



TGH

Making Integrations Simpler



Custom Function in OIC

Author

Sourav Karmakar



Custom Function in OIC

- **What is Custom Function?**
 - ➔ Extension libraries provide a means to register and organize JavaScript for use in integrations. Library functions are automatically available for you to drag from the Actions palette to your orchestration integrations.
- In this blog, we will see how to define a custom function in OIC and then use it inside our Integration.

Use Case:

We are required to develop a calculator Rest API in OIC using Custom Function. We will define a custom function (in JavaScript) and later use this function in an Integration.

Approach:

- Firstly, we will register the libraries that contain our custom JavaScript function. Once it gets registered, library functions are automatically available for us to drag and drop from the Actions palette for orchestration Integrations.
- Secondly, we will use this custom function in one integration.
- Thirdly, Testing of Integration.

JavaScript code: We have to save this JavaScript code in a file like calculator.js

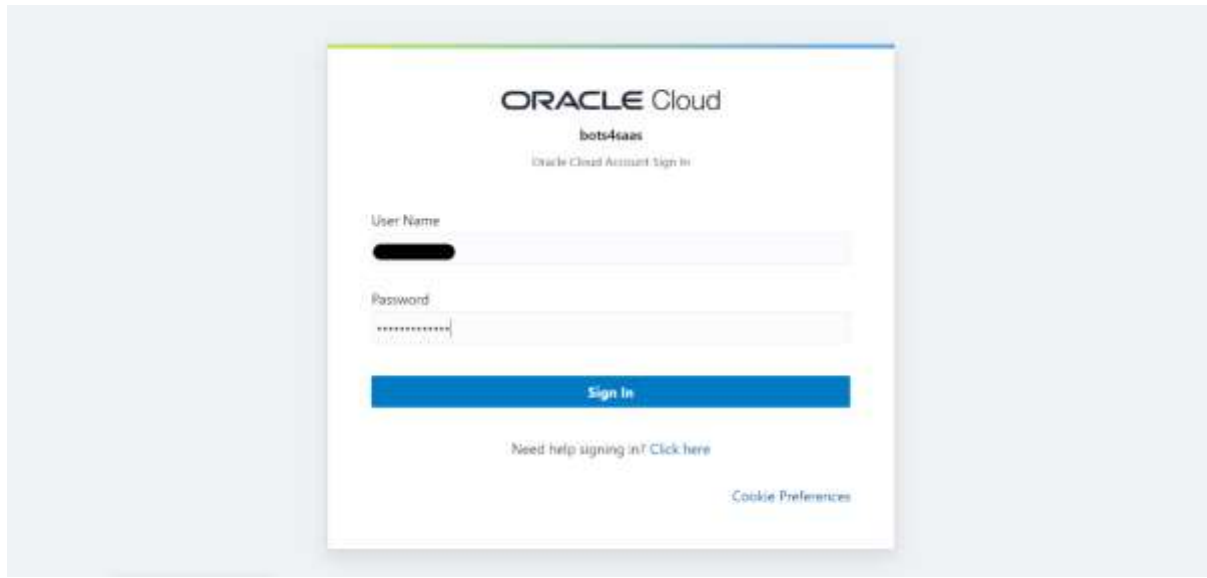
```
js calculator.js > ...
1  function calculator(number1, number2, operator) {
2      var output;
3      if (operator === "addition") {
4          output = number1 + number2;
5      } else if (operator === "subtract") {
6          output = number1 - number2;
7      } else if (operator === "multiply") {
8          output = number1 * number2;
9      } else if (operator === "divide") {
10         if (number2 !== 0) {
11             output = number1 / number2;
12         } else {
13             output = "Cannot divide by zero!";
14         }
15     } else {
16         output = "Invalid operator";
17     }
18     return output;
19 }
```

©[TGH Software Solutions Pvt. Ltd.](#)

