**Apply SOA Patches 30482761 and 32363659 FOR NXSD**

1. Download the following SOA Patches listed above
2. Navigate to Oracle\_Home
3. In this case C:\orafmw\
4. Navigate to OPatch folder
5. cd C:\orafmw\OPatch
6. Run the following code

opatch lsinventory

1. Unzip the files 30482761 and 32363659 onto the C: drive
2. Navigate into the first folder
3. cd C:\ 30482761
4. Inside the folder run the following code

C:\orafmw\OPatch\opatch.bat apply

1. Do the same for the folder 32363659
2. After installation then restart jdev using the code below

cd C:\orafmw\jdeveloper\jdev\bin

jdev -clean

**Downloading Jackson jar files**

1. Also Download the following files and place them in the following folder

com.fasterxml.jackson.databind-2.10.5.1.LIFERAY-PATCHED-1.jar  
jackson-core-2.13.1.jar  
jackson-jr-annotation-support-2.13.1.jar

1. Download the files from the following url below

[Maven Central Repository Search](https://search.maven.org/search)

1. Place the jar files in the folder below:

C:\orafmw\jdeveloper\maven-archetype

**POM File Creation**

1. Create a pom file in the project
2. Navigate to the following url and search for jackson-core

[Maven Central Repository Search](https://search.maven.org/search)

1. Download the pom file
2. Copy the contents of the pom file and use it to replace the contents of the pom file created in step 1 above
3. Under dependencies
4. Add the 3 jar files in the section “Downloading Jackson JAR files”

Executing the Composite

1. **Create JSONs**
2. Create a JSON file for the Native format Input – Request for the REST Adapter
3. Create a JSON file for the Native format Output – Response for the REST Adapter
4. **Create Schema Files for the BPEL’s Web Service**
5. Create a Schema File for the Project having – Request and Response for the Project
6. **Create and Configure the Rest Adapter**
7. Drag the Rest Adapter from the Components section to Create the Rest Adapter
8. The “Rest Binding Configuration Wizard – Step 1 of 2” dialog opens,
   1. Name the Rest Binding
   2. Select “Reference” in the “Type:” section
   3. select the Check box “Response will be invoked by components using WSDL interfaces”
   4. select the Check box “Enforce XML Schema Ordering”
9. Click Next
10. The “REST Binding Configuration Wizard – Step 2 of 2” appears
11. Under the “Resources” dialog box
12. In the “Base Url” Dialog enter the “Base Url: <http://camltest.azurewebsites.net>”
13. In the “Resources” section select the existing resource “/” and click “edit” icon at the center
14. The Update REST Resource dialog box appears.
15. In the “Relative Path: ” section, enter the relative path [“camluam/api/v1/auth/login](http://camluam/api/v1/auth/login)”
16. Click Ok
17. In the “Operations Binding” section, Click on the “+” icon to add an operation
18. In the “Operation:” section, enter “Login”
19. In the “Resource:” section, the resource is selected by default. Or select it if not selected
20. In the “HTTP Verb:” section select “POST”
21. In the “Request” tab, click the “wheel” icon
22. The “Native Format Builder: Welcome” dialog box opens
23. Click Next
24. The “File Name and Directory” dialog box appears
25. Leave default values (or rename accordingly) and click Next
26. “Choose Type” dialog box appears
27. In the “Select file type: ” section, select “JSON Interchange Format”
28. Click Next
29. “JSON File Description” dialog box appears
30. In the “File name: ” section select an existing JSON file

Since no input parameter is needed, simply create a json file with an empty tag {}

1. Leave defaults for the “Target namespace”, “Root Element”, “Character set”
2. Click Next
3. “Generate Native Format Schema File” dialog box appears

An xml is generated to convert the JSON objects entered earlier to xml format

1. Click Next
2. “Native Format Builder: Finish” dialog box appears
3. Click Finish
4. Now configure the “Response” tab.
5. Click on the “Response” tab
6. Select “JSON” check box
7. The “Schema URL:” section appears
8. Click on the “wheel icon” next to the “search” icon
9. “Native Format Builder: Welcome” dialog box appears.
10. Click Next
11. The “File Name and Directory” dialog box appears
12. Leave default values (or rename accordingly) and click Next
13. “Choose Type” dialog box appears
14. In the “Select file type: ” section, select “JSON Interchange Format”
15. Click Next
16. “JSON File Description” dialog box appears
17. In the “File name: ” section select an existing JSON file

