Oracle® Retail XBRi Cloud Services

Import Services
Release 18.1
F24058-02

August 2020

ORACLE

Oracle® Retail XBRi Cloud Services Import Services, Release 18.1

Primary Author:

Contributors:

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

The following restrictions and provisions only apply to the programs referred to in this section and licensed to you. You acknowledge that the programs may contain third party software (VAR applications) licensed to Oracle. Depending upon your product and its version number, the VAR applications may include:

(i) the **MicroStrategy** Components developed and licensed by MicroStrategy Services Corporation (MicroStrategy) of McLean, Virginia to Oracle and imbedded in the MicroStrategy for Oracle Retail Data Warehouse and MicroStrategy for Oracle Retail Planning & Optimization applications.

(ii) the **Wavelink** component developed and licensed by Wavelink Corporation (Wavelink) of Kirkland, Washington, to Oracle and imbedded in Oracle Retail Mobile Store Inventory Management.

(iii) the software component known as **Access Via™** licensed by Access Via of Seattle, Washington, and imbedded in Oracle Retail Signs and Oracle Retail Labels and Tags.

(iv) the software component known as **Adobe Flex™** licensed by Adobe Systems Incorporated of San Jose, California, and imbedded in Oracle Retail Promotion Planning & Optimization application.

You acknowledge and confirm that Oracle grants you use of only the object code of the VAR Applications. Oracle will not deliver source code to the VAR Applications to you. Notwithstanding any other term or condition of the agreement and this ordering document, you shall not cause or permit alteration of any VAR Applications. For purposes of this section, "alteration" refers to all alterations, translations, upgrades, enhancements, customizations or modifications of all or any portion of the VAR Applications including all reconfigurations, reassembly or reverse assembly, re-engineering or reverse engineering and recompilations or reverse compilations of the VAR Applications or any derivatives of the VAR Applications. You acknowledge that it shall be a breach of the agreement to utilize the relationship, and/or confidential information of the VAR Applications for purposes of competitive discovery.

The VAR Applications contain trade secrets of Oracle and Oracle's licensors and Customer shall not attempt, cause, or permit the alteration, decompilation, reverse engineering, disassembly or other reduction of the VAR Applications to a human perceivable form. Oracle reserves the right to replace, with functional equivalent software, any of the VAR Applications in future releases of the applicable program.

Contents

Send Us Your Comments

Oracle Retail XBRi Cloud Services, Import Services, Release 18.1

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document.

Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

> **Note:** Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the Online Documentation available on the Oracle Technology Network Web site. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: retail-doc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at www.oracle.com.

Preface

This Import Services Guide provides critical information about the processing and operating details of XBRi Cloud Services, including the following:

- Data Import
- Web Services
- System configuration settings
- Technical architecture

Audience

This guide is for:

- Systems administration and operations personnel
- Integrators and implementers

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Retail XBRi Cloud Services Release 18.1 documentation set:

- Oracle Retail XBRi Cloud Services Implementation Guide
- Oracle Retail XBRi Cloud Services Release Notes
- Oracle Retail XBRi Cloud Services Administration Guide
- Oracle Retail XBRi Cloud Services Online Help

Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

https://support.oracle.com

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received
- Screen shots of each step you take

Improved Process for Oracle Retail Documentation Corrections

Improved Process for Oracle Retail Documentation Corrections

To more quickly address critical corrections to Oracle Retail documentation content, Oracle Retail documentation may be republished whenever a critical correction is needed. For critical corrections, the republication of an Oracle Retail document may at times **not** be attached to a numbered software release; instead, the Oracle Retail document will simply be replaced on the Oracle Technology Network Web site, or, in the case of Data Models, to the applicable My Oracle Support Documentation container where they reside.

This process will prevent delays in making critical corrections available to customers. For the customer, it means that before you begin installation, you must verify that you have the most recent version of the Oracle Retail documentation set. Oracle Retail documentation is available on the Oracle Technology Network at the following URL:

`http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html`

An updated version of the applicable Oracle Retail document is indicated by Oracle part number, as well as print date (month and year). An updated version uses the same part number, with a higher-numbered suffix. For example, part number E123456-**02** is an updated version of a document with part number E123456-**01**.

If a more recent version of a document is available, that version supersedes all previous versions.

Oracle Retail Documentation on the Oracle Technology Network

Oracle Retail product documentation is available on the following web site:

`http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html`

(Data Model documents are not available through Oracle Technology Network. You can obtain them through My Oracle Support.)

Conventions

Navigate: This is a navigate statement. It tells you how to get to the start of the procedure and ends with a screen shot of the starting point and the statement "the Window Name window opens."

```
This is a code sample
    It is used to display examples of code
```

1
Introduction

The Import Services ETL process is designed to give customers flexibility over how POSLOG data is imported into their application and in managing and monitoring data once it is imported. The POSLOG import package handles sales and operational transactions from a POS system. It is integrated with the ETL Module. It manages the work of importing POSLOG data once the component import services are created and run. These components are explained in detail in the following sections:

2

Configuring Import Services

The Web service import process uses a collection of contracts designed to process data In order to import data, it is necessary to first create profiles, assign roles to the profiles, and set up the import workflow and performance logic. The process is configurable by Customer Administrators.

Before you are allowed to use any of the defined REST services, you must first log in and obtain a token that will allow access to the import services.

Logging into REST Web Services

Login sessions for REST web services are limited by default to 10, so it may be necessary to log out of sessions to allow other users to log in. See Logging Out of Rest Web Services

The Login web service is REST based and requires the following:

Protocol -HTTPS

Method - GET

URL - <XBRi Application system hostname>/analytics/rest/ext/v1/login?project=XBRI

Header Attributes

Authorization – Basic <Base64 Encoded username:password>

The username and password are from an XBRi account. The username and password must be delimited by a colon before encoding with base64. The username must be prefixed with XBR_

Example: XBR_myuser:myPassword

Expected Response - Assuming a HTTP 200 status comes back, the response body will contain one of two things: If the user details are invalid, the message: "Error connecting to object" Or, a plain text version of the access token. Any further communications with XBRi web services will require this token to verify access.

Getting back an HTTP 500 generally means the format of the submission is incorrect, or there's a problem with the server.

Logging Out of REST Web Services

Login sessions for REST web services are limited by default to 10, so it may be necessary to log out of sessions to allow other users to log in:

The Logout REST service requires the following:

Protocol -HTTPS

Method - GET

URL - <XBRi Application system hostname>/analytics/rest/ext/v1/logoff?token=[Bearer Token]

Header Attributes

Authorization - Basic <Security Token>

<Security Token> is made up of XBRi username ":" and XBRi user password, for Example:

XBR_myuser:myPassword

Base64 encoded, example result: WEJSX215dXNlcjpteVBhc3N3b3JkCg==

Expected Response:

HTTP Status 200

The components that comprise the import process are described in the following sections in the order in which they are executed.

ETL Import Profiles REST Service

This service is for maintaining import profiles. Administrator users configure profile properties using this service. This is the first step in the import process. After creating a profile, you assign roles to the profiles, so that only users with the assigned role for a profile can perform the actions associated with that role and profile. This is covered in the next section, Import Roles Rest Service.

Profiles allow each different import to have its own properties defined: These are the basic properties that can be defined:

Table 2-1: Import Profile Basic Properties

Property	Description
Profile Code	A mandatory unique identifier of the import.
Description	A user friendly description of what the import profile does. This property is optional
Module	Identifies what kind of import package this import can connect to. This field is mandatory. . The import package currently available is POSLOG.
Processing Threads	The number of processes that are able to extract data from the import queue. This will default to 1 if no value is provided. Note: In most cases, only one process should be needed, but in certain situations it may need to be increased. The maximum value across all profiles and staging processes allowed is 40. If you enter a number greater than 40, The ETL module will not start up.

Creating and Managing Import Profiles

In order to perform an import request, you must create an import profile. Without it, any request will be rejected. This is because secure access to a particular import profile is only allowed according to the roles associated with the profile. See the next section: Import Roles Rest Service for information on creating roles.

Creating a Profile

From a ReST point of view creating an import profile is a HTTP POST method. The full schema details can be found by looking at the Swagger output, which includes the schemas: <hostname:port>/ETLServices/swagger.json

Service URL: <hostname:port>/ETLServices/admin/v1/profiles

Table 2-2 Header Details

Name	Description
Content-Type	This can be one of two types, json or xml. The full expected string depending on the type, is either of the following: application/json application/xml
xbri-mst-token	The token from the login process. See Logging in to REST Web Services
Accept	This can be one of two types, json or xml. If not provided, will assume xml. Examples: application/json, application/xml

Body

The schema details mentioned above provides all the properties. Here is an example:

```
{
"profileCode": "POSLOG",
"description": "New POSLOG Import",
"module": "POSLOG",
"ProcessingThreads": 1
}
```

> **Note**: All properties are case sensitive, except for note fields such as Description.

Returned Values

When submitted (as per the Swagger notes) the following values are expected back:

HTTP 200 Success - plus return of record that was submitted.

HTTP 401 Unauthorized - the xbri-mst-token is missing, invalid or the user in question does not have admin access

HTTP 400 Bad Request - the parameters in the body don't match the schema.

HTTP 409 Conflict - the profile code already exists, use update instead.

HTTP 415 Unknown Media type - the Content-Type in the header block hasn't been provided.

HTTP 422 – There a defect in the submitted parameters. A message will be returned with the body to aid in diagnostics.

Confirmation

The confirmation of the creation can be tested by using the GET/READ calls.

> **Note:** When using GET/READ calls, including anything that passes codes or text beyond the URL, such as Update, Delete, and Filterby, make sure that anything containing special characters is URL encoded before being placed on the URL.

Updating a Profile

You can update an existing profile using the same parameters as for creating a profile except that the HTTP method is PUT instead of POST. You can also use this call instead of create. It will create or update a profile

The HTTP 409 return value is not applicable to this method.

Reading a Profile (Read)

In order to call this method you must set the following properties:

HTTP GET method

HTTP URL - <hostname:port>/ETLServices/admin/v1/profiles/<profilecode> - where <profilecode> will be swapped out for the intended one.

HTTP headers:

- xbri-mst-token as per the usage from the create call.
- Accept optional parameter, can be application/json or application/xml. If not provided, will default to application/json.

Returned Values

When submitted (as per the Swagger notes) the following values are expected back:

HTTP 200 Success - plus return of record that was submitted.

HTTP 401 Unauthorized - the xbri-mst-token is missing, invalid or the user in question does not have admin access

HTTP 404 Not found - no profile code was found.

Listing Created Profiles (Read)

In order to call this method you must set the following properties:

HTTP GET method

HTTP URL - See the URL in Creating a Profile.

HTTP headers

- xbri-mst-token as per the usage from the create call.
- Accept optional parameter, can be application/json or application/xml. If not provided, will default to application/json.

Returned Values

When submitted (as per the Swagger notes) the following values are expected back:

HTTP 200 Success - plus return of record that was submitted.

HTTP 401 Unauthorized - the xbri-mst-token is missing, invalid or the user in question does not have admin access

Deleting a Profile (DELETE)

In order to call this method you must set the following properties:

HTTP DELETE method

HTTP URL - <hostname:port>/ETLServices/admin/v1/profiles/<profilecode> - where <profilecode> will be swapped out for the intended one.

HTTP headers:

- xbri-mst-token as per the usage from the create call.

Returned Values

When submitted (as per the Swagger notes) the following values are expected back:

HTTP 200 Success - plus return of record that was submitted.

HTTP 401 Unauthorized - the xbri-mst-token is missing, invalid or the user in question does not have admin access

HTTP 404 Not found - no profile code was found.

ETL, Import Roles REST Service

This service is for associating roles with import profiles. This allows security to be maintained and tied to specific profiles.

Roles allow security to be maintained and tied to particular profiles. You must have created a profile before using this service. Currently there are two roles available to associate with profiles:

-XBRAdministrator

-XBRTenantAdministrator

Creating and Managing Import Roles

In order to perform an import request you will need to create a role. Without the settings below, the CRUD operations will be assumed false.

A role allows the following to be defined (per role):

Table 2-3: Role Properties

Property	Description
Profile Code	The profile code that will be associated with the role
Role Name	The name of the role to associate with the profile
Create	Indicate true/false if that role is allowed to create new import requests
Read	Indicate true/false if that role is allowed to read existing requests.
Update	Indicate true/false if that role is allowed to update existing requests
Delete	Indicate true/false if that role is allowed to delete requests

Creating a Role

From a ReST point of view this is effectively a HTTP POST method. The full schema details can be found by looking at the Swagger output, which includes the schemas: <hostname:port>/ETLServices/swagger.json

Service URL: <hostname:port>/ETLServices/admin/v1/roles

Table 2-4 Role Header Details

Name	Description
Content-Type	This can be one of two types, json or xml. The full expected string depending on the type, is either of the following: application/json application/xml
xbri-mst-token	The token from the login process. See Logging in to REST Web Services
Accept	This can be one of two types, json or xml. If not provided, will assume xml. Examples: application/json, application/xml

Body

The schema mentioned above provides all the properties; here is an example (JSON):(

```
{
"RoleName": "XBRAdministrator",
```

```
 "RoleSettings":{ "RoleSetting"}
}
ProfileCode": "POSLOG",
                          "Create": true,
                          "Read": true,
                          "Update": true,
                          "Delete": true
                      }
        ]
  }
}
```

Note: All properties are case sensitive.