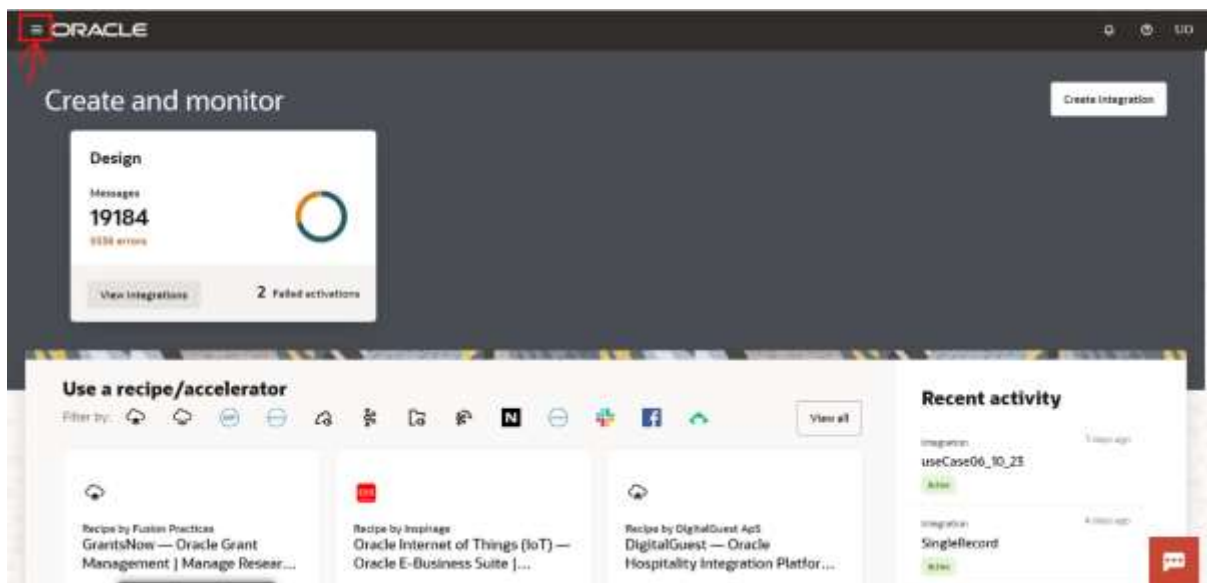
Now go to the OIC Integration Platform to develop a Calculator API

Step 1: Go to the Sign-in page.

Use Username and Password to log into the OIC Instance.



Step 2: Click on the hamburger button.

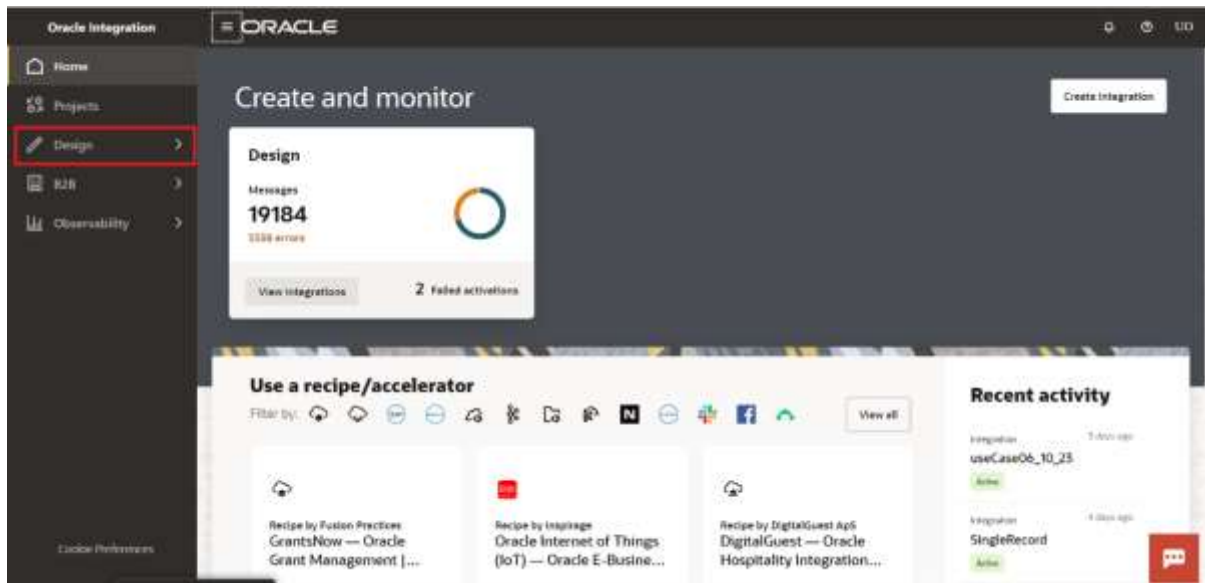


©TGH Software Solutions Pvt. Ltd.

