Role	CreateRequest	Role
User	CreateRequest	User

Response

BatchCreateResponse—Will include one or more **CreateResponses** depending on the number of base objects you passed into the operation.

Faults

This operation does not throw a fault. Instead, the `FaultDetail` is wrapped in the response. Any faults will be returned in the `BatchCreateResponse`. You will need to check each individual `CreateResponse` for the exception.

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- `authorizationFaultDetail`
- `invalidObjectFaultDetail`
- `objectNotFoundFaultDetail`
- `unexpectedFaultDetail`
- `unrecognizedObjectTypeFaultDetail`

Recommended Defaults

When you set up a mailing, there are several parameters that you will need to define. In Message Studio, some of these parameters have default values. If you are not sure what value to use, you may consider using the defaults listed below.

Templates

Before a mailing can be sent, the template must have the following elements. You can set these elements against the template or against the mailing:

- `bounceAddressID`
- `fromAddressId`
- `fromName`

In addition, you will have to set the following elements; again, these can be set against the template or against the mailing, but must be set. You may want to use the suggested values for the defaults:

- `bodyEncoding—7 bit`
- `headerEncoding—7 bit`
- `outputBodyCharSet—UTF-8`
- `outputHeaderCharSet—UTF-8`

All templates must be approved (`isApproved`) in order for a mailing to launch.

Standard Mailing Recommendations

Before a mailing can be sent, the mailing must have the following elements (plus, the ones mentioned above). You may want to use the suggested values for the defaults:

- `priority`— `normal`
- `rowDelimiter`— `_sm_row_delim_\n`
- `fieldDelimiter`— `::`

In most cases, you will want to include the `mailingClassId`, which identifies the IP addresses the EDS should use to send the mailing.

Other notes:

- All mailings must include the compliancy check (`isCompliant`) to be launched.
- All mailings must be approved (`isApproved`) to be launched.

Tracking Tags

Support is provided for the following tracking tags:

- | | |
|----------------------|-----------------------|
| • NONE | • FTAFTAG |
| • CLICKTAG | • OPENTAG |
| • ENCLICKTAG | • SHARETAG |
| • ENUNSUBSCRIBETAG | • SHORTCLICKTAG |
| • ENFTAFTAG | • SHORTUNSUBSCRIBETAG |
| • ENOPENTAG | • VIEWINBROWSERTAG |
| • ENSHARETAG | • UNSUBSCRIBETAG |
| • ENVIEWINBROWSERTAG | |

Transactional Mailing Recommendations

Before a mailing can be loaded, it must have the following elements (plus, the ones mentioned above in the Templates section). You may want to use the suggested values for the defaults:

- `priority—normal`
- `rowDelimiter—_sm_row_delim_\n`
- `fieldDelimiter—::`

Other notes:

- All mailings must include the compliancy check (`isCompliant`) to be loaded.
- All mailings must be approved (`isApproved`) to be loaded.

SMS Mailings

When you create a transactional mailings over SMS, you must set output character set to either ASCII or an ASCII-based character set. SMS does not support any other character sets.

Sample Code

```
/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
```

```

/*
 * Create a Transactional Mailing.
 *
 * @param mailingName the name of the mailing
 * @param templateId the template used for the mailing
 * @param targetId the target used for the mailing
 * @return TransactionalMailingId
 * @throws Exception any exception
 */

public TransactionalMailingId createTransactionalMailing(String mailingName, TemplateId
templateId, TargetId targetId) throws Exception {
    TransactionalMailing mailing = new TransactionalMailing();
    setMailingCommonFields(mailing, mailingName, bounceId, fromId, replyId, templateId);
    mailing.setDescription("TransactionalMailing from CXF");
    mailing.setType(MailingType.TRANSACTIONAL);
    mailing.setTargetId(targetId);
    mailing.setMessageType(MessageType.EMAIL);

    /* mailing.setMailingConfig(null); */
    /* Read only */
    /* mailing.getRecordStructure(); */
    /* Read only */

    return (TransactionalMailingId) helper.create(mailing);
}

private void setMailingCommonFields(Mailing mailing, String mailingName, SystemAddressId
bounceId, SystemAddressId fromId, SystemAddressId replyId, TemplateId templateId) {
    mailing.setBodyEncoding(Encoding.BASE_64);
    mailing.setBounceAddressId(bounceId);
    mailing.setCampaignId(null);
    mailing.setFieldDelimiter("::");
    mailing.setFromAddressId(fromId);
    mailing.setFromName("From Name");
    mailing.setHeaderEncoding(Encoding.EIGHT_BIT);
    mailing.setIsApproved(true);
    mailing.setIsCompliant(true);
    mailing.setName(mailingName);
    mailing.setOutputBodyCharSet(CharSet.UTF_8);
    mailing.setOutputBodyCharSetToken(null);

    /* can only set either charset or token */
    mailing.setOutputHeaderCharSet(CharSet.ASCII);
    mailing.setOutputHeaderCharSetToken(null);

```

```

mailing.setPriority(MailingPriority.NORMAL);
mailing.setReplyAddressId(replyId);
mailing.setRowDelimiter("\n");
mailing.setSubject("mailing subject");
mailing.setTemplateId(templateId);
mailing.getFormat().add(MessageFormat.HTML);
mailing.getHeader().add("mailing header");
}

```

dedupeRecords ()

Removes duplicate recipient records from an internal data source or extension. Choose between one and four fields in the data source. If all the data in the specified fields match, Message Studio removes the record.

Supported Assets

The following list shows the Message Studio assets that are supported for the dedupeRecords operation:

- Extension - Catalog (catalogId)
- Extension - Standard (extensionId)
- Internal Data Source (internalDataSourceId)

Recommendations

None.

Code Sequence

1. Get desired asset ID using `list (filter)` API operation.
2. `dedupeRecords (request)`

Supported Requests

The dedupeRecords () operation supports a single object that can be passed in as an argument. The object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

The following table maps the functional area to the object/request.

Functional area	Object/Request
Internal data source	DedupeDataSourceRecordsRequest

Response

dedupeRecordsResponse

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- objectNotFoundFault
- authorizationFault
- unrecognizedObjectTypeFault
- unexpectedFault
- dedupeRecordsResponse

Sample Code

```

/*
 * ----- *
 * Selligent, Inc. *
 * * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * * *
 * Visit http://www.strongview.com for more information. *
 * * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * * *
 * ----- *
 */

/*
 * Remove duplicate records for a data source based on field "name".
 * If duplicate records are found, keep the latest ones. *
 * @param idsId the InternalDataSourceId that will be deduped
 * @return if dedupe is successful

```



```

* @throws Exception any exception */

public boolean dedupeRecords(InternalDataSourceId idsId) throws Exception {
    DedupeDataSourceRecordsRequest request = new DedupeDataSourceRecordsRequest();
    request.setDataSourceId(idsId);
    request.setOption(DataSourceDedupeOption.KEEP_NEWEST);
    request.getMatchField().add("name");
    return service.dedupeRecords(request).isSuccess();
}

```

delete()

Deletes one or more assets from the database. This is a batch operation that allows you to pass multiple ids in a single call.

Supported Assets

The following list shows the Message Studio assets that are supported for the delete() operation:

- Internal Data Source (internalDataSourceId)
- External Data Source (externalDataSourceId)
- Target (targetId)
- Seed list (seedListId)
- Suppression list (suppressionListId)
- Template (templateId)
- (Assigned) Role (assignedRoleId)
- Role (roleId)
- Media Server (mediaServerId)
- Media Server Group (mediaServerGroupId)
- Extension - Catalog (catalogId)
- Extension - Standard (extensionId)
- Content block (contentBlockId)
- Attachment (attachmentId)
- Dynamic content (ruleId)
- Batch mailing (mailingId or standardMailingId)
- Transactional mailing (transactionalmailingId)
- Organization (organizationId)
- System Address (systemAddressId)
- User (userId)
- StrongTools
- Channel Template (channelTemplateId)
- Channel Standard mailing (channelStandardMailingId)
- Channel Transactional mailing (channelTransactionalMailingId)

Recommendations

None.

Code Sequence

1. Get the desired asset ID using **list(filter)** API operation.
2. `delete(array_of_ids)`

Supported Requests

The `delete()` operation supports multiple objects that can be passed in as arguments. Each object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

Note: You must pass in an array of objects for the `delete()` operation. The array of objects may be defined as a single asset or as multiple assets to be deleted.

The following table maps the functional area to the object/request.

Functional area	Object/Request	Specific Base Object
External data source	DeleteRequest	externalDataSourceId
Internal data source	DeleteRequest	internalDataSourceId
Target	DeleteRequest	targetId
Seed list	DeleteRequest	seedListId
Suppression list	DeleteRequest	suppressionListId
Template	DeleteRequest	templateId
Content blocks	DeleteRequest	contentBlockId
Attachment	DeleteRequest	attachmentId
Dynamic content	DeleteRequest	ruleId
Batch mailing	DeleteRequest	standardMailingId, channelStandardMailingId
Transactional mailing	DeleteRequest	transactionalMailingId, channelTransactionalMailingId

Response

`BatchDeleteResponse`

The `BatchDeleteResponse` will include one or more `DeleteResponses` depending on the number of base objects you passed into the operation.

Faults

This operation does not throw a fault. Instead, the `FaultDetail` is wrapped in the response. Any faults will be returned in the `BatchDeleteResponse`. You will need to check each individual `DeleteResponse` for the exception.

Refer to [Error Handling on page 35](#) for a description of the following faults that may be returned:

- `authorizationFaultDetail`
- `concurrentModificationFaultDetail`
- `objectNotFoundFaultDetail`
- `unexpectedFaultDetail`
- `unrecognizedObjectTypeFaultDetail`

Sample Code

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */

/*
 * Delete an object
 *
 * @param objectId the objectId that will be deleted
 * @throws Exception any exception
 */

```

```
public void delete(ObjectId objectId) throws Exception {
    List<ObjectId> objectIds = new ArrayList<ObjectId>();
    objectIds.add(objectId);
    delete(objectIds);
}
```

exportRecords ()

Exports the contents of the specified asset to file. You must specify a field delimiter and a row delimiter.

Supported Assets

The following list shows the Message Studio assets that are supported for the `exportRecords()` operation:

- Internal Data Source (internalDataSourceId)
- External Data Source (externalDataSourceId)
- Target (targetId)
- Extension - Catalog (catalogId)
- Extension - Standard (extensionId)
- Seed list (seedListId)
- Suppression list (suppressionListId)

Recommendations

Before exporting records to a data source, you should check the `DataSourceOperationStatus` to make sure it is `IDLE`; otherwise, only a partial records set will be exported. Possible status indicators include:

- `IDLE`—Indicates there is no operation in progress.
- `IMPORTING`—Indicates someone is adding records to the data source, either through the API or UI.
- `REFRESHING`—Indicates an external data source is currently refreshing the local copy.

Code Sequence

1. Get the desired asset ID using **list(filter)** API operation.
2. `exportRecords(request)`

Supported Requests

The `exportRecords()` operation supports multiple objects that can be passed in as arguments. Each object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

The following table maps the functional area to the object/request.

Functional area	Object/Request
Internal data source	ExportDataSourceRecordsRequest
External data source	ExportDataSourceRecordsRequest
Target	ExportTargetRecordsRequest
Seed list	ExportSeedListRecordsRequest
Suppression list	ExportSuppressionListRecordsRequest

Response

`exportRecordsResponse`

Faults

Refer to [Error Handling on page 35](#) for a description of the following faults that may be returned:

- `objectNotFoundFault`
- `authorizationFault`
- `unrecognizedObjectTypeFault`
- `unexpectedFault`

Sample Code

```
/*
 * -----
 * Selligent, Inc.
 *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved.
 *
 * Visit http://www.strongview.com for more information.
 *
 * You may incorporate this Source Code in your application only
 * if you own a valid license from Selligent, Inc.
 * and the copyright notices are not removed from the source code.
 */
```

```

*
* Distributing this source code outside your organization
* is strictly prohibited.
*
* -----
*/

/*
 * Export data source records to a file.
 * @param dataSourceId the data source to be exported
 * @param toFile the file to be written
 * @throws Exception any exception
 */

public void exportRecords(dataSourceId toFile) throws Exception {

    /* Create and fill in request */
    ExportDataSourceRecordsRequest request = new ExportDataSourceRecordsRequest();
    request.setDataSourceId(dataSourceId);
    request.setFieldDelimiter("|");
    request.setRowDelimiter("\n");

    /* Export regular records. Use true to export malformed records. */

    request.setUseMalformedRecords(false);
    toFile = new File(toFile);

    /* Wait for data source status to be idle before exporting */
    waitForIdle(dataSourceId);

    /* Make call and save data to file */
    ExportRecordsResponse response = service.exportRecords(request);
    InputStream is = response.getData().getInputStream();
    OutputStream os = new FileOutputStream(toFile);
    IOUtils.copy(is, os);
}

```

fetchLinks()

Fetches all the URLs from the message template or content block content, so that you can insert tracking tags in them.

Supported Assets

The following list shows the Message Studio assets that are supported for the `fetchlinks()` operation:

- Content block (`contentBlockId`)
- Channel Template (`channelTemplateId`)
- Template (`templateId`)

Code Sequence

1. Initialize the `template` or `contentBlock` object to be sent to the `create` API, or, for an edit operation, fetch the `template` or `contentBlock` object using `list(filter)` API operation.
2. Call `fetchLinks(request)` using the `template` or `contentBlock` object to fetch the list of URLs in the form of `NamedLink` objects.
3. Modify the tracking tags in the URLs and set them in the `namedLinks` property of the `template` or `contentBlock`.
4. Save the `template` or `contentBlock`.

Supported Requests

The `fetchlinks()` operation supports multiple objects that can be passed in as arguments. Each object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

The following table maps the functional area to the object/request.

Functional area	Object/Request
Channel Template	<code>NamedLink/fetchLinks()</code>
Template	<code>NamedLink/fetchLinks()</code>
Content block	<code>NamedLink/fetchLinks()</code>

Response

`FetchLinksResponse`

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- authorizationFaultDetail
- invalidObjectFaultDetail
- objectNotFoundFaultDetail
- unexpectedFaultDetail
- unrecognizedObjectTypeFaultDetail

Sample code

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */

private List<NamedLink> fetchLinksForTemplate(Template template, MessageFormat format)
throws Exception {
    List<NamedLink> links = new ArrayList<NamedLink>();
    FetchLinksTemplateRequest request = new FetchLinksTemplateRequest();
    request.setTemplate(template); request.setMessageFormat(format);
    FetchLinksResponse response = service.fetchLinks(request);
    List<FetchLinkResponse> individualLinkResponse = response.getFetchLinkResponse();

    if (individualLinkResponse != null && individualLinkResponse.size() > 0)
    {
        for (FetchLinkResponse individualLink : individualLinkResponse)

```



```

        {
            links.add(individualLink.getNamedLink());
        }
    }

    return links;
}

private List<NamedLink> fetchLinksForContentBlock(ContentBlock contentBlock) throws
Exception {
    List<NamedLink> links = new ArrayList<NamedLink>();
    FetchLinksContentBlockRequest request = new FetchLinksContentBlockRequest();
    request.setContentBlock(contentBlock);
    FetchLinksResponse response = service.fetchLinks(request);
    List<FetchLinkResponse> individualLinkResponse = response.getFetchLinkResponse();

    if (individualLinkResponse != null && individualLinkResponse.size() > 0)
    {
        for (FetchLinkResponse individualLink : individualLinkResponse)
        {
            links.add(individualLink.getNamedLink());
        }
    }
    return links;
}

private void insertTrackingTagsInTemplate() {

    /*
    * Assuming the template object is ready, in create/edit operation.          *
    * The following call will fetch all URLs in the form of named link objects *
    * for the HTML message part of this template.                             */

    List<NamedLink> links = this.fetchLinksForTemplate(template, MessageFormat.HTML);

    /* By default, each a CLICKTAG will be assigned to each url. The assignment */
    /* can be changed as follows. */

    Map<String, TrackingTag> urlToTagMap = new HashMap<String, TrackingTag>();
    urlToTagMap.put("http://www.somedomain.com/link1", TrackingTag.ENCLICKTAG);
    urlToTagMap.put("http://www.somedomain.com/link2", TrackingTag.UNSUBSCRIBETAG);

    for (int i = 0 ; i < links.size() ; i++)
    {
        if (urlToTagMap.get(links.get(i).getUrl()) != null)

```

```

    {
        links.get(i).setTrackingTag(urlToTagMap.get(links.get(i).getUrl()));
        links.get(i).setName("link" + (i+1));
    }

/* Other properties like web analytics, link name, etc can also be set in the */
/* named link object. */
}

/* Assign the updated links to the message part. */
template.getMessagePart().get(0).getNamedLinks().clear();
template.getMessagePart().get(0).getNamedLinks().addAll(links);

/* Then save the contentBlock object the edit/create api. */
}

private void insertTrackingTagsInContentBlock() {

    /* Assuming the contentBlock object is ready, in create/edit operation. */

    List<NamedLink> links = this.fetchLinksForContentBlock(contentBlock);

/* By default, each a CLICKTAG will be assigned to each url. The assignment can be */
/* changed as follows. */

    Map<String, TrackingTag> urlToTagMap = new HashMap<String, TrackingTag>();
    urlToTagMap.put("http://www.somedomain.com/link1", TrackingTag.ENCLICKTAG);
    urlToTagMap.put("http://www.somedomain.com/link2", TrackingTag.UNSUBSCRIBETAG);
    for (int i = 0 ; i < links.size() ; i++)
    {
        if (urlToTagMap.get(links.get(i).getUrl()) != null)
        {
            links.get(i).setTrackingTag(urlToTagMap.get(links.get(i).getUrl()));
            links.get(i).setName("link" + (i+1));
        }
    }

/* Other properties like web analytics, link name etc can also be set in the named */
/* link object. */

}

/* Assign the updated links to the message part. */

contentBlock.getNamedLinks().clear();
contentBlock.getNamedLinks().addAll(links);

```

```

/* Then save the contentBlock object the edit/create api. */
}
}

```

findImport()

Returns the list of recurringImportId of given IDS/Extension.

Supported Assets

The findImport() operation supports the following Message Studio assets:

- Data Source (dataSourceId)

Recommendations

None.

Code Sequence

1. Get desired dataSourceId using get API operation.
2. Use the above dataSourceId in the findImport (findRecurringImportRequest).

Supported Requests

The findImport() function supports a single object:
findRecurringImportRequest.

Functional area	Object/Request
Data Source	InternalDataSourceId or ExtensionId/findRecurringImportRequest

Response

FindRecurringImportResponse—Contains list of RecurringImportIds.

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- objectNotFoundFault
- authorizationFault
- unexpectedFault

Sample code

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
/*
 * This method returns list of recurringImportId corresponding to the dataSourceId
 * @param dataSourceId
 * @return List of RecurringImportId corresponding to dataSourceId
 */
public List<RecurringImportId> findImport(DataSourceId dataSourceId) {
    FindRecurringImportRequest findRecurringImportRequest = new FindRecurringImportRequest
();
    if (dataSourceId instanceof ExtensionId) {
        findRecurringImportRequest.setExtensionId((ExtensionId) dataSourceId);
    }
    else {
        findRecurringImportRequest.setInternalDataSourceId((InternalDataSourceId)
dataSourceId);
    }
}

```

```

FindRecurringImportResponse findRecurringImportResponse = service.findImport
(findRecurringImportRequest)
return findRecurringImportResponse.getImportId();
}

```

get ()

Returns the object referenced by a specified `objectId`. In most cases, you will use the `list ()` operation to identify a set of `objectIDs`, and then use the `get ()` operation to return the requested object.

Supported Assets

The following list shows the Message Studio assets that are supported for the `get ()` operation:

- Internal Data Source (`internalDataSourceId`)
- External Data Source (`externalDataSourceId`)
- Campaign (`campaignId`)
- Target (`targetId`)
- Seed list (`seedListId`)
- Suppression list (`suppressionListId`)
- Template (`templateId`)
- (Assigned) Role (`assignedRoleId`)
- Role (`roleId`)
- Media Server (`mediaServerId`)
- Media Server Group (`mediaServerGroupId`)
- Program (`programId`)
- Extension - Catalog (`catalogId`)
- Extension - Standard (`extensionId`)
- Content block (`contentBlockId`)
- Attachment (`attachmentId`)
- Dynamic content (`ruleId`)
- Batch mailing (`standardMailingId`)
- Transactional mailing (`transactionalmailingId`)
- Tracking tags
- Organization (`organizationId`)
- System Address (`systemAddressId`)
- User (`userId`)
- StrongTools
- Channel (`channelId`)
- Channel Template (`channelTemplateId`)
- Channel Standard mailing (`channelStandardMailingId`)
- Channel Transactional mailings (`channelTransactionalMailingId`)
- Web analytics

Recommendations

None.

Code Sequence

1. Get the desired asset ID using **list(filter)** API operation.
2. `get(array_of_ids)`

Supported Requests

The `get()` operation supports multiple objects that can be passed in as arguments. Each object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

Note: You must pass in an array of objects for the `get()` operation. The array of objects may be defined as a single asset or as multiple assets.

The following table maps the functional area to the object/request.

Functional area	Object/Request	Specific Base Object
Organization	GetRequest	OrganizationId
User	GetRequest	UserId
Role	GetRequest	RoleId
Campaign	GetRequest	CampaignId
System address	GetRequest	SystemAddressId
Media server	GetRequest	MediaServerId
Web analytics	GetRequest	WebAnalyticsId
Mailing class	GetRequest	MailingClassId
External data source	GetRequest	ExternalDataSourceId
Internal data source	GetRequest	InternalDataSourceId
Target	GetRequest	TargetId
Seed list	GetRequest	SeedListId
Suppression	GetRequest	SuppressListId

Functional area	Object/Request	Specific Base Object
list		
Template	GetRequest	TemplateId, ChannelTemplateId
Content blocks	GetRequest	ContentBlockId
Attachment	GetRequest	AttachmentId
Dynamic content	GetRequest	RuleId
Batch mailing	GetRequest	Standard MailingId, ChannelStandardMailingId
Transactional mailing	GetRequest	TransactionalMailingId, ChannelTransactionalMailingId

Response

BatchGetResponse

The `BatchGetResponse` will include one or more `GetResponses` depending on the number of base objects you passed into the operation.

Faults

This operation does not throw a fault. Instead the `FaultDetail` is wrapped in the response. Any faults will be returned in the `BatchGetResponse`. You will need to check each individual `GetResponse` for the exception.

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- `authorizationFaultDetail`
- `objectNotFoundFaultDetail`
- `unexpectedFaultDetail`
- `unrecognizedObjectTypeFaultDetail`

Sample Code

```
/*
 * -----
 * Selligent, Inc.
 *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved.
 */
```

```

* Visit http://www.strongview.com for more information. *
*
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
*
* Distributing this source code outside your organization *
* is strictly prohibited. *
*
* ----- *
*/

/**
 * Get a list of objects based on give IDs. *
 * @param objectIds list of IDs that user want to a get
 * @return list of BaseObjects
 * @throws Exception any exception
 */

public List<BaseObject> get(List<? extends ObjectId> objectIds) throws Exception {

    /* Create request and set objectIds */
    GetRequest request = new GetRequest();
    request.getObjectId().addAll(objectIds);

    /* Create list of BaseObject with the same size as list of IDs */

    List<BaseObject> result = new ArrayList<BaseObject>(objectIds.size());

    /* Make call */

    BatchGetResponse batchGetResponse = service.get(request);

    /* Go through each response inside batch response and get the object if it is successful
    */
    /* Print out error message if any object is not retrieved successfully */
    /* and set the object to null. */

    for (int i = 0; i < objectIds.size(); i++)
    {
        if (batchGetResponse.getSuccess().get(i))
        {
            result.add(batchGetResponse.getGetResponse().get(i).getBaseObject());
        }
        else {

```



```

        FaultDetail faultDetail = batchGetResponse.getFault().get(i);
        System.out.println(faultDetail.getFaultCode() + ": " + fault-
Detail.getFaultMessage());
        result.add(null);
    }
}
return result;
}

```

getAllEasByMailingId()

This operation gets a list of EAS servers details (Host IP address, mailing configuration name, serial number, EAS ID).

Supported Assets

- Batch Mailing (standardMailingId)
- Transactional Mailing (transactionalMailingId)
- Channel Standard Mailing (channelStandardMailingId)
- Channel Transactional Mailing (channelTransactionalMailingId)

Recommendations

None.

Code Sequence

1. Get the desired asset ID using **list(filter)** API operation.
2. Use the asset ID in GetAllEasByMailingIdRequest and call GetAllEasByMailingId(request).

Supported Requests

Functional area	Object/Request
Mailing	GetAllEasByMailingId/GetAllEasByMailingIdRequest

Response

- GetAllEasByMailingResponse—Returns the host_ip, mailing_config_name, serial_no, eas_id.

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- UnexpectedFault

Sample Code

