White Paper

Exposing pl/sql as REST Web Services using Integrated SOA Gateway

Author: Dipti Kaul Sep-2015 Fujitsu Consulting India Private Limited

Table of Contents

Overview	3
Integrated SOA Gateway	3
Major Features Architecture	
REST Web Services	4
Integration repository Customization	4
Case Studies	5
Exposing Standard API as REST Web Service and Invoking it	5
Exposing Pl/Sql as REST Web Service (Function Returning single value)	
Exposing Pl/Sql as REST Web Service (Procedure Returning Table Type value)	17
Conclusion	
References	

Overview

This document describes the process of exposing Standard and Custom API as Rest Web service using Integrated SOA Gateway. This document will help developers in understanding the ISG Customization, Rest Web Services and its development. This document is complete guide to ISG Customization with screenshots .It covers following case studies:

1. Exposing Standard API as REST Web Service and invoking it

2. Exposing **Pl/Sql Function** in Package as REST Web Service returning **single value a**nd invoking it.

3. Exposing **PI/Sql Procedure** in Package as REST Web Service **with Table Type as OUT Variable** and invoking it.

REST Web Services are available in Oracle E-Business Suite Release 12.2.3 and onwards only. REST based web services are deployed on Oracle E-Business Suite's application server. They do not dependent on Oracle SOA Suite and Oracle Applications Adapter.

Integrated SOA Gateway

Oracle E-Business Suite Integrated SOA Gateway (ISG) is a complete set of service infrastructure to provide, consume, and administer Oracle E-Business Suite Web services. You can use this tool to easily discover and search on interfaces, regardless of custom or Oracle seeded ones.

Major Features of ISG

Oracle E-Business Suite Integrated SOA Gateway can do the following:

- Display all Oracle E-Business Suite integration interface definitions through Oracle Integration Repository
- Support custom integration interfaces from Oracle Integration Repository
- Provide service enablement capability (SOAP and REST services) for seeded and custom integration interfaces within Oracle E-Business Suite
- Use the Integration Repository user interface to perform design-time activities such as generate and deploy Oracle E-Business Suite Web services
- Support synchronous interaction pattern for REST-based Web services
- Support multiple authentication types for inbound service requests in securing Web service content
- Enforce function security and role-based access control security to allow only authorized users to execute administrative functions
- Provide centralized, user-friendly logging configuration for Web services generated through Oracle E-Business Suite Integrated SOA Gateway's service provider
- Audit and monitor Oracle E-Business Suite inbound service operations from Service Monitor
- Leverage Oracle Workflow Business Event System to enable Web service invocation from Oracle E-Business Suite

REST Web Services

The acronym REST stands for Representational State Transfer, this basically means that each unique URL is a representation of some object. **REST** describes a set of architectural principles by which data can be transmitted over a standardized interface (such as HTTP). You can get the contents of that object using an HTTP GET, to delete it, you then might use a POST, PUT, or DELETE to modify the object (in practice most of the services use a POST for this).

Key features of EBS REST Services are as follows:

- Supports POST HTTP Verb
- Supports XML and JSON message format
- Described in WADL, XSDs
- No expensive tools require to interact with the Web service
- Smaller learning curve
- Efficient (SOAP uses XML for all messages, REST can use smaller message formats)
- Fast (no extensive processing required)
- REST-based implementation is simple compared to SOAP.

Integration Repository Customization

The term Customization of Integration Repository means option to add custom interfaces into Integration Repository. The term interface stands for all types of programs and technologies in Integration Repository, eg. plsql, java, business events, xml gateway etc.

Integration developers create and annotate custom integration interfaces based on the Integration Repository annotation standards. Integration administrators use a standalone design-time tool to validate these annotated source files against the annotation standards. After validation, a loader file is generated and then uploaded to the Integration Repository through backend processing. These custom interfaces are displayed based on the interface types to which they belong and displayed together with Oracle seeded ones from the Integration Repository user interface.

Process Flow



Enabling Custom Integration Interfaces and Services

Annotation Standards

Please refer the below link for details on Annotating Pl/Sql code.

http://docs.oracle.com/cd/E18727_01/doc.121/e12065/T511473T545912.htm

Setting Up Oracle E-Business Suite Integrated SOA Gateway

Please refer to below link for details:

http://docs.oracle.com/cd/E18727_01/doc.121/e12169/T511175T517162.htm

Also, please refer to the Doc ID 1311068.1 for ISG installation for REST Services.

Case Studies

Exposing Standard API as REST Web Service and Invoking it.

Perform following steps to expose Pl/Sql as a web service

- 1. Search API in Integration Repository
- 2. Deploy API as REST Service
- 3. Create Grants for REST Service
- 4. Invoke EBS REST Service (in JSON message format)
- 5. Verify the results

Here we will take an example of creating a user by exposing FND_USER_PKG Interface API and then invoking it to create user.

Following steps will explain the process of exposing the standard API and then invoking it.

Step 1: Navigate to Integration Repository Responsibility, Click on Search.



Then on search Page enter "FND_USER_PKG" in Internal Name and click on Go

ORACLE [®] Int	egration Reposito	νγ											_
Integration Repository						ĥ	\otimes	★	₽	2	Logged In As SYSADMI	м ?	
	Administration												
Search													
												E	Brows
Interface Name		_ Q	Internal Name	FND_USER_P	KG								
Product Family A	1	•	Interface Type	All	•								
Product A			Business Entity			<u> </u>							
	Go Clear All												
Show More Search Opt	ons												
Export 🔟 🎜 🕞	¢												
Name 🔺 Internal	Name	Product	Туре 🛆	Source 🛆	Status 🛆	Descr	iption						
			DL IOOI	Oracle	Active	Toble	Hondler!	to incort	or undate	data in ENF	D. LIOCO Ashis		

Step 2: Click on the search result, it will display List of methods available in the Interface Package.

