Oracle Financial Services Know Your Customer

Utilities Guide

Release 8.0.8.0.0

October 2019

F24323-01

ORACLE

Document Control

Version Number	Revision Date	Change Log
8.0.8.0.0	October 2019	First draft.

Table of Contents

1 Preface

This Preface provides supporting information for the Oracle Financial Services Know Your Customer (OFS KYC) Application Pack Utilities Guide and includes the following topics:

1.1 Purpose of this Document

The Utilities Guide details the table and table elements that form a part of the Table to JSON and JSON to Table utilities' web services which can be deployed on any web server.

1.2 Intended Audience

This document is intended for all those users who want to understand the utilities used in the OFS KYC onboarding service so that they can integrate with our service from their onboarding systems.

1.3 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.

1.4 Access to Oracle Support

Oracle customers 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.

1.5 Related Documents

This section identifies additional documents related to OFS KYC. You can access these documents from the Documentation Library (OHC).

- Oracle Financial Services Know Your Customer Administration Guide
- Oracle Financial Services Know Your Customer Risk Assessment Guide
- Oracle Financial Services Know Your Customer Service Guide
- Oracle Financial Services Know Your Customer API Data Elements Guide

1.6 Prerequisites

JNDI data source configured for the ATOMIC schema. After you deploy the web service, replace the `##JNDI_NAME##` placeholder with the data source name in the following files:

- `connection.properties` file under `WEB-INF/classes` folder
- `Web.xml` file under `WEB-INF` folder

2 Overview

2.1 Table to JSON Utility

The Table to JSON utility creates a JSON according to the configurations specified by a user. The user can define the JSON structure and the JSON fields to be constructed along with an indication of where the data of each of the fields are available. Once the field values are provided, the utility creates the JSON according to the specified configurations and picks the values against each field as defined during configuration.

> **NOTE** The utility can support only two levels of array in the JSON. To see an example, refer Viewing the JSON. Example are also provided in the KYC Service Guide.

The Table to JSON utility is used in KYC to:

- prepare the inputs of the third party verifications to be provided as a part of onboarding service. The JSONs can vary based on data such as country, data sources, and customer type. Ready-to-use mappings have been created to use this JSON.
- prepare the input JSON for the onboarding service. This JSON is then provided as a response for KYC Onboarding.
- create risk assesments.
- prepare the input for Customer Screening (CS).
- prepare the input of the internal watch list.

The Table to JSON utility is used in Enterprise Case Management (ECM) to:

- Generate the JSON response for the Common Gateway service.
- Generate the JSON response which will be made available in the KYC setup.

> **NOTE** The Table to JSON utility must ONLY be used in conjunction with the Financial Crime and Compliance Management (FCCM) application packs. Usage with any external applications is not allowed.

2.2 JSON to Table Utility

The JSON to Table utility allows a user to define where the data of each of the fields are available. These values are captured in the database in a tabular format. This utility captures the information of individual fields of JSON as per the configuration.

> **NOTE** The utility can support only two levels of array in the JSON. To see an example, refer Viewing the JSON. Example are also provided in the KYC Service Guide.

The JSON to Table utility is used in KYC to:

- Save the JSON response.
- Display the JSON response on your User Interface.

> **NOTE** The Table to JSON utility must ONLY be used in conjunction with the Financial Crime and Compliance Management (FCCM) application packs. Usage with any external applications is not allowed.

3 Tables and Table Elements

As a part of the general configuration, there are some tables which need to be updated to get the JSONs constructed by the utility. Currently these configurations must be defined by the user in the tables. These tables and their respective columns are explained in the sections that follow.