Returned Values

When submitted (as per the Swagger notes) the following values are expected back:

HTTP 200 - Success, plus return of record that was submitted.

HTTP 401 - Unauthorized, the xbri-mst-token is missing, invalid or the user in question does not have admin access.

HTTP 400 - Bad Request - the parameters in the body do not match the schema.

HTTP 409 - Conflict, role code already exists, use update instead.

HTTP 415 - Unknown Media type - the Content-Type in the header block has not been provided.

Confirmation

The confirmation of the creation can be tested by using the GET/READ calls.

Note: When using GET/READ calls, including anything that passes codes or text beyond the URL, such as Update, Delete, and Filterby, make sure that anything containing special characters is URL encoded before being placed on the URL.

Updating a Role

You can update an existing role using the same parameters as for creating a role except that the HTTP method is PUT instead of POST. You can also use this call instead of create. It will create or update a role

The HTTP 409 return value is not applicable to this method.

Reading a Role (Read)

In order to call this method you must set the following properties:

HTTP GET method

HTTP URL - <hostname:port>/ETLServices/admin/v1/roles/<role code> - where <role code> will be swapped out for the intended one.

HTTP headers:

- xbri-mst-token as per the usage from the create call.
- Accept optional parameter, can be application/json or application/xml. If not provided, will default to application/json.

Returned Values

When submitted (as per the Swagger notes) the following values are expected back:

HTTP 200 Success - plus return of record that was submitted.

HTTP 401 Unauthorized - the xbri-mst-token is missing, invalid or the user in question does not have admin access

HTTP 404 Not found - no role code was found.

Listing Created Roles (Read)

In order to call this method you must set the following properties:

HTTP GET method

HTTP URL - See the URL in Creating a Role.

HTTP headers

- xbri-mst-token as per the usage from the create call.
- Accept optional parameter, can be application/json or application/xml. If not provided, will default to application/json.

Returned Values

When submitted (as per the Swagger notes) the following values are expected back:

HTTP 200 Success - plus return of record that was submitted.

HTTP 401 Unauthorized - the xbri-mst-token is missing, invalid or the user in question does not have admin access

Deleting a Role (DELETE)

In order to call this method you must set the following properties:

HTTP DELETE method

HTTP URL - <hostname:port>/ETLServices/admin/v1/roles/<role code> - where <role code> will be swapped out for the intended one.

HTTP headers:

- xbri-mst-token as per the usage from the create call.

Returned Values

When submitted (as per the Swagger notes) the following values are expected back:

HTTP 200 Success - plus return of record that was submitted.

HTTP 401 Unauthorized - the xbri-mst-token is missing, invalid or the user in question does not have admin access

HTTP 404 Not found - no role code was found.

ETL, Import Staging Module REST Service

This service allows you to control and configure staging-based import modules. Administrator users can configure properties such as how often pending records are moved from the staging area to the application.

The following properties are definable (per module, such as POSLOG):

Table 2-5: Staging Properties

Property	Description
Module	The module that you are editing. This defaults to POSLOG
Description	A user-friendly description of the settings. This is optional and defaults to POSLOG Staging Module

Processing Threads	The number of processes that are able to extract data from the import queue and place them into the staging area. This is optional and will default to 1 if no value is provided. Note: The maximum value allowed for all profiles and staging processors is 40. If you enter a number greater than 40, The ETL module will not start up.
Stored Procedure	The stored procedure to trigger when a specified threshold has been met. This is optional, and there is no default value. If left blank, the module assumes an external process is controlling the area. If you are using the POSLOG module, the recommended stored procedure is "xbr_etl_immediate.run;BatchNo"
Max *R*ecords	The number of records that can be queued before the stored procedure is executed. This is optional, and defaults to 5000 rows.
No Activity Threshold	The number of seconds to remain idle (no new incoming queue entries) before the stored procedure is executed. This is optional, and defaults to 10?

If a staging record is not set up, then the records processed by the module will be transferred into the staging areas and no stored procedure will be triggered.

Creating a Staging Module

From a ReST point of view this is effectively a HTTP POST method. The full schema details can be found by looking at the Swagger output, which includes the schemas: <hostname:port>/ETLServices/swagger.json

Service URL: <hostname:port>/ETLServices/admin/v1/stagings

Table 2-6: Staging Header Details

Name	Description
Content-Type	This can be one of two types, json or xml. The full expected string depending on the type, is either of the following: application/json application/xml
xbri-mst-token	The token from the login process. See Logging in to REST Web Services
Accept	This can be one of two types, json or xml. If not provided, will assume xml. Examples: application/json, application/xml

Body:

The schema mentioned above provides all the properties; here is an example (JSON):

```
{
  "Module": "POSLOG",
  "Description": "POSLOG Staging Module",
  "StoredProcedure": " xbr_etl_immediate.run;BatchNo",
  "MaxRecords": 5000,
  "NoActivityThreshold": 1,
  "ProcessingThreads": 10
}
```

> **Note**: All properties are case sensitive, except for note fields such as Description.

Returned Values

When submitted (as per the Swagger notes) the following values are expected back:

HTTP 200 - Success, plus return of record that was submitted.

HTTP 401 - Unauthorized, the xbri-mst-token is missing, invalid or the user in question does not have admin access.

HTTP 400 - Bad Request - the parameters in the body do not match the schema.

HTTP 409 - Conflict, module code already exists, use update instead.

HTTP 415 - Unknown Media type - the Content-Type in the header block has not been provided.

Confirmation

The confirmation of the creation can be tested by using the GET/READ calls.

> **Note:** When using GET/READ calls, including anything that passes codes or text beyond the URL, such as Update, Delete, and Filterby, make sure that anything containing special characters is URL encoded before being placed on the URL.

Updating a Staging Module

You can update an existing staging module using the same parameters as for creating a module except that the HTTP method is PUT instead of POST. You can also use this call instead of create. It will create or update a staging module.

The HTTP 409 return value is not applicable to this method

Reading a Created Staging Module (Read)

In order to call this method you must set the following properties:

HTTP GET method

HTTP URL - <hostname:port>/ETLServices/admin/v1/stagings/<module code> - where <module code> will be swapped out for the intended one.

HTTP headers:

- xbri-mst-token as per the usage from the create call.
- Accept optional parameter, can be application/json or application/xml. If not provided, will default to application/json.

Returned Values

When submitted (as per the Swagger notes) the following values are expected back:

HTTP 200 Success - plus return of record that was submitted.

HTTP 401 Unauthorized - the xbri-mst-token is missing, invalid or the user in question does not have admin access

HTTP 404 Not found - no staging module code was found.

Listing Created Staging Modules (Read)

In order to call this method you must set the following properties:

HTTP GET method

HTTP URL - See the URL in Creating a Staging Module.

HTTP headers

- xbri-mst-token as per the usage from the create call.
- Accept optional parameter, can be application/json or application/xml. If not provided, will default to application/json.

Returned Values

When submitted (as per the Swagger notes) the following values are expected back:

HTTP 200 Success - plus return of record that was submitted.

HTTP 401 Unauthorized - the xbri-mst-token is missing, invalid or the user in question does not have admin access

Deleting a Staging Module (DELETE)

In order to call this method you must set the following properties:

HTTP DELETE method

HTTP URL - <hostname:port>/ETLServices/admin/v1/stagings/<module code> - where <module code> will be swapped out for the intended one.

HTTP headers:

- xbri-mst-token as per the usage from the create call.

Returned Values

When submitted (as per the Swagger notes) the following values are expected back:

HTTP 200 Success - plus return of record that was submitted.

HTTP 401 Unauthorized - the xbri-mst-token is missing, invalid or the user in question does not have admin access.

HTTP 404 Not found - no staging module code was found.

> **Note:** If you have changed settings, they will not be updated until you stop and re-start the Import Engine.

3

Importing Data

Once profiles have been created and associated with roles, and the Staging module has been configured, customer administrators can use the methods described in this section to import data.

ETL, Import REST Service

This service allows the import to be placed on the import engine. The import engine accepts import requests, which submit the data into a queue and return a reference that can be used to track the data in other REST services.

The following properties constitute a normal submission:

Table 2-7: Import Submission Properties

Property	Description
ID	A unique ID for the import
Name	A user-defined name for the import
Description	An optional user-defined description of what is being imported
Profile	What profile to use for import
Data	The data to be imported
data_type	The type of data being imported. These depend on the modules linked to the provided profile
Encoding	The type of encoding that has been used on the data
Priority	An optional rating of the priority of the import. Values available are: CRITICAL, UI, HIGH, MEDIUM, NORMAL, LOW, IDLE. If not provided, NORMAL is the default.
Dependencies	An optional list of IDs that are dependencies on this import.

Notes:

The information in this section relates to import requests and Not the import plugin modules. The ability to transform data into its final destination (Plugin Modules) is covered in other sections. The use cases in this section apply to the generic import engine.

There are a vast number of possible ways of using the Import REST service. Some of the possible scenarios have been listed below but it is not a definitive list.

All of the ReST statements mentioned below can also be found by looking at the Swagger output, which includes schemas: <hostname:port>/ETLServices/swagger.json

Import Request (Standard)

HTTP Method: GET

Service URL: <hostname:port>/ETLServices/import/v1

Table 2-8: Import Header Parameters

Name	Description
Content-Type	This can be one of two types, json or xml. The full expected string depending on the type, is either of the following: application/json application/xml
xbri-mst-token	The token from the login process. See Logging in to REST Web Services

Body

The following is an example import request using the JSON content type:

Simple JSON

```
{
  "id":"import.reprint.01",
  "name":"Test Receipt Reprint",
  "description":"Simple JSON packet showing receipt reprint",
  "data_type":"json",
  "encoding":"UTF-8",
  "data":"{\"header\":{\"common\":{\"businessDate\":\"2019-07-
30\",\"cashierNum\":\"1\",\"code\":\"1\",\"customer\":\"100\",\"division\":1,\"emp
loyeeSaleFlag\":\"Y\",\"employeeNum\":\"1\",\"regNum\":1,\"storeNum\":311,\"curren
cyCode\":\"USD\",\"trainingFlag\":\"N\",\"transDate\":\"2019-07-
30\",\"transNum\":1,\"transStat\":\"transStat\",\"transTime\":\"125848\",\"transTy
pe\":\"ADMIN\"},\"detail\":{\"recType\":\"HDR\",\"recCode\":\"608\",\"lineNum\":1,
\"origCashierNum\":\"100\",\"origRegNum\":1,\"origStoreNum\":311,\"origTransDate\"
:\"2019-07-30\",\"origTransNum\":99}}",
  "priority":"NORMAL",
  "profile":"POSLOG"
}
```

Zipped example of the same transaction:

```
{
  "id":"import.reprint.02",
  "name":"Test Receipt Reprint",
  "description":"Simple JSON packet showing receipt reprint",
  "data_type":"zip;json",
  "encoding":"base64",
"data":"UEsDBBQAAAAIADNXXE/sdiwh9gAAAN4BAAAFAAAAci50eHR1ULtuwzAM/BfNCWAlfcTZihhFO9
RDnA4dVZlwCOgRSHIBI/C/l7LkuECRjbw7Hnm8sjOIFhzbX5m0WlsTq+/eowHvKxGA7dmm4OW6eF5vC7Zi
Uvgzgqt7TQyPgG1hLnsfrCY3aosobvEHPUZTvmKgL8oOAI1Q8KpER6ovtsCLo4OOmmmG/FxitpzHBc6Bkc
Mh7fxsKpIHJ9Cg6bJnnSDz//qZSOa5aYIIpFrqjJ9QT7E2j7uH3Q0cLhF8qT7eazZSPggCFb2MbpaZfKuO
UwaZj3wq4riih857rcPucHtjflaCj3P01DZ/4mfodCfaQk76shzH8RdQSwECPwAUAAAACAAzV1xP7HYsIf
YAAADeAQAABQAkAAAAAAAAACAAAAAAAAAAAci50eHQKACAAAAAAAAEAGACNA++Bfo3VAY0D74F+jdUB4YnR
gH6N1QFQSwUGAAAAAAEAAQBXAAAAGQEAAAAA",
  "priority":"NORMAL",
  "profile":"POSLOG"
}
```

Returned Values

When submitted (as per the Swagger notes) the following values are expected back:

HTTP 200 - The request passed basic validation but the response will give the real indication to success, see example response below

HTTP 201 – HTTP 200 has indicated that the request structure is valid, but the validation fails.