```

/*
 * -----
 * Selligent, Inc.
 *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved.
 *
 * Visit http://www.strongview.com for more information.
 *
 * You may incorporate this Source Code in your application only
 * if you own a valid license from Selligent, Inc.
 * and the copyright notices are not removed from the source code.
 *
 * Distributing this source code outside your organization
 * is strictly prohibited.
 *
 * -----
 */

public GetAllEasByMailingIdResponse getAllEasByMailingId(MailingId mailingId) {
    etAllEasByMailingIdRequest request = new GetAllEasByMailingIdRequest();
    request.setMailingId(mailingId);
    return svc.getAllEasByMailingId(request);
}

```

getImportDetails()

Returns the import details (data source, import name, and list of import results) of the given RecurringImportId.

Supported Assets

The `getImportDetails()` operation supports the following Message Studio assets:

- Internal Data Source (`internalDataSourceId`)

Recommendations

None.

Code Sequence

1. Set `RecurringImportId` property. The setting can be obtained using `findImport` API.
2. `getImport(getRecurringImportRequest)`

Supported Requests

The `getImportDetails()` function supports a single object:
`GetRecurringImportRequest`.

Functional Area	Object/Request
Data Source	<code>getRecurringImportRequest</code>

Response

`getRecurringImportResponse`—Returns import details.

Faults

Refer to [Error Handling on page 35](#) for a description of the following faults that may be returned:

- `objectNotFoundFault`
- `authorizationFault`
- `unexpectedFault`

Sample code

```
/*
 * ----- *
```

```

* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *
* *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* *
* Distributing this source code outside your organization *
* is strictly prohibited. *
* *
* ----- *
*/

FindRecurringImportRequest findRecurringImportRequest = new
FindRecurringImportRequest();
InternalDataSourceId dsId = new InternalDataSourceId();
dsId.setId("100");

findRecurringImportRequest.setInternalDataSourceId(dsId);
FindRecurringImportResponse recurringImportResponse = service.findImport
(findRecurringImportRequest);
RecurringImportId recurringImportId = recurringImportResponse.getImportId().get(0);
GetRecurringImportRequest importRequest = new GetRecurringImportRequest();
GetRecurringImportResponse importResponse = service.getImportDetails(importRequest);
RecurringImport recurringImport = importResponse.getImportDetails();

System.out.println(recurringImport.getImportResult().get(0).getStartTime());
System.out.println(recurringImport.getName());

```

getImportStatus()

This operation requests the import status of an Internal Data Source or Extension.

Supported Assets

- Internal Data Source (internalDataSourceId)
- Extension - Standard (extensionId)
- Extension - Catalog (catalogId)

Recommendations

None.

Code Sequence

1. Get desired asset ID using `list (filter)` API operation.
2. Use above extensionId/internalDatasourceId in `GetRecurringImportStatusRequest` and call `getImportStatus(request)`.

Supported Requests

Functional area	Object/Request
Extension/IDS	GetRecurringImportStatusRequest/getImportStatus ()

Response

- `GetRecurringImportStatusResponse`—Returns the `RecurringImportResult` that contains `ImportType`, `ImportFrom`, `status`, `startTime`, `endTime`, `recordsAdded`, etc.

Faults

Refer to [Error Handling on page 35](#) for a description of the following faults that may be returned:

- `objectNotFoundFault`
- `authorizationFault`
- `unexpectedFault`

Sample Code

```
/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
```

```

* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* *
* Distributing this source code outside your organization *
* is strictly prohibited. *
* *
* ----- *
*/

public List<RecurringImportResult> getImportStatus(DataSourceId id) throws
UnexpectedFault, AuthorizationFault, ObjectNotFoundFault {
    GetRecurringImportStatusRequest request = new GetRecurringImportStatusRequest();
    request.setInternalDataSourceId((InternalDataSourceId)id);
    return svc.getImportStatus(request).getImportResult();
}

```

getGlobalUndeliverable()

Searches for one or more undeliverable recipients from the database. You must provide a filter criteria for the search.

Supported Assets

The following list shows the Message Studio assets that are supported for the getGlobalUndeliverable() operation:

- Global undeliverable recipient address globalUndeliverable

Recommendations

None.

Code Sequence

1. getGlobalUndeliverable(globalUndeliverableRequest)

Supported Requests

The getGlobalUndeliverable() operation supports a single object: globalUndeliverableRequest.

Response

globalUndeliverableResponse—Contains list of GlobalUndeliverableInfo (recipientAddress, channelType, status, lastBounceTime, failureCode).

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- objectNotFoundFault
- authorizationFault
- invalidObjectFault

Sample Code

```
/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */

public void getGlobalUndeliverable (String messageDestination, String channelType, String
organizationId) throws Exception{
    System.out.println("Start of getGlobalUndeliverables");
    GlobalUndeliverableRequest request = prepareGlobalUndeliverableRequest(channelType,
organizationId);
    request.setRecipientAddress(messageDestination);
    GlobalUndeliverableResponse response = service.getGlobalUndeliverable(request);
}
```

```

    if (response != null && response.getGlobalUndeliverableInfo() != null &&
response.getGlobalUndeliverableInfo().size()>0) {
    printRecords(response.getGlobalUndeliverableInfo());
    } else {
        System.out.println("No Records found");
    }
    System.out.println("End of getGlobalUndeliverables");
}

```

getProgramNodes ()

Returns a list of nodes used in a specified Lifecycle Marketing Program asset.

Supported Assets

The `getProgramNodes()` operation supports the following Message Studio asset:

- Program (`programId`)

Recommendations

None.

Code Sequence

1. Get desired asset ID using `list (filter)` API operation.
2. `getProgramNodes(request)`

Supported Requests

The `getProgramNodes()` operation supports a single object that can be passed in as an argument. The object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

The following table maps the functional area to the object/request.

Functional area	Object/Request
Program	<code>programId/GetProgramNodesRequest</code>

Response

```
getProgramNodesResponse.getProgramNodes
list<ProgramNode>
```

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- authorizationFault
- invalidObjectFault
- objectNotFoundFault
- unexpectedFault

Sample Code

```
/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
public void getProgramNodeList(ProgramId programId) throws Exception
{
    GetProgramNodesRequest nodesRequest = new GetProgramNodesRequest();
    nodesRequest.setProgramId(programId);
    GetProgramNodesResponse nodes = service.getProgramNodes(nodesRequest);
    System.out.println("Program Nodes:");
    for(ProgramNode node : nodes.getProgramNodes())
    {
```

```

        System.out.println(
            "\t" + node.getObjectId().getId() + " " + node.getName()
            + " Modified:" + node.getModifiedTime()
            + " Version:" + node.getVersion()
        );
    }
}

```

getRecords ()

This operation retrieves up to 10 data source records that match a specified criteria.

The `getRecords` operation accepts a list of fields to be matched and will retrieve up to 10 matching records. It throws an exception if more than 10 records are matched.

Each of the match fields specified must be one of the following:

- Primary Key field
- Email Address field
- SMS Address field

The response may contain no records if the match fields do not match any record in the data source. In other words, the presence of the `dataSourceRecords` field is optional in the response.

Supported Assets

The following list shows the Message Studio assets that are supported for the `getRecords ()` operation:

- Internal data sources (`internalDataSourceId`)
- Extension - Standard (`extensionId`)
- Extension - Catalog (`catalogId`)

Recommendations

None.

Faults

Refer to **Error Handling on page 35** for a description of any faults that may be returned.

The `getRecords` operation does not allow the following scenarios and throws an `InvalidObjectFault` exception:

- The specified data source (`dataSourceId`) is not of type: `Internal`.
- A non-existent column is specified in the match fields.
- A column specified in the match fields is neither the primary key, nor an email address field nor an SMS address field.
- A column is specified more than once in the match fields.
- The response contains more than 10 records that satisfy the match criteria.

Get Records Sample Code

```
/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */

/* Get records using the getRecords API. */

public void getRecord(InternalDataSourceId dataSourceId) throws Exception
{

/* Wait for data source status to be idle before adding */
```

```

    helper.waitForIdle(dataSourceId);

/* Use primary key (id) to fetch the records */

    GetDataSourceRecordsRequest getRecordRequest1 = new GetDataSourceRecordsRequest();
    getRecordRequest1.setDataSourceId(dataSourceId);
    NameValuePair idCol = new NameValuePair();
    idCol.setName("id");
    idCol.setValue("903");
    getRecordRequest1.getMatchFields().add(idCol);
    GetDataSourceRecordsResponse getRecordsResponse1 =
(GetDataSourceRecordsResponse) service.getRecords(getRecordRequest1);
    if (getRecordsResponse1.getDataSourceRecord() != null &&
getRecordsResponse1.getDataSourceRecord().size() > 0)
    {
        for (DataSourceRecord record : getRecordsResponse1.getDataSourceRecord())
        {
            printDataSourceRecord(record);
        }
    }

/* Wait for data source status to be idle before adding */
    helper.waitForIdle(dataSourceId);

/* Use the email field to fetch the records */
    GetDataSourceRecordsRequest getRecordRequest2 = new Get- DataSourceRecordsRequest();
    getRecordRequest2.setDataSourceId(dataSourceId);
    NameValuePair emailCol = new NameValuePair();
    emailCol.setName("email_address");
    emailCol.setValue("email_for_cxf_upsert-updated@cxf.com");
    getRecordRequest2.getMatchFields().add(emailCol);
    GetDataSourceRecordsResponse getRecordsResponse2 =
(GetDataSourceRecordsResponse) service.getRecords(getRecordRequest2);
    if (getRecordsResponse2.getDataSourceRecord() != null &&
getRecordsResponse2.getDataSourceRecord().size() > 0)
    {
        for (DataSourceRecord record : getRecordsResponse2.getDataSourceRecord())
        {
            printDataSourceRecord(record);
        }
    }
}

```

Request Sample Code

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * ----- *
 */
/*
 * Validate if a template's xsl content is valid.
 *
 * @param templateId the TemplateId whose XSL content will be validated.
 * @return if the content is valid XSL
 * @throws Exception any exception
 */

public boolean validateXsl(TemplateId templateId) throws Exception {
    ValidateXslRequest request = new ValidateXslRequest();
    request.setMessageFormat(MessageFormat.HTML);
    request.setTemplateId(templateId);
    return service.validateXsl(request).isValid();
}

<xs:complexType name="GetDataSourceRecordsRequest">
  <xs:complexContent>
    <xs:extension base="tns:GetRecordsRequest">
      <xs:sequence>
        <xs:element name="dataSourceId" type="tns- :DataSourceId"/>
        <xs:element name="matchFields" type="tns- :NameValuePair" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

Response Sample Code

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * ----- *
 */
/*
 * Validate if a template's xsl content is valid.
 *
 * @param templateId the TemplateId whose XSL content will be validated.
 * @return if the content is valid XSL
 * @throws Exception any exception
 */

public boolean validateXsl(TemplateId templateId) throws Exception {
    ValidateXslRequest request = new ValidateXslRequest();
    request.setMessageFormat(MessageFormat.HTML);
    request.setTemplateId(templateId);
    return service.validateXsl(request).isValid();
}

<xs:complexType name="GetDataSourceRecordsResponse">
  <xs:complexContent>
    <xs:extension base="tns:GetRecordsResponse">
      <xs:sequence>
        <xs:element name="dataSourceRecord" type="tns- :DataSourceRecord"
maxOccurs="unbounded" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</Faults

```

```
</xs:complexType>
```

getStatistics()

This operation returns basic statistics about a particular asset. This operation is supported for data sources, extensions, and mailings. The `getStatistics()` operation can return the following information:

For Data Sources/Extensions:

- Total number of invalid email addresses
- Total number of malformed records
- Total number of records
- Total number of unsubscribed recipients
- Time of the last data refresh

For Mailings:

- Serial number of the launched mailing
- Amount of elapsed time since the mailing launched
- Launch time
- Completion time
- Number of deferred deliveries
- Number of successful deliveries
- Number of failed deliveries
- Number of invalid email addresses included in the target
- Number of sent emails (this successful deliveries + failed deliveries) | number of targeted recipients
- Total number of clicks
- Total number of complaints
- Total number of opens

- Total number of unsubscribes
- number of clicks by unique recipients
- Number of complaints by unique recipients
- Number of opens by unique recipients
- Number of unsubscribes by unique recipients

Supported Assets

The following list shows the Message Studio assets that are supported for the `getStatistics()` operation:

- | | |
|-----------------------------------------------------------------|------------------------------------------------------------------------------------|
| • External Data Source
(<code>externalDataSourceId</code>) | • Batch mailing (<code>mailingId</code>) |
| • Internal Data Source
(<code>internalDataSourceId</code>) | • Transactional mailing
(<code>transactionalMailingId</code>) |
| • Extension - Catalog (<code>catalogId</code>) | • Channel Standard mailing
(<code>channelStandardMailingId</code>) |
| • Extension - Standard
(<code>extensionId</code>) | • Channel Transactional mailings
(<code>channelTransactionalMailingId</code>) |

Recommendations

None.

Code Sequence

1. Get desired asset ID using `list (filter)` API operation.
2. `getStatistics(request)`

Supported Requests

The `getStatistics()` operation supports multiple objects that can be passed in as arguments. Each object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

Note: You must pass in an array of objects for the `getStatistics()` operation. The array of objects may be defined as a single asset or as multiple assets.

The following table maps the functional area to the object/request.

Functional area	Object/Request
Internal data source	GetDataSourceStatisticsRequest
External data source	GetDataSourceStatisticsRequest
Batch mailing	GetMailingStatisticsRequest
Transactional mailing	GetMailingStatisticsRequest
Channel Standard mailing	GetMailingStatisticsRequest
Channel Transactional mailing	GetMailingStatisticsRequest
Extension - Catalog	GetDataSourceStatisticsRequest
Extension - Standard	GetDataSourceStatisticsRequest

Response

getStatisticsResponse

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- objectNotFoundFault
- authorizationFault
- unrecognizedObjectTypeFault
- unexpectedFault

Sample Code

```
/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
```

```

* ----- *
*/

/**
 * Retrieve and print out a data source.
 *
 * @param dataSoruceId the data source ID that will be retrieved and printed.
 * @throws Exception any exception
 */

public void getAndPrintStatistics(DataSourceId dataSourceId) throws Exception {

    /* Create request */
    GetDataSourceStatisticsRequest request = new GetDataSourceStatisticsRequest();
    request.setDataSourceId(dataSourceId);

    /* Make call */
    GetDataSourceStatisticsResponse response = (GetDataSourceStatisticsResponse)
service.getStatistics(request);
    System.out.println("\nStatistics for data source " + dataSourceId.getId());
    System.out.println("Total records: " + response.ge- tTotalRecords());
    System.out.println("Total invalid records: " + response.getTotalInvalidRecords());
    System.out.println("Total malformed records: " + response.getTotalMalformedRecords());
    System.out.println("Total unsub records: " + response.getTotalUnsubscribedRecords());
}

```

getTxnEasInfo()

Returns the EAS details and transactional mailing details (i.e., host name, mailing handle) of the given asset ID.

Supported Assets

The following list shows the Message Studio assets that are supported for the getTxnEasInfo() operation:

- Transactional Mailing (transactionalMailingId)
- Channel Transactional Mailing (channelTransactionalMailingId)

Code Sequence

1. Set TransactionMailingId property.
2. getTxnEasInfo (getTxnEasInfoRequest)

Supported Requests

The getTxnEasInfo () function supports a single object: GetTxnEasInfoRequest.

Functional area	Object/Request
Transactional Mailing	TransactionalMailingId/GetTxnEasInfoRequest
Channel Transactional Mailing	ChannelTransactionalMailingId/GetTxnEasInfoRequest

Response

GetTxnEasInfoResponse

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- objectNotFoundFault
- authorizationFault
- unexpectedFault

Sample Code

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *

```

```

* Distributing this source code outside your organization          *
* is strictly prohibited.                                         *
*                                                                  *
* ----- *
*/

GetTxnEasInfoRequest infoRequest = new GetTxnEasInfoRequest();
TransactionalMailingId mailingId = new TransactionalMailingId();
mailingId.setId("100");
infoRequest.setMailingId(mailingId);
GetTxnEasInfoResponse infoResponse = svc.getTxnEasInfo(infoRequest);
System.out.println(infoResponse.getHostname());
System.out.println(infoResponse.getMailingHandle());
System.out.println(infoRequest.getMailingId());

```

getTxnMailingHandle()

This operation returns the mailing handle of the mailing to be used in **txnSend()** API operation.

Supported Assets

The following list shows the Message Studio assets that are supported for the getTxnMailingHandle() operation:

- Transactional Mailing (transactionalMailingId)
- Channel Transactional Mailing (channelTransactionalMailingId)

Code Sequence

1. Set TransactionMailingId property.
2. getTxnMailingHandle(getTxnEasInfoRequest);
3. getImport(getRecurringImportRequest);

Supported Requests

The getTxnMailingHandle() function supports a single object: GetTxnEasInfoRequest.

Functional area	Object/Request
Transactional Mailing	TransactionalMailingId/GetTxnMailingHandleRequest

Functional area	Object/Request
Channel Transactional Mailing	ChannelTransactionalMailingId/GetTxnMailingHandleRequest

Response

GetTxnMailingHandleResponse

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- objectNotFoundFault
- authorizationFault
- unexpectedFault

Sample code

```

/*
 * -----
 * Selligent, Inc.
 *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved.
 *
 * Visit http://www.strongview.com for more information.
 *
 * You may incorporate this Source Code in your application only
 * if you own a valid license from Selligent, Inc.
 * and the copyright notices are not removed from the source code.
 *
 * Distributing this source code outside your organization
 * is strictly prohibited.
 *
 * -----
 */
GetTxnMailingHandleRequest handleRequest = new
GetTxnMailingHandleRequest();
TransactionalMailingId mailingId = new TransactionalMailingId();
mailingId.setId("100");
handleRequest.setMailingId(mailingId);
/* OR we can set handleRequest.setMailingName("demo_tx_email_name"); */
GetTxnMailingHandleResponse handleResponse = svc.getTxnMailingHandle(handleRequest);
System.out.println(handleResponse.getHandle());

```

```
System.out.println(recurringImport.getImportResult().get(0).getStartTime());
System.out.println(recurringImport.getName());
```

importContent()

Imports the actual attachment after you create an attachment using the `create()` operation.

Supported Assets

The following list shows the Message Studio assets that are supported for the `importContent()` operation:

- Attachment (attachmentId)

Recommendations

None.

Code Sequence

1. Get desired asset ID using `list (filter)` API operation.
2. `importContent (request)`

or

1. `create (array_of_objects)`
2. `importContent (request)`

Supported Requests

The `importContent()` operation supports a single object that can be passed in as an argument. The object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

The following table maps the functional area to the object/request.

Functional area	Object/Request
Attachment	attachmentId/ImportAttachmentContentRequest

Response

```
importContentResponse
```

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- objectNotFoundFault
- authorizationFault
- unrecognizedObjectTypeFault
- unexpectedFault

Sample Code

```
/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */

/**
 * Import attachment.
 *
 * @param attachmentId the AttachmentId that will be imported to.
 * @param attachmentFile the attachment file that will be imported.
 * @return if the importing operation is successful.
 * @throws Exception any exception
 */
```

```

public boolean importContent(AttachmentId attachmentId, File attachmentFile) throws
Exception {
    ImportAttachmentContentRequest request = new ImportAttachmentContentRequest();
    request.setAttachmentId(attachmentId);
    request.setFileName(attachmentFile.getName());

    /* Set content */

    InputStream is = new FileInputStream(attachmentFile);
    DataHandler dh = new DataHandler(new ByteArrayDataSource(is, "application/octet-
stream"));
    request.setContent(dh);
    return service.importContent(request).isSuccess();
}
}

```

importFtpFile()

This operation imports an FTP file that resides on the server.

Message Studio looks for the file in the directory:

<Installation directory>/data/ftp_upload/<organization>/<department>

Supported Assets

- Internal Data Source (dataSourceId)
- Extension - Standard (extensionId)

Recommendations

None.

Code Sequence

1. Get desired asset ID using **list (filter)** API operation.
2. Specify above dataSourceId, ftp-filename, contains_field_name (optional), field_delimiter, import_mode in (ADD_RECORDS, etc.) in ImportFtpFileRequest.
3. Call importFtpFile(request) API operation.

Supported Requests

Functional area	Object/Request
Data Source	ImportFtpFileRequest/importFtpFile

Response

- ImportFtpFileResponse—Returns a boolean (true/false) indication of success.

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- authorizationFault
- invalidObjectFault
- concurrentModificationFault
- unrecognizedObjectTypeFault
- objectNotFoundFault
- unexpectedFault

Sample Code

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */

```

```

public boolean importFtpFile(DataSourceId dataSourceId,String ftpFileName)
{
    ImportFtpFileRequest ftpFileRequest = new ImportFtpFileRequest();
    ftpFileRequest.setDataSourceId(dataSourceId);
    ftpFileRequest.setContainsFieldNames(true);
    ftpFileRequest.setFieldDelimiter("::");
    ftpFileRequest.setImportMode("ADD_RECORDS");
    return svc.importFtpFile(ftpFileRequest).isSuccess();
}

```

importMessagePart()

This operation imports Template content.

Supported Assets

The following list shows the Message Studio assets that are supported for the `importMessagePart()` operation:

- Template (`templateId`)
- Channel Template (`channelTemplateId`)

Recommendations

None.

Code Sequence

1. Get desired asset ID using `list (filter)` API operation.
2. `importMessagePart (request)`

or

1. `create (array_of_objects)`
2. `importMessagePart (request)`

Supported Requests

The `importMessagePart()` operation supports a single object that can be passed in as an argument. The object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

The following table maps the functional area to the object/request.

Functional area	Object/Request
Template	templateId/ImportMessagePartRequest
Channel Template	channelTemplateId/ImportMessagePartRequest

Response

`importMessagePartResponse`

Faults

Refer to [Error Handling on page 35](#) for a description of the following faults that may be returned:

- `objectNotFoundFault`
- `authorizationFault`
- `unexpectedFault`

Sample Code

```
/*
 * -----
 * Selligent, Inc.
 *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved.
 *
 * Visit http://www.strongview.com for more information.
 *
 * You may incorporate this Source Code in your application only
 * if you own a valid license from Selligent, Inc.
 * and the copyright notices are not removed from the source code.
 *
 * Distributing this source code outside your organization
 * is strictly prohibited.
 */
```

```

* ----- *
*/

/**
 * Import @param added to message part using a zip file and media server. *
 * templateId the template that message part will be @param the image files.*
 * mediaServerId the media server that is used to store @param zipFile the *
 * zip file that contains HTML and image files. @param folderName the *
 * directory to unzip the zip file * @return if the importing operation is *
 * successful @throws Exception any exception *
 */

public boolean importMessagePart(TemplateId templateId, MediaServerId mediaServerId, File
zipFile, String folderName) throws Exception {

    /* Create request */
    ImportMessagePartRequest request = new Import- MessagePartRequest();
    request.setFolderName(folderName);
    request.setIsXsl(false);
    request.setMediaServerId(mediaServerId);
    request.setTemplateId(templateId);

    /* Set zip file */
    InputStream is = new FileInputStream(zipFile);
    DataHandler dh = new DataHandler(new ByteArrayDataSource(is, "application/octet-
stream"));
    request.setZipFile(dh);
    return service.importMessagePart(request).isSuccess();
}

```

launch()

This operation launches an approved mailing.

Supported Assets

The following list shows the Message Studio assets that are supported for the launch () operation:

- Batch Mailing (standardMailingId)
- Channel Standard Mailing (channelStandardMailingId)

Recommendations

None.

Code Sequence

1. Get desired asset ID using `list (filter)` API operation.
2. `launch(request)`

Supported Requests

The `launchMailing()` operation supports multiple objects that can be passed in as arguments. Each object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

The following table maps the functional area to the object/request.

Functional area	Object/Request
Batch Mailing	standardMailingId/LaunchRequest
Channel Standard Mailing	channelStandardId/LaunchRequest

Response

`launchResponse`

Faults

Refer to [Error Handling on page 35](#) for a description of the following faults that may be returned:

- `objectNotFoundFault`
- `authorizationFault`
- `unexpectedFault`

Sample Code

```
/*
 * -----
 * Selligent, Inc.
 *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved.
```

```

*                                                                    *
* Visit http://www.strongview.com for more information.             *
*                                                                    *
* You may incorporate this Source Code in your application only      *
* if you own a valid license from Selligent, Inc.                   *
* and the copyright notices are not removed from the source code.   *
*                                                                    *
* Distributing this source code outside your organization           *
* is strictly prohibited.                                           *
*                                                                    *
* ----- *
*/

/*                                                                    *
* Launch a Standard Mailing.                                         *
* @param mailingId the ID of Standard Mailing that will be launched *
* @return if launch operation is successful                          *
* @throws Exception any exception                                    *
*/
public boolean launchMailing(StandardMailingId mailingId) throws Exception {
    LaunchRequest request = new LaunchRequest();
    request.setMailingId(mailingId);
    return service.launch(request).isSuccess();
}

```

list()

Use the `list()` operation to query Message Studio for a list of ids associated with a specific or series of specific assets. The `list()` operation requires you to set a filter that will return the ids of specific assets. You will basically set a series of conditions that you expect the returned list to conform to. For example, the sample code below searches all system addresses for a specific email address.