								-			Log Configuration D	
	Type Product Sitatus	PL/SQL Oracle Application Ob Active	ect Library				Interfa	ce Bource	Graele			
	Online Help	See the related online f	eta:									
II Desc	ription	Grand										
able Ha	Indier to insert or	r update data in FND_US	IR table.									
	Source File Source Version Source Product	Patch/115/3gl/AFSC 120.16.12020000.8	ISRS.pla									
~ ~	5 ()			Olation -	Description						4 Previous	
eate/Up	date User	CHANGE_USE LOAD_ROW DERIVE PERS	NAME	Active	This api changes Greates or update	username, de	eals with encry Ys User data a Lother package	ption chang s appropriat	es and update foreigne.	a keysthat were using t	he old username.	
	Interna	al Name FND Type PL/S Product Orac Status Active ss Entity User	USER_PK QL le Applicati e	9 on Object Lil	brary						Scope Interface Source	Public Oracle
Over Sen	Onli view RE: vice Alias	Status Not De	Grants	nline help								
Oven Sen RES Servi	Onli view RE vice Alias T Service S ice Operation	Ine Help See 1 ST Web Service Status Not De Lions	Grants	<u>nline help</u>	4. Provious	1.10	Nov	+ 10 N				
Oven Ser RES Servi	Onli view RE vice Alias T Service S ice Operati	Ine Help See 1 ST Web Service Status Not De tions	Grants	<u>nline help</u>	Previous	1-10	Nex	t 10 🕨				
Oven RES Servi	Onli view RE vice Alias T Service S ice Operati C C Service t All Select t Name	Ine Help See 1 ST Web Service Status Not De tions	Grants	nline help		1-10	▼ Nex	tt 10 D				
Oven RES Servi Select Select	Onli view RE vice Alias ST Service S ice Operati C S C t All Select t Name Change L	Ine Help See 1 ST Web Service Status Not De tions tt None User Name	Grants	nline help nternal Name :HANGE_USI	Previous A ER_NAME	1-10	• Nex	tt 10 ▶ Grant				
Over RES Servi Select Select	Onli vice Alias T Service S ice Operati C S Service S t All Select t Name A Change L Create/Uj	Ine Help See I ST Web Service Status Not De dons It None Jser Name podate User	be related c Grants bloyed	nternal Name HANGE_USI	Previous A ER_NAME	1-10	• Nex	tt 10 ▶ Grant				
Over RES Servi Select Select	Onli view RE vice Alias T Service S ice Operati t All Select t Name Create/Uj Derive Pe	Ine Help See 1 ST Web Service Status Not De tions tt None User Name pdate User erson Party Id	Grants Grants bloyed	ntiernal Name HANGE_USI OAD_ROW	Previous A ER_NAME SON_PARTY	1-10 	• Nex	tt 10 ▶ Grant				
Over RES Servi Select Select	View RE vice Alias T Service S ice Operation t All Select t Name A Create/Up Derive Per LDAP Wr Name	Ine Help See I ST Web Service Status Not De tions It None Jser Name pdate User prison Party Id rapper Change U	Grants Grants bloyed	nternal Name HANGE_USI OAD_ROW ERIVE_PER: DAP_WRAPF	Previous A R_NAME SON_PARTY PER_CHANG	1-10 ID be_USER	Nex	t 10 D Grant				
Over RES Servi Select	View RE: vice Alias T Service S ice Operation C Ali Select t Ali Select t Name Create/Uj Derive Pe LDAP Wr Name LDAP Wr	Ine Help See I ST Web Service Status Not De ions It None Jser Name pdate User erson Party Id rapper Change U	Grants Joyed	nternal Name HANGE_USI OAD_ROW DAP_WRAPF DAP_WRAPF	Previous A FR_NAME SON_PARTY PER_CHANG PER_CREATI	1-10 '_ID E_USER	Nex	t 10 ▶ Grant				
Over RES Servi Select	onii view RE vice Alias T Service S ice Operati C S S Change L Change L Change L Change L Derive Pe LDAP Wr LDAP Wr LDAP Wr	ST Web Service Status Not De tions tt None User Name potate User arson Party Id apper Change User apper Create Us	Grants Grants bloyed	ntiernal Name HANGE_USI OAD_ROW ERIVE_PER: DAP_WRAPE DAP_WRAPE DAP_WRP_L	Previous ER_NAME SON_PARTY PER_CHANG PER_CREATI UPDATE_USI	1-10 '_ID E_USER E_USER ER_LOAI	Nex	t 10 D Grant				
Over RES Servi Select Select	Onli view RE vice Alias CT Service S ice Operation t All Select t Name Change L Create/Up Derive Per LOAP Wr LOAP Wr LOAP Wr	ST Web Service Status Not De tions It None Jser Name pdate User erson Party Id rapper Create Us rapper Update U: rapper Update U:	Grants Joyed	ntienal Name HANGE_USI OAD_ROW DAP_WRAPP DAP_WRAPP DAP_WRAPI DAP_WRAPI	Previous ER_NAME SON_PARTY PER_CHANG PER_OREATI JPDATE_USIG PER_UPDATI	1-10 '_ID E_USER E_USER ER_LOAR E_USER	Nex NAME DER	tt 10 ₽ Grant				
Over RES Servi	Onli View RE Vice Alias To Service S ice Operation Change L Change L Change L Change L Change L Dap Wr Name LDAP Wr LDAP Wr LDAP Wr LDAP Wr	Ine Help See I ST Web Service Status Not De ions It None Jser Name pdate User erson Partv Id rapper Change U rapper Update U: rapper Update U:	Grants Oloyed	nternal Name HANGE_USI OAD_ROW IERIVE_PERI DAP_WRAPF DAP_WRAPF DAP_WRAPF DAP_WRAPF DAP_WRAPF DAP_WRAPF DAP_WRAPF	4 Previous ER_NAME SON_PARTY PER_CHANG PER_CREATI JPDATE_USE PER_UPDATT USER	1-10 '_ID IE_USER E_USER ER_LOAI E_USER	NAME DER	tt 10 ▶ Grant				
Over RES Servi Selec	View RE vice Alias to Centre S to Certain t All Select t Name A Change L DAP Wr LDAP Wr LDAP Wr Remove 1 Set Old P	ST Web Service Status Not De tions Status Not	Grants Grants ployed I C C C C C C C C C C C C C C C C C C	nternal Name HANGE_USI OAD_ROW ERIVE_PER: DAP_WRAPF DAP_WRAPF DAP_WRAPF EMOVE_PIL EMOVE_PIL EMOVE_PIL	Previous Previous Previous Previous SON_PARTY Previous Previous	1-10 '_ID IE_USER E_USER E_USER ISER_IOAL E_USER	Nex	t 10 ₽ Grant				
Over RES Servi	View RE vice Alias To Service S ice Operati C S C S Create/Ui Derive Pe LDAP Wr LDAP Wr LDAP Wr Remove I Set Old P	ST Web Service Status Not De tions tt None Jser Name pdate User erson Party Id rapper Create Us rapper Update U: rapper Opdate U: PII user informat Person Party Id Ser GUID	Grants Grants ployed I Ser L er L er L er L er L er L er L er	ntienal Name HANGE_USI OAD_ROW DAP_WRAPP DAP_WRAPP DAP_WRAPI DAP_WRAPI EMOVE_PII_ EMOVE_PII_ ET_OLD_PE ET_OLD_US	Previous ER_NAME SON_PARTY PER_CHANG PER_CREATI JUSER RSON_PARTI LUSER RSON_PARTI ER_GUID	1-10 '_ID DE_USER E_USER E_USER E_USER TY_ID	NAME	t 10 ▶ Grant				

This page will display List of methods that can be deployed.