HTTP 401 - Unauthorized, the xbri-mst-token is missing, invalid or the user in question does not have admin access.

HTTP 400 - Bad Request - the request data provided doesn't meet the required format.

HTTP 415 - The content-type has not been set up with a known type.

HTTP 416 - The provided profile code is invalid.

Example Response

```
{ "id": "myId4", "code": 1, "description": "Duplicate request" }
```

List of Available Responses:

0 - Successful submission into queue

1 - Duplicate Request

2 - Mandatory fields missing

3 - Unknown request

4 - Unknown data type

5 - Unknown profile code (seen when no plugin module have been defined or engine is stopped, meaning no plugin modules have been loaded).

6 - Unknown encoding

7 - Unknown priority

8 - Not currently able to handle request

9 - Data does not match the given profile or encoding

10 - Auth token expired or invalid

11 - Access Denied

12 - Unable to cancel request, job beyond rollback point

13 - Invalid Data

14 - Unable to retry request

15 - Unable to find any content in compressed request

Example: Successful Submission into Queue:

```
{ "id": "myId5", "code": 0, "description": "Request successfully added to the
queue", "token": "b1a36886-0930-4e8d-a35c-18f358b2efc8" }
```

The token generated can be used with other web services, see below:

Import Request (Callback)

This call is identical to the standard request, with some extra properties that must be added to the request.

Table 2-9: Import Request (Callback) Properties

Name	Description
Token_type	authentication token type. User definable, but if using the internal callback expected to be Bearer.
Authentication_token	user definable token.
Callback_frequency	optional, the frequency in quick callbacks are made. Can be either 'COMPLETE' or 'STAGE'. If not provided will default to COMPLETE.

Example Import Request (Callback):


```
{ "id":"myId10", "name":"myTest", "description":"myDescription",
"data_type":"zip;xml", "encoding":"base64",   "encoding":"base64",
"data":"UEsDBBQAAAAIADNXXE/sdiwh9gAAAN4BAAAFAAAAci50eHR1ULtuwzAM/BfNCWAlfcTZihhFO9
RDnA4dVZlwCOgRSHIBI/C/l7LkuECRjbw7Hnm8sjOIFhzbX5m0WlsTq+/eowHvKxGA7dmm4OW6eF5vC7Zi
Uvgzgqt7TQyPgG1hLnsfrCY3aosobvEHPUZTvmKgL8oOAI1Q8KpER6ovtsCLo4OOmmmG/FxitpzHBc6Bkc
Mh7fxsKpIHJ9Cg6bJnnSDz//qZSOa5aYIIpFrqjJ9QT7E2j7uH3Q0cLhF8qT7eazZSPggCFb2MbpaZfKuO
UwaZj3wq4riih857rcPucHtjflaCj3P01DZ/4mfodCfaQk76shzH8RdQSwECPwAUAAAACAAzV1xP7HYsIf
YAAADeAQAABQAkAAAAAAAAACAAAAAAAAAAci50eHQKACAAAAAAAAEAGACNA++Bfo3VAY0D74F+jdUB4YnR
gH6N1QFQSwUGAAAAAAEAAQBXAAAAGQEAAAAA",
  "priority":"NORMAL",
  "profile":"POSLOG"
}
, "authentication_token": "myToken", "token_type": "Bearer", "callback_frequency":
"STAGE" }
```


When submitted (as per the Swagger notes) the result code will be the same as the standard request.

> **Note:** In order to test a callback, you must create a remote service that will accept the callback data.

Example Callback Message

This is a typical message that is sent when a call back is made


```
{ "stage": "FAILED", "state": "FAILED", "percent_complete": 100 }
```


The stage and state depend on the module processing them.

Expected Response: HTTP 200

Get Submitted Log

Once a request has been successfully submitted onto the queue, a token will be generated. This token can be used to retrieve the current logs.

HTTP Method: GET

Service URL: <hostname:port>/ETLServices/import/v1/log/<token> - where <token> will be taken from the request submission.

Header Parameters:

Table 2-10: Import Get Header Parameters

Name	Description
xbri-mst-token	The token from the login process. See Logging in to REST Web Services
Accept	Optional, can be either XML or JSON, by default it will use JSON. For example: application/json or application/xml

Returned Values

When submitted (as per the Swagger notes) the following values are expected back:

HTTP 200 – Successfully retrieved the logs.

HTTP 401 - Unauthorized, the xbri-mst-token is missing, invalid or the user in question does not have admin access.

Examples of Results

Example (JSON):


```
{
"stage":"FAILED",
```

```
 "entries":[

{ "sequence":1, "category":"STATE_CHANGE", "summary":"02 November 2018 10:15:04
GMT: State Change", "percent_complete":20, "description":"State changed to
PENDING." }
```

Example (XML):

```
<?xml version="1.0" encoding="UTF-8"?>
<logResponse>
<stage>FAILED</stage>
<entries>
<sequence>1</sequence>
<category>STATE_CHANGE</category>
<summary>02 November 2018 10:15:04 GMT: State Change</summary>
<percent_complete>20</percent_complete>
<description>State changed to PENDING.</description>
</entries>
```

The entries provide the current stage of processing, in this case FAILED. Each entry indicates how it arrived at the status. A state of FAILED or COMPLETED means the entry has come to the end of the run.

Get Submitted Status

Once a request has been successfully submitted onto the queue, a token is generated. This token can be used to retrieve the current status of the import.

HTTP Method: GET

Service URL: <hostname:port>/ETLServices/import/v1/status/<token> - where <token> will be taken from the request submission.

Header parameters:

- xbri-mst-token - taken from the v18 login process

- accept - optional, can be either XML or JSON, by default it will use JSON. For example: application/json or application/xml

Returned Values

When submitted (as per the Swagger notes) the following values are expected back:

HTTP 200 - Successfully retrieved the current status.

HTTP 401 - Unauthorized, the xbri-mst-token is missing, invalid or the user in question does not have admin access.

HTTP 404 - Not found, the given token.

Examples of Results

Example (JSON):

```
{ "stage": "FAILED", "state": "FAILED", "percent_complete": 100 }
```

Example (XML):

```
<?xml version="1.0" encoding="UTF-8"?>
<statusResponse>
<stage>FAILED</stage>
<state>FAILED</state>
<percent_complete>100</percent_complete>
</statusResponse>
```

Get Submitted Request

Once a request has been successfully submitted onto the queue, a token will be generated. This token can be used to retrieve back the original request.

HTTP Method: GET

Service URL: <hostname:port>/ETLServices/import/v1/requests/<token> - where <token> will be taken from the request submission.

Header Parameters

Table 2-11: Import Get Submitted Request Header Parameters

Name	Description
xbri-mst-token	The token from the login process. See Logging in to REST Web Services
Accept	Optional, can be either XML or JSON, by default it will use JSON. For example: application/json or application/xml

Returned Values

When submitted (as per the Swagger notes) the following values are expected back:

HTTP 200 - Successfully retrieved the current status.

HTTP 401 - Unauthorized, the xbri-mst-token is missing, invalid or the user in question does not have admin access.

HTTP 404 - Not found, the given token.

Examples of Results

If successful, here are a few example results:

Example (JSON):

```
 [
{ "id":"myId16", "name":"myTest", "data_type":"zip;xml", "encoding":"base64",
"data":"UEsD[cut down]¿, priority":"NORMAL", "profile":"POSLOG", "dependencies":[]
}
]
```

Example (XML):

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<importRequests>
<importRequest>
<id>myId16</id>
<name>myTest</name>
<data_type>zip;xml</data_type>
<encoding>base64</encoding>
<data>UEsD¿[cut down]</data>
<priority>NORMAL</priority>
<profile>POSLOG</profile>
</importRequest>
</importRequests>
```

Cancel Submitted Request

Once a request has been successfully submitted onto the queue, a token will be generated. This token can be used to cancel a request.

HTTP Method: PATCH

Service URL: <hostname:port>/ETLServices/import/v1/cancel/<token> - where <token> will be taken from the request submission.

Header Parameters

Table 2-12: Import Cancel Submitted Request Header Parameters

Name	Description
xbri-mst-token	The token from the login process. See Logging in to REST Web Services
Accept	Optional, can be either XML or JSON, by default it will use JSON. For example: application/json or application/xml

Returned Values

When submitted (as per the Swagger notes) the following values are expected back:

HTTP 200 - Successfully retrieved the current status.

HTTP 401 - Unauthorized, the xbri-mst-token is missing, invalid or the user in question does not have admin access.

If the state was successful (HTTP 200), the same import states mentioned in the standard import request will be returned. In particular for this call:

Code 0 - successfully cancelled.

Code 3 - Unknown request - if a token doesn't exist.

Code 12 - Unable to cancel request, job beyond rollback point - Meaning the job has moved beyond a certain stage in its processing and can't be cancelled. Completed jobs are primary examples of a job that can't be cancelled.

Examples of Results

If successful, here are examples of some possible results:

Example (JSON):

```
{ "id": "b1f5bc81-6327-42e9-a629-edd4eb77506f", "code": 12, "description": "Unable to cancel request, job beyond rollback point" }
```

Example (XML):

```
<importResponseFailure>
<id>b1f5bc81-6327-42e9-a629-edd4eb77506f</id>
<code>12</code>
<description>Unable to cancel request, job beyond rollback point</description>
</importResponseFailure>
```

Retry Submitted Request

Once a request has been successfully submitted onto the queue, a token will be generated. If the import fails, this token can be used to retry a request. This request is only likely to be used in a few scenarios, for example unresolved dependency problems or some environmental corrections. In most cases problems with the data would simply be submitted again as a different request.

HTTP Method: PATCH

Service URL: <hostname:port>/ETLServices/import/v1/retry/<token> - where <token> will be taken from the request submission.

Header Parameters:

Table 2-13: Import Retry Submitted Request Header Parameters

Name	Description
xbri-mst-token	The token from the login process. See Logging in to REST Web Services
Accept	Optional, can be either XML or JSON, by default it will use JSON. For example: application/json or application/xml '

Returned Values

When submitted (as per the Swagger notes) the following values are expected back:

HTTP 200 - Successfully retrieved the current status.

HTTP 401 - Unauthorized, the xbri-mst-token is missing, invalid or the user in question does not have admin access.

If the state was successful (HTTP 200), the same import states mentioned in the standard import request will be returned. In particular for this call:

Code 0 - successfully resubmitted.

Code 3 - Unknown request - if a token doesn't exist.

Code 14 - Unable to retry request. The most common reason for this being the request has already completed.

Examples of Results

If successful, here are examples of some possible results:

Example (JSON):

```
{ "id": "myId19", "code": 0, "description": "Request successfully resubmitted",
"token": "dee88b44-7655-4b14-ab39-c33d0e649e45" }
```

Example (XML):

```
<importResponse>
<id>myId19</id>
<code>0</code>
<description>Request successfully resubmitted</description>
<token>dee88b44-7655-4b14-ab39-c33d0e649e45</token>
</importResponse>
```

Get Submitted Requests (for a given user)

The following method allows for submitted requests to be reviewed.

HTTP Method: GET

Service URL: <hostname:port>/ETLServices/import/v1/requests

Header Parameters:

Table 2-14: Import Get Submitted Request (for a given user) Header Parameters

Name	Description
xbri-mst-token	The token from the login process. See Logging in to REST Web Services
Accept	Optional, can be either XML or JSON, by default it will use JSON. For example: application/json or application/xml

Returned Values

When submitted (as per the Swagger notes) the following values are expected back:

HTTP 200 - Successfully retrieved the logs.

HTTP 401 - Unauthorized, the xbri-mst-token is missing, invalid or the user in question does not have admin access.

Examples of Results

Example (Snippet) JSON:

```
 [
{"id":"myId16","name":"myTest","data_type":"zip;xml","encoding":"base64","data":"¿
","priority":"NORMAL","profile":" POSLOG ","dependencies":["myId13"]}
,
```

{"id":"myId17","name":"myTest","data_type":"zip;xml","encoding":"base64","data":"¿
","priority":"NORMAL","profile":" POSLOG ","dependencies":["myId13"]}
¿

Example (Snippet) XML:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<importRequests>
<importRequest><id>myId16</id><name>myTest</name><data_type>zip;xml</data_type><en
coding>base64</encoding><data>¿</data><priority>NORMAL</priority><profile> POSLOG
</profile><dependencies>myId17</dependencies></importRequest>
¿

ETL, Xstore POSLOG Import

This section describes using the Import Engine plugin module POSLOG for use with XStore POSLOG formats. This module can be configured in the ETL Import Engine. Administrator users can configure this plugin to accept import data from Xstore. The process consists of three parts:

1. Setting up a Profile for Xstore POSLOG Import
2. Setting up permissions against the Xstore POSLOG Import profile
3. Submitting POSLOG Requests

1 Setting up a Profile for Xstore POSLOG Import

Set up a profile that includes these characteristics:

- Link to the POSLOG module,
- Set-up of a pre-processor of XPATH
- Example XPATH mapping configuration

Sample REST Call

Service URL: <host:port>/ETLServices/admin/v1/profiles

HEADER:

```
-Content-Type: application/json
-Xbri-mst-token: <token from the login process>
```

BODY:

```
{
  "ProfileCode": "XStorePOSLOG",
  "Description": "XStore POSLOG Import",
  "Module": "POSLOG",
  "Parameters": "PRE-PROCESSOR=XPATH",
  "ProcessingThreads": 1,
  "BaseXPathLoaderConfig": data,
  "CustXPathLoaderConfig" : null,
  "ParentSchema": null,
  "ChildSchema": null,
  "SFTPSettings": null
}
```

The properties in the body of the REST are defined in the table below

Table 2-15: Import Profile Properties for Xstore POSLOG Imports

Property	Description
Profile Code	A mandatory unique identifier of the import.
Description	A user friendly description of what the import profile does. This property is optional.

Module	Identifies what kind of import package this import can connect to. This field is mandatory.
Parameters	Parameters that only the plugin module will understand. Each property should be seen as a key value store and delimited with a semicolon. Currently the available parameter is PRE-PROCESSOR=XPATH
Processing Threads	The number of processes that are able to extract data from the import queue. This will default to 1 if no value is provided. Note: In most cases, only one process should be needed, but in certain situations it may need to be increased. The maximum value allowed across all profiles and staging processors is 40. If you enter a number greater than 40, The ETL module will not start up.
BaseXPathLoaderConfig	This configuration setting is an alternative to the provided XBR Loader configuration. It provides the ability to configure an XPATH configuration against a profile. This allows different XPATH configurations to be set-up under different profiles.
CustXPathLoaderConfig	This is the same as BaseXPathLoaderConfig, except it allows customer override configurations.
ParentSchema	Optional property to allow an XSD definition to be associated with the import. If provided, the data will be tested using it.
ChildSchema	Optional, a secondary XSD definition related to ParentSchema, seen as a child schema from the parent.
SFTPSettings	Optional property stating whether the module to which the import is connected should be pulling all data in from an SFTP server. If this is not specified, all data will come from the web service.

Note: All properties are case sensitive, except for note fields such as Description.

2 Setting up Permissions against the Profile for Xstore POSLOG Import

After you set up the profile, you must set-up permissions against it.

Sample REST Call

Service URL: <host:port>/ETLServices/admin/v1/ roles

HEADER:

-Content-Type: application/json

-xbri-mst-token: <token from the login process>

BODY:

```
{
  "RoleName": "XBRAdministrator",
  "RoleSettings":{
    "RoleSetting":[
      {"ProfileCode": "XStorePOSLOG", "Create": true, "Read": true, "Update":
true, "Delete": true}
    ]
  }
}
```

Important: Before you can submit any XStore POSLOGs you must first stop and start the server. This will allow the changes made to be picked up.

3 Submitting POSLOG Requests

Once you have set up a profile and permissions on that profile, you can submit POSLOG requests.

Sample POSLOG file

Import Request REST call:

Service URL: <host:port>/ETLServices/admin/v1/import

HEADER:

-Content-Type: application/json

-Xbri-mst-token: <token from the login process>

BODY:

```
{
  "id":"xstore-1",
  "name":"XStore POSLOG #1",
  "description":"Test of XStore POSLOG Import",
  "data_type":"zip;xml",
  "encoding":"base64",

"data":"UEsDBBQAAAAIAKJOuUxjhmnA6AUAAP0hAAAKAAAAUE9TTG9nLnhtbO1aW5PSMBR+d8b/EPuqva
aUwpQqclEUQYH18uTENst2LC22Yd3113vSFnsDBNHRGXd3YMu55eR8J/kyZK3HNysfXdMo9sKgI6iSIiAa
OKHrBcuOcLEYiqbw2LZeT+fjcInANog7whVj67Ysf/36VQqiS5FELJbCaCmP3s8oI54vB2RF4zVxqCykPm
2XXZf8XMLINQkYWVInjNaSE67km5iFUe5yUx7pK07G0BRFld+/Gs+dK7oiohfEjAQOFVCcCMahQxhM5aQk
UTo96SZ2BQSZtsuxjk0c9dl1IZJ9/x5CyFpEJIiJwwOhHk/VH/oEantJ/BjSnl5e+l5ASzJw8QIA4FXoFh
Vpbt31+u0PuBqSIqmSaphIRKqEpZYKD4qkpKaFoRe3a9oRZoNFdzT+OO+OB1l6PENuOo2WJPC+JfMd9W3V
kuvS3COt4JzPmhuDdUmSG74Lo88xK0QtCXK7hef78BmriqpiEyBW+Euz5FSRGz4QxWRqPRJf9SPylUajPl
p5cQzlQqKYG1aN7F5/1Lfkqjh3mNMvGwrwTDarTzSyVaWJm5ZclubWTzcwIo3jPrntE0ZtTVGbomKKWsOS
q7rciS69gIsW3qroslCbbV1tYyw1Gjr4F81y70Hg7vPV2rwTdMgXjOqe0zWNCAv5dBtYb+qAQy7KzXqbKI
K53vag7ezBxcySi5Iq+sW+LjzPAdwNrNw+9T3YVqgLS4o3Hw/REQqGWQNmMcdQsRGjKzQIWHT7irKr0O0I
L+ktBf+3oecW1kHuuQe6vbAdj4Nu7MPh52hskdyNRt7IkDjxKRpN+FKGpeN8FuzkE/JiROA9uCa+5yLCWO
R92jCaNXj+k0XgheM13kb5YZTbcRPA2sBYa2GMFV3RoVMzad28T2Mn8tYcJ7snTl6i7lBF4yl6J8OvJRfU
qXNtal3XpW6yUode7BA/HelRRVKIAxN20UXgsV4Ys9eR5ySzrQevhrSHfFlXpT91LIzMIwzmvWqQw3MsZW
qrkoYtuSyr+8zocuOTiEMWg21m19SlVsuSdyp3F/dHpMTm/TVhqm2oUsuEKDUFGmVdBIy1u6Jdh22Ivzur
VHdcUoMbRgNAvbsKNwGzSRaiIkaLKy+uNniikiRpV4K1wDvj7ktqQllqwKuxrVJZuHvQNxsgeY/d8t1k+7
yreHEcOt6PXX6vmq89E5YebAoFWd2poN7Tw69p5NDk/DG95H26Natb7hq8Ik+dDw/DK1CW7Er7uNSsBblB
8Ep3K+7zlrD0sPKTTT73726AHCIOx2RsyUXBXhfyyafl7ikJd/tlOhVLcBThP5Z8yDybvK1I2tY8E+0vMm
SxPWSOXBs3VFV8sbDkmupghGdRuFmDTQZTLql7JfPeIX7mXbIZdShsdxwCO0HAkiviPQ05d0gQUDfbeE3N
0FQ4wTWMBhxHavoqf8l8b+HCXLI9Cvyx84F27vnAkDRTP+t8ABFa5dPamcROWi3lSD6fD5GqKWg2vZj00d
PxS/nd88Udpf9OSsfnMzrEwABoTXwCheNTGLzCpviXKVZsgesdxf7PFItPZFhFMlp3BHssweImn7thqoph
qP8ow+IzGbahSbqJz2FYiNBQ8O9k2JahK5rWhLPN8VS7ip48HI7RbCBezLtPxwM0fz59/Xowu6Pb30m3qq
SfTbd8/PPoFrL4ZboF31+kW573Hdv+z2wLvXMi22rGHdsey7alXf9PsO2JzKqfxaxaW4HjOT6DWeGmQYfv
0MyDzLrg21iE0j/pyuB3LsWLAF6mdJn0rkiwpAdKkEcc9fm1xMfexWw2mPQ+WPJWnHnUux1tbzA6AngKtq
lKext/G+4Potc4Ez1+y6NoZ6HHI+A/gx6LNn8UPFGRGvjvgWecBR586aNLjab56+BtI1TB+210dCoXVYjI
VCS9+VMmKnyt2qrx0AESgv23QkK/l4FOpZ9kokd02yJkxEfJO4BSvgJ9Bo/ZaU/Yli+xLEWYgR5OhalCSf
avTFS2BXH1YhZ0iaIssuT0vxTs71BLAQI/ABQAAAAIAKJOuUxjhmnA6AUAAP0hAAAKACQAAAAAAAAAgAAA
AAAAAABQT1NMb2cueG1sCgAgAAAAAAABABgAAJjmygX00wEAmObKBfTTATx56kCTZNQBUEsFBgAAAAABAA
EAXAAAABAGAAAAAA==",
  "priority":"NORMAL",
  "profile":"XStore"
}
```

Note:

XStore POSLOG calls are expected to come through as either XML or zipped, base64 or hex encoded strings. The above sample is using zipped/base64 encoded data. If raw XML is used, it is assumed the encoding would be UTF-8. It is suggested that you push the XML through a function, for example: https://www.freeformatter.com/json-escape.html#ad-output or any of the many others available in order to avoid any unwanted characters. This will allow the result set to be encapsulated in the request packet.

ETL, Xstore POSLOG External SFTP Import

The POSLOG import module can be configured to provide SFTP settings for customers who need to process data via SFTP. This module will work with both old and new style Xstore POSLOG formats. Access to use this feature will depend on the user's access to a pre-defined SFTP area, or the ability to maintain SFTP plugins.

Important: the external SFTP must be whitelisted by Oracle Cloud services.

This section explains how to configure the ETL, Xstore POSLOG Import profile described in the previous section to enable importing through SFTP.

Setting up a Profile for External SFTP Import

First you must set up a profile with SFTPSettings defined. SFTPSettings is an optional property stating whether the module to which the import is connected should be pulling data in from an SFTP server instead of accepting data from the REST service. This property is available for the POSLOG module. The SFTPSettings are defined in Table 2-16, below.

Sample REST call:

Service URL: <host:port>/ETLServices/admin/v1/profiles

HEADER:

-Content-Type: application/json

-Xbri-mst-token: <token from the login process>

BODY:

```
{
"ProfileCode": "POSLOGSFTP",
"Description": "New POSLOG SFTP Import",
"Module": "POSLOG",
"ProcessingThreads": 1,
"BaseXPathLoaderConfig": null,
"CustXPathLoaderConfig" : null,
"ParentSchema": null,
"ChildSchema": null,
"SFTPSettings":(

{ "Host":"mycompany.com", "Filespec": "newpl*.json", "Username":"myusername",
"Password":"mypassword" }

}
```

Table 2-16: SFTP Settings Properties

Property	Description
Host	The host name for the SFTP server, optional field. If not provided, the application will assume the local machine rather than a remote SFTP site.
Username	Username to be used with the SFTP server - if using a blank host name, or your remote SFTP doesn't require login than this becomes an optional parameter.
Password	Password to be used with the SFTP server - if using a blank host name, or your remote SFTP doesn't require login than this becomes an optional parameter.
Filespec	By default this will use 'PL*.xml'. The property can contain multiple file specifications, if multiple are required they would need to be delimited by ';'. For example pl*.xml;pl*.json. This is optional See: File Specifications, below.
ImportDirectory	Optional sub directory off the SFTP home directory. For example 'etl' would equate to '/etl'. By default it will assume the current directory you were located in after the login.
SuccessFolder	For any successful files process, the folder in which to move them. If not provided, will assume 'success' This is optional.
FailureFolder	As above, but for failed files. Will default to 'failures' if not provided. This is optional.
ArchiveSuccessAge	Archive any successful files after defined days, default 1 days. This is optional.
ArchiveFailureAge	Archive any failed files after defined days, default 7 days. This is optional.
TidySuccessAge	Remove any successful files after defined days. A value of zero means that the tidy functions won't be in operation. Any files left there will be managed by another person or process. This is optional.
TidyFailureAge	Remove any failed files after defined days. A value of zero means that the tidy functions won't be in operation. Any files left there will be managed by another person or process. This is optional
ImportFrequency	Frequency in which an attempt to import SFTP files, by default 1 minute. This is optional.

Note:

After any updates to the settings, you must stop and re-start the Import Engine.

Once the Import engine is started, if the SFTP options have been given, it will start to check the defined systems based on the ImportFrequency.

File Specifications

File extensions are important because they are used when setting up an import request. The following extensions will be recognized.

<filename>.json - data type = 'json'

<filename>.xml - data type = 'xml'

<filename>.json.zip - data type = 'zip;json'

<filename.zip - data type (assumed) 'zip;xml'

Once the file has been picked up, the file extension (.json, .xml or .zip) will be renamed to the machine name that is processing it.

After that, it will be moved to the defined success or failure folders.

Review Considerations

Any tickets imported by this SFTP process will be allocated to the username 'SFTP-POSLOG' and therefore will not appear on any standard request lists.

The easiest way to look at these imports would be to review the current queue entries. See ETL, XBRi Import Queue for further details.

Archiving Example

If you have the archive of successful files set-up for 1 day and the tidy files also set up for one day, the application sequentially runs the archive operation before tidy so, for example:

If you have a successful import file dated 13th Jan, on the 14th that file will be archived as if it is a day old. The tidy files operation will not find anything to do as the newly created archive was created on the 14th, it is not yet a day old. On the 15th, that archive file will be a day old and will be removed.

Data Import from XBRi SFTP Server

This profile configuration allows you to import data from the XBRi SFTP server. This import option is provided for files that can bypass the ETL process.

This will work in exactly the same way as the ETL, Xstore POSLOG External SFTP Import module, with the same settings, except that you do not need to set the HOST, username or password in the SFTP settings

Sample REST call:

Service URL: <host:port>/ETLServices/admin/v1/profiles

HEADER:

-Content-Type: application/xml

-Xbri-mst-token: <token from the login process>

BODY:

```
{
"ProfileCode": "POSLOGSFTP",
"Description": "New POSLOG SFTP Import",
"Module": "POSLOG",
"ProcessingThreads": 1,
"BaseXPathLoaderConfig": null,
"CustXPathLoaderConfig" : null,
"ParentSchema": null,
"ChildSchema": null,
"SFTPSettings":(

{ "ImportFrequency": 1”, "ImportDirectory": "/etl", "Filespec": "newpl*.xml",
"SuccessFolder": "success", "FailureFolder": "failure", "ArchiveSuccessAge": 1,
"ArchiveFailureAge": 7, "TidySuccessAge": 14,  "TidyFailureAge": 30,  }
```

The SFTPSettings in the example above let you configure the following import features:

ImportFrequency - Configure the frequency of checks made

ImportDirectory - Configure initial folder location. The import directory must be confirmed by the Oracle Cloud team by means of an SR ticket.

Filespec - Configure filespec location.

Note: the ImportDirectory and Filespec locations must pass Oracle white and blacklist checks

ArchiveSuccessAge - any files beyond this age (in days) will be moved into an archive.

ArchiveFailureAge - as above, only for failed imports.

TidySuccessAge - this is a new parameter that will remove any successful archives/files that are older than the given days. If this parameter isn't provided, successful archives will not be removed.

TidyFailureAge - this is a new parameter that will remove any failed files that are older than the given days. If this parameter isn't provided, failed archives will not be removed.

Notes:

If any of the SFTP related properties are passed, validation will be performed based on it being an SFTP import.

The new tidy properties will be useable in the SFTP settings/module.

4

Managing Import Services

This section describes methods provided to support and control the ETL environment. See the XBRi Administration guide for descriptions of options in the XBRi application for managing event subscriptions, controlling ETL services, and extending ODI configuration.

ETL, XBRi Import Queue

This REST service allows the user to search the queue to see what has been processed, currently or historically.

The current option is to see everything, without sorting or filtering

Sample REST call:

Service URL: <host:port> /ETLServices/import/v1/queue/search

HEADER:

-Content-Type: application/json

-Xbri-mst-token: <token from the login process>

Returned Values

When submitted (as per the Swagger notes) the following values are expected back:

HTTP 200 - Success, queue request has been successfully submitted.

HTTP 401 - Unauthorized, the xbri-mst-token is missing, invalid or the user in question does not have admin access

Example Output (JSON):

```
[{
"token": "648abdc4-4c52-4e52-8179-07702ab8e2eb",
"id": "SFTP-POSLOG-POSLOG-PL1234",
"name": "SFTP-POSLOG Import",
"description": "SFTP Import",
"profile": "POSLOG",
"data_type": "xml",
"priority": "NORMAL",
"entryCreated": "2018-11-06T11:57:52.87Z",
"user": "SFTP-POSLOG",
"state": "COMPLETE"
}
```

Example Output (XML):

```
<queueEntryResponses>
<queueEntryResponse>
<token>648abdc4-4c52-4e52-8179-07702ab8e2eb</token>
<id>SFTP-POSLOG-POSLOG-PL1234</id>
<name>SFTP-POSLOG Import</name>
<description>SFTP Import</description>
<profile>POSLOG</profile>
<data_type>xml</data_type>
<priority>NORMAL</priority>
<entryCreated>2018-11-06T11:57:52.87Z</entryCreated>
<user>SFTP-POSLOG</user>
<state>COMPLETE</state>
</queueEntryResponse>
```

It is assumed that this information would go on to call other services, for example extract a log file or retry a particular job.

ETL, XBRi Queue List

This service allows administrator users to search the data on the current ETL queue. The data returned can be filtered and sorted.

Fields that Can Be Extracted

Here is a snippet of the fields that can be extracted:

* token - token generated at the time of submission/acceptance.

* id - id given on the request.

* name - Import - name given on the request.

* description - Description given on the request.

* profile - Profile given on the request.

* dataType - XML - data type given on the request.

* priority - priority given on the request.

* entryCreated: - Date/Timestamp the request was submitted and accepted on to the queue.

* user: - user name that submitted the request.

* userData: - state of of the queue entry.

* state: - overall state of the job.

Extract in JSON format:


```
 {
 "items":[

{         "token": "e0951bf5-59f3-4cf0-89d6-a64637b23b67",          "id":
"T9c1d61f2-988a-45e9-a823-d8b80f77166b",          "name": "new POSLOG",
"description": "Testing new POSLOG format",          "profile": "POSLOG",
"dataType": "zip;json",           "encoding": "base64;utf-8",
"priority": "NORMAL",           "entryCreated": "2019-02-12T12:47:26.678+0000",
"user": "auser",           "userData": "COMPLETE",           "state": "COMPLETE",
"callbackFrequency": "COMPLETE",           "mediaType": "application/json" }