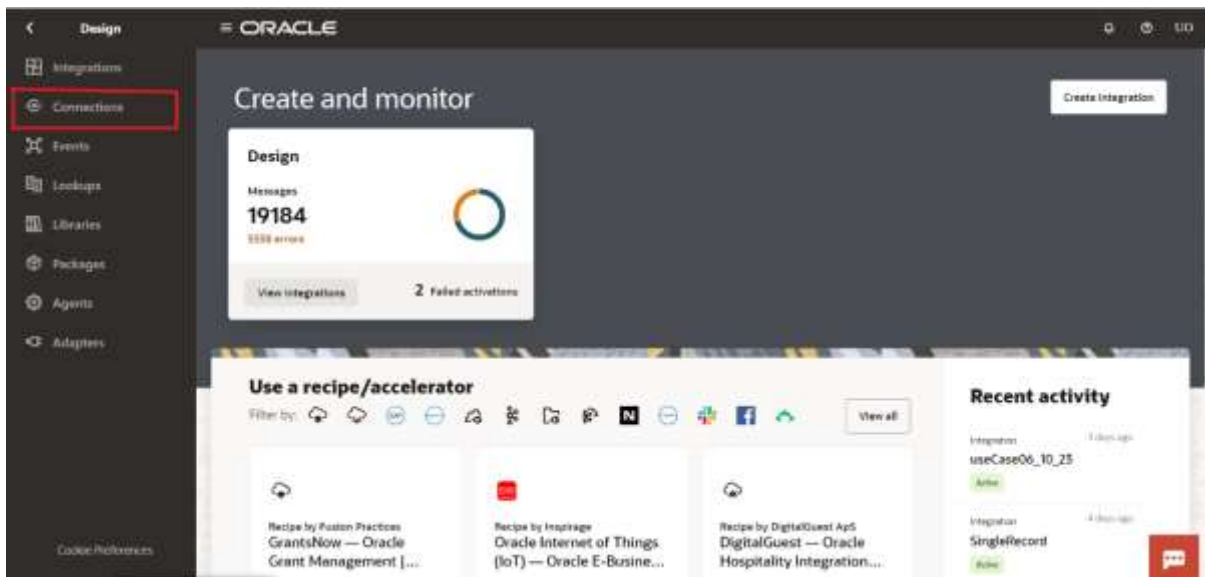
No part of this document may be copied, reproduced, republished, uploaded, posted, publicly displayed, encoded, translated, transmitted or distributed in any way to any other computer, server, website or other medium for publication or distribution, without TGH's prior written consent



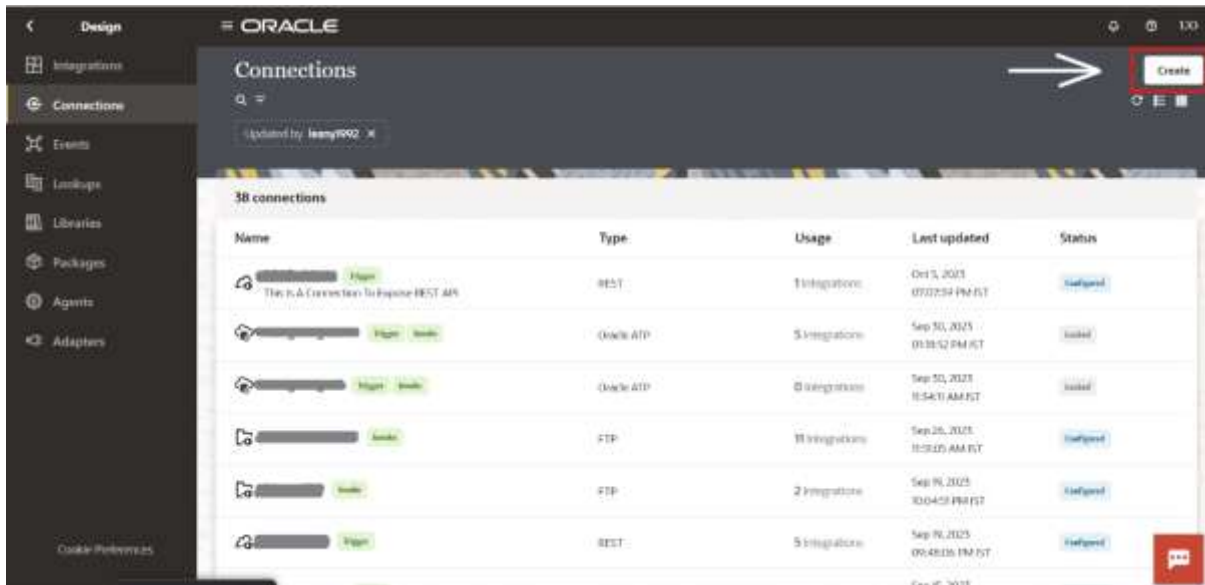
Step 3: Click on the Design tab.