Step 3: Now select Create/Update User (we will create a user by invoking this Service once Deployed.) and enter Service Alias (This is mandatory) and then Click on Deploy button at bottom. You will receive message that service is successfully deployed and the status will be deployed.

In	tegration Repository >								
	 Confirmation Successfully deployed Web Service for interface 'User' to Application Server Oracle E-Business Suite Integrated SOA Gateway is not configured for SOAP Services. Please refer My Oracle Support Note 1311068.1 to configure for SOAP Services. 								
Р	LSQL Interface : Us	er							
	Internal Name Type	FND_U PL/SQ	JSER_PKG L	Ir	Scope nterface Source	Public Oracle			
	Status Business Entity	Active	Application Object Library						
	Online Help	See th	e related online help						
(Overview REST Web	Service	Grants						
	Service Alias Create_ REST Service Status Verb POST	User Deploye	d <u>View WADL</u>						

Step 4: Now you will see the link to WADL file, Click on the link. It shows the physical location of the service endpoint where the service is hosted.

<?xml version="1.0" encoding="UTF-8"?>
<application
xmlns:tns1="http://xmlns.oracle.com/apps/fnd/rest/create user/load row/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://wadl.dev.java.net/2009/02"
xmlns:tns="http://xmlns.oracle.com/apps/fnd/soaprovider/plsql/rest/fnd user p
kg/"
targetNamespace="http://xmlns.oracle.com/apps/fnd/soaprovider/plsql/rest/fnd
user_pkg/"
name="FND_USER_PKG"><grammars><include
xmlns="http://www.w3.org/2001/XMLSchema"</pre>

href="http://xxx-xxxapdv01.cli.ad:8005/webservices/rest/Create_User/?XSD=LOAD_ROW_SYNCH_TYPEDEF.x sd"/>

<resources< th=""><th></th><th></th><th>base="http://xxx-xxx-</th></resources<>			base="http://xxx-xxx-
apdv01.cli.ad:	8005/webservices/rest	Create_User/	"> <resource path="load_row/"></resource>
<method< th=""><th>name="POST"</th><th>id="LOAD</th><th>ROW"><request><representation< th=""></representation<></request></th></method<>	name= "POST"	id="LOAD	ROW"> <request><representation< th=""></representation<></request>
type="tns1:Inp	outParameters" mediaTy	pe=" applicati	on/xml"/>
<representation< td=""><td>on type="tns1:InputPar</td><td>ameters" medi</td><td>aType="application/json"/></td></representation<>	on type="tns1:InputPar	ameters" medi	aType="application/json"/>
	>		

Now open the link highlighted in Yellow in new Browser window, it will show the **XSD file** with input parameters that will be used later for invoking the Web Service.

Step 5.Now Click on the Grants Tab, select Create/Update User and then click on Create Grant button

-	ation				
Integration Repository >					
Information Oracle E-Business Suite Integra	ted SOA Gateway is not configured for SOAP Serv	ices. Please refer My Oracle S	upport Note	∋ 1311068.1 to confi	gure for SOAP Services.
PLSQL Interface : User					
Internal Name FND_US PType PL/SQL States Business Entity User Online Help See the	ER_PKG pplication Object Library elated online help			Scope Interface Source	Public Oracle
Overview REST Web Service					
Overview REST Web Service C Select Object and Create Grant	Revoke Grant 📜 🎜 📧 🌣	Previous 1-10	Next 10 🕨		
Overview REST Web Service Overview Select Object and Create Grant Select All Select None	Revoke Grant	Previous 1-10	Next 10 D		
Overview REST Web Service O Select Object and Create Grant Select All Select None Select Name Object All Select None Select Name	Revoke Grant 🔯 🗢 👼	4 Previous 1-10 ■ I REST Service Operation △	Next 10 ▶ Grant △		
Overview REST Web Service Overview Select Object and Create Grant Select Ail Select None Select Name A Change User Name Select Name	Revoke Grant I 💢 🎜 🕞 🍪	Previous 1-10 REST Service Operation	Next 10 ▶ Grant △		
Overview REST Web Service C Select All Select None Select All Select None Select All Select None Create/Update User	Revoke Grant I 🖾 🌫 🕞 🅸 Internal Name 스 CHANGE_USER_NAME LOAD_ROW	Previous 1-10 Incompared to the second sec	Next 10 ▶ Grant △		
Overview REST Web Service O Select Object and Create Grant Select Name Select Name Change User Name Create/Update User Orrester/Update User Name Change User Name	Revoke Grant I I Z 2	Previous 1-10 REST Service Operation ✓	Next 10 ▶ Grant △		
Overview REST Web Service O Select Object and Create Grant Select Name Select Name Change User Name Create/Update User Derive Person Party Id LDAP Wrapper Change User	Revoke Grant I I III Image: IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Previous 1-10 REST Service Operation	Grant △		

Step 6: Select a grantee type.

Grantee Type can be one of the following values:

- USER The grantee is an individual user who was selected directly.
- GROUP The grantee is a group of users or a member of a group of users.
- GLOBAL The grant was issued to all users.

If you selected Specific User or Group of Users, specify the user or group for which to create the grants in the Grantee Name field.

In our Case Study we will enter SYSADMIN and then click on Create Grant Button.

Integratio	n Repositor	y Administr	ation	
Integration	n Repository	PLBQL In	terface : User 🚿	
Create C	Grants			
				Gancel Greate Grant
Selected	d Methods			
202 67 1	-0 4 0			
Name	-	Internal Name		
Create/U	pdate L	OAD_ROW		
Grant Al	II Selected			
Gran	ntee Type	Specific Use		
Grant	tee Name	SYSADMIN		
				Cancel Create Grant
1				

It will display confirmation message:

L	Integration Repository Administration
	Integration Repository >
	Information
	 Grant has been created successfully for operation 'Create/Update User'. Oracle E-Business Suite Integrated SOA Gateway is not configured for SOAP Services. Please refer My Oracle Support Note 1311068.1 to configure for SOAP Services.

To **Revoke Grants** click on Revoke Grants button under Grants Tab ,it will display List of users with access for revoking the Grants.

To **Undeploy** the Web services select the method and click on undeploy under the REST Web Services Tab

Step 7: Now we will invoke this web service using Advanced REST Client apps available in Google Chrome.