],
 "hasMore": true,
 "totalResults": 2587,
 "limit": 1,
 "count": 1,
 "encodedKeys":[]
 }
```


The *items* key holds the collection of rows returned from the search

The *hasMore* key is a boolean indication that there is more data than has been searched for.

The *totalRecords* key is an optional property that states how many records are available for extract. It will only be provided if requested, see below.

The *limit* key states whether there has been any limit on the data returned.

The *count* key shows how many rows are in the items collection.

The *encodedKeys* key is not currently used and will always be blank.

The data returned is primarily controlled by parameters used on the web service.

Using Parameters to Filter Data

Calling this web service without any parameters will bring back all records from the queue. This is not recommended due to the queue holding current and historical jobs. It is expected that users will apply a filter or restrict the data being returned by using some of the following parameters:

Table 2-17: XBRi Queue List Parameters

Property	Description
fromId	filter the results from a certain Id
toId	filter the result to a certain Id*
since	filter the results since a certain date & time. This date is the creation date of the entry on the queue.
until	filter the results until a certain date & time. This date is the creation date of the entry on the queue.
offset	only return rows after the given offset. Only really used to page the data.
limit	limit the number of rows returned. It's recommended that this is used for most operations to stop large volumes of data coming back
totalResults	a boolean parameter controlling if the total results is calculated. By default this result is false.
filterBy	provide a filter for the results, the details on what this filter can be is provided below **
orderBy	provide an order to the results. By default this based on the natural sorting order of the storage, currently the token. Details of the storage can be found below ***.

Notes:

These parameters are case sensitive.

When filtering on the Id field, the comparisons being made are based on a string field, not a numeric.

Logic Statements

The following logic statements can be used for filtering strings

Table 2-17: XBRi Queue List Logic Statements

Statement	Description
<field name> *is null*	filter for a field being null
<field name> *is not null*	filter for a field being not null
<field name> *lt* <value>	filter for a field being less than a defined value
<field name> *lteq* <value>	filter for a field being less than or equal to a defined value
<field name> *gt* <value>	filter for a field being greater than a defined value

<field name> *gteq* <value>	filter for a field being greater than or equal to a defined value
<field name> *not equal* <value>	filter for a field being not equal to a defined value
<field name> *not eq* <value>	same as above
<field name> *equals* <value>	filter for a field being equal to a defined value
<field name> *eq* <value>	same as above
<field name> *like* <value>	- filter on a field being like a defined value. It's expected that % symbols will be used to provide a wildcard.
<field name> *not like* <value>	filter on a field being not like a defined value. % symbols are assumed to be used in the defined value
<field name> *!like* <value>	same as above****
<field name> *contains* <value>	very similar to like, the difference is you don't provide % symbols, it automatically wraps the given value with % on each side
<field name> *!EQ* <value>	same as the not equal****
<field name> *!===* <value>	same as the not equal****
<field name> *!==* <value>	same as the not equal****
<field name> *!=* <value>	same as the not equal****
<field name> *===* <value>	same as the equal****
<field name> *==* <value>	same as the equal****
<field name> *=* <value>	same as the equal****
<field name> *<* <value>	same as less than****
<field name> *>* <value>	- same as greater than *****
<field name> *<=* <value>	same as less than equal****
<field name> *>=* <value>	same as greater than equal****

**** the above mentioned logic patterns contain special characters and therefore may cause confusion to the application if passed in without encoding. In summary, these parameters are being passed into a URL/web service. It is expected that any special characters used including single/double quotes may cause problems with the parsing routines and therefore recommended that the filtering strings be pushed through a URL encoder before trying to pass in the values.

Sample REST Calls

HTTP Method: POST

Service URL: HTTPS://[ETL Server]/ETLServices/import/v1/queue/search

HEADER:

Xbri-mst-token: <token from the login process>

Accept: application/json or application/xml (optional)

Search on Parameters Only

/ETLServices/import/v1/queue/search?filterBy="token = '7583636e-4c60-47ff-b24b-f00d2552fafd'"

/ETLServices/import/v1/queue/search?filterBy="id = 'T694e19dd-57fe-48de-ab14-2c39757d35f4'"

/ETLServices/import/v1/queue/search?filterBy="name = '1234'"

/ETLServices/import/v1/queue/search?filterBy="description contains 'new'"

/ETLServices/import/v1/queue/search?filterBy="data_type like '%xml%'"

/ETLServices/import/v1/queue/search?filterBy="encoding like '%utf%'"

/ETLServices/import/v1/queue/search?filterBy="priority = 'normal'"

/ETLServices/import/v1/queue/search?filterBy="entryCreated= '2019-02-11T11%3A30%3A32.935%2B0000'"

/ETLServices/import/v1/queue/search?filterBy="user='auser'"

/ETLServices/import/v1/queue/search?filterBy="userData = 'FAILED'"

/ETLServices/import/v1/queue/search?filterBy="state= 'FAILED'"

/ETLServices/import/v1/queue/search?filterBy="callbackFrequency= 'COMPLETE'"

/ETLServices/import/v1/queue/search?filterBy="mediaType= 'application/json'"

/ETLServices/import/v1/queue/search?filterBy="state!= 'FAILED'

Notes:

Each of the above examples shows the fields in the web service being filtered. Some data items do not require URL encoding, but dates, contain columns + characters which will be interpreted as spaces, so those characters must be converted.

The like and contains statements can only be used on String fields, the following fields aren't Strings: priority, entryCreated, callbackFrequency, and State.

Search on Parameters Using Logic

The logic statements mentioned above can also be combined, for example:

/ETLServices/import/v1/queue/search?filterBy="state!= 'FAILED' and data_type == 'zip;json'"

/ETLServices/import/v1/queue/search?filterBy="state!= 'FAILED' & data_type == 'zip;json'"

/ETLServices/import/v1/queue/search?filterBy="state!= 'FAILED' && data_type == 'zip;json'"

Ways of performing OR logic:

/ETLServices/import/v1/queue/search?filterBy="state!= 'FAILED' or data_type == 'zip;json'"

/ETLServices/import/v1/queue/search?filterBy="state!= 'FAILED' | data_type == 'zip;json'"

/ETLServices/import/v1/queue/search?filterBy="state!= 'FAILED' || data_type == 'zip;json'"

This comparison logic can be repeated any number of times to make more complex queries, for example:

/ETLServices/import/v1/queue/search?filterBy="state!= 'FAILED' and data_type == 'something' or token = '7583636e-4c60-47ff-b24b-f00d2552fafd'"

Or, if the statement needs more complexity:

/ETLServices/import/v1/queue/search?filterBy="(state equals 'FAILED' and token not equals '21587600-c93b-41ef-a3a2-ba5b83f6fe31') or (token equals '7583636e-4c60-47ff-b24b-f00d2552fafd' and state not equals 'FAILED')"

> **Note:** For the great majority of scenarios it is better to use textual logic statements than symbols when performing manual tests, because it reduces the need to URL encode statements. Using the special character logic will reduce the overall size of the statement but may need encoding.

Ordering Data Syntax

Ordering the data can be done using the following syntax:

<field name>[;asc/;desc],<....>

The sorting order ;asc or ;desc is optional. If not provided the application will assume +asc+ending. The semicolon character separates the field name from the asc/desc option. More sub sorting-options can be provided by delimiting the option by comma.

Ordering Data Example

orderBy=name,entryCreated;desc

The example above would order the results by name in ascending order, then by descending entry created date/time.

Example Using since and util Filters

/ETLServices/import/v1/queue/search?since=2019-02-11T11%3A30%3A32.935%2B0000&until=2019-02-11T11%3A30%3A32.935%2B0000

Combined Filter and Sorting Example:

/ETLServices/import/v1/queue/search?filterBy="state!= 'FAILED' and data_type == 'zip;json'"&orderBy=name,entryCreated;desc

Example Using fromId and toId

/ETLServices/import/v1/queue/search?fromId=T261aa16c-7f2c-4db9-bcba-82c98e6efe44&toId=T261aa16c-7f2c-4db9-bcba-82c98e6efe44

Example Using totalResults:

/ETLServices/import/v1/queue/search?since=2019-02-11T11%3A30%3A32.935%2B0000&until=2019-02-11T11%3A30%3A32.935%2B0000&totalResults=true

Setting totalResults parameter to true on execution causes the totalResults property in the results to be populated. It also provided the boolean setting hasMore based on the difference between the rows returned and the total count. If the above totalResults is turned on, the count property and totalResults will always be the same, the only reason they will differ is when limit or offset will be used. The totalResults provides the total number of records available. The count is the number of rows returned, plus the offset/limit restrictions.

Example Using totalResults with limit:

/ETLServices/import/v1/queue/search?since=2019-02-11T11%3A30%3A32.935%2B0000&until=2019-02-13T23:59:32.935%2B0000&totalResults=true&limit=5

Example Using totalResults with offset:

/ETLServices/import/v1/queue/search?since=2019-02-11T11%3A30%3A32.935%2B0000&until=2019-02-13T23:59:32.935%2B0000&totalResults=true&limit=5&offset=5

Returned Values

When submitted, the following values are expected back:

HTTP 200 - The request passed basic validation. The response will give the results.

HTTP 401 - Unauthorized, the xbri-mst-token is missing, invalid or the user in question does not have admin access.

HTTP 400 - Bad Request - the request data provided doesn't meet the required format.

ETL Queue Reports

The following two reports let you see the ETL Queue status and details from within the application. The reports are found in the Shared Reports > Admin > SQL Research Reports folder:

ETL Queue Detail Report – Provides details on the records passing through the ETL queue, with prompts on Date, Status, and Profile

ETL Queue Status Report – This report summarizes the number of records passing through the ETL queue, including a count of entries for each category such as Success, Failed, and so on.

Appendix A

POSLOG Schema

The following section covers the new schema and example data that will be used for POSLOG imports. This new definition has been created based on the current staging tables. The only difference is that instead of a flat structure, the information has been made relational in order to cut down on overall packet size, plus reducing network traffic.

Schema

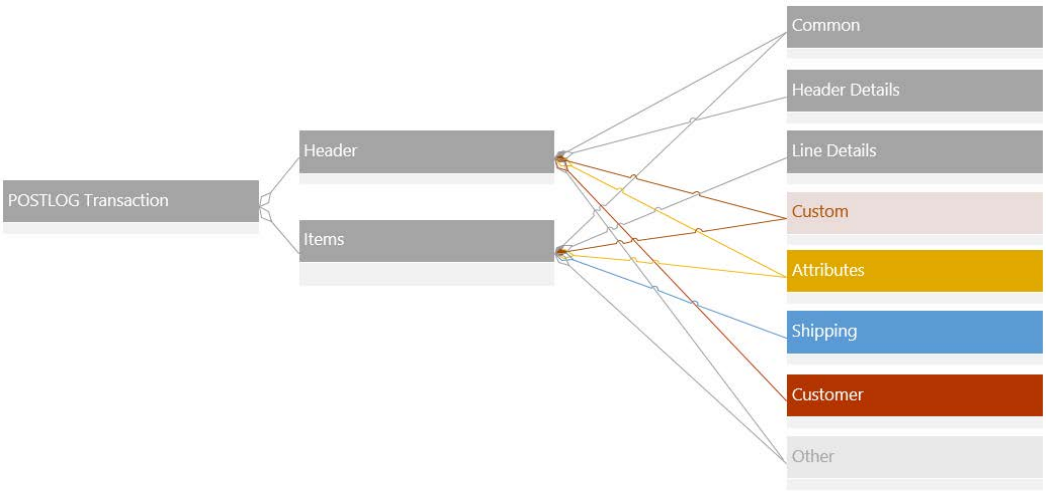
In order to simplify the staging schema, the properties are broken out into eight data categories:

- Header details
- Line details
- Attributes
- Shipping
- Customer
- Custom
- Other

These categories combine to make two primary record types:

- A single 'Header' record
- Multiple 'Line' records

The diagram below shows the connectivity of the data categories:



POSLOG Transaction Data Categories

Header Record

A header record is made up of:

- Header Details

- Common
- Attributes
- Customer
- Custom
- Other

Item Records

An item record is made up of:

- Line Details
- Common
- Attributes
- Shipping
- Custom
- Other

Both header and item records make up a transaction. The import process will accept a single transaction or a collection of transactions.

Table A-1: POSLOG Transaction

Name	Type	Field Info	Comment
header	Object 'Header'	Mandatory	See 'Header' section
items	Array of 'Item'	Optional**	See 'Item' Section

** This record is mandatory for certain record types, see the XBRi Core Field Mapping guide for details.

Table A-2: Header

Name	Type	Field Info	Comment
common	Object 'Common'	Mandatory	See 'Common' section
detail	Object 'Header Details'	Mandatory	See 'Header Details' section
custom	Object 'Custom'	Optional**	See 'Custom' Section
attributes	Object 'Attributes'	Optional**	See 'Attributes' section
customer	Object 'Customer'	Optional**	See 'Customer' section
other	Object 'Other'	Optional	See 'Other' section

** Property is optional only for certain record conditions, see the XBRi Core Field Mapping guide for details.

Table A-3: Item

Name	Type	Field Info	Comment
common	Object 'Common'	Mandatory	See 'Common' section
detail	Object 'Item Details'	Mandatory	See 'Item Details' section
custom	Object 'Custom'	Optional**	See 'Custom' Section
attributes	Object 'Attributes'	Optional**	See 'Attributes' section
shipping	Object 'Shipping'	Optional**	See 'Shipping Section'
other	Object 'Other'	Optional	See 'Other' section

** This record is mandatory for certain record conditions, see the XBRi Core Field Mapping guide for details.

Table A-4: Common

Name	Type	Size	Field Info	Comment
businessDate	Date		Mandatory, yyyy-mm-dd	
cashierNum	String	20		
code	String	3		
configVersion	String	10		
customer	String	60		
division	Numeric	(10,0)		Default 1
employeeSaleFlag	String	1	Mandatory	
employeeNum	String	20		
fbNoSaleFlag	String	1		
loyaltyNum	String	60		
regnum	Numeric	(10,0)	Mandatory	
storeNum	Numeric	(10,0)	Mandatory	
currencyCode	String			Default USD
trainingFlag	String	1	Mandatory	
transDate	Date		Mandatory, yyyy-mm-dd	
transNum	numeric	(10,0)	Mandatory	
transStat	String	10	Mandatory	
transTime	Time	6	Mandatory	HHMMSS

transType	String	10	Mandatory	

Table A-5: Header Details

Name	Type	Size	Field Info	Comment
recType	String	20	Mandatory	
recCode	String	10	Mandatory	
accountNum	String	20		Default 0
authCode	String	20		
clockInTime	DateStamp		YYYY-MM-DDThh:mm:ssZ	Example assuming Zulu Time zone
clockOutTime	DateStamp		YYYY-MM-DDThh:mm:ssZ	Example assuming Zulu Time zone
couponAmount	Numeric	15,5		Default 0
deleteFlag	String	1		
depositAmount	Numeric	15,5		Default 0
duration	Numeric	6,2		Default 0
extendedAmount	Numeric	15,5		Default 0
feeAmount	Numeric	15,5		Default 0
hoursWorked	Numeric	6,2		
lineDiscountAmount	Numeric	15,5		
manualAuthCode	Numeric	5		Default 0
manualKeyedCode	Numeric	5		Default 0
origCashierNum	String	20		
origRegNum	Numeric	10		
origStoreNum	Numeric	10		
origTransDate	Date		yyyy-mm-dd	
origTransNum	Numeric	10		
overrideAmount	Numeric	15,5		Default 0
postVoidTimeDiff	String	6		
quantity	Numeric	5,3		Default 0
reasonType	String	60		
sellingFlag	String	1		
taxAmount	Numeric	15,5		Default 0
taxCode	Numeric	10		Default 0

tenderAmount	Numeric	15,5		Default 0
timeCardEntryId	Numeric	10		
transDiscountAmount	Numeric	15,5		Default 0
voidCode	Numeric	5		Default 0
comments	String	2000		

Table A-6: Line Details

Name	Type	Size	Field Info	Comment
recType	String	20	Mandatory	
recCode	String	10	Mandatory	
item	String	60		
lineNum	Numeric	10	Mandatory	
accountNum	String	20		Default 0
accountNumHash	String	64		
authCode	String	20		
balanceAmount	Numeric	15,5		
foreignAmount	Numeric	15,5		
foreignCurrencyCode	String	3		
extendedAmount	Numeric	15,5		Default 0
manualAuthCode	Numeric	5		Default 0
manualKeyedCode	Numeric	5		Default 0
merchDept	String	60		
merchClass	String	60		
origCashierNum	String	20		
origRegNum	Numeric	10		
origStoreNum	Numeric	10		
origTransDate	Date		yyyy-mm-dd	
origTransNum	Numeric	10		
origAmount	Numeric	15,5		Default 0
tenderAmount	Numeric	15,5		Default 0
overrideType	String	10		
overrideAmount	Numeric	15,5		Default 0
quantity	Numeric	5,3		Default 0
returnFlag	String	1		
salesPersonNum	String	20		

taxModifyCode	Numeric	10		Default 0
taxRate	Numeric	3,6		
taxAmount	Numeric	15,5		Default 0
volume	Numeric	5,5		
weight	Numeric	5,5		
lineDiscountAmount	Numeric	15,5		Default 0
lineTime	DateTime		YYYY-MM-DDThh:mm:ssZ	Example assuming Zulu Time zone
returnVerifType	String	1		Default '-'
comments	String	2000		
reasonType	String	60		
transDiscountAmount	Numeric	15,5		Default 0
voidCode	Numeric	5		Default 0

Table A-7: Attributes

Name	Type	Size	Field Info	Comment
Char1	String	254		
Char2	String	254		
Char3	String	254		
Char4	String	254		
Char5	String	254		
Code1	Numeric	5		Default 0
Code2	Numeric	5		Default 0
Code3	Numeric	5		Default 0
Code4	Numeric	5		Default 0
Code5	Numeric	5		Default 0

Table A-8: Customer

Name	Type	Size	Field Info	Comment
firstName	String	60		
lastName	String	60		
address1	String	254		
address2	String	254		
address3	String	254		
city	String	254		

stateProvince	String	30		
postalCode	String	20		
country	String	254		
phone	String	32		
phone2	String	32		
emailAddress	String	254		
custom1	String	60		
custom2	String	60		
custom3	String	60		

Table A-9: Custom

Name	Type	Size	Field Info	Comment
char1	String	100		
char2	String	100		
char3	String	100		
char4	String	100		
char5	String	100		
char6	String	100		
char7	String	100		
char8	String	100		
char9	String	100		
char10	String	100		
date1	DateTime		YYYY-MM-DDThh:mm:ssZ	Example assuming Zulu Time zone
date2	DateTime		YYYY-MM-DDThh:mm:ssZ	Example assuming Zulu Time zone
date3	DateTime		YYYY-MM-DDThh:mm:ssZ	Example assuming Zulu Time zone
num1	Numeric	15,5		
num2	Numeric	15,5		
num3	Numeric	15,5		
num4	Numeric	15,5		
num5	Numeric	15,5		
num6	Numeric	15,5		

num7	Numeric	15,5		
num8	Numeric	15,5		
num9	Numeric	15,5		
num10	Numeric	15,5		
code1	String	30		
code2	String	30		
code3	String	30		
code4	String	30		
code5	String	30		
code6	String	30		
code7	String	30		
code8	String	30		
code9	String	30		
code10	String	30		
flag1	String	1		
flag2	String	1		
flag3	String	1		
flag4	String	1		
flag5	String	1		
flag6	String	1		
flag7	String	1		
flag8	String	1		
flag9	String	1		
flag10	String	1		

Table A-10: Shipping

Name	Type	Size	Field Info	Comment
firstName	String	254		
lastName	String	254		
address1	String	254		
address2	String	254		
address3	String	254		
city	String	254		
stateProvince	String	30		
postalCode	String	20		

country	String	254		
phone	String	32		
phone2	String	32		
emailAddress	String	254		

Table A-11: Other

Name	Type	Size	Field Info	Comment
orgId	Numeric	10		Default '1'
version	Numeric	10	Mandatory	Always '0'
accrualFlag	String			Always null
postdate	Date		yyyy-mm-dd	Always null
dataSource	String	3		Value will always be 'P'
runId	Numeric	10		Always null
transCount	String	60		Always null
pvTransCount	String	60		Always null
userModified	String	32		Always null
dateModified	Date		yyyy-mm-dd	
userReleased	String	20		Always null
dateReleased	Date		yyyy-mm-dd	
userAssigned	varchar			Always null
conversationRate	numeric			Default '1'
otherAmount	numeric			Default '0'
matchCode	numeric	1		Always null
batchNo	Numeric	10		Always null
dateLoaded	Date		yyyy-mm-dd	
transRecordCount	Numeric	15,0		Always null
status	String	2		Always 'OK'
accountNumEncrypt	String	254		Always null
accountNumKeyId	Numeric	10		Always null

The submission will allow data_type of either XML or JSON.

Example JSON

This section shows example data in JSON format with samples of all the data in the schema tables in the previous section.