3.1 Table to JSON

The following tables must be configured so that the Table to JSON utility can construct the JSON:

- fcc_tpg_table_json_mapping Table
- fcc_tpg_table_json_query Table
- fcc_tpg_table_json_query_params Table

The table elements are described below:

3.1.1 fcc_tpg_table_json_mapping Table

This table allows you to configure the JSON that is constructed by the utility, and is the first table that needs to be configured. Use this table to:

- configure the structure of the JSON
- indicate whether the value of the JSON is from a column in the table or a user input
- configure the value of the JSON field
- configure the data type of the field/array of the JSON

To view the table values, run the following query:

```
select * from fcc_tpg_table_json_mapping where mapping_id
='parent_mapping_id value';
```

The values are described in the table below:

Value Name	Description
Mapping ID	Mapping IDs are used in queries to construct a JSON. Each table has a mapping ID assigned to it and is case sensitive. For example, TEST_ARRAY_ENH1. Mapping IDs are unique for every JSON which is constructed.
Key	A unique key is assigned for every mapping ID which is generated.
Parent Field	Parent of the key that is generated in the JSON. Parent field of the first layer of keys must be kept as *parent* in this field. For the second layer of keys, the parent field is the fields in the first layer of keys.
Field	This field captures the exact label of the field name which is to be generated in the JSON.

Value Name	Description
Field Type	This field captures the field type. The types can be `ELEMENT, OBJECT` or `ARRAY`.
Field Data Type	This field captures the field data type of the JSON. The default is `STRING`. Other acceptable values are `STRING, NUMBER, BOOLEAN`.
Array Data Type	This field captures the array data type. The default is `OBJECT` (JSON object). Other acceptable values are `OBJECT, STRING, NUMBER, BOOLEAN`.
Data Source	The data source has to be either `TABLE` or `REQUEST`. The data source is `TABLE` if the value comes from a table or expression. The data source is `REQUEST` if the value comes from the request parameters. For information on the request parameters, see the Know Your Customer API Services Guide.
Table Name	This field captures the name of each table which contains the JSON field values. This has to be only in the schema where the web services are deployed. If the value of the field is not available in a table, then this will be blank. If the value is a static value, then this value can be provided in the Expression column. The table name must have the physical table name as defined in the database.
Column Name	This field captures the physical name of the column corresponding to the table which contains the field values of the JSON.
Expression	SQL expression that generates the field value. It overrides the Table Name and Column Name fields. It must be as is from the SQL query that generates the value. For example, in the case of `SELECT SampleValue FROM DUAL`, the expression field would be `SampleValue`. In the case of `SELECT UPPER(TABLE_NAME.COLUMN_NAME) from CUST`, the expression field would be `UPPER(TABLE_NAME.COLUMN_NAME)`.
Key Source	Provide the key source value as `REQUEST` to make a particular JSON field value dynamic. This means that the key is substituted by the value of the request parameters in the URL.
Related Mapping ID	Provides the mapping ID for the child array element which is inside the parent array. For each child array value in Field, there is a corresponding Related Mapping ID. Default is *null*.
Primary Key	Provides the primary key for the parent array element. For each parent array value in Parent Field, there is a corresponding Mapping ID. Default is *null*.

3.1.2 fcc_tpg_table_json_query Table

This table contains the `FROM` and `WHERE` clauses of the queries used to construct the JSON. For each mapping, this table has to be updated to reflect the `FROM` and `WHERE` clauses.

To view the table values, run the following query:

```
select * from fcc_tpg_table_json_query where mapping_id = ‘parent_mapping_id value’
```

The values are described in the table below:

Value Name	Description
Mapping ID	This field captures the mapping IDs which are case sensitive. This mapping ID has to be the same as that of the previous table mapping ID.
Parent Field	Parent of the key that is generated in the JSON. Parent field of the first layer of keys must be kept as *parent* in this field.
From Clause	FROM clause of the query that provides the data for the JSON. It should be defined at the parent level and every time we need to define a mapping ID.
Where Clause	WHERE clause of the query that provides the data for the JSON. There should only be one record being returned for OBJECT and ELEMENT types. ARRAYS may have one or more records, and each of those records will become a separate ARRAY element. It should be defined at the parent level and every time we need to pass request parameters for the URL and replace the question mark.
Foreign Key	Provides the foreign key for the child array element. For each child array value in Field, there is a corresponding Related Mapping ID. You must define the foreign key for that child array. Default is *null*.