If not available download it from following link:

https://chrome.google.com/webstore/detail/advanced-rest-client/hgmloofddffdnphfgcellkdfbfbjeloo

Now, after installing the REST client we will enter following details to invoke the web service:

http://socializade/	http://90080309000003008080005/webservices/rest/Create_User/load_row/							
GET ® POST © PUT © PATCH © DELETE © HEAD © OPTIONS © Other								
Raw Form	Headers							
Add new header								
Authorization	ralue							
	2	Construct						

Step 8: Enter the following header details:

Authorization by clicking on Add new header link then enters the application username/password by clicking on construct.

http://	****	8005/webse	rvices/rest/Create_User/lo	ad_row/	
GET ® P	OST © PUT	PATCH	OELETE OHEAD	OPTIONS Other	
Raw	Form	Headers			
dd new hea	der				
Authorization	'n	valu			
			Basic OAuth		
Raw	Form	Files	Login:		
Encode payl	oad Decode p	ayload	your username		
			Password:		
			your password	show password	
					OK Cancel
application/x-	www-form-urlen	coded			

Then enter Accept, Content language

formenteel							
http:// XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	8005/webservices/rest/Create_User/load_row/						
© GET ● POST ● PUT	PATCH DELETE HEAD OPTIONS Other						
Raw Form	Headers						
Add new header							
Authorization	Basic c3lzYWRtaW46c3lzYWRtaW4wMQ==						
Accept	application/json						
Content-Langauge	en-US						
Raw Form	Files (0) Payload						

Select application/json in the Content Type below.

Step 9: Enter the following Link and payload information and click on send. The payload is constructed using the xsd file generated at Step 4.

Link : http://xxx-xxx-apdv01.cli.ad:8005/webservices/rest/Create_User/load_row/

Payload:

```
{
"Create User": {
"@xmlns": "http://xmlns.oracle.com/apps/fnd/rest/create_user/load_row",
"RESTHeader": {
"xmlns": "http://xmlns.oracle.com/apps/fnd/rest/Create_User/header",
"Responsibility":"US_HRMS_MANAGER",
"RespApplication":"PER",
"SecurityGroup":"STANDARD",
"NLSLanguage":"AMERICAN",
"Org_Id" :"201"
},
"InputParameters": {
"X_USER_NAME": "TEST_USER",
"X_ENCRYPTED_USER_PASSWORD": "Welcome123",
"X_START_DATE": "14/09/2015",
"X_EMAIL_ADDRESS": "test_user@test.com"
}
}
}
```

http://svr-ebs	-apdv01.cli.ad:8	005/webserv	ces/rest/Create_User/load_row/	
⊖ GET ● PC	ST PUT	РАТСН	DELETE HEAD OPTIONS Other	
Raw	Form	Headers		
Add new heade	er -			
Authorization		Basic c3lz1	WRtaW46c3lzYWRtaW4wMQ==	×
Accept		application	'json	×
Content-Langa	auge	en-US		×
Raw	Form	Files (0)	Payload	
Encode paylo	ad Decode pa	yload		*
"@xmlns": "h	ttp://xmlns.	oracle.com	/apps/ <u>fnd</u> /rest/create_user/load_row",	
"xmlns": "ht "Responsibil	tp://xmlns.o lity":"US_HRM	racle.com/ S_MANAGER"	apps/ <u>fnd</u> /rest/Create_User/header",	•
application/jsor	ı	▼ Set	"Content-Type" header to overwrite this value.	

Clear Send

Step 10: Now to verify if user is created query in FND_USER table to check the results.

ଚଜା	boll New 1							
		· · · · · · · · · ·	¹⁰	20		50	°····· ⁸ °····· ⁹ °·····	
1	•	select * f	rom fnd_u	ser wher	'e user_name='TEST_0	SER'		
- Dat	a Grid							
	Messa		Data Grid	🔀 Auto Ti	race R DBMS Output (dis	abled)	Explain Plan	Output
+	*							ouput
:= L	JSER_	ID USER	_NAME		LAST_UPDATE_DATE	LAST_UPDATED_B	CREATION_DATE	CREATED_BY LAS
	123	70 TEST_	USER		9/15/2015 9:26:42 AM		9/15/2015 9:18:08 AM	0

Exposing Pl/Sql as REST Web Service (Function Returning single value)

Perform following steps to expose Pl/Sql as a web service

- 1. Create a Custom Package , annotate it
- 2. Create iLDT File

- 3. Upload the iLDT File to Integration Repository.
- 4. Search API in Integration Repository
- 5. Deploy API as REST Service
- 6. Create Grants for REST Service
- 7. Invoke EBS REST Service (in JSON message format)
- 8. Verify the results

Here we will take an example of Getting Employee Number based on Employee details provided when invoking the Web service.

Following steps will explain the process of exposing the Custom API that returns varchar variable as output when invoked.

Step 1: First Create a custom package xxtrc_get_emp_numer_pkg with .pls extension and annotate it Only annotate the package specification. Below code was used for this Case study.

Package Specification:

CREATE OR REPLACE PACKAGE xxtrc_get_emp_number_pkg AS

/* \$Header: \$ */

/*#

- * This interface returns the Employee Number.
- * @rep:scope public
- * @rep:product PER
- * @rep:displayname xxtrc_get_emp_number_pkg
- * @rep:lifecycle active
- * @rep:compatibility S
- * @rep:category BUSINESS_ENTITY PER_EMPLOYEE

*/

/*#

- * Returns Employee Number
- * @param P_First_name VARCHAR2 First Name
- * @param P_Last_name VARCHAR2 Last Name

* @param P_Email_address VARCHAR2 Email Address

- * @return Employee_Number
- * @rep:scope public
- * @rep:lifecycle active
- * @rep:displayname Return Employee Number
- */

FUNCTION get_emp_number (P_First_name IN VARCHAR2,

P_Last_name IN VARCHAR2,

P_Email_address IN VARCHAR2

) RETURN VARCHAR2;

END xxtrc_get_emp_number_pkg;

Package Body:

```
CREATE OR REPLACE PACKAGE BODY xxtrc get emp number pkg
AS
   FUNCTION get_emp_number (P_First_name IN VARCHAR2,
P_Last_name IN VARCHAR2,
                             P Email address IN VARCHAR2)
      RETURN VARCHAR2
   AS
      l emp num VARCHAR2 (30);
   BEGIN
      SELECT employee number
        INTO 1 emp num
        FROM per all people f
       WHERE NVL (UPPER (first name), 'X') = NVL (UPPER
(P First name), 'X')
             AND UPPER (last name) = UPPER (last name)
             AND NVL (email address, 'X') = NVL (P Email address,
'X');
      RETURN 1 emp num;
   EXCEPTION
      WHEN NO DATA FOUND
      THEN
```

```
RETURN 'No Data Found';
WHEN OTHERS
THEN
RETURN 'Invalid Data';
END get_emp_number;
END xxtrc get emp number pkg;
```

Step 2: Upload the .pls file to Oracle server.For our case study file was uploaded to custom_top Bin folder

Step 3: Create iLDT file by using following command and current Directory is where the .pls file was uploaded :

\$IAS_ORACLE_HOME/perl/bin/perl \$FND_TOP/bin/irep_parser.pl -g -v -username=sysadmin PER:patch/115/sql:xxtrc_get_emp_number_pkg.pls:12.0=xxtrc_get_emp_number_pkg.pls

Following message will appear after executing the command and it will generate iIDT file.