Supported Assets

The following list shows the Message Studio assets that are supported for the `list()` operation:

- Internal Data Source (internalDataSourceId)
- External Data Source (externalDataSourceId)
- Content block (contentBlockId)
- Attachment (attachmentId)
- Dynamic content (ruleId)

- Campaign (campaignId)
- Target (targetId)
- Seed list (seedListId)
- Suppression list (suppressionListId)
- Template (templateId)
- (Assigned) Role (assignedRoleId)
- Role (roleId)
- Media Server (mediaServerId)
- Media Server Group (mediaServerGroupId)
- Program (programId)
- Extension - Catalog (catalogId)
- Extension - Standard (extensionId)
- Batch mailing (standardMailingId)
- Transactional mailing (transactionalmailingId)
- Tracking tags
- Organization (organizationId)
- System Address (systemAddressId)
- User (userId)
- StrongTools
- Channel (channelId)
- Channel Template (channelTemplateId)
- Channel Standard mailing (channelStandardMailingId)
- Channel Transactional mailings (channelTransactionalMailingId)
- Web analytics

Recommendations

Each `listRequest` takes a filter and there is one filter for each object type. Then, you can set one or more conditions for each filter. There are two condition types:

- **Scalar**—Used for a single value
- **Array**—Used for a list of values, such as mailing types, or mailing status

Each condition has an operator, such as `EQUAL` and a value. Based on the condition, a value may be a string, ID, integer, or boolean. In one filter, you can set multiple conditions.

This may include page number, number of records per page, ordering type, or order by fields for each filter. If you do not choose to set these, the default values will be `pagenumber=0` (first page), `number of records=10` per page, and ascending order. The max number of records you can set is 200.

Code Sequence

1. `list(filter)`

Supported Requests

The `list()` operation supports multiple objects that can be passed in as arguments. Each object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

The following table maps the functional area to the object/request.

Functional area	Object/Request	Specific Base Object
Organization	ListRequest	OrganizationId
User	ListRequest	UserId
Role	ListRequest	RoleFilter
Campaign	ListRequest	CampaignFilter
System address	ListRequest	SystemAddressFilter
Media server	ListRequest	MediaServerFilter
Web analytics	ListRequest	WebAnalyticsFilter
Mailing Server Group	ListRequest	MailingClassFilter
External data source	ListRequest	DataSourceFilter
Internal data source	ListRequest	DataSourceFilter
Target	ListRequest	TargetFilter
Seed list	ListRequest	SeedListFilter
Suppression list	ListRequest	SuppressListFilter
Template	ListRequest	TemplateFilter, ChannelTemplateFilter
Content blocks	ListRequest	ContentBlockFilter
Attachment	ListRequest	AttachmentFilter
Dynamic Content	ListRequest	RuleFilter
Batch mailing	ListRequest	MailingFilter, ChannelMailingFilter
Transactional mailing	ListRequest	MailingFilter, ChannelMailingFilter
Program	ListRequest	ProgramFilter
Assigned Role	ListRequest	AssignedRoleFilter
Channel	ListRequest	ChannelFilter

Response

`listResponse`

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- unrecognizedObjectTypeFault
- unexpectedFault
- invalidObjectFault

Sample Code

Sample 1: Filter system addresses by type and a specific email address

```

/*
 * -----
 * Selligent, Inc.
 *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved.
 *
 * Visit http://www.strongview.com for more information.
 *
 * You may incorporate this Source Code in your application only
 * if you own a valid license from Selligent, Inc.
 * and the copyright notices are not removed from the source code.
 *
 * Distributing this source code outside your organization
 * is strictly prohibited.
 * -----
 */

/*
 * List SystemAddress by type (bounce/from/reply) and email address.
 * There is no filter by email address, so we need to get all address
 * for one type and filter them again.
 *
 * @param type the SystemAddressType to filter on
 * @param emailAddress the email address to match
 * @return the SystemAddress that belongs to the type and matches
 * the email address
 * @throws Exception any exception
 */

```

```

public SystemAddress listByEmailAddress(SystemAddressType type, String emailAddress)
throws Exception {

    /* Create and set attribute for filter */

    SystemAddressFilter filter = new SystemAddressFilter();
    helper.setFilter(filter);

    /* Create type condition */

    ScalarStringFilterCondition typeCondition = new ScalarStringFilterCondition();
    typeCondition.setOperator(FilterStringScalarOperator.EQUAL);

    /* Set type condition value */

    if (type.equals(SystemAddressType.BOUNCE))
    {
        typeCondition.setValue(SystemAddressType.BOUNCE.value());
    }
    else if (type.equals(SystemAddressType.FROM))
    {
        typeCondition.setValue(SystemAddressType.FROM.value());
    }
    else if (type.equals(SystemAddressType.REPLY))
    {
        typeCondition.setValue(SystemAddressType.REPLY.value());
    }
    filter.setTypeCondition(typeCondition);

    /* Create request and make call */
    ListRequest request = new ListRequest();
    request.setFilter(filter);
    List<ObjectId> addressIds = service.list(request).getObjectId();

    /* Go through all addresses to get the one that matches email address. */

    for (ObjectId each: addressIds)
    {
        SystemAddress address = (SystemAddress) helper.get((SystemAddressId) each);
        if (address.getEmailAddress().equals(emailAddress))
        {
            return address;
        }
    }
    return null;
}

```

```

    }
}

```

Sample 2: Lists mailings by name and returns one mailing if the name is matched

```

/*
 * ----- *
 * Selligent, Inc. *
 * * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * * *
 * Visit http://www.strongview.com for more information. *
 * * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * * *
 * ----- *
 */
/*
 * List mailings by name: this should return one mailing if name is matched.
 *
 * @param name the mailing name to be listed
 * @throws Exception any exception
 */

public void listByName(String name) throws Exception {
    MailingFilter filter = new MailingFilter();
    helper.setFilter(filter);

    /* Create name condition */
    ScalarStringFilterCondition nameCondition = new ScalarStringFilterCondition();
    nameCondition.setOperator(FilterStringScalarOperator.EQUAL);
    nameCondition.setValue(name);
    filter.setNameCondition(nameCondition);

    /* Set the filter to request */
    ListRequest request = new ListRequest();
    request.setFilter(filter);

    /* Make call and print results */

```

```
    printListResponse(service.list(request));
}
```

Sample 3: Lists mailings in a specific organization

```
/*
 * ----- *
 * Selligent, Inc. *
 * * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * * *
 * Visit http://www.strongview.com for more information. *
 * * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * * *
 * ----- *
 */
/*
 * List mailing in an organization: this can be done using an empty filter.
 * The organization is the one from header.
 * @throws Exception any exception
 */
public void listByOrganization() throws Exception {

    /* Create empty filter: no conditions */

    MailingFilter filter = new MailingFilter(); helper.setFilter(filter);

    /* Set the filter to request */

    ListRequest request = new ListRequest();
    request.setFilter(filter);

    /* Make call and print results */

    printListResponse(service.list(request));
}
```

Sample 4: Lists mailings by type and status

```
/*
```

```

* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *
* *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* *
* Distributing this source code outside your organization *
* is strictly prohibited. *
* *
* ----- *
*/
/**
 * List mailings by mailing type and status.
 *
 * @throws Exception any exception
 */

public void listByTypeAndStatus() throws Exception {

    /* Create a mailing filter and set attribute and order by list */
    MailingFilter filter = new MailingFilter(); helper.setFilter(filter); filter.getOrderBy()
    ().add(MailingOrderBy.NAME); filter.getOrderBy().add(MailingOrderBy.MODIFIED_TIME);

    /* Create status condition: status in (ACTIVE, EDITIN, COMPLETED) */
    ArrayStringFilterCondition statusCondition = new ArrayStringFilterCondition();
    statusCondition.setOperator(FilterArrayOperator.IN);
    statusCondition.getValue().add(MailingStatus.ACTIVE.value());
    statusCondition.getValue().add(MailingStatus.EDITING.value());
    statusCondition.getValue().add(MailingStatus.COMPLETED.value());
    filter.setStatusCondition(statusCondition);

    /* Create Type Condition: type in (RECURRING, TRANSACTIONAL) */
    ArrayStringFilterCondition typeCondition = new ArrayStringFilterCondition();
    typeCondition.setOperator(FilterArrayOperator.IN);
    typeCondition.getValue().add(MailingType.RECURRING.value());
    typeCondition.getValue().add(MailingType.TRANSACTIONAL.value());
    filter.setTypeCondition(typeCondition);

    /* Set the filter to request */
    ListRequest request = new ListRequest();

```

```

    request.setFilter(filter);

/* Make call and print results */
    printListResponse(service.list(request));
}

private void printListResponse(ListResponse response) throws Exception {
    List<ObjectId> mailingIds = response.getObjectId();
    System.out.println("\nList Response:");
    for (ObjectId each: mailingIds) {
        Mailing mailing = (Mailing) helper.get(each);
        System.out.println("\t" + each.getId() + " " + mailing.getName());
    }
}
}

```

load()

This operation loads an approved transactional mailing.

Supported Assets

The following list shows the Message Studio assets that are supported for the `load()` operation:

- Transactional Mailing (`transactionalMailingId`)
- Channel Transactional Mailing (`channelTransactionalMailingId`)

Recommendations

None.

Code Sequence

1. Get desired asset ID using `list (filter)` API operation.
2. `load(request)`

Supported Requests

The `loadMailing()` operation supports multiple objects that can be passed in as arguments. Each object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

The following table maps the functional area to the object/request.

Functional area	Object/Request
Transactional mailing	transactionalMailingId/loadRequest
Channel Transactional mailing	channelTransactionalMailingId/loadRequest

Response

`loadResponse`

Faults

Refer to [Error Handling on page 35](#) for a description of the following faults that may be returned:

- `objectNotFoundFault`
- `authorizationFault`
- `unexpectedFault`

Sample Code

```
/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
```

```

* ----- *
*/

/**
 * Load a Transactional Mailing.
 * @param mailingId the ID of Transactional Mailing that will be loaded
 * @return if load operation is successful
 * @throws Exception any exception
 */
public boolean loadMailing(TransactionalMailingId mailingId) throws Exception {
    LoadRequest request = new LoadRequest();
    request.setMailingId(mailingId);
    return service.load(request).isSuccess();
}

```

pause ()

This operation pauses a standard or transactional mailing. Pausing allows you to update template, content, or headers. Paused mailings can be resumed using the [resume](#) API option. You cannot modify the schema when you pause a mailing.

Supported Assets

The following list shows the Message Studio assets that are supported for the `pause ()` operation:

- Batch mailing (`standardMailingId`)
- Transactional mailing (`transactionalMailingId`)
- Channel Standard mailing (`channelStandardMailingId`)
- Channel Transactional mailing (`channelTransactionalMailingId`)

Recommendations

None.

Code Sequence

1. Get desired asset ID using `list (filter)` API operation.
2. `pause (request)`

Supported Requests

The `pause()` operation supports multiple objects that can be passed in as arguments. Each object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

The following table maps the functional area to the object/request.

Functional area	Object/Request
Batch mailing	standardMailingId/PauseRequest
Transactional mailing	transactionalMailingId/PauseRequest
Channel Standard Mailing	channelStandardMailingId/PauseRequest
Channel Transactional Mailing	channel transactionalMailingId/PauseRequest

Response

`pauseResponse`

Faults

Refer to [Error Handling on page 35](#) for a description of the following faults that may be returned:

- `objectNotFoundFault`
- `authorizationFault`
- `unexpectedFault`

Sample Code

```
/*
 * -----
 * Selligent, Inc.
 *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved.
 *
 * Visit http://www.strongview.com for more information.
 *
 * You may incorporate this Source Code in your application only
 * if you own a valid license from Selligent, Inc.
 * and the copyright notices are not removed from the source code.
 *
 * Distributing this source code outside your organization
```

```

* is strictly prohibited. *
* *
* ----- *
*/

/**
 * Pause a standard or transactional mailing. *
 * @param mailingId the ID of the Mailing that will be paused
 * @return if close operation is successful
 * @throws Exception any exception */

public boolean pause(standardMailingId mailingId) throws Exception {
    PauseRequest request = new PauseRequest();
    request.setMailingId(mailingId);
    return service.pause(request).isSuccess();
}

```

personalize()

This operation returns the content associated with a mailing or template after replacing the tokens in the content with the specified values. The operation also performs XSL translations of the tokens.

You can use this operation for testing a personalized mailing or a template that has tokens already embedded in it.

Note: If no tokens are specified in a request, the tokens in the message will be replaced by blanks. This behavior is consistent with UI. However, this operation does not allow the user to specify a target row as they can in the UI.

Supported Assets

The `personalize()` operation supports the following Message Studio assets:

- Batch mailing (`standardMailingId`)
- Transactional mailing (`transactionalMailingId`)
- Channel Standard mailing (`channelStandardMailingId`)
- Channel Transactional mailing (`transactionalMailingId`)

- Template (`templateId`)
- Channel Template (`channelTemplateId`)

Recommendations

None.

Code Sequence

1. Get the desired asset ID using **list(filter)** API operation.
2. `personalize(request)`

Supported Requests

The `personalize()` operation supports multiple objects that can be passed in as arguments. Each object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

The `personalize()` operation accepts two different requests:

- `PersonalizeMailingRequest`—Specifies the *MailingID* and other details of a mailing and optionally specifies tokens to personalize in the mailing. This request requires that you specify the following:
 - `messageFormat`—Specifies the mailing format (text, html or mime)
 - `useMultiPart`—Set to "true" if you are specifying multiple message formats (Text, HTML, etc.).
 - `personalizationTokens`—Specifies the tokens to be personalized using this operation.
- `PersonalizeTemplateRequest`—Specifies the *TemplateId* of an existing template and optionally specifies tokens to personalize in the template.

Response

`personalizeResponse`

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- authorizationFault
- invalidObjectFault
- objectNotFoundFault
- unexpectedFault

Sample Code

The following are samples that show the use of the following requests:

- [PersonalizeMailingRequest Sample Code on page 132](#)
- [PersonalizeTemplateRequest Sample Code on page 133](#)

PersonalizeMailingRequest Sample Code

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */

public String personalizeMailing(MailingId mailingId) throws InvalidObjectFault,
UnexpectedFault, ObjectNotFoundFault, AuthorizationFault
Supported requests{
    PersonalizeMailingRequest request = new PersonalizeMailingRequest();
    // Specify the mailing ID
    request.setMailingId(mailingId);

    // Specify MIME type (html/text/mime) using getMessageFormat request
    // NOTE: You may use multiple getMessageFormat().add() statements
    // to specify multiple MIME types, but they must be part of
    // the associated message_template.

```

```

request.getMessageFormat().add("html");

// Add personalization token and values
NameValuePair userNameToken = new NameValuePair();
    userNameToken.setName("user_name");
    userNameToken.setValue("John");
NameValuePair yearToken = new NameValuePair();
    yearToken.setName("yrs");
    yearToken.setValue("5");
request.getPersonalizationTokens().add(userNameToken);
request.getPersonalizationTokens().add(yearToken);

// If you are using multiple MIME types, set the UseMultiPart setting to "true":
request.setUseMultiPart(false);

PersonalizeResponse response = this.service.personalize(request);
// Prints the personalized message to console
System.out.println(response.getPersonalizedMessage());
return response.getPersonalize
}

```

PersonalizeTemplateRequest Sample Code

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
public String personalizeTemplate(TemplateId templateId) throws InvalidObjectFault,
UnexpectedFault, ObjectNotFoundFault, AuthorizationFault
Supported requests{
    PersonalizeTemplateRequest request = new PersonalizeTemplateRequest();
    // Specify the Template ID

```

```

request.setTemplateId(templateId);

// Specify MIME type (html/text/mime) using getMessageFormat request
// NOTE: You may use multiple getMessageFormat().add() statements
// to specify multiple MIME types, but they must be part of the message_template.
request.getMessageFormat().add("html");

// Set personalize token and values
NameValuePair userNameToken = new NameValuePair();
userNameToken.setName("user_name");
userNameToken.setValue("John");
NameValuePair yearToken = new NameValuePair();
yearToken.setName("yrs");
yearToken.setValue("5");

// request.getPersonalizationTokens() returns a List Object and
// we use it's add() method to specify the value(s).
request.getPersonalizationTokens().add(userNameToken);
request.getPersonalizationTokens().add(yearToken);

// If you are using multiple MIME types, set the UseMultiPart value to "true":
request.setUseMultiPart(false);
PersonalizeResponse response = service.personalize(request);

// Prints the personalize message to console
System.out.println(response.getPersonalizedMessage());
return response.getPersonalizedMessage();
}

```

reactivateGlobalUndeliverable()

Reactivates a mailing recipient (i.e., removes them from the the GlobalUndeliverable table).

Supported Assets

The following list shows the Message Studio assets that are supported for the `reactivateGlobalUndeliverable()` operation:

- GlobalUndeliverable

Recommendations

Before reactivating the mailing recipient, make sure *global.undeliverable.status = true*, or else a validation exception will be thrown.

Code Sequence

```
1. reactivateGlobalUndeliverable
   (reactivateGlobalUndeliverableRequest)
```

Supported Requests

The `reactivateGlobalUndeliverable()` operation supports a single object: `reactivateGlobalUndeliverableRequest`.

Response

`ReactivateGlobalUndeliverableResponse`—Contains summary for reactivation.

Faults

Refer to [Error Handling on page 35](#) for a description of the following faults that may be returned:

- `objectNotFoundFault`
- `authorizationFault`
- `invalidObjectFault`

Sample Code

```
/*
 * -----
 * Selligent, Inc.
 *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved.
 *
 * Visit http://www.strongview.com for more information.
 *
 * You may incorporate this Source Code in your application only
 * if you own a valid license from Selligent, Inc.
 * and the copyright notices are not removed from the source code.
 */
```

```

*                                                                 *
* Distributing this source code outside your organization        *
* is strictly prohibited.                                         *
*                                                                 *
* ----- *
*/

public void reactivateGlobalUndeliverable(String recipientAddress, String channelType,
String orgId) throws Exception {
    System.out.println("Start of reactivateGlobalUndeliverables");
    ReactivateGlobalUndeliverableRequest reactivateGlobalUndeliverableRequest = new
    ReactivateGlobalUndeliverableRequest();
    List<GlobalUndeliverableInfo> globalUndeliverableInfoList = new
    ArrayList<GlobalUndeliverableInfo>();
    GlobalUndeliverableInfo globalUndeliverableInfo = new GlobalUndeliverableInfo();

    globalUndeliverableInfo.setRecipientAddress(recipientAddress);
    globalUndeliverableInfo.setChannelType(channelType);

    // Ignore OrgId if config value set as across organization
    // OrgId mandatory if config value set as per organization

    OrganizationId organizationId = new OrganizationId();
    organizationId.setId(orgId);
    reactivateGlobalUndeliverableRequest.setOrganizationId(organizationId);
    globalUndeliverableInfoList.add(globalUndeliverableInfo);
    reactivateGlobalUndeliverableRequest.getGlobalUndeliverableInfo().addAll
    (globalUndeliverableInfoList);
    ReactivateGlobalUndeliverableResponse response =
    service.reactivateGlobalUndeliverable(reactivateGlobalUndeliverableRequest);
    System.out.println("End of reactivateGlobalUndeliverable");
}

```

refreshRecords ()

This operation refreshes the local copy of an external data source.

Supported Assets

The following list shows the Message Studio assets that are supported for the refreshRecords () operation:

- External Data Source (externalDataSourceId)

Recommendations

None.

Code Sequence

1. Get the asset ID using **list(filter)** API operation.
2. `refreshRecords(request)`

Supported Requests

The `refreshRecords()` operation supports a single object that can be passed in as an argument. The object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

The following table maps the functional area to the object/request.

Functional area	Object/Request
External data source	externalDataSourceId/RefreshRecordsRequest

Response

`refreshRecordsResponse`

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- `objectNotFoundFault`
- `authorizationFault`
- `unexpectedFault`
- `concurrentModificationFault`

Sample Code

```
/*
 * -----
 * Selligent, Inc.
 *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved.
```

```

*
* Visit http://www.strongview.com for more information.
*
* You may incorporate this Source Code in your application only
* if you own a valid license from Selligent, Inc.
* and the copyright notices are not removed from the source code.
*
* Distributing this source code outside your organization
* is strictly prohibited.
*
* -----
*/

/**
 * Refresh External Data Source records.
 *
 * @param edsId the EIS ID to be refreshed.
 * @return if refresh is successful
 * @throws Exception any exception
 */

public boolean refreshRecords(ExternalDataSourceId edsId) throws Exception {
    RefreshRecordsRequest request = new RefreshRecordsRequest();
    request.setDataSourceId(edsId);
    return service.refreshRecords(request).isSuccess();
}

```

removeRecords ()

Use the `removeRecords()` operation to remove recipient data from a data source, seed list, or suppression list.

Supported Assets

The following list shows the Message Studio assets that are supported for the `removeRecords()` operation:

- Internal data source (internalDataSourceId)
- External data source (externalDataSourceId)
- Seed List (seedListId)
- Extension - Standard (extensionId)
- Extension - Catalog (catalogId)
- Suppression List (suppressionListId)

Recommendations

In most cases, you should delete records from your external data source directly from your database and not through the API. Selligent does not recommend deleting records from your external data source through the API or the UI.

The maximum number of records that can be specified inline in a `removeRecords` request is not unlimited. The latest version 3.1.7 of Apache CXF has introduced a limit of 50,000 child nodes (default) to protect against various DoS (Denial of Service) attacks.

Code Sequence

1. Get desired asset ID using `list (filter)` API operation.
2. `removeRecords (request)`

Supported Requests

The `removeRecords()` operation supports multiple objects that can be passed in as arguments. Each object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

The following table maps the functional area to the object/request.

Functional area	Object/Request
Internal data source	<code>RemoveDataSourceRecordsRequest</code>
External data source	<code>RemoveDataSourceRecordsRequest</code>
Seed list	<code>RemoveSeedListRecordsRequest</code>
Suppression list	<code>RemoveSuppressionListRequest</code>

Response

`removeRecordsResponse`

Faults

Refer to [Error Handling on page 35](#) for a description of the following faults that may be returned:

- `objectNotFoundFault`
- `authorizationFault`

- unrecognizedObjectTypeFault
- unexpectedFault
- concurrentModificationFault

Sample Code

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */

/**
 * Remove one record that was created in addRecordsInline() from the SeedList.
 *
 * @param seedListId the SeedListId whose record will be removed.
 * @return the number of record that was removed.
 * @throws Exception any exception
 */

public int removeRecordsInline(SeedListId seedListId) throws Exception {
    RemoveSeedListRecordsRequest request = new RemoveSeedListRecordsRequest();
    request.setSeedListId(seedListId);
    request.getAddress().add("seed1@seed.cxf.com");
    return service.removeRecords(request).getRecordsRemoved();
}

```

resume ()

Use the `resume ()` operation to resume delivery of a paused mailing. If you made modifications to the mailing content or headers, the modifications are used when the mailing is resumed.

Supported Assets

The following list shows the Message Studio assets that are supported for the `resume ()` operation:

- Batch Mailing (`standardMailingId`)
- Transactional Mailing (`transactionalMailingId`)
- Channel Standard Mailing (`channelStandardMailingId`)
- Channel Transactional Mailing (`channelTransactionalMailingId`)

Recommendations

None.

Code Sequence

1. Get the desired asset ID using **list(filter)** API operation.
2. `resume (request)`

Supported Requests

The `resume ()` operation supports multiple objects that can be passed in as arguments. Each object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

The following table maps the functional area to the object/request.

Functional area	Object/Request
Batch mailing	ResumeRequest
Transactional mailing	ResumeRequest
Channel Standard mailing	ResumeRequest
Channel Transactional mailing	ResumeRequest

Response

resumeResponse

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- objectNotFoundFault
- authorizationFault
- unexpectedFault

Sample Code

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */

/**
 * Resume a standard or transactional mailing. *
 * @param mailingId the ID of Mailing that will be closed
 * @return if close operation is successful
 * @throws Exception any exception */

public boolean resume(MailingId mailingId) throws Exception {
    ResumeRequest request = new ResumeRequest();
    request.setMailingId(mailingId);
    return service.resume(request).isSuccess();
}

```

```
}
```

schedule()

This operation schedules a mailing to launch at a specific interval.

Supported Assets

The following list shows the Message Studio assets that are supported for the `schedule()` operation:

- Batch Mailing (`standardMailingId`)
- Channel Standard Mailing (`channelStandardMailingId`)

Recommendations

None.

Code Sequence

1. `create(mailing)`—Set the schedule in the mailing while saving it.
2. `schedule(request)`

Supported Requests

The `schedule()` operation supports a single object that can be passed in as an argument. The object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

The following table maps the functional area to the object/request.