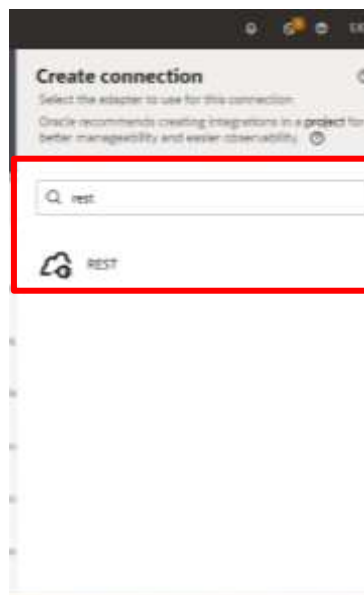
Step 4: Click on the Connections tab to create a Rest Connector.



Step 5: Click on the Create button to configure a Rest Connector.



Step 6: Search Rest in the search box and select the Rest Adapter.

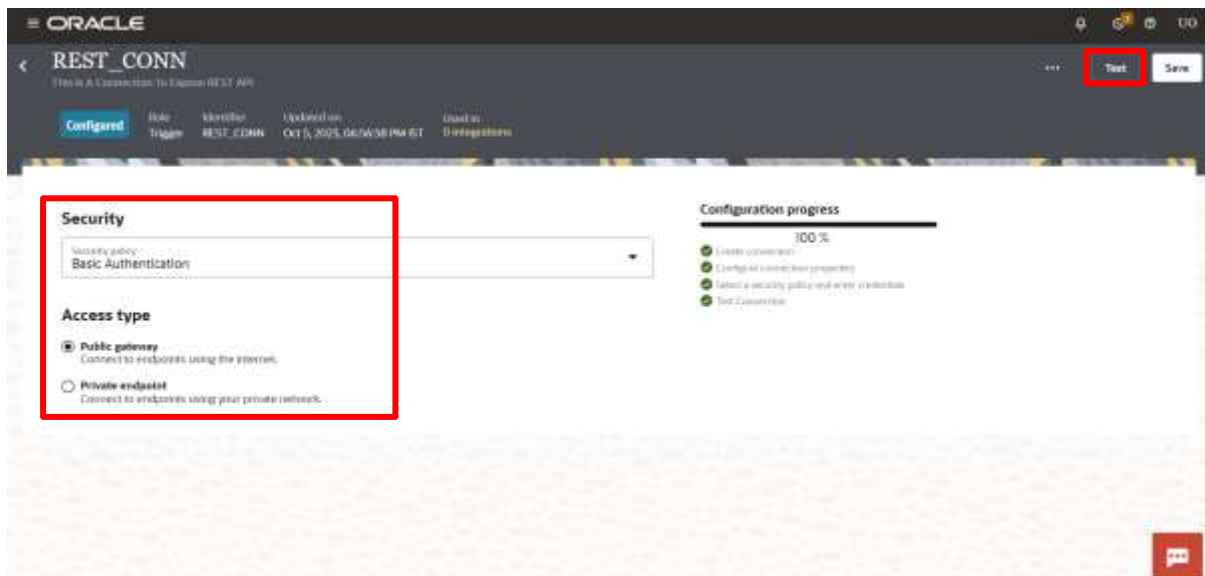


Step 7: Give a suitable name to the Rest Connector and the role should be Trigger as it is an inbound Integration and click on the Create button.

