

STRONGVIEW

SOAP API V2

BEFORE AND AFTER CHANGES

5/2014

Overview

- SOAP API enhancements
 - New v2 endpoint
 - Deprecated old send API
 - Published personalizeOnly API
 - Enums to String Conversion for Types/States
 - Template Updates
 - Mailing Updates
- Compatibility and upgrade requirements

New v2 endpoint

v2 new path is available in 8.2 and above:

<https://<hostname>:4443/sm/services/mailing/v2>

Misc Enhancements

- Deprecated send API since everyone has already moved to the txSend
- Exposed personalizeOnly API as personalize() to generate dynamic preview. Originally was only available for internal developers. Feature will be available in *both* v1 (2009/03/02) and v2.
- Samples updated to support latest versions of:
 - Java (CXF)
 - PHP
 - Ruby
 - C#

SOAP API v2 Features

Enums converted to String

- DatabaseType
- DataSourceType
- DataSourceDataType
- DataSourceFieldType
- DataSourceOperationStatus
- DataSourceDedupeOption
- ImportType
- RecurringImportType
- TargetType
- TrackingTag
- OpenTag
- MessageFormat
- Message Type
- Mailing Type
- MailingStatus
- MailingPriority
- DailyOccurrence
- WeeklyOccurrence
- AssetExpiryTimeInterval
- ProgramStatus
- ProgramNodeType
- SystemAddressType
- DataSourceImportFrequency
- DataSourceImportMode
- StrongtoolOpenAs
- Encoding
- FaultCode

Remaining Enums

- AssignedRoleOrderBy
- AttachmentOrderBy
- CampaignOrderBy
- CharSet
- ComparisonOperation
- ContentBlockOrderBy
- DataSourceOrderBy
- DayOfWeek
- FilterArrayOperator
- FilterBooleanScalarOperator
- FilterIntegerScalarOperator
- FilterStringScalarOperator
- LogicalOperation
- MailingClassOrderBy
- OrganizationOrderBy
- ProgramOrderBy
- RecurringImportStatus
- RoleOrderBy
- RuleOrderBy
- SeedListOrderBy
- StrongtoolOrderBy
- SuppressionListOrderBy
- SystemAddressOrderBy
- TargetOrderBy
- UserOrderBy
- WebAnalyticsOrderBy

Template Updates

Tracking Tags – Specifications

- Tracking tags will be broken down into two parts: tagType (e.g. click, unsub, share, etc.) and tagFormat (e.g. none, encoded, etc.)

BEFORE

```
<xs:element name="trackingTag" type="tns:TrackingTag"/>
```

AFTER

```
<xs:element name="tagType" type="tns:TrackingTagType"/>
```

```
<xs:element name="tagFormat" type="tns:TrackingTagFormat" minOccurs="0"/>
```

- Applicable to tags in message template and content blocks
- Old “Short” format will now be mapped to Obfuscated

Tracking Tag Property Specification

- Tracking will be expanded from Share to include Facebook tags (Like, Recommend, Comment)

Tracking Tags – Property Specification

Field for TrackingTagProperty in NamedLink

BEFORE

```
<xs:element minOccurs="0" name="trackingTagProperties"  
type="tns:TrackingTagProperties"/>
```

AFTER

```
<xs:element maxOccurs="unbounded" name="trackingTagProperties"  
type="tns:NameValuePair"/>
```

Tag Type	Allowed Property Names
SHARE	imageurl, offerurl, desc, ttl
FBLIKE	posturl, pagettl
FBRECOMMEND	posturl, pagettl
FBCOMMENT	posturl, commentttl, pagettl

Mailing Updates

Mailings

- **Pre-targeting Options** - Represented in the UI as: “Start Targeting x hours before mailing starts” option which is available for Batch mailings for the “Scheduled” and “Recurring” launch types
<xs:element name =“preTargetingPeriod” type=“xs:integer” minOccurs=“0”>
- **Max/min recipient limits** – represents the value of the “Abort if fewer than x recipients” option in the UI
<xs:element name=“minRecipientLimit” type=“xs:integer” minOccurs=“0”/>
<xs:element name=“maxRecipientLimit” type=“xs:integer” minOccurs=“0”/>
- **Use header from template option** – represents the “User Header from Template” checkbox on mailing wizard Content tab
<xs:element name=“useHeaderFromTemplate” type=“xs:boolean”/>
- **CC/BCC specification** –present under Content tab > Advanced Options on Batch and Transactional mailing wizards.
<xs:element minOccurs=“0” name=“ccAddress” type=“xs:string”/>
<xs:element minOccurs=“0” name=“bccAddress” type=“xs:string”/>
- **Notification options**

Notification Options

The object definition for StandardMailing will be updated to contain the following field:

```
<xs:element minOccurs="0" name="notificationOption" type="tns:StandardMailingNotification"/>
```

The object definition for TransactionalMailing will be updated to contain the following field:

```
<xs:element minOccurs="0" name="notificationOption" type="tns: MailingNotification"/>
```

The notificationOption field will correspond to the notification options available in the Schedule tab of the Batch and Transactional mailing wizards in the UI.