3.1.3 fcc_tpg_table_json_query_params Table

This table contains the request parameters used by the queries in the fcc_tpg_table_json_query table.

To view the table values, run the following query:

```
select * from fcc_tpg_table_json_query_param where mapping_id = ‘parent_mapping_id value’;
```

The values are described in the table below:

Value Name	Description
Mapping ID	Mapping IDs are used in queries to construct a JSON. Each table has a mapping ID assigned to it and is case sensitive. For example, TEST_ARRAY_ENH1. Mapping IDs are unique for every JSON which is constructed.
Parent Field	Parent of the key that is generated in the JSON. Parent field of the first layer of keys must be kept as *parent* in this field.
Parameter Order	Order of the parameter within the WHERE clause of the previous table. The *?* values in the WHERE clause will be replaced by these values in this order. For example, if the WHERE clause is `where ob_cust_seq_id = ? and request_id = ?`, the first *?* is replaced by the value of the parameter with order 1.

Value Name	Description
Parameter Name	Name of the query parameter that will be passed through the URL.
Parameter Source	Source of the parameter. REQUEST is the only supported value now.

After the configurations are done, enter the URL for the JSON in the POSTMAN client based on the following format:

```
http://domain:port/ TabletoJSONService/createtabletojson?mappingId= ” ”
```

For more information, see Viewing the JSON.

3.2 JSON to Table

The following tables are used in this utility:

- fcc_ob_json_table_config Table
- fcc_ob_json_table_map Table

The table elements are described below.

3.2.1 fcc_ob_json_table_config Table

This table defines the structure of the input JSON which needs to be persisted in the KYC tables.

To view the table values, run the following query:

```
select * from fcc_tpg_json_table_config where
parent_mapping_id='parent_mapping_id value' order by order_used;
```

The values are described in the table below:

Value Name	Description
PARENT_MAPPING_ID	Parent mapping ID of the JSON. The parent_mapping_id value can be one of the following: • SCORING_RESPONSE • CS_WLS_RESPONSE • INT_WLS_RESPONSE • TRULIOO_ENTITY_RESPONSE TRULIOO_RESPONSE
JSON_ELEMENT	This field captures the JSON ID value for each JSON element. This value should match the element names in the JSON.
PARENT_JSON_ELEMENT	This field captures the name of the parent JSON element. If there is no parent element, *(null)* is displayed by default.
DATA_TYPE	This field captures the logical data type of the JSON element. The default is STRING. Other Data types are String array, Array, or JSON.

Value Name	Description
ORDER_USED	This field provides the order of the JSON elements which need to be captured among the complete JSON structure. The order of the first element to be captured is 1. The order of each subsequent element increases by 1 whenever a particular JSON element is captured.

3.2.2 fcc_ob_json_table_map Table

This table contains the mappings between the configured JSON elements and the corresponding table elements.

To view the table values, run the following query:

```
select * from fcc_tpg_json_table_map where parent_mapping_id='<
parent_mapping_id value>' order by order_used;
```

The values are described in the table below:

Value Name	Description
MAPPING_ID	Child mapping ID of the parent mapping ID. For example, if the corresponding parent mapping ID is mapped to three different tables, then three unique mapping IDs will be used.
PARENT_MAPPING_ID	Mapping ID which groups all the configuration and the corresponding mappings for the JSON. This must be the same in both config and map tables. The `parent_mapping_id` value can be one of the following: • SCORING_RESPONSE • CS_WLS_RESPONSE • INT_WLS_RESPONSE • IDV_ENTITY_RESPONSE IDV_RESPONSE
JSON_ELEMENT	The JSON elements configured in the config table. Only captures JSON elements of type `String` or `String array`. If a JSON element of type JSON or array is defined in the config table, then that JSON element must not be displayed in the map table.
TARGET_TABLE	Table in to which data is persisted.
TARGET_COLUMN	Column in to which data is persisted.
EXPRESSION	Values can be the service name, system date, or another query expression. In case you do not know the actual expression value, provide the value `NOTNULL`.
ORDER_USED	The same order used for a particular JSON element in the config table. Only captures JSON elements of type `string`.