Step 4: Upload the generated iLDT file to Integration repository by using following command:

\$FND_TOP/bin/FNDLOAD apps/apps 0 Y UPLOAD \$FND_TOP/patch/115/import/wfirep.lct
xxtrc_get_emp_number_pkg_pls.ildt

[applmgr@svr-ebs-apdv01 bin]\$ \$FND_TOP/bin/FNDLOAD apps/apps 0 Y UPLOAD \$FND_TOP/patch/115/import/wfirep.lct xxtrc_get_emp_number_pkg_pls.ildt Log filename : L551303.log

Report filename : 0551303.out [applmgr@svr-ebs-apdv01 bin]\$

Step 5: Navigate to Integration Repository Responsibility, Click on Search.

			*	0	*	⇔		Lo	gged in As SYSADMIN	•	-1
Integration Repository Administration											
View By Product Family Welcome to the Orac	cle Integration Repository										
Cracle Applications Stute Cracle Applications Technology Family Oracle Contracts Family	y is an integral part of Oracle E-Business Suite. It is business service interface from the catalog for in	provides a complete catalo tegration with any system,	g of Ora applica	tion, or l	usiness	Suite's Bu partner.	isiness Servic	e interface	s. The tool lets users ea	sity discov	er
Oracle Discrete Manufacturing Pamily Cracle Financial Globalization Cracle Financial Globalization										Sear	ich.
Gracle Financial Payables Family Gracle Financial Receivables Gracle Financial Bervices Gracle Financial Bervices											
Cracle Funnan Resources Family Cracle Funnan Resources Family Cracle Interaction Center Family Cracle Interaction Center Family											
Oracle Marketing Family Oracle Marketing and Bales Family											
Oracle Order Management Pamily Oracle Process Manufacturing Family											

Then on search Page enter "xxtrc_get_emp_number_pkg" in Internal Name and click on Go

Interface Name Product Family Product All		2	E	Internal Name Interface Type Business Entity	xxtrc_get_emp_num All	ber_pkg	_			
Show More Search C	Go Clear All Show More Search Options									
Export 1 🔯 🗢 🖷 🔅										
Name 📥		Internal Name		Product		Туре 🛆	Source $ riangle$	Status 🛆	Description	
xtrc get emp number pkg		XXTRC_GET_EMP_NUM	IBER_PKG	Oracle Human Resources		PL/SQL	Custom	Active	This interface returns the Employee Number.	

Step 6: Click on the search result Link, it will display List of method available in the Interface Package.

PLSQL Interface : xxtrc_get_emp_number_pkg								
Internal Name XXTRC_GET_EMP_NUMBER_PKG Type PL/SQL Product Oracle Human Resources Status Business Entity Employee	Scope Interface Source	Public Custom						
Overview REST Web Service Grants								
Full Description								
This interface returns the Employee Number.								
Source Information								
Source File patch/115/sql/xxtrc_get_emp_number_pkg.pls Source Version 12.0 Source Product PER								
Procedures and Functions								
X 2 5 \$								
Name Internal Name Status Description								
Return Employee Number GET_EMP_NUMBER Active Returns Employee Number								

Click on **REST Web Service** Tab

PLSQL Interface : xxtrc_get_emp_number_pkg								
Internal Name XXTRC_GET_EMP_NUMBER_PKG Type PL/SQL Product Oracle Human Resources Status Active Business Entity Employee								
CVEIVIEW REST Web service	Grants							
* Service Alias								
REST Service Status Not De	ployed							
Service Operations								
📜 😂 🕞 🌣								
Select All Select None								
Select Name A	Internal Name	Grant						
Return Employee Number	GET_EMP_NUMBER	2						
TIP To apply any changes in C	peration, Undeploy the s	service.						
REST Service Security								

This page will display the Return Employee Number method.

Step 7: Now select Return Employee Number method (we will create a user by invoking this Service once Deployed.) and enter Service Alias (This is mandatory) and then Click on Deploy button at bottom. You will receive message that service is successfully deployed and the status will be deployed.

Integration Repository >		
Confirmation 1. Successfully deployed Web Service for interface 'xxtrc_get_emp_number_pkg' to Application Server 2. Oracle E-Business Suite Integrated SOA Gateway is not configured for SOAP Services. Please refer	My Oracle Support Note 1311068.1 to confi	gure for SOAP Services.
PLSQL Interface : xxtrc_get_emp_number_pkg		
Internal Name XXTRC_GET_EMP_NUMBER_PKG Type PL/SQL Produc Oracle Human Resources Status Active Business Entity Employee Overview REST Web Service Grants	Scope Interface Source	Public Custom
Service Alias Get_Emp_Num REST Service Status Deployed <u>View WADL</u> Verb POST		

Step 8: Now you will see the link to WADL file , Click on the link . It shows the physical location of the service endpoint where the service is hosted.

xml version="1.0" encoding="UTF-8"?	
<application< th=""><th></th></application<>	
xmlns:tns1="http://xmlns.oracle.com/apps/per/res	st/get emp num/get emp number/
" xmlns:xsd="htt	tp://www.w3.org/2001/XMLSchema"
<pre>xmlns="http://wadl.dev.java.net/2009/02"</pre>	
xmlns:tns="http://xmlns.oracle.com/apps/per/soap	provider/plsql/rest/xxtrc_get
emp number pkg/"	
targetNamespace="http://xmlns.oracle.com/apps/pe	er/soaprovider/plsql/rest/xxtr
c get emp number pkg/" name="XXTRC GET EMP NUMBE	IR_PKG" > <grammars></grammars>
<pre><include http:="" td="" xmlns="http://www.w3.org/2001/XMLSchema</pre></td><td>a''</td></tr><tr><td>href=" xxx-xxx-<=""><td></td></include></pre>	
apdv01.cli.ad:8005/webservices/rest/Get_Emp_Num/	'?XSD=GET_EMP_NUMBER_SYNCH_TYP
CDEF.msd"/>	
<resources< th=""><th>base="http://xxx-xxx-</th></resources<>	base="http://xxx-xxx-
apdv01.cli.ad:8005/webservices/rest/Get Emp Num/	/"> <resource< td=""></resource<>
path="get_emp_number/"> <method< td=""><td>name="POST"</td></method<>	name="POST"
id="GET_EMP_NUMBER"> <request><representation< td=""><td>type="tns1:InputParameters"</td></representation<></request>	type="tns1:InputParameters"
<pre>mediaType="application/xml"/><representation< pre=""></representation<></pre>	type="tns1:InputParameters"
<pre>mediaType="application/json"/><respons< pre=""></respons<></pre>	se> <representation< td=""></representation<>
type="tns1:OutputParameters" mediaType="app	<pre>plication/xml"/><representation< pre=""></representation<></pre>
type="tns1:OutputParameters"	
<pre>mediaType="application/json"/></pre>	od>
plication	
Piicacion	

Now open the link highlighted in Yellow in new Browser window, it will show the input parameters that will be used later for invoking the Web Service.