Field in webservice schema	Field in mailing table
recipientAddresses	email_notification
notifyStateChange	notification_types
notifyTargetingCompletion	Notification_types

Compatibility & Upgrade

Compatibility & Upgrade

- Existing version (2009/03/02) will remain exactly the same for backwards compatibility.
- Customers interested in using v2 would need to:
 - Upgrade to 8.2 or above release
 - Re-write existing client code to use the new endpoint
 - Regenerate code off of the new WSDL/Schema
 - Update enumerations
 - Update Tracking tag and property changes
 - Update Open tag changes
 - Check EDS connection changes

Enumeration to String Transition

Example API client code that would need to be updated following the v2 transition from enum to string:

BEFORE

```
MessagePart htmlMessagePart = new htmlMessagePart();  
htmlMessagePart.setFormat(MessageFormat.HTML);
```

AFTER

```
MessagePart htmlMessagePart = new htmlMessagePart();  
htmlMessagePart.setFormat("HTML");
```

Tracking Tag Changes

Example API client code that would need to be updated following the v2 tracking tag changes:

BEFORE

```
link.setTrackingTag(TrackingTag.ENCLICKTAG);
```

AFTER

```
link.setTrackingTagType("CLICK");  
link.setTrackingTagFormat("Encoded");
```

*trackingTagFormat is optional and does not need to be set. If no specified, the value is inherited from the Link Tracking property for the template and will default to Obfuscated type if the template property is not specified.

Tracking Tag Property Changes

Example API client code that would need to be updated following the v2 tracking tag property changes:

BEFORE

```
link.setTrackingTag(TrackingTag.ENSHARETAG);  
TrackingTagProperties properties = new TrackingTagProperties();  
properties.setTitle("some title");  
properties.setDesc("some description");  
properties.setOfferUrl("http://offerurl.com");  
properties.setImageUrl("http://imageurl.com");  
link.setTrackingTagProperties(properties);
```


Tracking Tag Property Changes

AFTER

```
link.setTrackingTagType("SHARE");  
link.setTrackingTagFormat("Encoded");  
// Set title.  
NameValuePair title = new NameValuePair();  
title.setName("ttl"); // property name  
title.setValue("some title");  
link.getTrackingTagProperties().add(title);  
// Set description  
NameValuePair desc = new NameValuePair();  
desc.setName("desc"); // property name  
desc.setValue("some description");  
link.getTrackingTagProperties().add(desc);
```

Tracking Tag Property Changes

AFTER

```
// Set offer url.  
NameValuePair offerUrl = new NameValuePair();  
offerUrl.setName("offerurl"); // property name  
offerUrl.setValue("http://offerurl.com");  
link.getTrackingTagProperties().add(offerUrl);  
  
// Set image url.  
NameValuePair imageUrl = new NameValuePair();  
imageUrl.setName("imageurl"); // property name  
imageUrl.setValue("http://imageurl.com");  
link.getTrackingTagProperties().add(imageUrl);
```

OpenTag Changes

Example API client code that would need to be updated following the v2 open tag changes:

BEFORE

```
messagePart.setOpenTag(OpenTag.ENOPENTAG);
```

AFTER

```
template.setOpenTracking(true);
```

*Format will default to Obfuscated if no value is specified for this property.

EDS Connection Info Changes

Example API client code that would need to be updated following the v2 connection information changes – using connectionParameters:

BEFORE

```
ExternalDataSource.ConnectionInfo info =  
new ExternalDataSource.ConnectionInfo();  
info.setDatabaseName(dbName);  
// .. and set other properties in info  
eds.setConnectionInfo(info); // eds is of type ExternalDataSource
```

AFTER

```
ExternalDataSource.ConnectionInfo info =  
new ExternalDataSource.ConnectionInfo();  
ExternalDataSource.ConnectionInfo.ConnectionParameters params =  
new ExternalDataSource.ConnectionInfo.ConnectionParameters();  
params.setDatabaseName(dbName);  
// .. and set other properties in params  
info.setConnectionParameters(params);  
eds.setConnectionInfo(info); // eds is of type ExternalDataSource
```

EDS Connection Info Changes

Example API client code that would need to be updated following the v2 connection information changes – using jdbcUrl:

BEFORE

```
ExternalDataSource.ConnectionInfo info =  
new ExternalDataSource.ConnectionInfo();  
info.setDatabaseName(dbName);  
// .. and set other properties in info  
eds.setConnectionInfo(info); // eds is of type ExternalDataSource
```

AFTER

```
ExternalDataSource.ConnectionInfo info =  
new ExternalDataSource.ConnectionInfo();  
ExternalDataSource.ConnectionInfo.JdbcParameters jdbcParams =  
new ExternalDataSource.ConnectionInfo.JdbcParameters();  
jdbcParams.setUrl(jdbcUrl);  
// .. and set other properties in jdbcParams  
info.setJdbcParameters(jdbcParams);  
eds.setConnectionInfo(info); // eds is of type ExternalDataSource
```




THANK YOU