4 Viewing the JSON

To view the JSON for the Table to JSON utility:

1. Open the POSTMAN client.
2. In the Builder tab, select the POST method.
3. Enter a URL in the method field. An example is given below:

   http://domain:port/TabletoJSONService/createtabletojson?mappingId= ‘ ‘
4. Click Send.

The JSON appears in the Request Body editor.

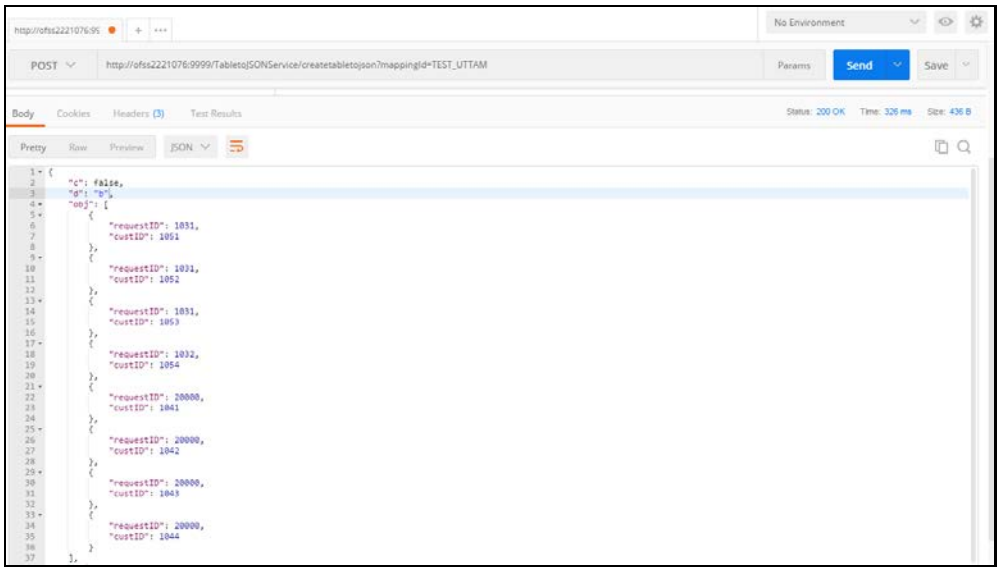
To view the JSON for the JSON to Table utility:

1. Open the POSTMAN client.
2. In the Builder tab, select the POST method.
3. Enter a URL in the method field. An example is given below:

   http://domain:port/JSONToTablePersistenceUtility/persistJSON?mappingID=’ ‘&requestid=’ ‘&OB_CUST_SEQ_ID=’ ‘
4. Click Send.

The JSON appears in the Request Body editor.

Below is a sample JSON:



If there are any errors, they are displayed in the Body editor after you send the request.

5 Mapping IDs and REST URLs for Utilities

The following table provides information on the mapping IDs, REST URLs and the applicable utilities. For information on the JSON preparation, see the Input preparation/ Hitting the individual services/Output capture section in the Know Your Customer Service Guide.

Mapping ID	Description	URL	Utility
CMMN_GATEWAY_INPUT	This mapping ID defines the structure of the JSON to be formed for the common gateway. It holds the definition of each data element and the value to be picked from the table for input creation.	http://#deployed_server#:#port#/TabletoJSONService/createtabletojson?mappingId=CMMN_GATEWAY_INPUT&requestId={OB_REQUEST_ID}&customerCounter={OB_CUST_COUNT}	Table to JSON
CREATE_ASSMNT	This mapping ID defines the structure of the JSON to be formed for creating an assessment. It holds the definition of each data element and the value to be picked from the table for input creation.	http://#deployed_server#:#port#/TabletoJSONService/createtabletojson?mappingId=CREATE_ASSMNT&requestId={OB_REQUEST_ID}	Table to JSON
CS_INPUT_IND	This mapping ID defines the structure of the JSON to be formed for customer screening where the customer type is individual. It holds the definition of each data element and the value to be picked from the table.	http://#deployed_server#:#port#/CommonGatewayService/createtabletojson?mappingId=CS_INPUT_IND&customerId={CUSTOMER_ID}&requestId={REQUEST_ID}	Table to JSON
CS_INPUT_NONIND	This mapping ID defines the structure of the JSON to be formed for customer screening where the customer type is entity. It holds the definition of each data element and the value to be picked from the table.	http://#deployed_server#:#port#/CommonGatewayService/createtabletojson?mappingId=CS_INPUT_NONIND&customerId={CUSTOMER_ID}&requestId={REQUEST_ID}	Table to JSON