```
{
        "header": {
                "common": {
```

```
                                        "businessDate": "2018-12-29",
                                        "cashierNum": "CFDUFOOQSPMUJLKNPEDL",
                                        "code": "cod",
                                        "configVersion": "configVer",
                                        "customer": "customer",
{
        "header": {
                "common": {
                        "businessDate": "2018-12-29",
                        "cashierNum": "CFDUFOOQSPMUJLKNPEDL",
                        "code": "cod",
                        "configVersion": "configVer",
                        "customer": "customer",
"address3": "customerAddr3",
                        "city": "customerCity",
                        "stateProvince": "customerStateProvince",
                        "postalCode": "customerPostalCode",
                        "country": "customerCountry",
                        "phone": "customerPhone",
                        "phone2": "customerPhone2",
                        "emailAddress": "customerEmailAddr",
                        "custom1": "customerCustom1",
                        "custom2": "customerCustom2",
                        "custom3": "customerCustom3"
                },
                "custom": {
                        "char1": "customChar1",
                        "char2": "customChar2",
                        "char3": "customChar3",
                        "char4": "customChar4",
                        "char5": "customChar5",
                        "char6": "customChar6",
                        "char7": "customChar7",
                        "char8": "customChar8",
                        "char9": "customChar9",
                        "char10": "customChar10",
                        "date1": "2018-12-29T16:03:41Z",
                        "date2": "2018-12-29T16:03:41Z",
                        "date3": "2018-12-29T16:03:41Z",
                        "num1": 13,
                        "num2": 14,
                        "num3": 15,
                        "num4": 16,
                        "num5": 17,
                        "num6": 18,
                        "num7": 19,
                        "num8": 20,
                        "num9": 21,
                        "num10": 22,
                        "code1": "customCode1",
                        "code2": "customCode2",
                        "code3": "customCode3",
                        "code4": "customCode4",
                        "code5": "customCode5",
                        "code6": "customCode6",
                        "code7": "customCode7",
                        "code8": "customCode8",
                        "code9": "customCode9",
                        "code10": "customCode10",
                        "flag1": "1",
                        "flag2": "2",
                        "flag3": "3",
                        "flag4": "4",
                        "flag5": "5",
                        "flag6": "6",
                        "flag7": "7",
                        "flag8": "8",
                        "flag9": "9",
                        "flag10": "0"
                },
```

```
                "other": {
                    "orgId": 7587907656,
                    "version": 5399098361,
                    "accrualFlag": "T",
                    "postDate": "2018-12-29",
                    "dataSource": "abc",
                    "runId": 8111598310,
                    "transCount": "transCount",
                    "pvTransCount": "pvTransCount",
                    "userModified": "userModified",
"dateModified": "2018-12-29",
                    "userReleased": "userReleased",
                    "dateReleased": "2018-12-29",
                    "userAssigned": "userAssign",
                    "conversionRate": 23,
                    "otherAmount": 23,
                    "matchCode": 1,
                    "batchNo": 5181301486,
                    "dateLoaded": "2018-12-29",
                    "transRecordCount": 23,
                    "status": "ok",
                    "accountNumEncrypt": "accountNumEncrypt",
                    "accountNumKeyId": 472138501
                }
            },
            "items": [
                {
                    "common": {
                        "businessDate": "2018-12-29",
                        "cashierNum": "KKAYKOAPIBNKLRMBAHKH",
                        "code": "cod",
                        "configVersion": "configVer",
                        "customer": "customer",
                        "division": 4572582868,
                        "employeeSaleFlag": "e",
                        "employeeNum": "employeeNum",
                        "fBNoSaleFlag": "f",
                        "loyaltyNum": "loyaltyNum",
                        "regNum": 2349801512,
                        "storeNum": 5942291615,
                        "currencyCode": "cur",
                        "trainingFlag": "t",
                        "transDate": "2018-12-29",
                        "transNum": 7615324170,
                        "transStat": "transStat",
                        "transTime": "040335",
                        "transType": "transType"
                    },
                    "detail": {
                        "recType": "VCNBRKVJDRIWLENIPTVA",
                        "recCode": "recCode",
                        "item": "item",
                        "lineNum": 2,
                        "accountNum": "accountNum",
                        "accountNumHash": "accountNumHash",
                        "authCode": "authCode",
                        "balanceAmount": 23,
                        "foreignAmount": 24,
                        "foreignCurrencyCode": "for",
                        "extendedAmount": 30,
                        "manualAuthCode": 1,
                        "manualKeyedCode": 1,
                        "merchDept": "merchDept",
                        "merchClass": "merchClass",
                        "origCashierNum": "origCashierNum",
                        "origRegNum": 9999,
                        "origStoreNum": 2636776421,
                        "origTransDate": "2018-12-29",
                        "origTransNum": 99,
                        "origAmount": 25,
```

```
                                        "tenderAmount": 34,
                                        "overrideType": "override",
                                        "overrideAmount": 31,
                                        "quantity": 32,
                                        "returnFlag": "r",
                                        "salesPersonNum": "salesPersonNum",
                                        "taxModifyCode": 1,
"taxRate": 26,
                                        "taxAmount": 33,
                                        "volume": 27,
                                        "weight": 28,
                                        "lineDiscountAmount": 29,
                                        "lineTime": "2018-12-29T16:03:41.137Z",
                                        "returnVerifType": "v",
                                        "comments": "comments",
                                        "reasonType": "reasonType",
                                        "transDiscountAmount": 35,
                                        "voidCode": 1
                                },
                                "attributes": {
                                        "char1": "attributeChar1",
                                        "char2": "attributeChar2",
                                        "char3": "attributeChar3",
                                        "char4": "attributeChar4",
                                        "char5": "attributeChar5",
                                        "code1": 1,
                                        "code2": 2,
                                        "code3": 3,
                                        "code4": 4,
                                        "code5": 5
                                },
                                "shipping": {
                                        "firstName": "shipToFirstName",
                                        "lastName": "shipToLastName",
                                        "address1": "shipToAddr1",
                                        "address2": "shipToAddr2",
                                        "address3": "shipToAddr3",
                                        "city": "shipToCity",
                                        "stateProvince": "shipToStateProvince",
                                        "postalCode": "shipToPostalCode",
                                        "country": "shipToCountry",
                                        "phone": "shipToPhone",
                                        "phone2": "shipToPhone2",
                                        "emailAddress": "shipToEmailAddr"
                                },
                                "custom": {
                                        "char1": "customChar1",
                                        "char2": "customChar2",
                                        "char3": "customChar3",
                                        "char4": "customChar4",
                                        "char5": "customChar5",
                                        "char6": "customChar6",
                                        "char7": "customChar7",
                                        "char8": "customChar8",
                                        "char9": "customChar9",
                                        "char10": "customChar10",
                                        "date1": "2018-12-29T16:03:41Z",
                                        "date2": "2018-12-29T16:03:41Z",
                                        "date3": "2018-12-29T16:03:41Z",
                                        "num1": 13,
                                        "num2": 14,
                                        "num3": 15,
                                        "num4": 16,
                                        "num5": 17,
                                        "num6": 18,
                                        "num7": 19,
                                        "num8": 20,
                                        "num9": 21,
                                        "num10": 22,
                                        "code1": "customCode1",
```

```
                                        "code2": "customCode2",
                                        "code3": "customCode3",
                                        "code4": "customCode4",
                                        "code5": "customCode5",
"code6": "customCode6",
                                        "code7": "customCode7",
                                        "code8": "customCode8",
                                        "code9": "customCode9",
                                        "code10": "customCode10",
                                        "flag1": "1",
                                        "flag2": "2",
                                        "flag3": "3",
                                        "flag4": "4",
                                        "flag5": "5",
                                        "flag6": "6",
                                        "flag7": "7",
                                        "flag8": "8",
                                        "flag9": "9",
                                        "flag10": "0"
                                },
                                "other": {
                                        "orgId": 7587907656,
                                        "version": 5399098361,
                                        "accrualFlag": "T",
                                        "postDate": "2018-12-29",
                                        "dataSource": "abc",
                                        "runId": 8111598310,
                                        "transCount": "transCount",
                                        "pvTransCount": "pvTransCount",
                                        "userModified": "userModified",
                                        "dateModified": "2018-12-29",
                                        "userReleased": "userReleased",
                                        "dateReleased": "2018-12-29",
                                        "userAssigned": "userAssign",
                                        "conversionRate": 23,
                                        "otherAmount": 23,
                                        "matchCode": 1,
                                        "batchNo": 5181301486,
                                        "dateLoaded": "2018-12-29",
                                        "transRecordCount": 23,
                                        "status": "ok",
                                        "accountNumEncrypt": "accountNumEncrypt",
                                        "accountNumKeyId": 472138501
                                }
                        }
                ]
}
```


Example XML

This section shows example data in XML format for all generated fields.


```
<?xml version="1.0" encoding="UTF-8"?>
<transaction>
        <header>
                <common>
                        <businessDate>2018-12-29</businessDate>
                        <cashierNum>KDVFVKEJWMNXPQHXOFOL</cashierNum>
                        <code>cod</code>
                        <configVersion>configVer</configVersion>
                        <customer>customer</customer>
                        <division>2658698322</division>
                        <employeeSaleFlag>e</employeeSaleFlag>
                        <employeeNum>employeeNum</employeeNum>
                        <fBNoSaleFlag>f</fBNoSaleFlag>
                        <loyaltyNum>loyaltyNum</loyaltyNum>
                        <regNum>6505292392</regNum>
                        <storeNum>9171336011</storeNum>
```