Step 9.Now Click on the Grants Tab, select return Employee Number and then click on Create Grant button

Information Oracle E-Business Suite Integrated SOA Gateway is not configured for SOAP Services. Please refer My Oracle Support Note 1311068.1 to configure for SOAP Services.									
PLSQL Interface : xxtrc_ge	PLSQL Interface : xxtrc_get_emp_number_pkg								
Internal Name XXTF Type PL/S Product Orac Status Business Entity Empl	RC_GET_EMP_NUMBE QL de Human Resources re loyee	R_PKG		Interface	Scope Source	Public Custom			
Overview REST Web Service	Grants								
Select Object and Create Grant Revoke Grant 🞇 😂 💌 🌣									
Select All Select None									
Select Name 📥	Internal Name	REST Service Operation	Grant 스						
Return Employee Number	GET_EMP_NUMBER	×							

Step 10: Select a grantee type. In our Case Study we will enter SYSADMIN and then click on Create Grant Button.

It will display confirmation message:



To **Revoke Grants** click on Revoke Grants button under Grants Tab, it will display List of users with access for revoking the Grants.

To **Undeploy** the Web services select the method and click on undeploy under the REST Web Services Tab

Step 11: Now we will invoke this web service using Advanced REST Client apps available in Google Chrome.

Please refer to the steps from **case study 1** to invoke the web service. For this case study following details were used:

Link:- http://xxx-xxx-apdv01.cli.ad:8005/webservices/rest/Get_Emp_Num/get_emp_number/

Payload:

{

```
"Get_Employee_Number": {
"@xmlns": "http://xmlns.oracle.com/apps/per/rest/Get_Emp_Num/get_emp_number",
"RESTHeader": {
"xmlns": "http://xmlns.oracle.com/apps/per/rest/Get Emp Num/header",
"Responsibility":"US_HRMS_MANAGER",
"RespApplication":"PER",
"SecurityGroup":"STANDARD",
"NLSLanguage":"AMERICAN",
"Org_Id" :"201"
},
"InputParameters": {
"P_FIRST_NAME": "Kaul",
"P_LAST_NAME": "Dipti"
ł
ł
}
```

Step 12: Enter the payload information and click on send. You will receive attached response.

```
{
    "OutputParameters" : {
        "@xmlns:xsi" : "http://www.w3.org/2001/XMLSchema-instance",
```

```
"@xmlns"
                                                                                                                                                                                                                                 :
"http://xmlns.oracle.com/apps/per/rest/Get Emp Num/get emp number/",
           "GET EMP NUMBER" : "1"
     }
}
                             Status
                                                      200 OK 🥝 Loading time: 880 ms
                     to
b
                     Scroll to to
                                                      Accept: application/json
Origin: chrome-extension://hgmloofddffdnphfgcelikdfofbjeloo
User-Agent: Mozilla/5,0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0.2454.85 Safari/537.36
Authorization: Basic U1ITQURNSU46c3lzYWRtaW4wMQ==
                              Request
                              headers
                                                       Content-Language: en-US
                                                       Content-Type: application/json 1
                                                      Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.8
                              Response
                                                      Date: Wed, 16 Sep 2015 13:13:13 GMT 💡
                                                      Date: Web, 16 Sep 2015 13:13:13 GMT *
Server: *
Content-Length: 207 *
X-ORACLE-DMS-ECID: 0057x0Fc8yy3R9_tPh8iX00061F00003M
X-Frame-Options: SAMEORIGIN
Keep-Alive: timeout=15
Connection: Keep-Alive
Content-Type: application/json *
Content-Type: application/json *
                                                      Content-Language: en
```

Exposing Pl/Sql as REST Web Service (Procedure Returning Table Type value)

Perform following steps to expose Pl/Sql as a web service

- 1. Create a Custom Package ,annotate it
- 2. Create iLDT File
- 3. Upload the iLDT File to Integration Repository.
- 4. Search API in Integration Repository
- 5. Deploy API as REST Service
- 6. Create Grants for REST Service
- 7. Invoke EBS REST Service (in JSON message format)
- 8. Verify the results

Here we will take an example of Getting Location details based on Location ID provided when invoking the Web service.

Following steps will explain the process of exposing the Custom API that returns **Table type** variable as output when invoked. The steps will remain same as used in **Case Study 2**. The package used for this Case Study is below:

Package Specification:

CREATE OR REPLACE PACKAGE xxtrc_hr_get_location_pkg

AS

```
/* $Header: $ */
```

/*#

* This is Public Interface for Location and gives location information on the basis of Location ID.

* @rep:scope public

* @rep:product PER

- * @rep:displayname xxtrc_hr_get_location
- * @rep:lifecycle active
- * @rep:compatibility S
- * @rep:category BUSINESS_ENTITY Location
- */

TYPE LOCATION_TBL IS TABLE OF HR_LOCATIONS%ROWTYPE

INDEX BY BINARY_INTEGER;

/*#

- * Returns Location Information
- * @param P_LOCATION_ID NUMBER LOCATION_ID
- * @param P_LOCATION_TBL LOCATION_TBL LOCATION_TBL
- * @rep:scope public
- * @rep:lifecycle active
- * @rep:displayname Returns Location Details
- */

PROCEDURE get_location_details (P_LOCATION_ID IN NUMBER , P_LOCATION_TBL OUT LOCATION_TBL);

END xxtrc_hr_get_location_pkg;

Package Body:

CREATE OR REPLACE PACKAGE BODY xxtrc_hr_get_location_pkg AS

```
PROCEDURE get location details (P LOCATION ID IN NUMBER
, P LOCATION TBL OUT LOCATION TBL)
   AS
      l_return_tbl LOCATION_TBL;
      l return tbl null LOCATION TBL;
                               NUMBER;
      l count
      CURSOR my cur (P LOCATION ID IN NUMBER)
      IS
         SELECT *
          FROM HR LOCATIONS
          WHERE LOCATION ID = P LOCATION ID;
   BEGIN
      FOR rec IN my cur (P LOCATION ID)
      LOOP
         l count := 1;
         l return tbl (l count).LOCATION ID := rec.LOCATION ID;
         l return tbl (l count).LOCATION CODE := rec.LOCATION CODE;
         l return tbl (l count).LOCATION USE := rec.LOCATION USE;
         l return tbl (l count).BUSINESS GROUP ID :=
rec.BUSINESS GROUP ID;
         l return tbl (l count).DESCRIPTION := rec.DESCRIPTION;
         l return tbl (l count).SHIP TO LOCATION ID :=
rec.SHIP TO LOCATION ID;
         l return tbl (l count).SHIP TO SITE FLAG :=
rec.SHIP TO SITE FLAG;
         l return tbl (l count).RECEIVING SITE FLAG :=
rec.RECEIVING SITE FLAG;
         l return tbl (l count).BILL TO SITE FLAG :=
rec.BILL TO SITE FLAG;
         l return tbl (l count).IN ORGANIZATION FLAG :=
rec.IN ORGANIZATION FLAG;
         l return tbl (l count).OFFICE SITE FLAG :=
rec.OFFICE SITE FLAG;
         l return tbl (l count).DESIGNATED RECEIVER ID :=
rec.DESIGNATED RECEIVER ID;
         l return tbl (l count).INVENTORY ORGANIZATION ID :=
            rec.INVENTORY ORGANIZATION ID;
         l return tbl (l count).TAX NAME := rec.TAX NAME;
         l return tbl (l count).INACTIVE DATE := rec.INACTIVE DATE;
         l return tbl (l count).STYLE := rec.STYLE;
         l return tbl (l count).ADDRESS LINE 1 := rec.ADDRESS LINE 1;
         1 return tbl (1 count).ADDRESS LINE 2 := rec.ADDRESS LINE 2;
         l return tbl (l count).ADDRESS LINE 3 := rec.ADDRESS LINE 3;
         l return tbl (l count).TOWN OR CITY := rec.TOWN OR CITY;
         l return tbl (l count).COUNTRY := rec.COUNTRY;
         l return tbl (l count).POSTAL CODE := rec.POSTAL CODE;
         l return tbl (l count).REGION 1 := rec.REGION 1;
         l return tbl (l count).REGION 2 := rec.REGION 2;
         l return tbl (l count).REGION 3 := rec.REGION 3;
```

```
1 return tbl (l count).TELEPHONE NUMBER 1 :=
rec.TELEPHONE NUMBER 1;
         l return tbl (l count).TELEPHONE NUMBER 2 :=
rec.TELEPHONE NUMBER 2;
         l return tbl (l count).TELEPHONE NUMBER 3 :=
rec.TELEPHONE NUMBER 3;
         l return tbl (l count).LOC INFORMATION13 :=
rec.LOC INFORMATION13;
         l return tbl (l count).LOC INFORMATION14 :=
rec.LOC INFORMATION14;
         l return tbl (l count).LOC INFORMATION15 :=
rec.LOC INFORMATION15;
         l return tbl (l count).LOC INFORMATION16 :=
rec.LOC INFORMATION16;
         l return tbl (l count).LOC INFORMATION17 :=
rec.LOC INFORMATION17;
         l return tbl (l count).LOC INFORMATION18 :=
rec.LOC INFORMATION18;
         l return tbl (l count).LOC INFORMATION19 :=
rec.LOC INFORMATION19;
         l return tbl (l count).LOC INFORMATION20 :=
rec.LOC INFORMATION20;
         l return tbl (l count).ATTRIBUTE CATEGORY :=
rec.ATTRIBUTE CATEGORY;
         l return tbl (l count).ATTRIBUTE1 := rec.ATTRIBUTE1;
         l return tbl (l count).ATTRIBUTE2 := rec.ATTRIBUTE2;
         l return tbl (l count).ATTRIBUTE3 := rec.ATTRIBUTE3;
         l return tbl (l count).ATTRIBUTE4 := rec.ATTRIBUTE4;
         l return tbl (l count).ATTRIBUTE5 := rec.ATTRIBUTE5;
         l return tbl (l count).ATTRIBUTE6 := rec.ATTRIBUTE6;
         l return tbl (l count).ATTRIBUTE7 := rec.ATTRIBUTE7;
         l return tbl (l count).ATTRIBUTE8 := rec.ATTRIBUTE8;
         l return tbl (l count).ATTRIBUTE9 := rec.ATTRIBUTE9;
         l return tbl (l count).ATTRIBUTE10 := rec.ATTRIBUTE10;
         l return tbl (l count).ATTRIBUTE11 := rec.ATTRIBUTE11;
         l return tbl (l count).ATTRIBUTE12 := rec.ATTRIBUTE12;
         l return tbl (l count).ATTRIBUTE13 := rec.ATTRIBUTE13;
         l return tbl (l count).ATTRIBUTE14 := rec.ATTRIBUTE14;
         l return tbl (l count).ATTRIBUTE15 := rec.ATTRIBUTE15;
         l return tbl (l count).ATTRIBUTE16 := rec.ATTRIBUTE16;
         l return tbl (l count).ATTRIBUTE17 := rec.ATTRIBUTE17;
         l return tbl (l count).ATTRIBUTE18 := rec.ATTRIBUTE18;
         l return tbl (l count).ATTRIBUTE19 := rec.ATTRIBUTE19;
         l return tbl (l count).ATTRIBUTE20 := rec.ATTRIBUTE20;
         l return tbl (l count).GLOBAL ATTRIBUTE CATEGORY :=
            rec.GLOBAL ATTRIBUTE CATEGORY;
         l return tbl (l count).GLOBAL ATTRIBUTE1 :=
rec.GLOBAL ATTRIBUTE1;
         l return tbl (l count).GLOBAL ATTRIBUTE2 :=
rec.GLOBAL ATTRIBUTE2;
         l return tbl (l count).GLOBAL ATTRIBUTE3 :=
rec.GLOBAL ATTRIBUTE3;
```

```
l return tbl (l count).GLOBAL ATTRIBUTE4 :=
rec.GLOBAL ATTRIBUTE4;
         l return tbl (l count).GLOBAL ATTRIBUTE5 :=
rec.GLOBAL ATTRIBUTE5;
         l return tbl (l count).GLOBAL ATTRIBUTE6 :=
rec.GLOBAL ATTRIBUTE6;
         l return tbl (l count).GLOBAL ATTRIBUTE7 :=
rec.GLOBAL ATTRIBUTE7;
         l return tbl (l count).GLOBAL ATTRIBUTE8 :=
rec.GLOBAL ATTRIBUTE8;
         l return tbl (l count).GLOBAL ATTRIBUTE9 :=
rec.GLOBAL ATTRIBUTE9;
         l return tbl (l count).GLOBAL ATTRIBUTE10 :=
rec.GLOBAL ATTRIBUTE10;
         l return tbl (l count).GLOBAL ATTRIBUTE11 :=
rec.GLOBAL ATTRIBUTE11;
         l return tbl (l count).GLOBAL ATTRIBUTE12 :=
rec.GLOBAL ATTRIBUTE12;
         l return tbl (l count).GLOBAL ATTRIBUTE13 :=
rec.GLOBAL ATTRIBUTE13;
         l return tbl (l count).GLOBAL ATTRIBUTE14 :=
rec.GLOBAL ATTRIBUTE14;
         l return tbl (l count).GLOBAL ATTRIBUTE15 :=
rec.GLOBAL ATTRIBUTE15;
         l return tbl (l count).GLOBAL ATTRIBUTE16 :=
rec.GLOBAL ATTRIBUTE16;
         l return tbl (l count).GLOBAL ATTRIBUTE17 :=
rec.GLOBAL ATTRIBUTE17;
         l return tbl (l count).GLOBAL ATTRIBUTE18 :=
rec.GLOBAL ATTRIBUTE18;
         l return tbl (l count).GLOBAL ATTRIBUTE19 :=
rec.GLOBAL ATTRIBUTE19;
         l return tbl (l count).GLOBAL ATTRIBUTE20 :=
rec.GLOBAL ATTRIBUTE20;
         l return tbl (l count).LAST UPDATE DATE :=
rec.LAST UPDATE DATE;
         l return tbl (l count).LAST UPDATED BY :=
rec.LAST UPDATED BY;
         l return tbl (l count).LAST UPDATE LOGIN :=
rec.LAST_UPDATE LOGIN;
         l return tbl (l count).CREATED BY := rec.CREATED BY;
         l_return_tbl (l_count).CREATION DATE := rec.CREATION DATE;
         l return tbl (l count).ENTERED BY := rec.ENTERED BY;
         l return tbl (l count).TP HEADER ID := rec.TP HEADER ID;
         l return tbl (l count).ECE TP LOCATION CODE :=
rec.ECE TP LOCATION CODE;
         l return tbl (l count).OBJECT VERSION NUMBER :=
rec.OBJECT VERSION NUMBER;
         l return tbl (l count).LEGAL ADDRESS FLAG :=
rec.LEGAL ADDRESS FLAG;
         1 return tbl (1 count).TIMEZONE CODE := rec.TIMEZONE CODE;
```

```
l_count := l_count + 1;
END LOOP;
P_LOCATION_TBL:= l_return_tbl;
EXCEPTION
WHEN NO_DATA_FOUND
THEN
P_LOCATION_TBL:= l_return_tbl_null;
WHEN TOO_MANY_ROWS
THEN
P_LOCATION_TBL:=l_return_tbl_null;
END get_location_details;
END xxtrc_hr_get_location_pkg;
```