Functional area	Object/Request
Batch mailing	<code>schedule()</code>
Channel Standard mailing	<code>schedule()</code>

Response

`ScheduleResponse`

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- authorizationFaultDetail
- invalidObjectFaultDetail
- objectNotFoundFaultDetail
- unexpectedFaultDetail
- unrecognizedObjectTypeFaultDetail

Sample Code

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */

mailing.setType(MailingType.RECURRING);

/* Set schedule: daily recurrence with every two days */

SchedulableMailing.Schedule.Recurrence.DailyRecurrence daily = new
SchedulableMailing.Schedule.Recurrence.DailyRecurrence();
daily.setInterval(2);
SchedulableMailing.Schedule.Recurrence recurrence = new
SchedulableMailing.Schedule.Recurrence();
recurrence.setEndAfterXMailings(100); /* launch 100 mailing */
recurrence.setDailyRecurrence(daily);

```



```

SchedulableMailing.Schedule schedule = new SchedulableMailing.Schedule();
schedule.setRecurrence(recurrence); /* one hour later */
schedule.setStartDateTime(new Date(System.currentTimeMillis() + 3600000));
mailing.setSchedule(schedule);
...
/* Now call the create mailing API to save the mailing and collect the */
/* mailingId from the create API.*/
/* Finally, schedule the mailing using its mailing id.*/
ScheduleRequest request = new ScheduleRequest();
request.setMailingId(mailingId);
service.schedule(request);

```

send()

Important: The `send()` operation is deprecated. Use `txnsend()` call to trigger the sending of messages for a transactional mailing. The `txnsend()` operation provides better send performance.

startImport()

Returns the status (true or false) of the import operation of the given `RecurringImportId`.

Support Assets

- `StartRecurringImportRequest`

Code Sequence

1. Set the following properties:
 - `RecurringImportId`—Can be obtained using the `findImport` API
 - `fullRefresh`—Boolean
2. `findImport(findImportRequest);`
3. `startImport(startReccuringImportRequest);`

Supported Requests

The `startImport()` function supports a the following object/requests:

Functional area	Object/Request
Data Source	RecurringImportId/StartRecurringImportRequest
Channel Transactional Mailing	ChannelTransactionalMailingId/GetTxnEasInfoRequest

Response

StartRecurringImportReponse—Returns a boolean flag (true/false) indicating the success of the operation.

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- objectNotFoundFault
- authorizationFault
- unexpectedFault

Sample code

```

/*
 * -----
 * Selligent, Inc.
 *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved.
 *
 * Visit http://www.strongview.com for more information.
 *
 * You may incorporate this Source Code in your application only
 * if you own a valid license from Selligent, Inc.
 * and the copyright notices are not removed from the source code.
 *
 * Distributing this source code outside your organization
 * is strictly prohibited.
 * -----
 */

FindRecurringImportResponse recurringImportResponse = service.findImport
(findRecurringImportRequest);
RecurringImportId recurringImportId = recurringImportResponse.getImportId().get(0);
StartRecurringImportRequest startImportRequest = new;
StartRecurringImportRequest();
startImportRequest.setImportId(recurringImportId);

```

```
startImportRequest.setFullRefresh(true);
StartRecurringImportResponse startImportResponse = svc.startImport(startImportRequest);
System.out.println("Import operation status "+ startImportResponse.isSuccess());
```

test()

This operation tests the dynamic content within a template or mailing by sending a test mailing.

Supported Assets

The following list shows the Message Studio assets that are supported for the `test()` operation:

- Batch Mailing (`standardMailingId`)
- Channel Standard Mailing (`channelStandardMailingId`)
- Template (`templateId`)
- Channel Template (`channelTemplateId`)

Recommendations

None.

Code Sequence

1. Get the desired asset ID using **list(filter)** API operation.
2. `test(request)`

Supported Requests

The `test()` operation supports multiple objects that can be passed in as arguments. Each object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

The following table maps the functional area to the object/request.

Functional area	Object/Request
Template	(<code>TemplateId</code> , <code>ChannelTemplateId/TestTemplateRequest</code>)
Batch mailing	(<code>StandardMailingId</code> , <code>ChannelStandardMailingId/TestMailingRequest</code>)

Response

testResponse

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- objectNotFoundFault
- authorizationFault
- unrecognizedObjectTypeFault
- unexpectedFault
- invalidObjectFault

Sample Code

```
/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */

/**
 * Test template for it dynamic content.
 * Email will be sent and tokens in the template will be replaced with values.
 *
 * @param templateId the TemplateId that will be sent out.
 * @return if the test request is sent successfully.
 * @throws Exception any exception
 */
```

```

*/

public boolean testTemplate(TemplateId templateId) throws Exception {
    TestTemplateRequest request = new TestTemplateRequest();
    request.setSubjectPrefix("Subject Prefix: ");
    request.setTemplateId(templateId);
    request.setTestListId(null);
    request.setUseMultiPart(false);
    request.getAddress().add("test_address@testtemplate.com");
    request.getFormat().add(MessageFormat.HTML);

    /* Make sure you have these two tokens in your template */
    NameValuePair pair1 = new NameValuePair(); pair1.setName("EMAIL");
    pair1.setValue("testemail@testtemplate.com");
    request.getTokenValue().add(pair1);
    NameValuePair pair2 = new NameValuePair();
    pair2.setName("NAME");
    pair2.setValue("testemail name");
    request.getTokenValue().add(pair2);
    return service.test(request).isSuccess();
}

```

txnSend()

There are two methods that can be used for sending transactional messages:

- `txnSend()` —Use this operation for moderate volume/frequency transactional mailing applications. The frequency of `txnSend()` requests should not exceed 500,000 per hour.
- `send()` —Use this API for very high volume or high frequency applications after you create the transactional mailings through the Message Studio API operation **`create()`** on page 64.

Note: `send()` —This method is still supported but has been deprecated due to performance constraints.

Supported Assets

The following list shows the Message Studio assets that are supported for the `txnSend()` operation:

- Transactional Mailing (`transactionalMailingId`)

- Channel Transactional Mailing (`channelTransactionalMailingId`)

Recommendations

For Message Studio 4.x or earlier, the maximum supported size of an API call is 64 KB, including any database records and message content.

In version 5.0 or greater, the maximum recommended size is 256 KB. Usually this is not a problem, since most transactional API calls will fall below this level.

Code Sequence

1. Get the desired asset ID using **list(filter)** API operation.
2. `txnSend(request)`

Supported Requests

The `txnSend()` operation supports a single object that can be passed in as an argument. The object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

The following table maps the functional area to the object/request.

Functional area	Object/Request
Transactional mailing	<code>transactionalMailingId/TxnSendRequest</code>
Channel Transactional mailing	<code>channelTransactionalMailingId/TxnSendRequest</code>

Response

`txnSendResponse`—Returns boolean flag (true/false) indicating success.

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

`BadHandleFault`

Sample Code

The following method uses name/value pairs, which take a string that contains all data to send the transactional email.

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
public boolean txnSend(TransactionalMailingId mailingId) {

    // Get transaction mailing handle
    GetTxnMailingHandleRequest handleRequest = new
        GetTxnMailingHandleRequest();
    handleRequest.setMailingId(mailingId);
    String mailingHandle = service.getTxnMailingHandle(handleRequest)
        .getHandle();
    List<SendRecord> sendRecords = new ArrayList<SendRecord>();

    SendRecord record1 = new SendRecord();
    NameValuePair emailCol;
    emailCol = new NameValuePair();
    emailCol.setName("email_address");
    emailCol.setValue("email_test@test.com");
    record1.getField().add(emailCol);
    emailCol = new NameValuePair();
    emailCol.setName("description");
    emailCol.setValue("description" + i);
    record1.getField().add(emailCol);

    sendRecords.add(record1);

    TxnSendRequest txnSendRequest = new TxnSendRequest();
    txnSendRequest.setHandle(mailingHandle);
    txnSendRequest.getSendRecord().addAll(sendRecords);
    TxnSendResponse txnSendResponse = service.txnSend(txnSendRequest);

    return txnSendResponse.isSuccess();
}

```

update ()

Use the `update ()` operation to edit and update the content of the specified asset. For example, you could do any of the following (and more):

- Rename an asset
- Change a target's query
- Modify a content block

Supported Assets

The following list shows the Message Studio assets that are supported for the `update ()` operation:

- Internal Data Source (`internalDataSourceId`)
- External Data Source (`externalDataSourceId`)
- Target (`targetId`)
- Seed list (`seedListId`)
- Suppression list (`suppressionListId`)
- Template (`templateId`)
- (Assigned) Role (`assignedRoleId`)
- Role (`roleId`)
- Media Server (`mediaServerId`)
- Media Server Group (`mediaServerGroupId`)
- Program (`programId`)
- Extension - Catalog (`catalogId`)
- Extension - Standard (`extensionId`)
- Content block (`contentBlockId`)
- Attachment (`attachmentId`)
- Dynamic content (`ruleId`)
- Batch mailing (`standardMailingId`)
- Transactional mailing (`transactionalmailingId`)
- Tracking tags
- Organization (`organizationId`)
- System Address (`systemAddressId`)
- User (`userId`)
- StrongTools
- Channel Template (`channelTemplateId`)
- Channel Standard mailing (`channelStandardMailingId`)
- Channel Transactional mailings (`channelTransactionalMailingId`)

Recommendations

- Updating a mailing or template to `isApproved=true` triggers validation checks. All objects must have all necessary attributes for launching or the server will return a fault.
- The following table lists the read-only fields that cannot be updated:

Asset	Read-only fields
Data source	objectId, createTime, operationStatus, modifiedTime (from BaseObject), version (from BaseObject), organizationId, ownerId.
Target	objectId, totalRecords, retargetingDataSourceId, createTime, modifiedTime (from BaseObject), version (from BaseObject), organizationId, ownerId.
Seed list	objectId, totalRecords, createTime, modifiedTime (from BaseObject), version (from BaseObject), organizationId, ownerId
Suppression list	objectId, totalRecords, createTime, modifiedTime (from BaseObject), version (from BaseObject), organizationId, ownerId
Template, ChannelTemplate	objectId, createTime, modifiedTime (from BaseObject), version (from BaseObject), organizationId, ownerId, channelId (for Channel mailings)
Content block	objectId, size, createTime, modifiedTime BaseObject), version (from BaseObject), organizationId, ownerId
Attachment	objectId, size, createTime, modifiedTime BaseObject), version (from BaseObject), organizationId, ownerId
Rule	objectId, createTime, modifiedTime (from BaseObject), version (from BaseObject), organizationId, ownerId
Batch, Channel, and Transactional mailings	objectId, createTime, modifiedTime (from BaseObject), version (from BaseObject), organizationId, ownerId, format, parentId, lastGoodStatus, serverErrorCode, serverErrorMessage, status, channelId (for Channel mailings)

Code Sequence

- Get the desired asset ID using **list(filter)** API operation.
- get** (*object_ids*)
- update** (*array_of_objects*)

Supported Requests

The `update()` operation supports multiple objects that can be passed in as arguments. Each object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

Note: You must pass in an array of objects for the `update()` operation. The array of objects may be defined as a single asset or as multiple assets to be updated.

The following table maps the functional area to the object/request.

Functional area	Object/Request	Specific Base Object
External data source	updateRequest	ExternalDataSource
Internal data source	updateRequest	InternalDataSource
Target	updateRequest	Target
Seed list	updateRequest	SeedList
Suppression list	updateRequest	SuppressList
Template	updateRequest	Template, ChannelTemplate
Content blocks	updateRequest	ContentBlock
Attachment	updateRequest	Attachment
Dynamic Content (Rule)	updateRequest	Rule
Batch mailing	updateRequest	StandardMailing, ChannelStandardMailing
Transactional mailing	updateRequest	TransactionalMailing, ChannelTransactionalMailing

Response

BatchUpdateResponse

The `BatchUpdateResponse` will include one or more `UpdateResponses` depending on the number of base objects you passed into the operation.

Faults

This operation does not throw a fault. Instead, the `FaultDetail` is wrapped in the response. Any faults will be returned in the `BatchUpdateResponse`. You will need to check each individual `UpdateResponse` for the exception.

Refer to [Error Handling on page 35](#) for a description of the following faults that may be returned:

- authorizationFaultDetail
- invalidObjectFaultDetail
- objectNotFoundFaultDetail
- staleObjectFaultDetail
- unexpectedFaultDetail
- unrecognizedObjectTypeFaultDetail

Tracking Tags

Support is provided for the following tracking tags:

- None
- OPENTAG
- ENOPENTAG
- CLICKTAG
- SHORTCLICKTAG
- ENCLICKTAG
- UNSUBSCRIBETAG
- SHORTUNSUBSCRIBETAG | ENUNSUBSCRIBETAG
- FTAFTAG
- ENFTAFTAG
- SHARETAG
- ENSHARETAG
- VIEWINBROWSERTAG
- ENVIEWINBROWSERTAG

Sample Code

```
/*
 * ----- *
 * Selligent, Inc. *
 * *
 * *
```

```

* Copyright © 2016 Selligent, Inc.. - All rights reserved.      *
*                                                                *
* Visit http://www.strongview.com for more information.        *
*                                                                *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc.              *
* and the copyright notices are not removed from the source code.*
*                                                                *
* Distributing this source code outside your organization       *
* is strictly prohibited.                                       *
*                                                                *
* ----- *
*/

/**
 * Update an data source with new name.
 *
 * @param dataSourceId the data source ID whose name will be updated
 * @param newName the new name for the data source
 * @throws Exception any exception
 */
public void updateDataSourceName(DataSourceId dataSourceId, String newName) throws
Exception {

    /* Get the data source first */
    List<ObjectId> dataSourceIds = new ArrayList<ObjectId>(1);
    dataSourceIds.add(dataSourceId);
    DataSource dataSource = (DataSource) helper.get(dataSourceIds).get(0);

    /* Change name and update */
    dataSource.setName(newName);
    helper.update(dataSource);
}

```

upsertRecord()

This operation executes a single record add or update in one step.

Important: This API operation is only supported for Internal Data Sources.

Supported Assets

The following list shows the Message Studio assets that are supported for the `upsertRecord()` operation:

- Internal data sources (`internalDataSourceId`)
- Extension - Standard (`extensionId`)
- Extension - Catalog (`catalogId`)

Recommendations

None.

Code Sequence

1. Get the desired asset ID using **list(filter)** API operation.
2. `copy(request)`

Faults

Refer to **Error Handling on page 35** for a description of any faults that may be returned.

The API does not allow the following scenarios and throws an `InvalidObjectFault`:

- No `dataSourceRecord` specified in the request.
- More than one `dataSourceRecord` specified in the request.
- A non-existent column in specified in the input record.
- A system column in specified in the input record.
- A column is specified more than one in the input record.

Sample Code

```
/*
 * -----
 * Selligent, Inc.
 *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved.
```

```

*                                                                    *
* Visit http://www.strongview.com for more information.    *
*                                                                    *
* You may incorporate this Source Code in your application only      *
* if you own a valid license from Selligent, Inc.                   *
* and the copyright notices are not removed from the source code.   *
*                                                                    *
* Distributing this source code outside your organization           *
* is strictly prohibited.                                           *
*                                                                    *
* ----- *
*/

public void upsertRecord(InternalDataSourceId dataSourceId) throws Exception
{
    addSampleRecordForUpsert(dataSourceId);

    /* Wait for data source status to be idle before adding */
    helper.waitForIdle(dataSourceId);

    /* A non-existent record, which will get added. */
    UpsertDataSourceRecordsRequest upsertRequest1 = new UpsertDataSourceRecordsRequest();
    upsertRequest1.setDataSourceId(dataSourceId);
    DataSourceRecord newRecord = new DataSourceRecord();
    NameValuePair idCol = new NameValuePair();
    idCol.setName("id");
    idCol.setValue("903");
    newRecord.getField().add(idCol);
    NameValuePair emailCol = new NameValuePair();
    emailCol.setName("email_address");
    emailCol.setValue("email_for_cxf_upsert3@cxf.com");
    newRecord.getField().add(emailCol);
    NameValuePair nameCol = new NameValuePair();
    nameCol.setName("name");
    nameCol.setValue("cxf_name_for_upsert3");
    newRecord.getField().add(nameCol);
    upsertRequest1.getDataSourceRecord().add(newRecord);
    service.upsertRecord(upsertRequest1);

    /* Wait for data source status to be idle before adding */
    helper.waitForIdle(dataSourceId);

    /* An existing record, which will get updated. */
    UpsertDataSourceRecordsRequest upsertRequest2 = new UpsertDataSourceRecordsRequest();
    upsertRequest2.setDataSourceId(dataSourceId);

```

```

DataSourceRecord updateRecord = new DataSourceRecord();
idCol = new NameValuePair();
idCol.setName("id");
idCol.setValue("901");
updateRecord.getField().add(idCol);
emailCol = new NameValuePair();
emailCol.setName("email_address");
emailCol.setValue("email_for_cxf_upsert-updat- ed@cxfr.com");
updateRecord.getField().add(emailCol);
upsertRequest2.getDataSourceRecord().add(updateRecord);
service.upsertRecord(upsertRequest2);
}

```

Request Sample Code

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */

<xs:complexType name="UpsertDataSourceRecordsRequest">
  <xs:complexContent>
    <xs:extension base="tns:UpsertRecordsRequest">
      <xs:sequence>
        <xs:element name="dataSourceId" type="tns- :DataSourceId"/>
        <xs:element name="dataSourceRecord" type="tns- :DataSourceRecord"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

Response Sample Code

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * ----- *
 */
<xs:complexType name="UpsertRecordsResponse">
  <xs:complexContent>
    <xs:extension base="tns:Response">
      <xs:sequence>
        <xs:element name="success" type="xs:boolean"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

validateXsl()

Use the `validateXsl()` operation to validate the XSL in an XSL template.

Supported Assets

The following list shows the Message Studio assets that are supported for the `validateXsl()` operation:

- Template (`templateId`)
- Channel Template (`channelTemplateId`)

Recommendations

None.

Code Sequence

1. Get the desired asset ID using **list(filter)** API operation.
2. `validateXsl(request)`

Supported Requests

The `validateXsl()` operation supports a single object that can be passed in as an argument. The object is associated with a functional area of the Message Studio UI, and is also mapped to a specific SOAP request.

The following table maps the functional area to the object/request.

Functional area	Object/Request
Template	templateId/ValidateXslRequest
Channel Template	channelTemplateId/ValidateXslRequest

Response

`validateXslResponse`

Faults

Refer to **Error Handling on page 35** for a description of the following faults that may be returned:

- `objectNotFoundFault`
- `authorizationFault`
- `unexpectedFault`

Sample Code

```
/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
```

```

*                                                                 *
* Visit http://www.strongview.com for more information.          *
*                                                                 *
* You may incorporate this Source Code in your application only   *
* if you own a valid license from Selligent, Inc.                *
* and the copyright notices are not removed from the source code. *
*                                                                 *
* Distributing this source code outside your organization         *
* is strictly prohibited.                                         *
*                                                                 *
* ----- *
*/
/**
 * Validate if a template's xsl content is valid.
 *
 * @param templateId the TemplateId whose XSL content will be validated.
 * @return if the content is valid XSL
 * @throws Exception any exception
 */

public boolean validateXsl(TemplateId templateId) throws Exception {
    ValidateXslRequest request = new ValidateXslRequest();
    request.setMessageFormat(MessageFormat.HTML);
    request.setTemplateId(templateId);
    return service.validateXsl(request).isValid();
}

```

Sample Java Code

```

/*
* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *
* *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* *
* Distributing this source code outside your organization *
* is strictly prohibited. *

```

```

*
* ----- *
*/

package com.strongmail.webserviceclient.samples
import com.strongmail.webserviceclient.samples.generated.MailingService;
import com.strongmail.webserviceclient.samples.generated.MailingService_Service;
import com.strongmail.webserviceclient.samples.generated.objects.*;

import org.apache.cxf.binding.soap.SoapMessage;
import org.apache.cxf.configuration.jsse.TLSClientParameters;
import org.apache.cxf.endpoint.Client;
import org.apache.cxf.endpoint.ClientImpl;
import org.apache.cxf.endpoint.Endpoint;
import org.apache.cxf.frontend.ClientProxy;
import org.apache.cxf.interceptor.Fault;
import org.apache.cxf.phase.Phase;
import org.apache.cxf.phase.PhaseInterceptor;
import org.apache.cxf.transport.http.HTTPConduit;
import org.apache.cxf.ws.security.wss4j.WSS4JOutInterceptor;
import org.apache.ws.security.SOAPConstants;
import org.apache.ws.security.handler.WSHandlerConstants;
import org.apache.ws.security.util.WSSecurityUtil;
import org.apache.ws.security.WSConstants;
import org.apache.ws.security.WSPasswordCallback;

import java.util.Scanner;
import java.util.List;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Map;
import java.util.HashMap;
import java.util.Set;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.URL;
import java.security.KeyStore;
import java.security.cert.X509Certificate;

import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import javax.net.ssl.X509TrustManager;

```

```

import javax.security.auth.callback.Callback;
import javax.security.auth.callback.CallbackHandler;
import javax.security.auth.callback.UnsupportedCallbackException;
import javax.xml.namespace.QName;
import javax.xml.ws.Binding;
import javax.xml.ws.BindingProvider;
import javax.xml.ws.soap.SOAPBinding;
import javax.xml.soap.SOAPException;
import javax.xml.soap.SOAPMessage;
import com.sun.xml.ws.developer.SchemaValidationFeature;
import org.w3c.dom.Element;
import org.w3c.dom.Text;

public class SampleProgram
{
    public static void main(String[] args) throws Exception
    {

        /* Organization, username, password */
        String organization = "admin"; /* Use "/" to specify suborganization. E.g.
admin/subadmin/onemore */
        String username = "admin";
        String password = "admin";

        /* Initialize the service object with the org, username and password details. */
        SampleProgram sample = new SampleProgram();

        try
        {
            MailingService svc = sample.createService(organization, username, password);

            /* Sample for listing users and roles. */
            describeUserListSample(svc);

            /* Sample for listing templates. */
            describeTemplateListSample(svc);

            /* Sample for listing mailings. */
            describeMailingListSample(svc);

            /* Sample for creating a template. */
            describeTemplateCreateSample(svc);
        }
        catch (Exception e)
        {

```

```

        System.out.println("Encountered exception: " + e.getMessage());
        throw e;
    }
}

private MailingService createService(String organization,
                                     String username,
                                     String password)
    throws Exception
{
    URL wsdlURL = new URL("https://hostname:4443/sm/services/mailing/2009/03/02?wsdl");

    QName SERVICE_QNAME = new QName(SERVICE_NAMESPACE, SERVICE_NAME);

    /* Create service with schema validation on */
    MailingService_Service mss = new MailingService_Service(wsdlURL, SERVICE_QNAME);
    SchemaValidationFeature feature = new SchemaValidationFeature();
    MailingService svc = mss.getMailingServicePort(feature);

    /* Set endpoint */
    Endpoint endpoint = ClientProxy.getClient(svc).getEndpoint();
    String endpointAddress = endpoint.getEndpointInfo().getAddress();

    ((BindingProvider)svc).getRequestContext().put(BindingProvider.ENDPOINT_ADDRESS_
PROPERTY,
                                                    endpointAddress);

    /* By default do not use MTOM since there is a bug in JAXB library. */
    /* See: https://jaxb.dev.java.net/issues/show_bug.cgi?id=588 */
    Binding binding = ((BindingProvider)svc).getBinding();
    ((SOAPBinding)binding).setMTOMEnabled(false);

    /* Next, create the security headers for authentication. */
    /* WSSE standard username token */
    Map<String, Object> outprops = new HashMap<String, Object>();
    outprops.put(WSHandlerConstants.ACTION, WSHandlerConstants.USERNAME_TOKEN);

    /* Set username and password */
    outprops.put(WSHandlerConstants.USER, username);
    outprops.put(WSHandlerConstants.PASSWORD_TYPE, WSConstants.PW_TEXT);
    outprops.put(WSHandlerConstants.PW_CALLBACK_REF, new ClientPasswordCallbackHandler
(password));

    /* Create client authentication interceptor */
    AuthenticationInterceptor authenticationInterceptor =

```

```

        new AuthenticationInterceptor(SERVICE_NAMESPACE_SCHEMA, outprops, organization);
authenticationInterceptor.setAllowMTOM(false);

/* Set client to use HTTPS - specify certificate and keystore password (read as JVM
arguments.). */
ClientImpl client = (ClientImpl)ClientProxy.getClient(svc);
String keystore = System.getProperty("javax.net.ssl.trustStore");
String keystorePassword = System.getProperty("javax.net.ssl.trustStorePassword");
KeyStore ks = KeyStore.getInstance("JKS");
ks.load(new FileInputStream(keystore), keystorePassword.toCharArray());
TrustManagerFactory tmf = TrustManagerFactory.getInstance(
    TrustManagerFactory.getDefaultAlgorithm());
tmf.init(ks);
TrustManager[] tms = tmf.getTrustManagers();
TLSClientParameters param = new TLSClientParameters();
param.setSecureSocketProtocol("SSL");
param.setTrustManagers(tms);
HTTPConduit https = (HTTPConduit)client.getConduit();
https.setTlsClientParameters(param);

/* Set authentication interceptor into the endpoint in the service object. */
Endpoint clientEndpoint = client.getEndpoint();
clientEndpoint.getOutInterceptors().add(authenticationInterceptor);

return svc;
}

private static void describeUserListSample(MailingService svc) throws Exception
{

/* List users using the list API with the UserFilter. */
ListRequest userListRequest = new ListRequest();
UserFilter userFilter = new UserFilter();
userFilter.setIsAscending(true);
userFilter.setPageNumber(0);
userFilter.setRecordsPerPage(10);
userFilter.setMaxRecordsPerPage(200);
userListRequest.setFilter(userFilter);
ListResponse userListResponse = svc.list(userListRequest);

/* Get details about each user using the get API call. */
List<User> users = new ArrayList<User>();
for (ObjectId id : userListResponse.getObjectId())
{
    UserId userId = (UserId)id;