Since an input parameter is needed, select a previously created JSON file

1. Leave defaults for the “Target namespace”, “Root Element”, “Character set”
2. Click Next
3. “Generate Native Format Schema File” dialog box appears

An xml is generated to convert the JSON objects entered earlier to xml format

1. Click Next
2. “Native Format Builder: Finish” dialog box appears
3. Click Finish
4. Click “OK” to exit the “REST Operation Binding” dialog box
5. Click “Finish” to exit the “REST Binding Configuration Wizard – Step 2 OF 2”
6. **Create and Configure the BPEL**
7. Right click on the center of the composite and insert a BPEL Process
8. The “BPEL Process” dialog box appears
9. In the “Name:” section enter a name for the BPEL
10. In the “Template: ” section, select “Synchronous BPEL Process”
11. Select “Expose as a SOAP service” Check box
12. In the “Input: ” box click on the “search” icon
13. The “Type Chooser” dialog box appears
14. Expand the “Project Schema Files”
15. Expand an initially created Schema
16. Select “Request” and click OK
17. In the “Output: ” box click on the “search” icon
18. The “Type Chooser” dialog box appears
19. Expand an initially created Schema
20. Select “Response” and click OK

Note:

If input from BPEL is being passed as input to REST, the schemas must match.

Also if output from BPEL is being passed as output to REST, the schemas must match.

1. Click Ok
2. Next Wire the BPEL Process to the REST Adapter

To do so in 16 above, Click on the Arrow on the right of the BPEL icon and drag it to the Arrow on the left of the REST Adapter

1. Double Click on BPEL Process icon
2. Drag and drop “Invoke” from the components into the region between “receiveInput” and “replyOutput”
3. Wire “Invoke” to the “RestENairaLogin” REST Adapter
4. The “Edit Invoke” dialog box appears
5. In the “Name:” section enter the name of Invoke
6. In the “Operation:” section select the operation (Login)
7. In the “Input” tab click on the “+” sign
8. Click Ok
9. In the “Output” tab, click on the “+” sign
10. Click Ok
11. In the “Partner Link:” section, the partner link is selected by default
12. Click “Apply” and then “Ok”
13. Drag the “Assign” component from the components to the region between “receiveInput” and the “Invoke”
14. Double click on the Assign
15. The “Edit Assign” dialog box appears
16. In the “Copy Rules” tab
17. Wire “inputVariable” of the BPEL to the “inputVariable” of the Invoke
18. Click “Apply” and click “OK”
19. Drag another “Assign” component from the components to the region between “Invoke” and the “replyOutput”
20. Double click on the Assign
21. The “Edit Assign” dialog box appears
22. Wire “outputVariable” of the “Invoke” to the “outputVariable” of the BPEL
23. Click “Apply” and click “OK”
24. Click Save
25. Click on the “Source” tab of the BPEL
26. Assign the Header Parameters for the REST Adapter
27. Copy the following code to the Invoke section

<Invoke….>  
 <bpelx:toProperties>  
 <bpelx:toProperty name="rest.binding.http.user\_id">"bulkpayment"</bpelx:toProperty>  
 <bpelx:toProperty name="rest.binding.http.password">"QWNjZXNzUEBzc3dvcmQxMjM$="</bpelx:toProperty>  
 </bpelx:toProperties>

</Invoke>

1. The two properties are the user\_id and password header parameters
2. Right click on the Project
3. Click Deploy
4. Click on the Project to deploy

Configure Rest Adapter

1. After configuring the Rest Adapter and Wiring the BPEL Process to the Rest Adapter, double click on the BPEL Process.
2. Open the Source Code of the BPEL process
3. Modify the Invoke section like below to add the header parameters (user\_id and password)

<Invoke….>

<bpelx:toProperties>

<bpelx:toProperty name="rest.binding.http.user\_id">"bulkpayment"</bpelx:toProperty>

<bpelx:toProperty name="rest.binding.http.password">"QWNjZXNzUEBzc3dvcmQxMjM$="</bpelx:toProperty>

</bpelx:toProperties>

</Invoke>

1. The two properties are the user\_id and password header parameters.