Step 1: First create a custom package xxtrc_hr_get_location_pkg with help of attachment below with .pls extension and annotate it.

Step 2: Upload the .pls file to Oracle server. For our case study file was uploaded to custom_top Bin folder

Step 3: Create iLDT file by using following command and current Directory is where the .pls file was uploaded:

\$IAS_ORACLE_HOME/perl/bin/perl \$FND_TOP/bin/irep_parser.pl -g -v -username=sysadmin PER:patch/115/sql:xxtrc_hr_get_location_pkg.pls:12.0=xxtrc_hr_get_location_pkg.pls

Step 4: Upload the generated iLDT file to Integration repository by using following command:

\$FND_TOP/bin/FNDLOAD apps/apps 0 Y UPLOAD \$FND_TOP/patch/115/import/wfirep.lct xxtrc_hr_get_location_pkg_pls.ildt

Step 5: Navigate to Integration Repository Responsibility, Click on Search. Then on search Page enter "xxtrc_hr_get_location_pkg" in Internal Name and click on Go

Step 6: Click on the search result Link, it will display List of method available in the Interface Package. Click on **REST Web Service** Tab.

Step 7: Now select returns Location method (we will create a user by invoking this Service once Deployed.) and enter Service Alias (This is mandatory) and then Click on Deploy button at bottom. You will receive message that service is successfully deployed and the status will be deployed.

Step 8: Now you will see the link to WADL file, Click on the link. It shows the physical location of the service endpoint where the service is hosted.

xmlns="http://wadl.dev.java.net/2009/02"

xmlns:tns="http://xmlns.oracle.com/apps/per/soaprovider/plsql/rest/xxtrc hr g
et location pkg/"

et location pkg/"
targetNamespace="http://xmlns.oracle.com/apps/per/soaprovider/plsql/rest/xxtr
c hr get location pkg/" name="XXTRC HR GET LOCATION PKG"><grammars><include
xmlns="http://www.w3.org/2001/XMLSchema"
href="medge//www.w3.org/2001/XMLSchema"</pre>

apdv01.cli.ad:8005/webservices/rest/Get Location Details/7XSD=GET LOCATION DE

CAILS SYNCH TYPEDEF.mcd"/> </grammars><resources

<resources< th=""><th>base="http://xxx-xxx-</th></resources<>	base="http://xxx-xxx-
apdv01.cli.ad:8005/webservices/rest/Get Lo	cation Details/"> <resource< th=""></resource<>
path="get_location_details/"> <method< th=""><th>name="POST"</th></method<>	name="POST"
id="GET LOCATION DETAILS"> <request><repres< td=""><td>entation</td></repres<></request>	entation
type="tns1:InputParameters" mediaType	e=" application/xml "/> <representation< td=""></representation<>
type="tns1:InputParameters"	
<pre>mediaType="application/json"/><r< pre=""></r<></pre>	esponse> <representation< td=""></representation<>
type="tns1:OutputParameters" mediaType	e=" application/xml "/> <representation< td=""></representation<>
type="tns1:OutputParameters"	
<pre>mediaType="application/json"/><</pre>	<pre>/method></pre>
plication> Now open the the link highlighted in Ye	llow in new Browser window, it will show the
input parameters that will be used later for invoking the	Web Service.

Step 9.Now Click on the Grants Tab, select return Employee Number and then click on Create Grant button

Step 10: Select a grantee type. In our Case Study we will enter SYSADMIN and then click on Create Grant Button.

Step 11: Now we will invoke this web service using Advanced REST Client apps available in Google Chrome.

Please refer to the steps from case study 1 to invoke the web service. For this case study following details was used:

Link:- http://xxx-xxx-apdv01.cli.ad:8005/webservices/rest/Get_Location_Details/get_location_details/

Payload:

"Get_Location_Details": {
 "@xmlns": "http://xmlns.oracle.com/apps/per/rest/Get_Location_Details/get_location_details",
 "RESTHeader": {
 "xmlns": "http://xmlns.oracle.com/apps/per/rest/Get_Location_Details/header",
 "Responsibility": "US_HRMS_MANAGER",
 "RespApplication": "PER",
 "SecurityGroup": "STANDARD",
 "NLSLanguage": "AMERICAN",
 "Org_Id" : "201"
 },
 "InputParameters": {
 "P_LOCATION_ID": "421" }}}

Step 12: Enter the payload information and click on send. You will receive status and response.

Conclusion

This document gives you an overview of the Integrated SOA Gateway and REST Web Services .It also gives you step by step instructions on exposing Pl/Sql as REST Web from scratch. You should now be able to create REST Web Service either using Standard API or Custom API.

References

- 1. Oracle E-Business Suite Integrated SOA Gateway Implementation Guide
- 2. <u>https://blogs.oracle.com/stevenChan/entry/how_to_publish_pl_sql</u>