```

```

        GetRequest getRequest = new GetRequest();
        getRequest.getObjectId().add(userId);
        User user = (User) (svc.get(getRequest).getGetResponse().get(0).getBaseObject());
        users.add(user);
    }

    /* Display all users */
    System.out.println("");
    System.out.println("-----");
    System.out.println("Users List");
    for (User user : users)
    {
        String roles = "";
        if (user.isIsSuperUser())
        {
            /* User is a super user. */
            roles = "Super User";
        }
        else if (user.isIsAdmin())
        {
            /* User is an admin. */
            roles = "Admin User";
        }
        else
        {
            /* Get the roles assigned to the user from the "access" field. */
            for (User.Access accessInfo : user.getAccess())
            {
                RoleId roleId = accessInfo.getRoleId();
                /* Get details about the role (e.g. its name) using the get API call. */
                GetRequest getRequest = new GetRequest();
                getRequest.getObjectId().add(roleId);
                Role role = (Role) (svc.get(getRequest).getGetResponse().get(0).getBaseObject
            ());
                roles += role.getName() + " ";
            }
            if (roles == "")
            {
                roles = "<None>";
            }
        }
    }

    System.out.println(user.getFirstName() + " " + user.getLastName() + ". Role(s): " +
roles);

```

```

    }
}

private static void describeTemplateListSample(MailingService svc) throws Exception
{
    /* List templates using the list API with the UserFilter. */
    ListRequest templateListRequest = new ListRequest();
    TemplateFilter templateFilter = new TemplateFilter();
    templateFilter.setIsAscending(true);
    templateFilter.setPageNumber(0);
    templateFilter.setRecordsPerPage(10);
    templateFilter.setMaxRecordsPerPage(200);
    templateListRequest.setFilter(templateFilter);
    ListResponse templateListResponse = svc.list(templateListRequest);
    /* Get details about each template using the get API call. */
    List<Template> templates = new ArrayList<Template>();
    for (ObjectId id : templateListResponse.getObjectId())
    {
        TemplateId templateId = (TemplateId)id;
        GetRequest getRequest = new GetRequest();
        getRequest.getObjectId().add(templateId);
        Template template = (Template)(svc.get(getRequest).getGetResponse().get
(0).getBaseObject());
        templates.add(template);
    }
    /* Display all templates */
    System.out.println("");
    System.out.println("-----");
    System.out.println("Templates List");
    for (Template template : templates)
    {
        System.out.println(template.getName());
    }
}

private static void describeMailingListSample(MailingService svc) throws Exception
{
    /* List batch mailings using the list API with the MailingFilter. */
    ListRequest mailingListRequest = new ListRequest();
    MailingFilter mailingFilter = new MailingFilter();
    mailingFilter.setIsAscending(true);
    mailingFilter.setPageNumber(0);
    mailingFilter.setRecordsPerPage(10);
    mailingFilter.setMaxRecordsPerPage(200);
    mailingListRequest.setFilter(mailingFilter);

```



```

ListResponse mailingListResponse = svc.list(mailingListRequest);
/* Get details about each mailing using the get API call. */
List<Mailing> mailings = new ArrayList<Mailing>();
for (ObjectId id : mailingListResponse.getObjectId())
{
    MailingId mailingId = (MailingId)id;
    GetRequest getRequest = new GetRequest();
    getRequest.getObjectId().add(mailingId);
    Mailing mailing = (Mailing)(svc.get(getRequest).getGetResponse().get
(0).getBaseObject());
    mailings.add(mailing);
}
/* Display all batch mailings */
System.out.println("");
System.out.println("-----");
System.out.println("Mailings List");
for (Mailing mailing : mailings)
{
    System.out.println(mailing.getName());
}
}

private static SystemAddressId createAddress(MailingService svc, String type) throws
Exception
{
    SystemAddress address = new SystemAddress();
    address.setEmailAddress(type+System.currentTimeMillis()+"@sm.com");
    if (type.equalsIgnoreCase("bounce"))
    {
        address.setIsBounce(true);
    }
    else if (type.equalsIgnoreCase("from"))
    {
        address.setIsFrom(true);
    }
    else
    {
        address.setIsReply(true);
    }
    CreateRequest createAddressRequest = new CreateRequest();
    createAddressRequest.getBaseObject().add(address);
    SystemAddressId addressId = (SystemAddressId)(svc.create
(createAddressRequest).getCreateResponse().get(0).getObjectId());
    return addressId;
}

```

```

private static void describeTemplateCreateSample(MailingService svc) throws Exception
{
    String name = "Sample_Template_" + System.currentTimeMillis();
    System.out.println("");
    System.out.println("-----");
    System.out.println("Template create");
    System.out.println("");

    Template sampleTemplate = new Template();

    /* Initialize template meta information and envelope properties. */
    sampleTemplate.setName(name);
    sampleTemplate.setDescription("A Sample Template created via the Message Studio API  
sample code");
    sampleTemplate.setIsApproved(true);
    sampleTemplate.setBodyEncoding(Encoding.SEVEN_BIT);
    sampleTemplate.setHeaderEncoding(Encoding.SEVEN_BIT);
    sampleTemplate.setOutputBodyCharSet(CharSet.UTF_8);
    sampleTemplate.setOutputHeaderCharSet(CharSet.UTF_8);

    /* Create and set system addresses for the template (you can use existing ones if  
needed). */
    sampleTemplate.setBounceAddressId(createAddress(svc, "bounce"));
    sampleTemplate.setFromAddressId(createAddress(svc, "from"));
    sampleTemplate.setReplyAddressId(createAddress(svc, "reply"));

    /* Set template content and related properties. */
    MessagePart templateContentObject = new MessagePart();
    templateContentObject.setFormat(MessageFormat.HTML);
    templateContentObject.setContent("<html><head>Sample Content</head><body>Hi ##first_  
name## ##last_name##!<br/>Welcome to our new users program! You can click <a  
href=\"http:www.yourdomain.com/newuser\">here</a> to get a tour of our  
services</body></html>");
    sampleTemplate.getMessagePart().add(templateContentObject);

    /* Make a call to the create API. */
    CreateRequest createTemplateRequest = new CreateRequest();
    createTemplateRequest.getBaseObject().add(sampleTemplate);
    TemplateId createdTemplateId = (TemplateId)(svc.create  
(createTemplateRequest).getCreateResponse().get(0).getObjectId());

    /* Get details about the just created template using the get API call. */
    GetRequest getRequest = new GetRequest();
    getRequest.getObjectId().add(createdTemplateId);

```

```

    Template createdTemplate = (Template) (svc.get (getRequest) .getGetResponse () .get
(0) .getBaseObject ());

    System.out.println("Created template '" + createdTemplate.getName () + "'");
}

/* Service name constant: defined in WSDL */
private static String SERVICE_NAME = "MailingService";
private static String SERVICE_NAMESPACE =
"http://www.strongmail.com/services/2009/03/02";
private static String SERVICE_NAMESPACE_SCHEMA =
"http://www.strongmail.com/services/2009/03/02/schema";

class ClientPasswordCallbackHandler implements CallbackHandler
{
    private String password;

    public ClientPasswordCallbackHandler(String password)
    {
        this.password = password;
    }

    public void handle(Callback[] callbacks)
    throws IOException, UnsupportedCallbackException
    {
        WSPasswordCallback pc = (WSPasswordCallback) callbacks[0];
        pc.setPassword(password);
    }
}

class AuthenticationInterceptor extends WSS4JOutInterceptor
{
    /* Schema namespace */
    private String schemaNS;

    /* Interceptor to set headers: organizationName */
    private PhaseInterceptor phaseInterceptor;

    /* organizationName to login */
    private String organizationName;
    private boolean isSSO;
    public AuthenticationInterceptor(String schemaNS,
    Map<String, Object> properties,
    String orgName)
    {

```

```

        super(properties);
        this.schemaNS = schemaNS;
        this.phaseInterceptor = new WSS4JOutInterceptorInternal();
        this.organizationName = orgName;
    }

    public void setOrganizationName(String orgName)
    {
        this.organizationName = orgName;
    }

    public void setUsername(String username)
    {
        getProperties().put(WSHandlerConstants.USER, username);
    }

    public void setPassword(String password)
    {
        getProperties().put(WSHandlerConstants.PW_CALLBACK_REF,
            new ClientPasswordCallbackHandler(password));
    }

    public void handleMessage(SoapMessage message) throws Fault
    {
        super.handleMessage(message);
        message.getInterceptorChain().add(phaseInterceptor);
    }

    /**
     * Set SOAP header before sending out message. Header includes username, password,
     * organizationName. This message set organizationName and parent class
     * will set username/password.
     */
    class WSS4JOutInterceptorInternal implements PhaseInterceptor<SoapMessage>
    {
        private static final String ORGANIZATION_TOKEN_ELEMENT_NAME = "OrganizationToken";
        private static final String ORGANIZATION_ELEMENT_NAME = "organizationName";

        public WSS4JOutInterceptorInternal()
        {
            super();
        }

        public void handleMessage(SoapMessage message)
        {

```

```

try
{
    Element securityHeader = getSecurityHeader(message);

    /* Create <OrganizationToken> and <organizationName> element, */
    /* and set <organizationName> as child of <OrganizationToken> */
    Element e = securityHeader.getOwnerDocument().createElementNS(schemaNS,
                                                                    ORGANIZATION_
TOKEN_ELEMENT_NAME);
    Element e2 = securityHeader.getOwnerDocument().createElementNS(schemaNS,
                                                                    ORGANIZATION_
ELEMENT_NAME);
    Text t = securityHeader.getOwnerDocument().createTextNode(
        organizationName);
    e2.appendChild(t);
    e.appendChild(e2);
    securityHeader.appendChild(e);
}
catch (Throwable e)
{
    e.printStackTrace();
}
}

private Element getSecurityHeader(SoapMessage message) throws SOAPException
{
    SOAPMessage doc = message.getContent(SOAPMessage.class);
    String actor = (String)getOption(WSHandlerConstants.ACTOR);

    SOAPConstants sc = WSSecurityUtil.getSOAPConstants(doc.getSOAPPart
().getDocumentElement());
    return WSSecurityUtil.getSecurityHeader(doc.getSOAPPart(), actor, sc);
}

public Set<String> getAfter()
{
    return Collections.emptySet();
}

public Set<String> getBefore()
{
    return Collections.emptySet();
}

public String getId()

```

```

    {
        return WSS4JOutInterceptorInternal.class.getName();
    }

    public String getPhase()
    {
        return Phase.POST_PROTOCOL;
    }

    public void handleFault(SoapMessage message)
    {
    }
}
}

```

Sample C# Code

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */

using System;
using System.Linq;
using System.Text;
using System.Collections.Generic;
using System.Web.Services;
using System.Xml;
using System.IO;

```

```

using System.Threading;
using System.Security.Cryptography.X509Certificates;
using Microsoft.Web.Services.Timestamp;
using Microsoft.Web.Services2;
using Microsoft.Web.Services2.Configuration;
using Microsoft.Web.Services2.Security;
using Microsoft.Web.Services2.Security.Tokens;
using Microsoft.Web.Services2.Policy;
using Microsoft.Web.Services2.Referral;

namespace ConsoleApplication1
{
    class SampleProgram
    {
        /* Namespace and endpoint URL: change to your server */
        public const string ns = "http://www.strongmail.com/services/2009/03/02/schema";
        public const string url = "https://127.0.0.1:4443/sm/services/mailing/2009/03/02";

        /* Generated MailingService stub */
        MailingService svc;

        public static void Main(string[] args)
        {
            /* Organization, username, password */
            string organization = "admin"; /* Use "/" to specify suborganization. E.g.
admin/subadmin/onemore */
            string username = "admin";
            string password = "admin";

            /* Initialize the service object with the org, username and password details. */
            MailingService svc = createService(username, password, organization);

            /* Sample for listing users and roles. */
            describeUserListSample(svc);

            /* Sample for listing templates. */
            describeTemplateListSample(svc);

            /* Sample for listing mailings. */
            describeMailingListSample(svc);

            /* Sample for creating a template. */
            describeTemplateCreateSample(svc);
        }
    }
}

```

```

}

/* Create a MailingService client with proper headers. */
private static MailingService createService(string username, string password, string
organization)
{

    /* Create service and set endpoint */
    MailingService svc = new MailingService();
    svc.Url = url;
    Console.WriteLine("Endpoint: " + svc.Url);

    /* Ignore Certificate by trusting all certificate */
    System.Net.ServicePointManager.CertificatePolicy = new TrustAllCertificatePolicy();

    /* Create security tokens for SOAP header */
    UsernameToken userToken = new UsernameToken(username, password,
PasswordOption.SendPlainText);
    SoapContext requestContext = svc.RequestSoapContext;
    requestContext.Security.Tokens.Add(userToken);

    /* Use filter to set organization token and get rid of timestamp since it does not
work with Java server. */
    MailingServiceOutFilter myfilter = new MailingServiceOutFilter(organization, ns);
    svc.Pipeline.OutputFilters.Insert(0, myfilter);
    return svc;
}

private static void describeUserListSample(MailingService svc)
{

    /* List users using the list API with the UserFilter. */
    ListRequest userListRequest = new ListRequest();
    UserFilter userFilter = new UserFilter();
    userFilter.isAscending = true;
    userFilter.isAscendingSpecified = true;
    userFilter.pageNumber = 0;
    userFilter.pageNumberSpecified = true;
    userFilter.recordsPerPage = 10;
    userFilter.recordsPerPageSpecified = true;
    userListRequest.filter = userFilter;
    ObjectId[] userIds = svc.list(userListRequest).objectId;

    /* Get details about each user using the get API call. */
    User[] users = new User[userIds.Length];

```



```

for (int i = 0; i < userIds.Length; i++)
{
    UserId userId = (UserId) (userIds[i]);
    ObjectId[] objectIds = new ObjectId[1];
    objectIds[0] = userId;
    User user = (User) (svc.get(objectIds).getResponse[0].baseObject);
    users[i] = user;
}

/* Display all users */
Console.WriteLine("");
Console.WriteLine("-----");
Console.WriteLine("Users List");
for (int i = 0; i < users.Length; i++)
{
    string roles = "";
    if (users[i].isSuperUser)
    {
        /* User is a super user. */
        roles = "Super User";
    }
    else if (users[i].isAdmin)
    {
        /* User is an admin. */
        roles = "Admin User";
    }
    else
    {
        User.Access assignedRoles = users[i].access;

        /* Get the roles assigned to the user from the "access" field. */
        for (int j = 0; j < assignedRoles.Length; j++)
        {
            RoleId roleId = assignedRoles[j].roleId;

            /* Get details about the role (e.g. its name) using the get API call. */
            ObjectId[] objectIds = new ObjectId[1];
            objectIds[0] = roleId;
            Role role = (Role) (svc.get(objectIds).getResponse[0].baseObject);
            roles += role.getName() + " ";
        }
        if (roles == "")
        {

```

```

        roles = "<None>";
    }
}

Console.WriteLine(users[i].firstName + " " + users[i].lastName + ". Role(s): " +
roles);
}
}

private static void describeTemplateListSample(MailingService svc)
{

    /* List users using the list API with the TemplateFilter. */
    ListRequest templateListRequest = new ListRequest();
    TemplateFilter templateFilter = new TemplateFilter();
    templateFilter.isAscending = true;
    templateFilter.isAscendingSpecified = true;
    templateFilter.pageNumber = 0;
    templateFilter.pageNumberSpecified = true;
    templateFilter.recordsPerPage = 10;
    templateFilter.recordsPerPageSpecified = true;
    templateListRequest.filter = templateFilter;
    ListResponse templateListResponse = svc.list(templateListRequest);

    /* Get details about each template using the get API call. */
    ObjectId[] templateIds = svc.list(templateListRequest).objectId;
    Template[] templates = new Template[templateIds.Length];
    for (int i = 0; i < templateIds.Length; i++)
    {

        TemplateId templateId = (TemplateId) (templateIds[i]);
        ObjectId[] objectIds = new ObjectId[1];
        objectIds[0] = templateId;
        Template template = (Template) (svc.get(objectIds).getResponse[0].baseObject);
        templates[i] = template;
    }

    /* Display all templates. */
    Console.WriteLine("");
    Console.WriteLine("-----");
    Console.WriteLine("Templates List");
    for (int i = 0; i < templates.Length; i++)
    {
        Console.WriteLine(templates[i].name);
    }
}

```

```

}

private static void describeMailingListSample(MailingService svc)
{
    /* List users using the list API with the MailingFilter. */
    ListRequest mailingListRequest = new ListRequest();
    MailingFilter mailingFilter = new MailingFilter();
    mailingFilter.isAscending = true;
    mailingFilter.isAscendingSpecified = true;
    mailingFilter.pageNumber = 0;
    mailingFilter.pageNumberSpecified = true;
    mailingFilter.recordsPerPage = 10;
    mailingFilter.recordsPerPageSpecified = true;
    mailingListRequest.filter = mailingFilter;
    ListResponse mailingListResponse = svc.list(mailingListRequest);

    /* Get details about each mailing using the get API call. */
    ObjectId[] mailingIds = svc.list(mailingListRequest).objectId;
    Mailing[] mailings = new Mailing[mailingIds.Length];
    for (int i = 0; i < mailingIds.Length; i++)
    {
        MailingId mailingId = (MailingId) (mailingIds[i]);
        ObjectId[] objectIds = new ObjectId[1];
        objectIds[0] = mailingId;
        Mailing mailing = (Mailing) (svc.get(objectIds).getResponse[0].baseObject);
        mailings[i] = mailing;
    }

    /* Display all mailings. */
    Console.WriteLine("");
    Console.WriteLine("-----");
    Console.WriteLine("Mailings List");
    for (int i = 0; i < mailings.Length; i++)
    {
        Console.WriteLine(mailings[i].name);
    }
}

private static SystemAddressId createAddress(MailingService svc, string type)
{
    SystemAddress address = new SystemAddress();
    address.emailAddress = type + "@sm.com";
    if (type == "bounce")
    {

```

```

        address.isBounce = true;
    }
    else if (type == "from")
    {
        address.isFrom = true;
    }
    else
    {
        address.isReply = true;
    }

    BaseObject[] objects = new BaseObject[1];
    objects[0] = address;
    return (SystemAddressId) (svc.create(objects).createResponse[0].objectId);
}

private static void describeTemplateCreateSample(MailingService svc)
{
    string name = "Sample_Template";
    Console.WriteLine("");
    Console.WriteLine("-----");
    Console.WriteLine("Template create");
    Console.WriteLine("");
    Template template = new Template();

    /* Initialize template meta information and envelope properties. */
    template.bodyEncoding = Encoding.BASE64;
    template.description = "Template from dotnet";
    template.fromName = "From Name";
    template.headerEncoding = Encoding.EIGHT_BIT;
    template.isApproved = true;
    template.name = templateName;
    template.subject = "template subject from dotnet";
    template.Item = CharSet.ASCII;
    template.Item1 = CharSet.UTF8;

    /* Create and set system addresses for the template (you can use existing ones if
needed). */
    template.bounceAddressId = createAddress(svc, "bounce");
    template.fromAddressId = createAddress(svc, "from");
    template.replyAddressId = createAddress(svc, "reply");

    /* Set template content and related properties. */
    MessagePart messagePart = new MessagePart();

```

```

        messagePart.content = "<html><head>Sample Content</head><body>Hi ##first_name##
##last_name##!<br/>Welcome to our new users program! You can click <a
href=\"http://www.yourdomain.com/newuser\">here</a> to get a tour of our
services</body></html>";

        messagePart.format = MessageFormat.HTML;
        messagePart.isXsl = false;
        template.messagePart = new MessagePart[] { messagePart };

        /* Make a call to the create API. */
        BaseObject[] objects = new BaseObject[1];
        objects[0] = template;
        TemplateId createdTemplateId = (TemplateId) (svc.create(objects).createResponse
[0].objectId);

        /* Get details about the just created template using the get API call. */
        ObjectId[] objectIds = new ObjectId[1];
        objectIds[0] = createdTemplateId;
        Template createdTemplate = (Template) (svc.get(objectIds).getResponse[0].baseObject);

        Console.WriteLine("Created template '" + createdTemplate.name + "'");
    }

    /* A policy to trust all certificate */
    public class TrustAllCertificatePolicy : System.Net.ICertificatePolicy
    {
        public TrustAllCertificatePolicy()
        { }
        public bool CheckValidationResult(System.Net.ServicePoint sp,
X509Certificate cert, System.Net.WebRequest req, int problem)
        {
            return true;
        }
    }

    public class MailingServiceOutFilter : SoapOutputFilter
    {
        private string _ns; /* namespace of organization name */
        private string _organizationName; /* organization name to login */
        public MailingServiceOutFilter() { }
        public MailingServiceOutFilter(string organizationName, string ns)
        {
            _organizationName = organizationName;
            _ns = ns;
        }

        public override void ProcessMessage(SoapEnvelope envelope)

```

```

{

    /* Add OrganizationName token */
    XmlElement OrgTokenElement = envelope.CreateElement("sm", "OrganizationToken", _
ns);

    XmlElement orgElement = envelope.CreateElement("sm", "organizationName", _ns);

    /* Set organizationName value */
    XmlText orgName = envelope.CreateTextNode(_organizationName);
    orgElement.AppendChild(orgName);
    OrgTokenElement.AppendChild(orgElement);

    /* Remove timestamp token */
    IEnumerator it = envelope.Header.GetEnumerator();
    while(it.MoveNext()){
        XmlElement el = (XmlElement) it.Current;

        if (el.Name.Equals("wsse:Security")) {
            el.AppendChild(OrgTokenElement);

            /* Remove Timestamp */
            XmlNodeList timestamp = el.GetElementsByTagName("wsu:Timestamp");
            el.RemoveChild(timestamp.Item(0));
        }
    }
}
}
}
}
}

```

Sample XML Code

The following are examples of the XML requests and responses for the calls shown in the Java and C# sample code. The examples are categorized by the Web Services API Version (1 or 2)

Web Services API v2 Examples
<ul style="list-style-type: none">• Channel Standard Mailing Create Request (v2) on page 185• Channel Standard Mailing Create Response (v2) on page 187• Channel Standard Mailing Get Request (v2) on page 188• Channel Standard Mailing Get Response (v2) on page 189• Channel Transactional Mailing Create Request (v2) on page 191• Channel Transactional Mailing Create Response (v2) on page 193• Channel Transactional Mailing Get Request (v2) on page 194• Channel Transactional Mailing Get Response (v2) on page 195• Channel Template Create Request (v2) on page 197• Channel Template Create Response (v2) on page 199• Channel Template Get Request (v2) on page 198• Channel Template Get Response (v2) on page 200• External Data Source Create Request (v2) on page 202• Standard Mailing Create Request (v2) on page 204• Standard Mailing Create Response (v2) on page 207• Standard Mailing Get Request (v2) on page 207• Standard Mailing Get Response (v2) on page 208• Standard Mailing List Request (v2) on page 211• Standard Mailing List Response (v2) on page 212• Transactional Mailing Create Request (v2) on page 213• Transactional Mailing Create Response (v2) on page 215• Transactional Mailing Get Request (v2) on page 216• Transactional Mailing Get Response (v2) on page 217

Web Services API v2 Examples

- [Template Create Request \(v2\) on page 219](#)
- [Template Create Response \(v2\) on page 221](#)
- [Template Get Request \(v2\) on page 222](#)
- [Template Get Response \(v2\) on page 223](#)
- [Template List Request \(v2\) on page 226](#)
- [Template List Response \(v2\) on page 227](#)
- [Template Update Response - Success \(v2\) on page 230](#)
- [Template Update Response - Failure \(v2\) on page 231](#)
- [User Get Request \(v2\) on page 232](#)
- [User Get Response \(v2\) on page 233](#)
- [User List Request \(v2\) on page 234](#)
- [User List Response \(v2\) on page 235](#)

Web Services API v1 Examples

- [Mailing Create Request \(v1\) on page 236](#)
- [Mailing Create Response \(v1\) on page 238](#)
- [Mailing Get Request \(v1\) on page 239](#)
- [Mailing Get Response \(v1\) on page 240](#)
- [Mailing List Request \(v1\) on page 241](#)
- [Mailing List Response \(v1\) on page 242](#)
- [Template Create Request \(v1\) on page 243](#)
- [Template Create Response \(v1\) on page 244](#)
- [Template Get Request \(v1\) on page 245](#)
- [Template Get Response \(v1\) on page 246](#)
- [Template List Request \(v1\) on page 248](#)
- [Template List Response \(v1\) on page 249](#)
- [User Get Request \(v1\) on page 250](#)
- [User Get Response \(v1\) on page 251](#)

Web Services API v1 Examples

- [User List Request \(v1\) on page 252](#)
- [User List Response \(v1\) on page 253](#)

Channel Standard Mailing Create Request (v2)

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * ----- *
 */
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken wsu:Id="UsernameToken-8DC6F0E182FEA86DA2139461428531132">
        <wsse:Username>admin</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">admin</wsse:Password>
      </wsse:UsernameToken>
      <OrganizationToken xmlns="http://www.strongmail.com/services/v2/schema">
        <organizationName>admin</organizationName>
        <subOrganizationId>
          <id>1</id>
        </subOrganizationId>
      </OrganizationToken>
    </wsse:Security>
  </SOAP-ENV:Header>
  <soap:Body>
    <create xmlns="http://www.strongmail.com/services/v2/schema">

