

Managing Oracle fusion custom object through REST webservice



Blog By – Shivam Upadhyay

Objective:

Objective of this article is to provide an overview on how we can manage the custom objects we create through REST webservices. This covers the custom object created under CRM as well as ERP and SCM area.

Business use case or Problem statement:

- How to manage your custom object records or keep them in sync with the external applications.
- To minimize the manual maintenance and efforts to manage your custom object.
- A realtime handshake between source application and Oracle fusion custom object.

Pre requisites and Considerations:

- In this article we are considering that the custom object is already created in your Oracle fusion application.
- No child custom object is created and considered while designing this content.
- We are using postman application to execute the webservices.
- A basic idea of following webservices methods, Get Post Put and Delete methods is needed.

Custom Objects:

If I need to say in very simple terms, the custom objects are user defined objects which stores the additional information specific to your company or entity. These are flexible in nature can be customized as per the specific requirements.

To know more and how to create custom objects multiple documents and resources are available on Oracle repository and MOS. I suggest to refer the same.

Create custom object:

Now we know what are the custom objects and why they are used. We will now see with an example how we can manage our custom object using REST webservices.

For this firstly, we need to define a custom object in Oracle fusion with some sample records. Below is the one which I created for this document purpose. This custom object is storing a data in simple tabular format, with no child objects and screen defined.

Combination Custom Object											
View	Combination Name	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Level 7	Level 8	Level 9	Level 10
	CBM-00004	PID Class	SMD	CLS	ASDF	L5	L6	L7	L8	L9	L10
	CBM-00006	PID Class	PLM	CLS	ASDF	L5-SH	L6-TEST-09012...	L7-SKL	L8-WUI	L9-ERE	L10-WUO
	CBM-00010	SEMICON	SMD	CLS	ASDF	L6	L6	L7	L8	L9	L10
	CBM-00011	SEMICON	Webservice -1	Memories-1	DRAM	Monolithic	In form of Moun...	L7	L8	L9	L10

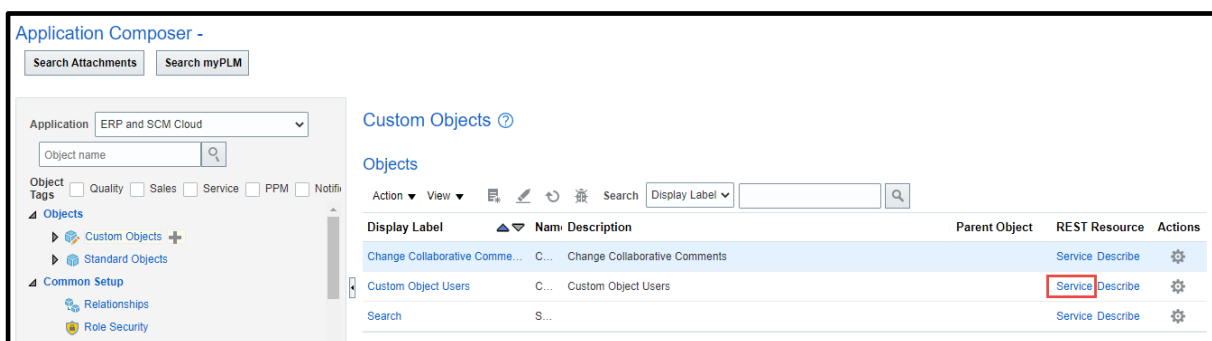
Get Custom object webservice endpoint URL

Once we have defined the custom object, we need to get the webservice Endpoint URL of the same to perform basic CRUD operations.

This we can get from the services options under the REST resource column under custom object section of Application composer.

Here is the example, how a link of the Custom Object REST endpoint should look like.

https://<CloudInstanceURL>/fscmRestApi/resources/latest/CustomObject_c



Perform Postman application setup:

Go to your postman application and create a connection with required authentication to run this webservice.

Within postman application we will perform GET , POST , PUT and DELETE actions on the webservice.


Retrieving existing records: Use GET method

In order to retrieve the custom object existing records, we use GET method.

Below is the example of performing the GET operation within the postman application.

For this, we need to pass a sample payload including the apinames of all the required and user defined values. We can ignore the autogenerated, sequence generated or any dynamic created values within the body response.

On successful creation of the record the success message “201 Created” will appear under status



```

2
3  ... "Level1_c": "SEMICON-POST WB",
4  ... "Level2_c": "SMD-POST WB",
5  ... "Level3_c": "SCIC-POST WB",
6  ... "Level4_c": "L4-POST WB",
7  ... "Level6_c": "L5-POST WB",
8  ... "Level5_c": "L6-POST WB",
9  ... "Level7_c": "L7-POST WB",
10 ... "Level8_c": "L8-POST WB",
11 ... "Level9_c": "L9-POST WB",
12 ... "Level10_c": "L10-POST WB",
13 ... "SelectedRow": null,

```

Body Cookies (1) Headers (17) Test Results Status: 201 Created Time: 719 ms Size: 2.37 KB Save Response

The response will be generated and shown below section of the body and can also be retrieved through GET operation.

Combination Custom Object

Actions ▾ Create

View ▾

Name	Combination	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Level 7	Level 8	Level 9	Level 10
CBM-00013	SEMICON-POST WB	SMD-POST WB	SCIC-POST ...	L4-POST WB	L6-POST WB	L5-POST WB	L7-POST WB	L8-POST WB	L9-POST WB	L10-POST WB	

The new record created can now be viewed on the custom object itself. You need to refresh the custom object to see the latest records. For ease, you can use filtered search to view the newly created record.

We can use this webservice and use POST method from external application and middleware to create new records within the custom object. Which makes sure that any new record added at the source should reflect on the custom object on Oracle side as well.

Updating a custom object record: PUT method

In order to update an existing record within your custom object use PUT method.

For this, we need to pass a sample payload including the apinames of all the required and user defined values. We can ignore the autogenerated, sequence generated or any dynamic created values within the body response.

On successful update of the record the success message “200 OK” will appear under status



The updated record can now be viewed on the custom object itself. You need to refresh the custom object to see the latest records. For ease, you can use filtered search to view the updated record.

We can use this webservice and use PUT method from external application and middleware to updated the records within the custom object. Which makes sure that any changes on the source side should reflect on the custom object on Oracle side as well.

Delete a custom object record: DELETE method:

In order to delete an existing record within your custom object use DELETE method.

For this, we need to pass a unique identifier within sample payload body. This can be the Id, recordName, recordNumber. These parameters help to identify which particular record should be fetched and deleted.

On successful update of the record the success message “204 No Content” will appear under status.



If we now try to search the deleted record, it will give no result found message as the record we trying to search doesn't exist anymore within your custom object.

Combination Custom Object

Find List

Actions

View

Name	Combination	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Level 7	Level 8	Level 9	Level 10
No results found.											

We can use this webservice and use DELETE method from external application and middleware to delete the obsolete or redundant records within the custom object. Which makes sure that any changes on the source side should reflect on the custom object on Oracle side as well.

Conclusion:

Using custom object REST webservices we can ensure the sync between source and destination application. Where Oracle fusion custom object can be either at source or at the destination system which can derive or reference the information as needed.

About Author:-

Shivam Upadhyay is a PLM Sr. Consultant with 9 years of experience in Oracle Agile PLM and PLM cloud. He has an PLM implementation experience over 10+ projects along with multiple PLM architecture experience across multiple domains.