```
                <currencyCode>cur</currencyCode>
                <trainingFlag>t</trainingFlag>
<transDate>2018-12-29</transDate>
                <transNum>4223885932</transNum>
                <transStat>transStat</transStat>
                <transTime>035918</transTime>
                <transType>transType</transType>
            </common>
            <detail>
                <recType>OTGJLGGIIEQPFBLNXADV</recType>
                <recCode>recCode</recCode>
                <lineNum>1</lineNum>
                <accountNum>accountNum</accountNum>
                <authCode>authCode</authCode>
                <clockInTime>2018-12-29T15:59:25Z</clockInTime>
                <clockOutTime>2018-12-29T15:59:25Z</clockOutTime>
                <couponAmount>2</couponAmount>
                <deleteFlag>d</deleteFlag>
                <depositAmount>3</depositAmount>
                <duration>4</duration>
                <extendedAmount>7</extendedAmount>
                <feeAmount>5</feeAmount>
                <hoursWorked>6</hoursWorked>
                <manualAuthCode>1</manualAuthCode>
                <manualKeyedCode>2</manualKeyedCode>
                <origCashierNum>origCashierNum</origCashierNum>
                <origRegNum>9999</origRegNum>
                <origStoreNum>5476614833</origStoreNum>
                <origTransDate>2018-12-29</origTransDate>
                <origTransNum>99</origTransNum>
                <overrideAmount>8</overrideAmount>
                <postVoidTimeDiff>PVTD</postVoidTimeDiff>
                <quantity>9</quantity>
                <reasonType>reasonType</reasonType>
                <sellingFlag>s</sellingFlag>
                <taxAmount>10</taxAmount>
                <taxCode>1234</taxCode>
                <tenderAmount>11</tenderAmount>
                <timeCardEntryId>4321</timeCardEntryId>
                <transDiscountAmount>12</transDiscountAmount>
                <voidCode>1</voidCode>
                <comments>comments</comments>
            </detail>
            <attributes>
                <char1>attributeChar1</char1>
                <char2>attributeChar2</char2>
                <char3>attributeChar3</char3>
                <char4>attributeChar4</char4>
                <char5>attributeChar5</char5>
                <code1>1</code1>
                <code2>2</code2>
                <code3>3</code3>
                <code4>4</code4>
                <code5>5</code5>
            </attributes>
            <customer>
                <firstName>customerFirstName</firstName>
                <lastName>customerLastName</lastName>
                <address1>customerAddr1</address1>
                <address2>customerAddr2</address2>
                <address3>customerAddr3</address3>
                <city>customerCity</city>
                <stateProvince>customerStateProvince</stateProvince>
                <postalCode>customerPostalCode</postalCode>
                <country>customerCountry</country>
                <phone>customerPhone</phone>
                <phone2>customerPhone2</phone2>
                <emailAddress>customerEmailAddr</emailAddress>
                <custom1>customerCustom1</custom1>
<custom2>customerCustom2</custom2>
```

```
                        <custom3>customerCustom3</custom3>
                    </customer>
                    <custom>
                        <char1>customChar1</char1>
                        <char2>customChar2</char2>
                        <char3>customChar3</char3>
                        <char4>customChar4</char4>
                        <char5>customChar5</char5>
                        <char6>customChar6</char6>
                        <char7>customChar7</char7>
                        <char8>customChar8</char8>
                        <char9>customChar9</char9>
                        <char10>customChar10</char10>
                        <date1>2018-12-29T15:59:25Z</date1>
                        <date2>2018-12-29T15:59:25Z</date2>
                        <date3>2018-12-29T15:59:25Z</date3>
                        <num1>13</num1>
                        <num2>14</num2>
                        <num3>15</num3>
                        <num4>16</num4>
                        <num5>17</num5>
                        <num6>18</num6>
                        <num7>19</num7>
                        <num8>20</num8>
                        <num9>21</num9>
                        <num10>22</num10>
                        <code1>customCode1</code1>
                        <code2>customCode2</code2>
                        <code3>customCode3</code3>
                        <code4>customCode4</code4>
                        <code5>customCode5</code5>
                        <code6>customCode6</code6>
                        <code7>customCode7</code7>
                        <code8>customCode8</code8>
                        <code9>customCode9</code9>
                        <code10>customCode10</code10>
                        <flag1>1</flag1>
                        <flag2>2</flag2>
                        <flag3>3</flag3>
                        <flag4>4</flag4>
                        <flag5>5</flag5>
                        <flag6>6</flag6>
                        <flag7>7</flag7>
                        <flag8>8</flag8>
                        <flag9>9</flag9>
                        <flag10>0</flag10>
                    </custom>
                    <other>
                        <orgId>3372150196</orgId>
                        <version>6491980267</version>
                        <accrualFlag>T</accrualFlag>
                        <postDate>2018-12-29</postDate>
                        <dataSource>abc</dataSource>
                        <runId>7833400239</runId>
                        <transCount>transCount</transCount>
                        <pvTransCount>pvTransCount</pvTransCount>
                        <userModified>userModified</userModified>
                        <dateModified>2018-12-29</dateModified>
                        <userReleased>userReleased</userReleased>
                        <dateReleased>2018-12-29</dateReleased>
                        <userAssigned>userAssign</userAssigned>
                        <conversionRate>23</conversionRate>
                        <otherAmount>23</otherAmount>
                        <matchCode>1</matchCode>
                        <batchNo>4607182991</batchNo>
                        <dateLoaded>2018-12-29</dateLoaded>
<transRecordCount>23</transRecordCount>
                        <status>OK</status>
                        <accountNumEncrypt>accountNumEncrypt</accountNumEncrypt>
                        <accountNumKeyId>3828391703</accountNumKeyId>
```

```
                </other>
        </header>
        <items>
                <item>
                        <common>
                                <businessDate>2018-12-29</businessDate>
                                <cashierNum>LDYEMHMFUKDXBOMRTOSV</cashierNum>
                                <code>cod</code>
                                <configVersion>configVer</configVersion>
                                <customer>customer</customer>
                                <division>1097732828</division>
                                <employeeSaleFlag>e</employeeSaleFlag>
                                <employeeNum>employeeNum</employeeNum>
                                <fBNoSaleFlag>f</fBNoSaleFlag>
                                <loyaltyNum>loyaltyNum</loyaltyNum>
                                <regNum>5127173067</regNum>
                                <storeNum>2965178626</storeNum>
                                <currencyCode>cur</currencyCode>
                                <trainingFlag>t</trainingFlag>
                                <transDate>2018-12-29</transDate>
                                <transNum>7566213694</transNum>
                                <transStat>transStat</transStat>
                                <transTime>035919</transTime>
                                <transType>transType</transType>
                        </common>
                        <detail>
                                <recType>KTUWLYQABJJHBFWEACYP</recType>
                                <recCode>recCode</recCode>
                                <item>item</item>
                                <lineNum>2</lineNum>
                                <accountNum>accountNum</accountNum>
                                <accountNumHash>accountNumHash</accountNumHash>
                                <authCode>authCode</authCode>
                                <balanceAmount>23</balanceAmount>
                                <foreignAmount>24</foreignAmount>
                                <foreignCurrencyCode>for</foreignCurrencyCode>
                                <extendedAmount>30</extendedAmount>
                                <manualAuthCode>1</manualAuthCode>
                                <manualKeyedCode>1</manualKeyedCode>
                                <merchDept>merchDept</merchDept>
                                <merchClass>merchClass</merchClass>
                                <origCashierNum>origCashierNum</origCashierNum>
                                <origRegNum>9999</origRegNum>
                                <origStoreNum>1637227287</origStoreNum>
                                <origTransDate>2018-12-29</origTransDate>
                                <origTransNum>99</origTransNum>
                                <origAmount>25</origAmount>
                                <tenderAmount>34</tenderAmount>
                                <overrideType>override</overrideType>
                                <overrideAmount>31</overrideAmount>
                                <quantity>32</quantity>
                                <returnFlag>r</returnFlag>
                                <salesPersonNum>salesPersonNum</salesPersonNum>
                                <taxModifyCode>1</taxModifyCode>
                                <taxRate>26</taxRate>
                                <taxAmount>33</taxAmount>
                                <volume>27</volume>
                                <weight>28</weight>
                                <lineDiscountAmount>29</lineDiscountAmount>
                                <lineTime>2018-12-29T15:59:25.68Z</lineTime>
                                <returnVerifType>v</returnVerifType>
                                <comments>comments</comments>
                                <reasonType>reasonType</reasonType>
<transDiscountAmount>35</transDiscountAmount>
                                <voidCode>1</voidCode>
                        </detail>
                        <attributes>
                                <char1>attributeChar1</char1>
                                <char2>attributeChar2</char2>
                                <char3>attributeChar3</char3>
```

```
                        <char4>attributeChar4</char4>
                        <char5>attributeChar5</char5>
                        <code1>1</code1>
                        <code2>2</code2>
                        <code3>3</code3>
                        <code4>4</code4>
                        <code5>5</code5>
                    </attributes>
                    <shipping>
                        <firstName>shipToFirstName</firstName>
                        <lastName>shipToLastName</lastName>
                        <address1>shipToAddr1</address1>
                        <address2>shipToAddr2</address2>
                        <address3>shipToAddr3</address3>
                        <city>shipToCity</city>
                        <stateProvince>shipToStateProvince</stateProvince>
                        <postalCode>shipToPostalCode</postalCode>
                        <country>shipToCountry</country>
                        <phone>shipToPhone</phone>
                        <phone2>shipToPhone2</phone2>
                        <emailAddress>shipToEmailAddr</emailAddress>
                    </shipping>
                    <custom>
                        <char1>customChar1</char1>
                        <char2>customChar2</char2>
                        <char3>customChar3</char3>
                        <char4>customChar4</char4>
                        <char5>customChar5</char5>
                        <char6>customChar6</char6>
                        <char7>customChar7</char7>
                        <char8>customChar8</char8>
                        <char9>customChar9</char9>
                        <char10>customChar10</char10>
                        <date1>2018-12-29T15:59:25Z</date1>
                        <date2>2018-12-29T15:59:25Z</date2>
                        <date3>2018-12-29T15:59:25Z</date3>
                        <num1>13</num1>
                        <num2>14</num2>
                        <num3>15</num3>
                        <num4>16</num4>
                        <num5>17</num5>
                        <num6>18</num6>
                        <num7>19</num7>
                        <num8>20</num8>
                        <num9>21</num9>
                        <num10>22</num10>
                        <code1>customCode1</code1>
                        <code2>customCode2</code2>
                        <code3>customCode3</code3>
                        <code4>customCode4</code4>
                        <code5>customCode5</code5>
                        <code6>customCode6</code6>
                        <code7>customCode7</code7>
                        <code8>customCode8</code8>
                        <code9>customCode9</code9>
                        <code10>customCode10</code10>
                        <flag1>1</flag1>
                        <flag2>2</flag2>
                        <flag3>3</flag3>
                        <flag4>4</flag4>
<flag5>5</flag5>
                        <flag6>6</flag6>
                        <flag7>7</flag7>
                        <flag8>8</flag8>
                        <flag9>9</flag9>
                        <flag10>0</flag10>
                    </custom>
                    <other>
```

```
                                        <orgId>3372150196</orgId>
                                        <version>6491980267</version>
                                        <accrualFlag>T</accrualFlag>
                                        <postDate>2018-12-29</postDate>
                                        <dataSource>abc</dataSource>
                                        <runId>7833400239</runId>
                                        <transCount>transCount</transCount>
                                        <pvTransCount>pvTransCount</pvTransCount>
                                        <userModified>userModified</userModified>
                                        <dateModified>2018-12-29</dateModified>
                                        <userReleased>userReleased</userReleased>
                                        <dateReleased>2018-12-29</dateReleased>
                                        <userAssigned>userAssign</userAssigned>
                                        <conversionRate>23</conversionRate>
                                        <otherAmount>23</otherAmount>
                                        <matchCode>1</matchCode>
                                        <batchNo>4607182991</batchNo>
                                        <dateLoaded>2018-12-29</dateLoaded>
                                        <transRecordCount>23</transRecordCount>
                                        <status>ok</status>

        <accountNumEncrypt>accountNumEncrypt</accountNumEncrypt>
                                        <accountNumKeyId>3828391703</accountNumKeyId>
                                </other>
                        </item>
                </items>
</transaction>
```

Provided here is an example with all the optional fields stripped out. The optional fields will use the defaults mentioned or will be null.

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction>
        <header>
                <common>
                        <businessDate>2018-12-29</businessDate>
                        <employeeSaleFlag>e</employeeSaleFlag>
                        <regNum>6505292392</regNum>
                        <storeNum>9171336011</storeNum>
                        <trainingFlag>t</trainingFlag>
                        <transDate>2018-12-29</transDate>
                        <transNum>4223885932</transNum>
                        <transStat>transStat</transStat>
                        <transTime>035918</transTime>
                        <transType>transType</transType>
                </common>
                <detail>
<recType>OTGJLGGIIEQPFBLNXADV</recType>
                        <recCode>recCode</recCode>
                        <lineNum>1</lineNum>
                </detail>
        </header>
        <items>
                <item>
                        <common>
                                <businessDate>2018-12-29</businessDate>
                                <employeeSaleFlag>e</employeeSaleFlag>
```

```
                    <regNum>5127173067</regNum>
                    <storeNum>2965178626</storeNum>
                    <trainingFlag>t</trainingFlag>
                    <transDate>2018-12-29</transDate>
                    <transNum>7566213694</transNum>
                    <transStat>transStat</transStat>
                    <transTime>035919</transTime>
                    <transType>transType</transType>
                </common>
                <detail>
                    <recType>KTUWLYQABJJHBFWEACYP</recType>
                    <recCode>recCode</recCode>
                    <lineNum>2</lineNum>
                </detail>
            </item>
        </items>
</transaction>
```

Example structure when wrapping up multiple transactions

```
<transactions>
<transaction>
        …
</transaction>
<transaction>
        …
</transaction>
</transactions>
```