```

```

<baseObject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ChannelStandardMailing">
  <bodyEncoding>BASE_64</bodyEncoding>
  <isApproved>true</isApproved>
  <isCompliant>true</isCompliant>
  <contentBlockId>
    <id>309</id>
  </contentBlockId>
  <fieldDelimiter>.:</fieldDelimiter>
  <format>HTML</format>
  <headerEncoding>EIGHT_BIT</headerEncoding>
  <name>channel_standard_mailing_20140902214800</name>
  <priority>NORMAL</priority>
  <outputBodyCharSetToken>##body token##</outputBodyCharSetToken>
  <outputHeaderCharSetToken>##header token##</outputHeaderCharSetToken>
  <templateId xsi:type="ChannelTemplateId">
    <id>1103</id>
  </templateId>
  <type>ONE_TIME</type>
  <description>StandardMailing from CXF</description>
  <useHeaderFromTemplate>false</useHeaderFromTemplate>
  <eliminateDuplicates>true</eliminateDuplicates>
  <seedListId>
    <id>210</id>
  </seedListId>
  <suppressionListId>
    <id>304</id>
  </suppressionListId>
  <minRecipientLimit>2</minRecipientLimit>
  <maxRecipientLimit>123</maxRecipientLimit>
  <notificationOption>
    <recipientAddresses>test1_test@strongmail.com</recipientAddresses>
    <recipientAddresses>test2_test@strongmail.com</recipientAddresses>
    <recipientAddresses>test3_test@strongmail.com</recipientAddresses>
    <notifyStateChange>true</notifyStateChange>
    <notifyTargetingCompletion>true</notifyTargetingCompletion>
  </notificationOption>
  <channelId>
    <id>2</id>
  </channelId>
  <includedTarget>
    <targetId>
      <id>1304</id>
    </targetId>
    <suppressionColumnName>id</suppressionColumnName>
  </includedTarget>
</baseObject>

```

```

        <seedColumnName>id</seedColumnName>
    </includedTarget>
    <channelHeader>
        <headerName>CustomUserHeader1</headerName>
        <headerValue>CustomUserHeaderValue1</headerValue>
    </channelHeader>
    <channelHeader>
        <headerName>CustomUserHeader2</headerName>
        <headerValue>CustomUserHeaderValue2</headerValue>
    </channelHeader>
    <channelHeader>
        <headerName>CustomUserHeader3</headerName>
        <headerValue>CustomUserHeaderValue3</headerValue>
    </channelHeader>
    <channelProfileAttribute>name</channelProfileAttribute>
    <channelProfileAttribute>id</channelProfileAttribute>
</baseObject>
</create>
</soap:Body>
</soap:Envelope>

```

Channel Standard Mailing Create Response (v2)

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <createResponse xmlns="http://www.strongmail.com/services/v2/schema">
      <success>true</success>
    </createResponse>
  </soap:Body>
</soap:Envelope>

```

```

<fault xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
<createResponse>
  <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ChannelStandardMailingId">
    <id>1201</id>
  </objectId>
</createResponse>
</createResponse>
</soap:Body>
</soap:Envelope>

```

Channel Standard Mailing Get Request (v2)

```

/*
* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *
* *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* *
* Distributing this source code outside your organization *
* is strictly prohibited. *
* *
* ----- *
*/
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken wsu:Id="UsernameToken-6255C29619274FAEDA13946313651284">
        <wsse:Username>admin</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">admin</wsse:Password>
      </wsse:UsernameToken>
      <OrganizationToken xmlns="http://www.strongmail.com/services/v2/schema">
        <organizationName>admin</organizationName>
        <subOrganizationId>
          <id>1</id>
        </subOrganizationId>

```

```

        </OrganizationToken>
    </wsse:Security>
</SOAP-ENV:Header>
<soap:Body>
    <get xmlns="http://www.strongmail.com/services/v2/schema">
        <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ChannelStandardMailingId">
            <id>1201</id>
        </objectId>
    </get>
</soap:Body>
</soap:Envelope>

```

Channel Standard Mailing Get Response (v2)

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <getResponse xmlns="http://www.strongmail.com/services/v2/schema">
            <success>true</success>
            <fault xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
            <getResponse>
                <baseObject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ChannelStandardMailing">
                    <modifiedTime>2014-03-12T14:21:25.350+05:30</modifiedTime>
                    <objectId xsi:type="ChannelStandardMailingId">
                        <id>1201</id>
                    </objectId>
                </baseObject>
            </getResponse>
        </getResponse>
    </soap:Body>
</soap:Envelope>

```

```

<version>6</version>
<bodyEncoding>BASE_64</bodyEncoding>
<isApproved>true</isApproved>
<isCompliant>true</isCompliant>
<contentBlockId>
  <id>309</id>
</contentBlockId>
<fieldDelimiter>::</fieldDelimiter>
<format>Push_Notification</format>
<headerEncoding>EIGHT_BIT</headerEncoding>
<lastGoodStatus>INITIATING_TARGETING</lastGoodStatus>
<name>Updated_channel_standard_mailing_20140902214800</name>
<priority>NORMAL</priority>
<outputBodyCharSetToken>##body token##</outputBodyCharSetToken>
<outputHeaderCharSetToken>##header token##</outputHeaderCharSetToken>
<rowDelimiter>>_sm_row_delim_</rowDelimiter>
<serverErrorCode>0</serverErrorCode>
<status>ERROR</status>
<templateId xsi:type="ChannelTemplateId">
  <id>1103</id>
</templateId>
<type>ONE_TIME</type>
<createdTime>2014-03-12T14:21:25.365+05:30</createdTime>
<description>StandardMailing from CXF</description>
<organizationId>
  <id>1</id>
</organizationId>
<ownerId>
  <id>1</id>
</ownerId>
<useHeaderFromTemplate>false</useHeaderFromTemplate>
<eliminateDuplicates>true</eliminateDuplicates>
<seedListId>
  <id>210</id>
</seedListId>
<suppressionListId>
  <id>304</id>
</suppressionListId>
<minRecipientLimit>2</minRecipientLimit>
<maxRecipientLimit>123</maxRecipientLimit>
<notificationOption>
  <recipientAddresses>test1_test@strongmail.com</recipientAddresses>
  <recipientAddresses>test2_test@strongmail.com</recipientAddresses>
  <recipientAddresses>test3_test@strongmail.com</recipientAddresses>
  <notifyStateChange>true</notifyStateChange>

```

```

        <notifyTargetingCompletion>true</notifyTargetingCompletion>
    </notificationOption>
    <channelId>
        <id>1</id>
    </channelId>
    <includedTarget>
        <targetId>
            <id>1304</id>
        </targetId>
        <suppressionColumnName>id</suppressionColumnName>
        <seedColumnName>id</seedColumnName>
    </includedTarget>
    <channelHeader>
        <headerName>CustomUserHeader1</headerName>
        <headerValue>CustomUserHeaderValue1</headerValue>
        <headerType>LOCAL</headerType>
    </channelHeader>
    <channelHeader>
        <headerName>CustomUserHeader2</headerName>
        <headerValue>CustomUserHeaderValue2</headerValue>
        <headerType>LOCAL</headerType>
    </channelHeader>
    <channelHeader>
        <headerName>CustomUserHeader3</headerName>
        <headerValue>CustomUserHeaderValue3</headerValue>
        <headerType>LOCAL</headerType>
    </channelHeader>
    <channelProfileAttribute>id</channelProfileAttribute>
    <channelProfileAttribute>name</channelProfileAttribute>
</baseObject>
</getResponse>
</getResponse>
</soap:Body>
</soap:Envelope>

```

Channel Transactional Mailing Create Request (v2)

```

/*
* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *

```

```

* Visit http://www.strongview.com for more information.
*
* You may incorporate this Source Code in your application only
* if you own a valid license from Selligent, Inc.
* and the copyright notices are not removed from the source code.
*
* Distributing this source code outside your organization
* is strictly prohibited.
*
* -----
*/
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken wsu:Id="UsernameToken-8DC6F0E182FEA86DA2139461428531132">
        <wsse:Username>admin</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">admin</wsse:Password>
      </wsse:UsernameToken>
      <OrganizationToken xmlns="http://www.strongmail.com/services/v2/schema">
        <organizationName>admin</organizationName>
        <subOrganizationId>
          <id>1</id>
        </subOrganizationId>
      </OrganizationToken>
    </wsse:Security>
  </SOAP-ENV:Header>
  <soap:Body>
    <create xmlns="http://www.strongmail.com/services/v2/schema">
      <baseObject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ChannelTransactionalMailing">
        <bodyEncoding>BASE_64</bodyEncoding>
        <isApproved>true</isApproved>
        <isCompliant>true</isCompliant>
        <fieldDelimiter>.:</fieldDelimiter>
        <format>HTML</format>
        <headerEncoding>EIGHT_BIT</headerEncoding>
        <name>channel_tx_mailing_20140902214800</name>
        <priority>NORMAL</priority>
        <outputBodyCharSetToken>##body token##</outputBodyCharSetToken>
        <outputHeaderCharSetToken>##header token##</outputHeaderCharSetToken>
        <templateId xsi:type="ChannelTemplateId">
          <id>1201</id>
        </templateId>
      </baseObject>
    </create>
  </soap:Body>
</soap:Envelope>

```



```

        </templateId>
        <type>TRANSACTIONAL</type>
        <description>TransactionalMailing from CXF</description>
        <useHeaderFromTemplate>false</useHeaderFromTemplate>
        <targetId>
            <id>1106</id>
        </targetId>
        <channelId>
            <id>2</id>
        </channelId>
    </baseObject>
</create>
</soap:Body>
</soap:Envelope>

```

Channel Transactional Mailing Create Response (v2)

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * ----- *
 */
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <createResponse xmlns="http://www.strongmail.com/services/v2/schema">
            <success>true</success>
            <fault xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
            <createResponse>
                <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ChannelTransactionalMailingId">

```

```

        <id>1303</id>
    </objectId>
</createResponse>
</createResponse>
</soap:Body>
</soap:Envelope>

```

Channel Transactional Mailing Get Request (v2)

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * ----- *
 */
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
    <s:Header>
        <wsse:Security mustUnderstand="1" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
                <wsse:Username>admin</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">admin</wsse:Password>
            </wsse:UsernameToken>
            <OrganizationToken xmlns="http://www.strongmail.com/services/v2/schema">
                <organizationName>admin</organizationName>
            </OrganizationToken>
        </wsse:Security>
    </s:Header>
    <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema">

```

```

    <get xmlns="http://www.strongmail.com/services/v2/schema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ChannelTransactionalMailingId">
        <id>2200</id>
    </objectId>
</get>
</soap:Body>
</soap:Envelope>

```

Channel Transactional Mailing Get Response (v2)

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getResponse xmlns="http://www.strongmail.com/services/v2/schema">
      <success>true</success>
      <fault xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
      <getResponse>
        <baseObject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ChannelTransactionalMailing">
          <modifiedTime>2014-03-12T14:21:25.350+05:30</modifiedTime>
          <objectId xsi:type="ChannelTransactionalMailingId">
            <id>1303</id>
          </objectId>
          <version>7</version>
          <bodyEncoding>BASE_64</bodyEncoding>
          <isApproved>true</isApproved>
          <isCompliant>true</isCompliant>

```

```

<fieldDelimiter>::</fieldDelimiter>
<format>Push_Notification</format>
<headerEncoding>EIGHT_BIT</headerEncoding>
<lastGoodStatus>INITIATING_LOADING</lastGoodStatus>
<name>Update_channel_tx_mailing_20140902214800</name>
<priority>NORMAL</priority>
<outputBodyCharSetToken>##body token##</outputBodyCharSetToken>
<outputHeaderCharSetToken>##header token##</outputHeaderCharSetToken>
<rowDelimiter>_sm_row_delim_</rowDelimiter>
<serverErrorCode>0</serverErrorCode>
<status>ACTIVE</status>
<templateId xsi:type="ChannelTemplateId">
  <id>1202</id>
</templateId>
<type>TRANSACTIONAL</type>
<createdTime>2014-03-12T14:21:25.365+05:30</createdTime>
<description>TransactionalMailing from CXF</description>
<organizationId>
  <id>1</id>
</organizationId>
<ownerId>
  <id>1</id>
</ownerId>
<useHeaderFromTemplate>false</useHeaderFromTemplate>
<mailingConfig>Mailing.1303.9655</mailingConfig>
<recordStructure>id</recordStructure>
<recordStructure>email_address</recordStructure>
<recordStructure>name</recordStructure>
<targetId>
  <id>1106</id>
</targetId>
<notificationOption>
  <notifyStateChange>false</notifyStateChange>
</notificationOption>
<channelId>
  <id>2</id>
</channelId>
</baseObject>
</getResponse>
</getResponse>
</soap:Body>
</soap:Envelope>

```

Channel Template Create Request (v2)

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken wsu:Id="UsernameToken-8DC6F0E182FEA86DA2139461428531132">
        <wsse:Username>admin</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">admin</wsse:Password>
      </wsse:UsernameToken>
      <OrganizationToken xmlns="http://www.strongmail.com/services/v2/schema">
        <organizationName>admin</organizationName>
      </OrganizationToken>
    </wsse:Security>
  </SOAP-ENV:Header>
  <soap:Body>
    <create xmlns="http://www.strongmail.com/services/v2/schema">
      <baseObject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ChannelTemplate">
        <description>Template from CXF</description>
        <bodyEncoding>BASE_64</bodyEncoding>
        <headerEncoding>EIGHT_BIT</headerEncoding>
        <isApproved>true</isApproved>
        <messagePart>

```

```

        <content>This is example of push notification. However to run this sample User
should MQ License</content>
        <format>push_notification</format>
        <isXsl>>false</isXsl>
    </messagePart>
    <outputBodyCharSet>ASCII</outputBodyCharSet>
    <outputHeaderCharSet>UTF-8</outputHeaderCharSet>
    <name>channel_std_mailing_cxf20140902214800</name>
    <linkTracking>Full</linkTracking>
    <channelId>
        <id>2</id>
    </channelId>
    <channelHeader>
        <headerName>CustomUserHeader1</headerName>
        <headerValue>CustomUserHeaderValue1</headerValue>
    </channelHeader>
    <channelHeader>
        <headerName>CustomUserHeader2</headerName>
        <headerValue>CustomUserHeaderValue2</headerValue>
    </channelHeader>
    <channelHeader>
        <headerName>CustomUserHeader3</headerName>
        <headerValue>CustomUserHeaderValue3</headerValue>
    </channelHeader>
</baseObject>
</create>
</soap:Body>
</soap:Envelope>

```

Channel Template Get Request (v2)

```

/*
* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *
* *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* *
* Distributing this source code outside your organization *

```

```

* is strictly prohibited. *
* *
* ----- *
*/
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken wsu:Id="UsernameToken-8DC6F0E182FEA86DA2139461428531132">
        <wsse:Username>admin</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">admin</wsse:Password>
      </wsse:UsernameToken>
      <OrganizationToken xmlns="http://www.strongmail.com/services/v2/schema">
        <organizationName>admin</organizationName>
        <subOrganizationId>
          <id>1</id>
        </subOrganizationId>
      </OrganizationToken>
    </wsse:Security>
  </SOAP-ENV:Header>
  <soap:Body>
    <get xmlns="http://www.strongmail.com/services/v2/schema">
      <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ChannelTemplateId">
        <id>1201</id>
      </objectId>
    </get>
  </soap:Body>
</soap:Envelope>

```

Channel Template Create Response (v2)

```

/*
* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *
* *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *

```

```

*                                                                 *
*   Distributing this source code outside your organization      *
*   is strictly prohibited.                                       *
*                                                                 *
* ----- *
*/
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <createResponse xmlns="http://www.strongmail.com/services/v2/schema">
      <success>true</success>
      <fault xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
      <createResponse>
        <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ChannelTemplateId">
          <id>1201</id>
        </objectId>
      </createResponse>
    </createResponse>
  </soap:Body>
</soap:Envelope>

```

Channel Template Get Response (v2)

```

/*
* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *
* *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* *
* Distributing this source code outside your organization *
* is strictly prohibited. *
* *
* ----- *
*/
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getResponse xmlns="http://www.strongmail.com/services/v2/schema">
      <success>true</success>
      <fault xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>

```



```

<getResponse>
  <baseObject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ChannelTemplate">
    <modifiedTime>2014-09-02T21:26:24.196+05:30</modifiedTime>
    <objectId xsi:type="ChannelTemplateId">
        <id>1201</id>
    </objectId>
    <version>0</version>
    <createdTime>2014-09-02T21:26:24.196+05:30</createdTime>
    <description>Template from CXF</description>
    <bodyEncoding>BASE_64</bodyEncoding>
    <headerEncoding>EIGHT_BIT</headerEncoding>
    <isApproved>true</isApproved>
    <messagePart>
        <content>This is example of push notification. However to run this sample
User should MQ License</content>
        <format>Push_Notification</format>
        <isXsl>false</isXsl>
    </messagePart>
    <outputBodyCharSet>ASCII</outputBodyCharSet>
    <outputHeaderCharSet>UTF-8</outputHeaderCharSet>
    <name>ChannelTemplate_channel_standard_mailing_20140902212556</name>
    <organizationId>
        <id>1</id>
    </organizationId>
    <ownerId>
        <id>1</id>
    </ownerId>
    <linkTracking>Full</linkTracking>
    <channelId>
        <id>2</id>
    </channelId>
    <channelHeader>
        <headerName>CustomUserHeader1</headerName>
        <headerValue>CustomUserHeaderValue1</headerValue>
        <headerType>LOCAL</headerType>
    </channelHeader>
    <channelHeader>
        <headerName>CustomUserHeader2</headerName>
        <headerValue>CustomUserHeaderValue2</headerValue>
        <headerType>LOCAL</headerType>
    </channelHeader>
    <channelHeader>
        <headerName>CustomUserHeader3</headerName>
        <headerValue>CustomUserHeaderValue3</headerValue>
        <headerType>LOCAL</headerType>
    </channelHeader>

```

```

        </baseObject>
    </getResponse>
</getResponse>
</soap:Body>
</soap:Envelope>

```

External Data Source Create Request (v2)

This example uses the JDBC URL.

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * ----- *
 */
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken wsu:Id="UsernameToken-8DC6F0E182FEA86DA2139461428554533">
        <wsse:Username>admin</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">admin</wsse:Password>
      </wsse:UsernameToken>
      <OrganizationToken xmlns="http://www.strongmail.com/services/v2/schema">
        <organizationName>admin</organizationName>
        <subOrganizationId>
          <id>1</id>
        </subOrganizationId>
      </OrganizationToken>
    </wsse:Security>
  </SOAP-ENV:Header>

```

```

<soap:Body>
  <create xmlns="http://www.strongmail.com/services/v2/schema">
    <baseObject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ExternalDataSource">
      <description>External Data Source from CXF</description>
      <field>
        <dataType>DATA</dataType>
        <fieldType>INTEGER</fieldType>
        <name>id</name>
        <isPrimaryKey>true</isPrimaryKey>
        <writebackEnabled>false</writebackEnabled>
      </field>
      <field>
        <dataType>DATA</dataType>
        <fieldType>EMAIL</fieldType>
        <name>email</name>
        <isPrimaryKey>false</isPrimaryKey>
        <writebackEnabled>false</writebackEnabled>
      </field>
      <field>
        <dataType>DATA</dataType>
        <fieldType>SMS</fieldType>
        <name>phone</name>
        <isPrimaryKey>false</isPrimaryKey>
        <writebackEnabled>false</writebackEnabled>
      </field>
      <field>
        <dataType>STATUS</dataType>
        <fieldType>TEXT</fieldType>
        <name>email_status</name>
        <isPrimaryKey>false</isPrimaryKey>
        <writebackEnabled>false</writebackEnabled>
      </field>
      <field>
        <dataType>UNSUBSCRIBE</dataType>
        <fieldType>TEXT</fieldType>
        <name>unsub</name>
        <isPrimaryKey>false</isPrimaryKey>
        <writebackEnabled>false</writebackEnabled>
      </field>
      <field>
        <dataType>DATA</dataType>
        <fieldType>TEXT</fieldType>
        <name>first_name</name>
        <isPrimaryKey>false</isPrimaryKey>

```

```

        <writebackEnabled>false</writebackEnabled>
    </field>
    <field>
        <dataType>DATA</dataType>
        <fieldType>TEXT</fieldType>
        <name>last_name</name>
        <isPrimaryKey>false</isPrimaryKey>
        <writebackEnabled>false</writebackEnabled>
    </field>
    <name>cxf_eds_jdbc_1394614280867</name>
    <jdbcParameters>
        <url>jdbc:oracle:thin:@10.0.0.0:strongmail_test:1521</url>
        <username>USERNAME</username>
        <password>PASSWORD</password>
    </jdbcParameters>
    <enableLocalCopy>true</enableLocalCopy>
    <allowRefreshAtLaunchTime>true</allowRefreshAtLaunchTime>
    <dailyRefresh>
        <startTime>14:21:25.542+05:30</startTime>
    </dailyRefresh>
    <writebackTable>YOUR_TABLE_TO_WRITE_BACK</writebackTable>
    <sourceTableName>YOUR_SOURCE_TABLE</sourceTableName>
</baseObject>
</create>
</soap:Body>
</soap:Envelope>

```

Standard Mailing Create Request (v2)

```

/*
* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *
* *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* *
* Distributing this source code outside your organization *
* is strictly prohibited. *
* *

```

```

* ----- *
*/
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken wsu:Id="UsernameToken-8DC6F0E182FEA86DA2139461428531132">
        <wsse:Username>admin</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">admin</wsse:Password>
      </wsse:UsernameToken>
      <OrganizationToken xmlns="http://www.strongmail.com/services/v2/schema">
        <organizationName>admin</organizationName>
        <subOrganizationId>
          <id>1</id>
        </subOrganizationId>
      </OrganizationToken>
    </wsse:Security>
  </SOAP-ENV:Header>
  <soap:Body>
    <create xmlns="http://www.strongmail.com/services/v2/schema">
      <baseObject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="StandardMailing">
        <attachmentId>
          <id>1100</id>
        </attachmentId>
        <bodyEncoding>BASE_64</bodyEncoding>
        <bounceAddressId>
          <id>98</id>
        </bounceAddressId>
        <isApproved>true</isApproved>
        <isCompliant>true</isCompliant>
        <contentBlockId>
          <id>1100</id>
        </contentBlockId>
        <fieldDelimiter>.:</fieldDelimiter>
        <format>HTML</format>
        <fromAddressId>
          <id>97</id>
        </fromAddressId>
        <fromName>From Name</fromName>
        <headerEncoding>EIGHT_BIT</headerEncoding>
        <header>mailing header</header>
        <name>cx_f_stmailing_demo</name>
        <priority>NORMAL</priority>
      </baseObject>
    </create>
  </soap:Body>
</soap:Envelope>

```

```

<outputBodyCharSet>UTF-8</outputBodyCharSet>
<outputHeaderCharSet>ASCII</outputHeaderCharSet>
<replyAddressId>
  <id>99</id>
</replyAddressId>
<rowDelimiter></rowDelimiter>
<subject>mailing subject</subject>
<templateId>
  <id>1101</id>
</templateId>
<type>RECURRING</type>
<description>StandardMailing from CXF</description>
<useHeaderFromTemplate>false</useHeaderFromTemplate>
<ccAddress>test_cc@strongview.com</ccAddress>
<bccAddress>test_bcc@strongview.com</bccAddress>
<schedule>
  <startDateTime>2014-03-13T15:21:25.308+05:30</startDateTime>
  <recurrence>
    <endAfterXMailings>100</endAfterXMailings>
    <dailyRecurrence>
      <interval>2</interval>
    </dailyRecurrence>
  </recurrence>
</schedule>
<eliminateDuplicates>true</eliminateDuplicates>
<includedTargetId>
  <id>1801</id>
</includedTargetId>
<seedListId>
  <id>1001</id>
</seedListId>
<suppressionListId>
  <id>1100</id>
</suppressionListId>
<minRecipientLimit>2</minRecipientLimit>
<maxRecipientLimit>123</maxRecipientLimit>
<notificationOption>
  <recipientAddresses>test1_test@strongview.com</recipientAddresses>
  <recipientAddresses>test2_test@strongview.com</recipientAddresses>
  <recipientAddresses>test3_test@strongview.com</recipientAddresses>
  <notifyStateChange>true</notifyStateChange>
  <notifyTargetingCompletion>true</notifyTargetingCompletion>
</notificationOption>
<preTargetingPeriod>5</preTargetingPeriod>
</baseObject>

```