Mapping ID	Description	URL	Utility
INTRL_WLS_INPUT	This mapping ID defines the structure of the JSON to be formed for the internal watch list. It holds the definition of each data element and its value to be picked from the table for input creation.	http://#deployed_server#:#port#/CommonGatewayService/createtabletojson?mappingId=INTRL_WLS_INPUT&requestId={REQUEST_ID}	Table to JSON
OB_RESP_1	This mapping ID defines the structure of the JSON to be formed as a final response after the onboarding customer data has been processed through all the services.	http://#deployed_server#:#port#/TabletoJSONService/createtabletojson?mappingId=OB_RESP_1&requestId=1003	Table to JSON
SCORING_INPUT	This mapping ID defines the structure of the JSON to be formed for scoring service. It holds the definition of each data element and the value to be picked from the table for input creation.	http://#deployed_server#:#port#/TabletoJSONService/createtabletojson?mappingId=SCORING_INPUT&requestId={OB_REQUEST_ID}	Table to JSON
SCORING_RESPONSE	This mapping ID defines where the values of the scoring output in the JSON format has to be captured in the KYC OB tables. This must be in the form of *table.column* mappings for each data element of the JSON.	http://#deployed_server#:#port#/JSONToTablePersistenceUtility/persistJSON?mappingID=SCORING_RESPONSE&requestid={OB_REQUEST_ID}	Table to JSON
CS_WLS_RESPONSE	This mapping ID defines where the values of the customer screening output in the form of JSON has to be captured in the KYC OB tables. This must be in the form of *table.column* mappings for each data element of the JSON.	http://#deployed_server#:#port#/JSONToTablePersistenceUtility/persistJSON?mappingID=CS_WLS_RESPONSE&requestId={OB_REQUEST_ID}&customerId={CUST_SEQ_ID}	JSON to table

Mapping ID	Description	URL	Utility
INT_WLS_RESPONSE	This mapping ID defines where the values of the internal watch list output in the form of JSON has to be captured in the KYC OB tables. This must be in the form of *table.column* mappings for each data element of the JSON.	http://#deployed_server#:#port#/JSONToTablePersistenceUtility/persistJSON?mappingID=INT_WLS_RESPONSE&requestid={OB_REQUEST_ID}&customerId={CUST_SEQ_ID}	JSON to table
CREATE_ASSMNT	This mapping ID defines which *table.column* will be updated based on the JSON formed when you create an assessment.	http://#deployed_server#:#port#/JSONToTablePersistenceUtility/persistJSON?mappingID=CREATE_ASSMNT	JSON to table
KYCOB_CASE_RESPONSE	The action taken on cases gives a JSON response. This mapping ID defines in which *table.column* the value will be updated.	http://#deployed_server#:#port#/JSONToTablePersistenceUtility/persistJSON?mappingID=KYCOB_CASE_RESPONSE	JSON to table

OFSAAI Support Contact Details

- Contact Infrastructure support at https://flexsupp.oracle.com if you have installed ERM and FCCM applications.
- Raise an SR in https://support.oracle.com if you have any queries related to EPM applications.

Send Us Your Comments

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, indicate the title and part number of the documentation along with the chapter/section/page number (if available) and contact the Oracle Support.

Before sending us your comments, you might like to ensure that you have the latest version of the document wherein any of your concerns have already been addressed. You can access My Oracle Support site which has all the revised/recently released documents.