```

    </create>
  </soap:Body>
</soap:Envelope>

```

Standard Mailing Create Response (v2)

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <createResponse xmlns="http://www.strongmail.com/services/v2/schema">
      <success>true</success>
      <fault xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
      <createResponse>
        <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="StandardMailingId">
          <id>2200</id>
        </objectId>
      </createResponse>
    </createResponse>
  </soap:Body>
</soap:Envelope>

```

Standard Mailing Get Request (v2)

```

/*
 * ----- *
 * Selligent, Inc. *

```

```

*                                                                 *
* Copyright © 2016 Selligent, Inc.. - All rights reserved.      *
*                                                                 *
* Visit http://www.strongview.com for more information.         *
*                                                                 *
* You may incorporate this Source Code in your application only  *
* if you own a valid license from Selligent, Inc.               *
* and the copyright notices are not removed from the source code.*
*                                                                 *
* Distributing this source code outside your organization       *
* is strictly prohibited.                                       *
*                                                                 *
* ----- *
*/
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken wsu:Id="UsernameToken-6255C29619274FAEDA13946313651284">
        <wsse:Username>admin</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">admin</wsse:Password>
      </wsse:UsernameToken>
      <OrganizationToken xmlns="http://www.strongmail.com/services/v2/schema">
        <organizationName>admin</organizationName>
        <subOrganizationId>
          <id>1</id>
        </subOrganizationId>
      </OrganizationToken>
    </wsse:Security>
  </SOAP-ENV:Header>
  <soap:Body>
    <get xmlns="http://www.strongmail.com/services/v2/schema">
      <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="StandardMailingId">
        <id>1</id>
      </objectId>
    </get>
  </soap:Body>
</soap:Envelope>

```

Standard Mailing Get Response (v2)

```
/*
```



```

* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *
* *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* *
* Distributing this source code outside your organization *
* is strictly prohibited. *
* *
* ----- *
*/
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getResponse xmlns="http://www.strongmail.com/services/v2/schema">
      <success>true</success>
      <fault xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
      <getResponse>
        <baseObject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="StandardMailing">
          <modifiedTime>2014-03-12T14:21:25.350+05:30</modifiedTime>
          <objectId xsi:type="StandardMailingId">
            <id>2100</id>
          </objectId>
          <version>0</version>
          <attachmentId>
            <id>1100</id>
          </attachmentId>
          <bodyEncoding>BASE_64</bodyEncoding>
          <bounceAddressId>
            <id>98</id>
          </bounceAddressId>
          <isApproved>true</isApproved>
          <isCompliant>true</isCompliant>
          <contentBlockId>
            <id>1100</id>
          </contentBlockId>
          <fieldDelimiter>::</fieldDelimiter>
          <format>HTML</format>
          <fromAddressId>
            <id>97</id>

```

```

</fromAddressId>
<fromName>From Name</fromName>
<headerEncoding>EIGHT_BIT</headerEncoding>
<header>mailing header</header>
<mailingClassId>
  <id>1</id>
</mailingClassId>
<name>cxf_stmailing1394614280867</name>
<priority>NORMAL</priority>
<outputBodyCharSet>UTF-8</outputBodyCharSet>
<outputHeaderCharSet>UTF-8</outputHeaderCharSet>
<replyAddressId>
  <id>99</id>
</replyAddressId>
<rowDelimiter></rowDelimiter>
<serverErrorCode>0</serverErrorCode>
<status>EDITING</status>
<subject>mailing subject</subject>
<templateId>
  <id>1101</id>
</templateId>
<type>RECURRING</type>
<createdTime>2014-03-12T14:21:25.365+05:30</createdTime>
<description>StandardMailing from CXF</description>
<organizationId>
  <id>1</id>
</organizationId>
<ownerId>
  <id>1</id>
</ownerId>
<useHeaderFromTemplate>false</useHeaderFromTemplate>
<ccAddress>test_cc@strongview.com</ccAddress>
<bccAddress>test_bcc@strongview.com</bccAddress>
<schedule>
  <startDateTime>2014-03-12T15:21:25.308+05:30</startDateTime>
  <recurrence>
    <endAfterXMailings>100</endAfterXMailings>
    <dailyRecurrence>
      <interval>2</interval>
    </dailyRecurrence>
  </recurrence>
</schedule>
<eliminateDuplicates>true</eliminateDuplicates>
<includedTargetId>
  <id>1801</id>

```

```

        </includedTargetId>
        <seedListId>
            <id>1001</id>
        </seedListId>
        <suppressionListId>
            <id>1100</id>
        </suppressionListId>
        <minRecipientLimit>2</minRecipientLimit>
        <maxRecipientLimit>123</maxRecipientLimit>
        <notificationOption>
            <recipientAddresses>test1_test@strongview.com</recipientAddresses>
            <recipientAddresses>test2_test@strongview.com</recipientAddresses>
            <recipientAddresses>test3_test@strongview.com</recipientAddresses>
            <notifyStateChange>true</notifyStateChange>
            <notifyTargetingCompletion>true</notifyTargetingCompletion>
        </notificationOption>
        <preTargetingPeriod>5</preTargetingPeriod>
    </baseObject>
</getResponse>
</getResponse>
</soap:Body>
</soap:Envelope>

```

Standard Mailing List Request (v2)

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

```

```

    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssesecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssesecurity-utility-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken wsu:Id="UsernameToken-6255C29619274FAEDA13946313601301">
        <wsse:Username>admin</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">admin</wsse:Password>
      </wsse:UsernameToken>
      <OrganizationToken xmlns="http://www.strongmail.com/services/v2/schema">
        <organizationName>admin</organizationName>
        <subOrganizationId>
          <id>1</id>
        </subOrganizationId>
      </OrganizationToken>
    </wsse:Security>
  </SOAP-ENV:Header>
  <soap:Body>
    <list xmlns="http://www.strongmail.com/services/v2/schema">
      <filter xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="MailingFilter">
        <isAscending>true</isAscending>
        <pageNumber>0</pageNumber>
        <recordsPerPage>10</recordsPerPage>
        <maxRecordsPerPage>200</maxRecordsPerPage>
      </filter>
    </list>
  </soap:Body>
</soap:Envelope>

```

Standard Mailing List Response (v2)

```

/*
* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *
* *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* *
* Distributing this source code outside your organization *
* is strictly prohibited. *

```

```

*
* ----- *
*/
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <listResponse xmlns="http://www.strongmail.com/services/v2/schema">
      <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="MailingId">
        <id>500</id>
      </objectId>
      <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="MailingId">
        <id>701</id>
      </objectId>
    </listResponse>
  </soap:Body>
</soap:Envelope>

```

Transactional Mailing Create Request (v2)

```

/*
* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *
* *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* *
* Distributing this source code outside your organization *
* is strictly prohibited. *
* *
* ----- *
*/
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken wsu:Id="UsernameToken-8DC6F0E182FEA86DA2139461428531132">
        <wsse:Username>admin</wsse:Username>
      </wsse:UsernameToken>
    </wsse:Security>
  </SOAP-ENV:Header>
  <body>
    <listResponse xmlns="http://www.strongmail.com/services/v2/schema">
      <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="MailingId">
        <id>500</id>
      </objectId>
      <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="MailingId">
        <id>701</id>
      </objectId>
    </listResponse>
  </body>
</soap:Envelope>

```

```

    <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">admin</wsse:Password>
  </wsse:UsernameToken>
  <OrganizationToken xmlns="http://www.strongmail.com/services/v2/schema">
    <organizationName>admin</organizationName>
    <subOrganizationId>
      <id>1</id>
    </subOrganizationId>
  </OrganizationToken>
</wsse:Security>
</SOAP-ENV:Header>
<soap:Body>
  <create xmlns="http://www.strongmail.com/services/v2/schema">
    <baseObject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="TransactionalMailing">
      <bodyEncoding>BASE_64</bodyEncoding>
      <isApproved>true</isApproved>
      <isCompliant>true</isCompliant>
      <fieldDelimiter>::</fieldDelimiter>
      <format>HTML</format>
      <headerEncoding>EIGHT_BIT</headerEncoding>
      <name>Transaction_Mailing_CURL</name>
      <priority>NORMAL</priority>
      <outputBodyCharSet>UTF-8</outputBodyCharSet>
      <outputHeaderCharSet>ASCII</outputHeaderCharSet>
      <rowDelimiter></rowDelimiter>
      <templateId xsi:type="TemplateId">
        <id>1701</id>
      </templateId>
      <type>TRANSACTIONAL</type>
      <description>TransactionalMailing from CXF</description>
      <useHeaderFromTemplate>false</useHeaderFromTemplate>
      <targetId>
        <id>1401</id>
      </targetId>
      <notificationOption>
        <recipientAddresses>notify_name@t.com</recipientAddresses>
        <notifyStateChange>true</notifyStateChange>
      </notificationOption>
      <subject>mailing subject</subject>
      <fromAddressId>
        <id>2902</id>
      </fromAddressId>
      <fromName>From Name</fromName>
      <header>mailing header</header>
    </baseObject>
  </create>
</soap:Body>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

```

        <bounceAddressId>
            <id>2900</id>
        </bounceAddressId>
        <replyAddressId>
            <id>2902</id>
        </replyAddressId>
    </baseObject>
</create>
</soap:Body>
</soap:Envelope>

```

Transactional Mailing Create Response (v2)

```

/*
 * -----
 * Selligent, Inc.
 *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved.
 *
 * Visit http://www.strongview.com for more information.
 *
 * You may incorporate this Source Code in your application only
 * if you own a valid license from Selligent, Inc.
 * and the copyright notices are not removed from the source code.
 *
 * Distributing this source code outside your organization
 * is strictly prohibited.
 * -----
 */
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <createResponse xmlns="http://www.strongmail.com/services/v2/schema">
      <success>true</success>
      <fault xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
      <createResponse>
        <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="TransactionalMailingId">
          <id>2200</id>
        </objectId>
      </createResponse>
    </createResponse>
  </soap:Body>
</soap:Envelope>

```

Transactional Mailing Get Request (v2)

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <wsse:Security mustUnderstand="1" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
        <wsse:Username>admin</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">admin</wsse:Password>
      </wsse:UsernameToken>
      <OrganizationToken xmlns="http://www.strongmail.com/services/v2/schema">
        <organizationName>admin</organizationName>
      </OrganizationToken>
    </wsse:Security>
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <get xmlns="http://www.strongmail.com/services/v2/schema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="TransactionalMailingId">
        <id>2200</id>
      </objectId>
    </get>
  </s:Body>

```



```
</soap:Envelope>
```

Transactional Mailing Get Response (v2)

```
/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getResponse xmlns="http://www.strongmail.com/services/v2/schema">
      <success>true</success>
      <fault xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
      <getResponse>
        <baseObject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="TransactionalMailing">
          <modifiedTime>2014-03-12T14:21:25.350+05:30</modifiedTime>
          <objectId xsi:type="TransactionalMailingId">
            <id>2200</id>
          </objectId>
          <version>0</version>
          <bodyEncoding>BASE_64</bodyEncoding>
          <isApproved>true</isApproved>
          <isCompliant>true</isCompliant>
          <fieldDelimiter>::</fieldDelimiter>
          <format>HTML</format>
          <headerEncoding>EIGHT_BIT</headerEncoding>
          <name>Transaction_Mailing_CURL</name>
          <priority>NORMAL</priority>
          <outputBodyCharSet>UTF-8</outputBodyCharSet>
          <outputHeaderCharSet>ASCII</outputHeaderCharSet>
```

```

<rowDelimiter></rowDelimiter>
<serverErrorCode>0</serverErrorCode>
<status>EDITING</status>
<templateId xsi:type="TemplateId">
  <id>1701</id>
</templateId>
<type>TRANSACTIONAL</type>
<createdTime>2014-03-12T14:21:25.365+05:30</createdTime>
<description>TransactionalMailing from CXF</description>
<organizationId>
  <id>1</id>
</organizationId>
<ownerId>
  <id>1</id>
</ownerId>
<useHeaderFromTemplate>false</useHeaderFromTemplate>
<mailingConfig>Mailing.2200.4357</mailingConfig>
<recordStructure>id</recordStructure>
<recordStructure>email_address</recordStructure>
<recordStructure>name</recordStructure>
<targetId>
  <id>1401</id>
</targetId>
<notificationOption>
  <recipientAddresses>notify_name@t.com</recipientAddresses>
  <notifyStateChange>true</notifyStateChange>
</notificationOption>
<mailingClassId>
  <id>1</id>
</mailingClassId>
<subject>mailing subject</subject>
<fromAddressId>
  <id>2901</id>
</fromAddressId>
<fromName>From Name</fromName>
<header>mailing header</header>
<bounceAddressId>
  <id>2900</id>
</bounceAddressId>
<replyAddressId>
  <id>2902</id>
</replyAddressId>
</baseObject>
</getResponse>
</getResponse>

```

```

    </soap:Body>
</soap:Envelope>

```

Template Create Request (v2)

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken wsu:Id="UsernameToken-8DC6F0E182FEA86DA2139461428485424">
        <wsse:Username>admin</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">admin</wsse:Password>
      </wsse:UsernameToken>
      <OrganizationToken xmlns="http://www.strongmail.com/services/v2/schema">
        <organizationName>admin</organizationName>
      </OrganizationToken>
    </wsse:Security>
  </SOAP-ENV:Header>
  <soap:Body>
    <create xmlns="http://www.strongmail.com/services/v2/schema">
      <baseObject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="Template">
        <attachmentId>
          <id>1501</id>
        </attachmentId>
        <description>Template from CURL</description>
      </baseObject>
    </create>
  </soap:Body>
</soap:Envelope>

```

```

<bodyEncoding>BASE_64</bodyEncoding>
<contentBlockId>
  <id>1401</id>
</contentBlockId>
<headerEncoding>EIGHT_BIT</headerEncoding>
<isApproved>true</isApproved>
<messagePart>
  <content>
    <html>
      <body>Hello, This is a simple HTML message with two
links<br/>This is <a href="http://www.somedomain.com/link1">Link
1</a><br/>This is <a href="http://www.somedomain.com/link2">Link
2</a><br/>This is <a href="#">Link 3</a>
      </body></html>
    </content>
    <format>HTML</format>
    <isXsl>false</isXsl>
    <namedLinks>
      <name>link1</name>
      <url>http://www.somedomain.com/link1</url>
      <linkId>2000</linkId>
      <tagType>CLICK</tagType>
      <tagFormat>Obfuscated</tagFormat>
    </namedLinks>
    <namedLinks>
      <name>link2</name>
      <url>http://www.somedomain.com/link2</url>
      <linkId>2001</linkId>
      <tagType>UNSUB</tagType>
      <tagFormat>Obfuscated</tagFormat>
    </namedLinks>
    <namedLinks>
      <name>link3</name>
      <url>#</url>
      <linkId>2002</linkId>
      <tagType>FBRECOMMEND</tagType>
      <tagFormat>Obfuscated</tagFormat>
      <trackingTagProperties>
        <name>pagettl</name>
        <value>test123</value>
      </trackingTagProperties>
      <trackingTagProperties>
        <name>posturl</name>
      </trackingTagProperties>
    </namedLinks>
  </messagePart>

```

```

</messagePart>
<outputBodyCharSet>ASCII</outputBodyCharSet>
<outputHeaderCharSet>UTF-8</outputHeaderCharSet>
<name>Template_XML_Request</name>
<linkTracking>Full</linkTracking>
<bounceAddressId>
  <id>2900</id>
</bounceAddressId>
<fromAddressId>
  <id>2901</id>
</fromAddressId>
<fromName>From Name</fromName>
<header>CXF template header</header>
<replyAddressId>
  <id>2902</id>
</replyAddressId>
<subject>CXF subject</subject>
<openTracking>true</openTracking>
</baseObject>
</create>
</soap:Body>
</soap:Envelope>

```

Template Create Response (v2)

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>

```

```

<createResponse xmlns="http://www.strongmail.com/services/v2/schema">
  <success>true</success>
  <fault xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true" />
  <createResponse>
    <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="TemplateId">
      <id>1800</id>
    </objectId>
  </createResponse>
</createResponse>
</soap:Body>
</soap:Envelope>

```

Template Get Request (v2)

```

/*
 * -----
 * Selligent, Inc.
 *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved.
 *
 * Visit http://www.strongview.com for more information.
 *
 * You may incorporate this Source Code in your application only
 * if you own a valid license from Selligent, Inc.
 * and the copyright notices are not removed from the source code.
 *
 * Distributing this source code outside your organization
 * is strictly prohibited.
 * -----
 */
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken wsu:Id="UsernameToken-6255C29619274FAEDA13946313651284">
        <wsse:Username>admin</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">admin</wsse:Password>
      </wsse:UsernameToken>
      <OrganizationToken xmlns="http://www.strongmail.com/services/v2/schema">
        <organizationName>admin</organizationName>
        <subOrganizationId>

```

```

        <id>1</id>
    </subOrganizationId>
</OrganizationToken>
</wsse:Security>
</SOAP-ENV:Header>
<soap:Body>
    <get xmlns="http://www.strongmail.com/services/v2/schema">
        <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="TemplateId">
            <id>1700</id>
        </objectId>
    </get>
</soap:Body>
</soap:Envelope>

```

Template Get Response (v2)

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <getResponse xmlns="http://www.strongmail.com/services/v2/schema">
            <success>true</success>
            <fault xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true" />
            <getResponse>
                <baseObject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="Template">
                    <modifiedTime>2014-03-12T14:20:20.542+05:30</modifiedTime>
                    <objectId xsi:type="TemplateId">

```

```

        <id>1700</id>
    </objectId>
    <version>0</version>
    <attachmentId>
        <id>1501</id>
    </attachmentId>
    <createdTime>2014-03-12T14:20:20.542+05:30</createdTime>
    <description>Template from CURL</description>
    <bodyEncoding>BASE_64</bodyEncoding>
    <contentBlockId>
        <id>1401</id>
    </contentBlockId>
    <headerEncoding>EIGHT_BIT</headerEncoding>
    <isApproved>true</isApproved>
    <messagePart>
        <content>
            &lt;html>
                &lt;body>Hello, This is a simple HTML message with two links&lt;br/>
                    This is &lt;a
href="##CLICKTAG##2000##ArgDelimiter####ArgDelimiter####ArgDelimiter##http://www.somedoma
in.com/link1">Link 1&lt;/a>&lt;br/>
                    This is &lt;a
href="##UNSUBSCRIBETAG##http://www.somedomain.com/link2">Link 2&lt;/a>&lt;br/>This is
&lt;a class="SM_FACEBOOKRECOMMENDBUTTON"
href="##FBRECOMMENDTAG##2002##ArgDelimiter##eu=##_MS_ORG_
ID####ArgDelimiter####ArgDelimiter####ArgDelimiter###">Link 3&lt;/a>
                &lt;/body>
            &lt;/html>
        </content>
        <format>HTML</format>
        <isXsl>false</isXsl>
        <namedLinks>
            <name>link1</name>
            <url>http://www.somedomain.com/link1</url>
            <linkId>2000</linkId>
            <tagType>CLICK</tagType>
            <tagFormat>Full</tagFormat>
        </namedLinks>
        <namedLinks>
            <name></name>
            <url>http://www.somedomain.com/link2</url>
            <linkId>2001</linkId>
            <tagType>UNSUB</tagType>
            <tagFormat>Full</tagFormat>
        </namedLinks>
        <namedLinks>

```



```

        <name>link3</name>
        <url>#</url>
        <linkId>2002</linkId>
        <tagType>FBRECOMMEND</tagType>
        <tagFormat>Full</tagFormat>
        <trackingTagProperties>
            <name>pagett1</name>
            <value>test123</value>
        </trackingTagProperties>
        <trackingTagProperties>
            <name>posturl</name>
            <value>##ENVIEWINBROWSSERTAG##2001##ArgDelimiter##eu=##_MS_ORG_
ID##</value>
        </trackingTagProperties>
    </namedLinks>
</messagePart>
<outputBodyCharSet>ASCII</outputBodyCharSet>
<outputHeaderCharSet>UTF-8</outputHeaderCharSet>
    <name>Template_XML_Request</name>
    <organizationId>
        <id>1</id>
    </organizationId>
    <ownerId>
        <id>1</id>
    </ownerId>
    <linkTracking>Full</linkTracking>

    <bounceAddressId>
        <id>2900</id>
    </bounceAddressId>
    <fromAddressId>
        <id>2901</id>
    </fromAddressId>
    <fromName>From Name</fromName>
    <header>CXF template header</header>
    <replyAddressId>
        <id>2902</id>
N    </replyAddressId>

    <subject>CXF subject</subject>
    <openTracking>true</openTracking>
</baseObject>
</getResponse>
</getResponse>
</soap:Body>
</soap:Envelope>

```

Template List Request (v2)

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="UsernameToken-
14858725">
        <wsse:Username xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">admin</wsse:Username>
        <wsse:Password xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">admin</wsse:Password>
      </wsse:UsernameToken>
      <OrganizationToken xmlns="http://www.strongmail.com/services/v2/schema">
        <organizationName>admin</organizationName>
        <subOrganizationId>
          <id>1</id>
        </subOrganizationId>
      </OrganizationToken>
    </wsse:Security>
  </SOAP-ENV:Header>
  <soap:Body>
    <list xmlns="http://www.strongmail.com/services/v2/schema">
      <filter xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="TemplateFilter">

```

```

        <isAscending>true</isAscending>
        <pageNumber>0</pageNumber>
        <recordsPerPage>10</recordsPerPage>
        <maxRecordsPerPage>200</maxRecordsPerPage>
    </filter>
</list>
</soap:Body>
</soap:Envelope>

```

Template List Response (v2)

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <listResponse xmlns="http://www.strongmail.com/services/v2/schema">
      <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="TemplateId">
        <id>100</id>
      </objectId>
      <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="TemplateId">
        <id>200</id>
      </objectId>
      <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="TemplateId">
        <id>300</id>
      </objectId>
      <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="TemplateId">

```

```

        <id>500</id>
    </objectId>
</listResponse>
</soap:Body>
</soap:Envelope>

```

Template Update Request (v2)

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * ----- *
 */

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken wsu:Id="UsernameToken-8DC6F0E182FEA86DA2139461428485424">
        <wsse:Username>admin</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">admin</wsse:Password>
      </wsse:UsernameToken>
      <OrganizationToken xmlns="http://www.strongmail.com/services/v2/schema">
        <organizationName>admin</organizationName>
      </OrganizationToken>
    </wsse:Security>
  </SOAP-ENV:Header>
  <soap:Body>
    <create xmlns="http://www.strongmail.com/services/v2/schema">
      <baseObject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="Template">
        <objectId>

```

```

        <id>1800</id>
    </objectId>
    <version>0</version>
    <attachmentId>
        <id>1501</id>
    </attachmentId>
    <description>Template from CURL update</description>
    <bodyEncoding>BASE_64</bodyEncoding>
    <contentBlockId>
        <id>1401</id>
    </contentBlockId>
    <headerEncoding>EIGHT_BIT</headerEncoding>
    <isApproved>true</isApproved>
    <messagePart>
        <content>&lt;html&gt;&lt;body&gt;Hello, This is a simple HTML message with two
links&lt;br/&gt;This is &lt;a
href="http://www.somedomain.com/link1"&gt;Link 1&lt;/a&gt;&lt;br/&gt;This is &lt;a
href="http://www.somedomain.com/link2"&gt;Link 2&lt;/a&gt;&lt;br/&gt;This
is &lt;a href="#"&gt;Link 3&lt;/a&gt; &lt;/body&gt;&lt;/html&gt;</content>
        <format>HTML</format>
        <isXsl>false</isXsl>
        <namedLinks>
            <name>link1</name>
            <url>http://www.somedomain.com/link1</url>
            <linkId>2000</linkId>
            <tagType>CLICK</tagType>
            <tagFormat>Obfuscated</tagFormat>
        </namedLinks>
        <namedLinks>
            <name>link2</name>
            <url>http://www.somedomain.com/link2</url>
            <linkId>2001</linkId>
            <tagType>UNSUB</tagType>
            <tagFormat>Obfuscated</tagFormat>
        </namedLinks>
        <namedLinks>
            <name>link3</name>
            <url>#</url>
            <linkId>2002</linkId>
            <tagType>FBRECOMMEND</tagType>
            <tagFormat>Obfuscated</tagFormat>
            <trackingTagProperties>
                <name>pagett1</name>
                <value>test123</value>
            </trackingTagProperties>
    </messagePart>

```

```

        <trackingTagProperties>
            <name>posturl</name>
        </trackingTagProperties>
    </namedLinks>
</messagePart>
<outputBodyCharSet>ASCII</outputBodyCharSet>
<outputHeaderCharSet>UTF-8</outputHeaderCharSet>
<name>Template_XML_Request_update</name>
<linkTracking>Full</linkTracking>
<bounceAddressId>
    <id>2900</id>
</bounceAddressId>
<fromAddressId>
    <id>2901</id>
</fromAddressId>
<fromName>From Name</fromName>
<header>CXF template header</header>
<replyAddressId>
    <id>2902</id>
</replyAddressId>
<subject>CXF subject</subject>
<openTracking>true</openTracking>
</baseObject>
</create>
</soap:Body>
</soap:Envelope>

```

Template Update Response - Success (v2)

```

/*
 * -----
 * Selligent, Inc.
 *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved.
 *
 * Visit http://www.strongview.com for more information.
 *
 * You may incorporate this Source Code in your application only
 * if you own a valid license from Selligent, Inc.
 * and the copyright notices are not removed from the source code.
 *
 * Distributing this source code outside your organization
 * is strictly prohibited.
 *
 */

```

```

* ----- *
*/
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <updateResponse xmlns="http://www.strongmail.com/services/v2/schema">
      <success>true</success>
      <fault xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
      <updateResponse>
        <success>true</success>
      </updateResponse>
    </updateResponse>
  </soap:Body>
</soap:Envelope>

```

Template Update Response - Failure (v2)

```

/*
* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *
* *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* *
* Distributing this source code outside your organization *
* is strictly prohibited. *
* *
* ----- *
*/
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <updateResponse xmlns="http://www.strongmail.com/services/v2/schema">
      <success>false</success>
      <fault xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="StaleObjectFaultDetail">
        <faultCode>STALE_OBJECT</faultCode>
        <faultMessage>update operation found stale object: 1800</faultMessage>
      </fault>
    <updateResponse>
      <success>false</success>

```

```

    </updateResponse>
  </updateResponse>
</soap:Body>
</soap:Envelope>

```

User Get Request (v2)

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * ----- *
 */
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken wsu:Id="UsernameToken-6255C29619274FAEDA13946313651284">
        <wsse:Username>admin</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">admin</wsse:Password>
      </wsse:UsernameToken>
      <OrganizationToken xmlns="http://www.strongmail.com/services/v2/schema">
        <organizationName>admin</organizationName>
        <subOrganizationId>
          <id>1</id>
        </subOrganizationId>
      </OrganizationToken>
    </wsse:Security>
  </SOAP-ENV:Header>
  <soap:Body>
    <get xmlns="http://www.strongmail.com/services/v2/schema">
      <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="UserId">

```



```

        <id>1</id>
    </objectId>
</get>
</soap:Body>
</soap:Envelope>

```

User Get Response (v2)

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * ----- *
 */
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getResponse xmlns="http://www.strongmail.com/services/v2/schema">
      <success>true</success>
      <fault xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true" />
      <getResponse>
        <baseObject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="User">
          <modifiedTime>2012-10-14T19:26:52.375-07:00</modifiedTime>
          <objectId xsi:type="UserId"><id>1</id></objectId>
          <version>2</version>
          <description>Default Super User</description>
          <email>admin@strongview.com</email>
          <firstName>Admin</firstName>
          <isAdmin>true</isAdmin>
          <isSuperUser>true</isSuperUser>
          <lastName>Emp</lastName>
          <login>admin</login>
          <organizationId>

```

```

        <id>1</id>
      </organizationId>
    </baseObject>
  </getResponse>
</getResponse>
</soap:Body>
</soap:Envelope>

```

User List Request (v2)

```

/*
* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *
* *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* *
* Distributing this source code outside your organization *
* is strictly prohibited. *
* *
* ----- *
*/

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken wsu:Id="UsernameToken-6255C29619274FAEDA13946313601301">
        <wsse:Username>admin</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">admin</wsse:Password>
      </wsse:UsernameToken>
      <OrganizationToken xmlns="http://www.strongmail.com/services/v2/schema">
        <organizationName>admin</organizationName>
        <subOrganizationId>
          <id>1</id>
        </subOrganizationId>
      </OrganizationToken>
    </wsse:Security>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <UserListRequest v2="http://www.strongmail.com/services/v2/schema">
      <subOrganizationId>
        <id>1</id>
      </subOrganizationId>
    </UserListRequest>
  </SOAP-ENV:Body>
</soap:Envelope>

```

```

</SOAP-ENV:Header>
<soap:Body>
<list xmlns="http://www.strongmail.com/services/v2/schema">
  <filter xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="UserFilter">
    <isAscending>true</isAscending>
    <pageNumber>0</pageNumber>
    <recordsPerPage>10</recordsPerPage>
    <maxRecordsPerPage>200</maxRecordsPerPage>
  </filter>
</list>
</soap:Body>
</soap:Envelope>

```

User List Response (v2)

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */

```

```

User List Response
-----
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <listResponse xmlns="http://www.strongmail.com/services/v2/schema">
      <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="UserId">
        <id>1</id>
      </objectId>
    </listResponse>
  </soap:Body>
</soap:Envelope>

```

Mailing Create Request (v1)

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-utility-1.0.xsd"
wsu:Id="UsernameToken-20671040">
        <wsse:Username>admin</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">admin</wsse:Password>
      </wsse:UsernameToken>
      <OrganizationToken xmlns="http://www.strongmail.com/services/2009/03/02/schema">
        <organizationName>admin</organizationName>
      </OrganizationToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <create xmlns="http://www.strongmail.com/services/2009/03/02/schema">
      <baseObject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="StandardMailing">
        <attachmentId>
          <id>3300</id>
        </attachmentId>
        <bodyEncoding>BASE64</bodyEncoding>
        <bounceAddressId>

```

```

    <id>98</id>
  </bounceAddressId>
  <isApproved>true</isApproved>
  <isCompliant>true</isCompliant>
  <contentBlockId>
    <id>4100</id>
  </contentBlockId>
  <fieldDelimiter>::</fieldDelimiter>
  <format>HTML</format>
  <fromAddressId>
    <id>97</id>
  </fromAddressId>
  <fromName>From Name</fromName>
  <headerEncoding>EIGHT_BIT</headerEncoding>
  <header>mailing header</header>
  <name>cxf_stmailing12345</name>
  <priority>NORMAL</priority>
  <outputBodyCharSet>UTF-8</outputBodyCharSet>
  <outputHeaderCharSet>ASCII</outputHeaderCharSet>
  <replyAddressId>
    <id>99</id>
  </replyAddressId>
  <rowDelimiter>
</rowDelimiter>
  <subject>mailing subject</subject>
  <templateId>
    <id>3600</id>
  </templateId>
  <type>RECURRING</type>
  <description>StandardMailing from CXF</description>
  <schedule>
    <startDateTime>2014-04-25T14:06:23.825+05:30</startDateTime>
    <recurrence>
      <endAfterXMailings>100</endAfterXMailings>
      <dailyRecurrence>
        <interval>2</interval>
      </dailyRecurrence>
    </recurrence>
  </schedule>
  <eliminateDuplicates>true</eliminateDuplicates>
  <includedTargetId>
    <id>5700</id>
  </includedTargetId>
  <seedListId>
    <id>3200</id>

```

```

        </seedListId>
        <suppressionListId>
            <id>3300</id>
        </suppressionListId>
    </baseObject>
</create>
</soap:Body>
</soap:Envelope>

```

Mailing Create Response (v1)

```

/*
* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *
* *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* *
* Distributing this source code outside your organization *
* is strictly prohibited. *
* *
* ----- *
*/
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getResponse xmlns="http://www.strongmail.com/services/2009/03/02/schema">
      <success>true</success>
      <fault xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
      <createResponse>
        <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="StandardMailingId">
          <id>2200</id>
        </objectId>
      </createResponse>
    </createResponse>
  </soap:Body>
</soap:Envelope>

```

Mailing Get Request (v1)

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="UsernameToken-
14858725">
        <wsse:Username xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">admin</wsse:Username>
        <wsse:Password xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">admin</wsse:Password>
      </wsse:UsernameToken>
      <OrganizationToken xmlns="http://www.strongmail.com/services/2009/03/02/schema">
        <organizationName>admin</organizationName>
      </OrganizationToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <get xmlns="http://www.strongmail.com/services/2009/03/02/schema">
      <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="MailingId">
        <id>100</id>
      </objectId>
    </get>
  </soap:Body>
</soap:Envelope>

```

```

    </soap:Body>
</soap:Envelope>

```

Mailing Get Response (v1)

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getResponse xmlns="http://www.strongmail.com/services/2009/03/02/schema">
      <success>true</success>
      <fault xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true" />
      <getResponse>
        <baseObject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="StandardMailing">
          <modifiedTime>2012-10-14T20:10:24.798-07:00</modifiedTime>
          <objectId xsi:type="StandardMailingId"><id>101</id></objectId>
          <version>7</version>
          <bodyEncoding>SEVEN_BIT</bodyEncoding>
          <bounceAddressId><id>100</id></bounceAddressId>
          <isApproved>true</isApproved>
          <isCompliant>true</isCompliant>
          <fieldDelimiter>::</fieldDelimiter>
          <format>HTML</format>
          <fromAddressId><id>200</id></fromAddressId>
          <fromName>Dummy From Name</fromName>
          <headerEncoding>SEVEN_BIT</headerEncoding>
          <lastGoodStatus>INITIATING_SAVING</lastGoodStatus>
          <mailingClassId><id>1</id></mailingClassId>

```



```

    <name>m1</name>
    <priority>NORMAL</priority>
    <outputBodyCharSet>UTF-8</outputBodyCharSet>
    <outputHeaderCharSet>UTF-8</outputHeaderCharSet>
    <replyAddressId><id>101</id></replyAddressId>
    <rowDelimiter>_sm_row_delim_
  </rowDelimiter>
  <serverErrorCode>0</serverErrorCode>
  <status>COMPLETED</status>
  <subject>Sample template</subject>
  <templateId><id>100</id></templateId>
  <type>ONE_TIME</type>
  <createTime>2012-10-14T20:09:18.610-07:00</createTime>
  <description></description>
  <organizationId><id>1</id></organizationId>
  <ownerId><id>1</id></ownerId>
  <eliminateDuplicates>true</eliminateDuplicates>
  <includedTargetId><id>100</id></includedTargetId>
</baseObject>
</getResponse>
</getResponse>
</soap:Body>
</soap:Envelope>

```

Mailing List Request (v1)

```

/*
 * ----- *
 * Selligent, Inc. *
 * *
 * Copyright © 2016 Selligent, Inc.. - All rights reserved. *
 * *
 * Visit http://www.strongview.com for more information. *
 * *
 * You may incorporate this Source Code in your application only *
 * if you own a valid license from Selligent, Inc. *
 * and the copyright notices are not removed from the source code. *
 * *
 * Distributing this source code outside your organization *
 * is strictly prohibited. *
 * *
 * ----- *
 */
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

```

```

<soap:Header>
  <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" soap:mustUnderstand="1">
    <wsse:UsernameToken xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="UsernameToken-
14858725">
      <wsse:Username xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">admin</wsse:Username>
      <wsse:Password xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">admin</wsse:Password>
    </wsse:UsernameToken>
    <OrganizationToken xmlns="http://www.strongmail.com/services/2009/03/02/schema">
      <organizationName>admin</organizationName>
    </OrganizationToken>
  </wsse:Security>
</soap:Header>
<soap:Body>
  <list xmlns="http://www.strongmail.com/services/2009/03/02/schema">
    <filter xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="MailingFilter">
      <isAscending>true</isAscending>
      <pageNumber>0</pageNumber>
      <recordsPerPage>10</recordsPerPage>
      <maxRecordsPerPage>200</maxRecordsPerPage>
    </filter>
  </list>
</soap:Body>
</soap:Envelope>

```

Mailing List Response (v1)

```

/*
* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *
* *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* *
* Distributing this source code outside your organization *

```

```

* is strictly prohibited. *
* *
* ----- *
*/
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <listResponse xmlns="http://www.strongmail.com/services/2009/03/02/schema">
      <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="MailingId">
        <id>101</id>
      </objectId>
      <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="MailingId">
        <id>200</id>
      </objectId>
    </listResponse>
  </soap:Body>
</soap:Envelope>

```

Template Create Request (v1)

```

/*
* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *
* *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* *
* Distributing this source code outside your organization *
* is strictly prohibited. *
* *
* ----- *
*/
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" soap:mustUnderstand="1">

```

```

    <wsse:UsernameToken xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="UsernameToken-
14858725">
        <wsse:Username xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">admin</wsse:Username>
        <wsse:Password xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">admin</wsse:Password>
    </wsse:UsernameToken>
    <OrganizationToken xmlns="http://www.strongmail.com/services/2009/03/02/schema">
        <organizationName>admin</organizationName>
    </OrganizationToken>
</wsse:Security>
</soap:Header>
<soap:Body>
    <create xmlns="http://www.strongmail.com/services/2009/03/02/schema">
        <baseObject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="Template">
            <description>A Sample Template created via the Message Studio API sample
code</description>
            <bodyEncoding>SEVEN_BIT</bodyEncoding>
            <bounceAddressId><id>17300</id></bounceAddressId>
            <fromAddressId><id>17301</id></fromAddressId>
            <headerEncoding>SEVEN_BIT</headerEncoding>
            <isApproved>true</isApproved>
            <messagePart>
                <content>&lt;html>&lt;head>Sample Content&lt;/head>&lt;body>Hi ##first_name##
##last_name##!&lt;br/>Welcome to our new users program! You can click &lt;a
href="http:www.yourdomain.com/newuser">here&lt;/a> to get a tour of our
services&lt;/body>&lt;/html></content>
                <format>HTML</format>
                <isXsl>false</isXsl>
            </messagePart>
            <outputBodyCharSet>UTF-8</outputBodyCharSet>
            <outputHeaderCharSet>UTF-8</outputHeaderCharSet>
            <name>Sample_Template_1350498788685</name>
            <replyAddressId><id>17302</id></replyAddressId>
        </baseObject>
    </create>
</soap:Body>
</soap:Envelope>

```

Template Create Response (v1)

/*

```

* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *
* *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* *
* Distributing this source code outside your organization *
* is strictly prohibited. *
* *
* ----- *
*/
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <createResponse xmlns="http://www.strongmail.com/services/2009/03/02/schema">
      <success>true</success>
      <fault xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true" />
      <createResponse>
        <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="TemplateId">
          <id>200</id>
        </objectId>
      </createResponse>
    </createResponse>
  </soap:Body>
</soap:Envelope>

```

Template Get Request (v1)

```

/*
* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *
* *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *

```

```

*                                                                 *
*   Distributing this source code outside your organization      *
*   is strictly prohibited.                                     *
*                                                                 *
* ----- *
*/
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="UsernameToken-
14858725">
        <wsse:Username xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">admin</wsse:Username>
        <wsse:Password xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">admin</wsse:Password>
      </wsse:UsernameToken>
      <OrganizationToken xmlns="http://www.strongmail.com/services/2009/03/02/schema">
        <organizationName>admin</organizationName>
      </OrganizationToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <get xmlns="http://www.strongmail.com/services/2009/03/02/schema">
      <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="TemplateId">
        <id>100</id>
      </objectId>
    </get>
  </soap:Body>
</soap:Envelope>

```

Template Get Response (v1)

```

/*
* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *
* *
* You may incorporate this Source Code in your application only *

```

```

* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* * *
* Distributing this source code outside your organization *
* is strictly prohibited. *
* * *
* ----- *
*/
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getResponse xmlns="http://www.strongmail.com/services/2009/03/02/schema">
      <success>true</success>
      <fault xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true" />
      <getResponse>
        <baseObject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="Template">
          <modifiedTime>2012-10-14T20:08:46.233-07:00</modifiedTime>
          <objectId xsi:type="TemplateId"><id>100</id></objectId>
          <version>8</version>
          <createdTime>2012-10-14T20:05:22.284-07:00</createdTime>
          <bodyEncoding>SEVEN_BIT</bodyEncoding>
          <bounceAddressId><id>100</id></bounceAddressId>
          <fromAddressId><id>200</id></fromAddressId>
          <fromName>Dummy From Name</fromName>
          <headerEncoding>SEVEN_BIT</headerEncoding>
          <isApproved>true</isApproved>
          <messagePart>
            <content>&lt;html>
&lt;head>
  &lt;title>&lt;/title>
&lt;/head>
&lt;body style="background-color: white;">
  &lt;br />
  &lt;p> Hi ##name##! We are glad that you subscribed. Your personal info is: ID: ##id##
and Name: ##name##&lt;br />
Content block token personalization - ##cbToken##&lt;br />
Rules personalization - ##SM_RULE_rule1##&lt;br /> &lt;br /> &lt;br />
clicktag - simple link &lt;a href="##ENCLICKTAG##2000##ArgDelimiter##
##ArgDelimiter####ArgDelimiter##www.google.com" target="_blank"> Google &lt;/a> to go to
the search page.&lt;br /> &lt;br />
shortclicktag - link with one param in it &lt;a
href="##ENCLICKTAG##2001##ArgDelimiter####ArgDelimiter####ArgDelimiter##www.yahoo.com?sp=
value"> Yahoo &lt;/a> to go to the search page.&lt;br /> &lt;br />
encodedclicktag - link with some params in it &lt;a
href="##ENCLICKTAG##2002##ArgDelimiter####ArgDelimiter####ArgDelimiter##www.hotmail.com?s
p=v1&amp;ap=v2"> Hotmail &lt;/a> to go to the search page.&lt;br />

```

```

    </body>
</html>

    </content>
    <format>HTML</format>
    <isXsl>false</isXsl>
    <namedLinks>
        <url>www.google.com</url>
        <linkId>2000</linkId>
        <trackingTag>ENCLICKTAG</trackingTag>
    </namedLinks>
    <namedLinks>
        <url>www.yahoo.com?sp=value</url>
        <linkId>2001</linkId>
        <trackingTag>ENCLICKTAG</trackingTag>
    </namedLinks>
    <namedLinks>
        <url>www.hotmail.com?sp=v1&amp;ap=v2</url>
        <linkId>2002</linkId>
        <trackingTag>ENCLICKTAG</trackingTag>
    </namedLinks>
    </messagePart>
    <outputBodyCharSet>UTF-8</outputBodyCharSet>
    <outputHeaderCharSet>UTF-8</outputHeaderCharSet>
    <name>tpl</name>
    <organizationId><id>1</id></organizationId>
    <ownerId><id>1</id></ownerId>
    <replyAddressId><id>101</id></replyAddressId>
    <subject>Sample template</subject>
    </baseObject>
    </getResponse>
</getResponse>
</soap:Body>
</soap:Envelope>

```

Template List Request (v1)

```

/*
* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *
* *

```



```

* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* *
* Distributing this source code outside your organization *
* is strictly prohibited. *
* *
* ----- *
*/
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="UsernameToken-
14858725">
        <wsse:Username xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">admin</wsse:Username>
        <wsse:Password xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">admin</wsse:Password>
      </wsse:UsernameToken>
      <OrganizationToken xmlns="http://www.strongmail.com/services/2009/03/02/schema">
        <organizationName>admin</organizationName>
      </OrganizationToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <list xmlns="http://www.strongmail.com/services/2009/03/02/schema">
      <filter xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="TemplateFilter">
        <isAscending>true</isAscending>
        <pageNumber>0</pageNumber>
        <recordsPerPage>10</recordsPerPage>
        <maxRecordsPerPage>200</maxRecordsPerPage>
      </filter>
    </list>
  </soap:Body>
</soap:Envelope>

```

Template List Response (v1)

```

/*
* ----- *
* Selligent, Inc. *

```

```

*                                                                    *
* Copyright © 2016 Selligent, Inc.. - All rights reserved.          *
*                                                                    *
* Visit http://www.strongview.com for more information.             *
*                                                                    *
* You may incorporate this Source Code in your application only      *
* if you own a valid license from Selligent, Inc.                   *
* and the copyright notices are not removed from the source code.   *
*                                                                    *
* Distributing this source code outside your organization           *
* is strictly prohibited.                                           *
*                                                                    *
* ----- *
*/
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <listResponse xmlns="http://www.strongmail.com/services/2009/03/02/schema">
      <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="TemplateId">
        <id>100</id>
      </objectId>
    </listResponse>
  </soap:Body>
</soap:Envelope>

```

User Get Request (v1)

```

/*
* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *
* *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* *
* Distributing this source code outside your organization *
* is strictly prohibited. *
* *
* ----- *
*/

```

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="UsernameToken-
14858725">
        <wsse:Username xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">admin</wsse:Username>
        <wsse:Password xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">admin</wsse:Password>
      </wsse:UsernameToken>
      <OrganizationToken xmlns="http://www.strongmail.com/services/2009/03/02/schema">
        <organizationName>admin</organizationName>
      </OrganizationToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <get xmlns="http://www.strongmail.com/services/2009/03/02/schema">
      <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="UserId">
        <id>1</id>
      </objectId>
    </get>
  </soap:Body>
</soap:Envelope>

```

User Get Response (v1)

```

/*
* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *
* *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* *
* Distributing this source code outside your organization *
* is strictly prohibited. *
* *

```

```

* ----- *
*/
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getResponse xmlns="http://www.strongmail.com/services/2009/03/02/schema">
      <success>true</success>
      <fault xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true" />
      <getResponse>
        <baseObject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="User">
          <modifiedTime>2012-10-14T19:26:52.375-07:00</modifiedTime>
          <objectId xsi:type="UserId"><id>1</id></objectId>
          <version>2</version>
          <description>Default Super User</description>
          <email>admin@strongview.com</email>
          <firstName>Admin</firstName>
          <isAdmin>true</isAdmin>
          <isSuperUser>true</isSuperUser>
          <lastName>Emp</lastName>
          <login>admin</login>
          <organizationId>
            <id>1</id>
          </organizationId>
        </baseObject>
      </getResponse>
    </getResponse>
  </soap:Body>
</soap:Envelope>

```

User List Request (v1)

```

/*
* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *
* *
* You may incorporate this Source Code in your application only *
* if you own a valid license from Selligent, Inc. *
* and the copyright notices are not removed from the source code. *
* *
* Distributing this source code outside your organization *

```

```

* is strictly prohibited. *
* *
* ----- *
*/

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" soap:mustUnderstand="1">
      <wsse:UsernameToken xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="UsernameToken-
14858725">
        <wsse:Username xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">admin</wsse:Username>
        <wsse:Password xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">admin</wsse:Password>
      </wsse:UsernameToken>
      <OrganizationToken xmlns="http://www.strongmail.com/services/2009/03/02/schema">
        <organizationName>admin</organizationName>
      </OrganizationToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <list xmlns="http://www.strongmail.com/services/2009/03/02/schema">
      <filter xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="UserFilter">
        <isAscending>true</isAscending>
        <pageNumber>0</pageNumber>
        <recordsPerPage>10</recordsPerPage>
        <maxRecordsPerPage>200</maxRecordsPerPage>
      </filter>
    </list>
  </soap:Body>
</soap:Envelope>

```

User List Response (v1)

```

/*
* ----- *
* Selligent, Inc. *
* *
* Copyright © 2016 Selligent, Inc.. - All rights reserved. *
* *
* Visit http://www.strongview.com for more information. *

```

```

*
* You may incorporate this Source Code in your application only
* if you own a valid license from Selligent, Inc.
* and the copyright notices are not removed from the source code.
*
* Distributing this source code outside your organization
* is strictly prohibited.
*
* -----
*/

```

User List Response

```

-----
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <listResponse xmlns="http://www.strongmail.com/services/2009/03/02/schema">
      <objectId xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="UserId">
        <id>1</id>
      </objectId>
    </listResponse>
  </soap:Body>
</soap:Envelope>

```

Index

~ A ~

- addGlobalUndeliverable 48
- addRecords 44
- addtrackingtags 45
- API clients
 - C# 22, 28
 - Java CXF 15
- API Code Samples
 - C# code 174
 - Java code 162
 - XML code 183
- API Error Handling 35
- API Resources 11
- API SOAP
 - About 7
 - Object Descriptions 41
 - Operations 47
 - txnSend() 149
- archiveMailing 54
- Attachments
 - SOAP API max size 37

~ C ~

- C# Clients 22
- cancel 45
- cancelRefreshRecords 44
- close 45

~ D ~

- dedupeRecords 44
- Dynamic Content
 - API category 45

~ E ~

- exportRecords 44

~ F ~

- fetchLinks 45

- findImport 44

~ G ~

- getGlobalUndeliverable 94
- getImportDetails 44
- getImportStatus 89, 92
- getProgramNodes 44, 96
- getRecords 44
- getStatistics 44
- getTxnEasInfo 46
- getTxnMailingHandle 46

~ I ~

- importContent 45
- importFtpFile 44, 112
- importMessagePart 45

~ J ~

- Java CXF Clients 15

~ L ~

- launch 45
- Limit on removeRecords 139

~ M ~

- Mailing Server Groups 42
- Max size Tx Mailling API 150

~ P ~

- pause 45
- personalize 45

~ R ~

- reactivateGlobalUndeliverable 134
- refreshRecords 44
- removeRecords 44
- removeRecords limit 139
- resume 45
- Ruby Clients 28

Rules

- API category 45

~ S ~

- schedule 45

SMS

- CLI create mailing 69

SOAP API 7

- Attachment size 37

- Object Descriptions 41

- startImport 44

~ T ~

- test 45

Transactional mailings

- Max size in API 150

- txnSend() 46, 149

~ U ~

- upsertRecords 44

~ V ~

- validateXsl 45