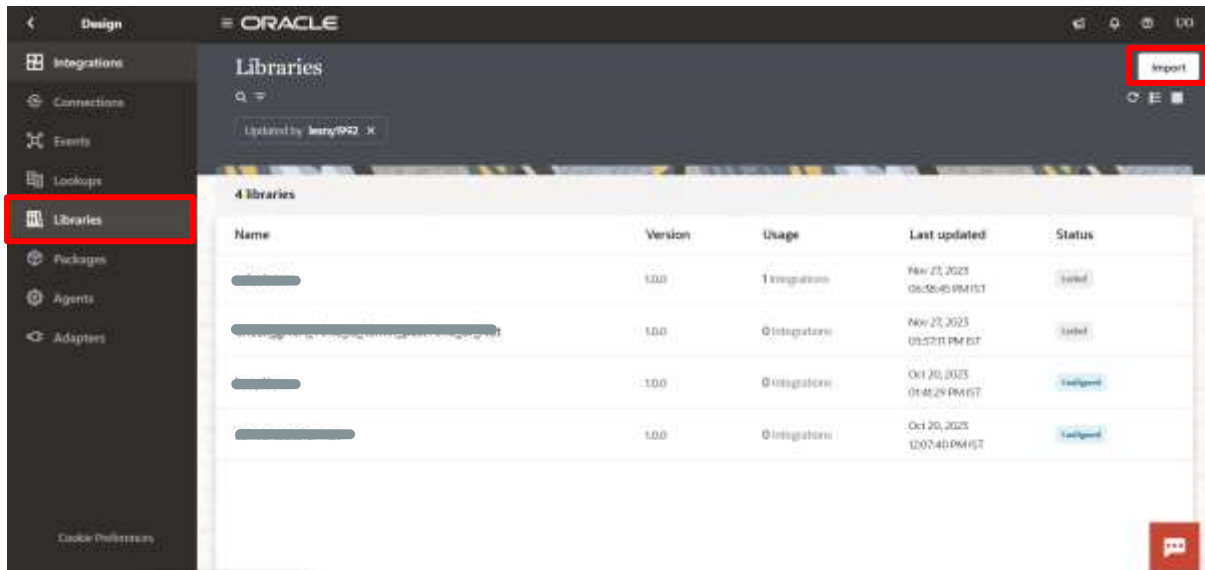


NOTE: Identifier should populate automatically but we can change this and need to give any unique name.

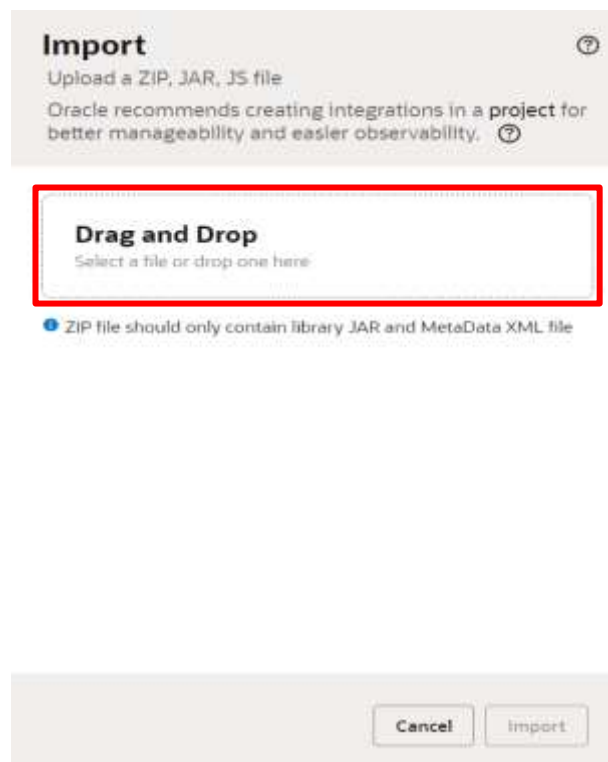
Step 8: Fill the required credentials and click on Test.



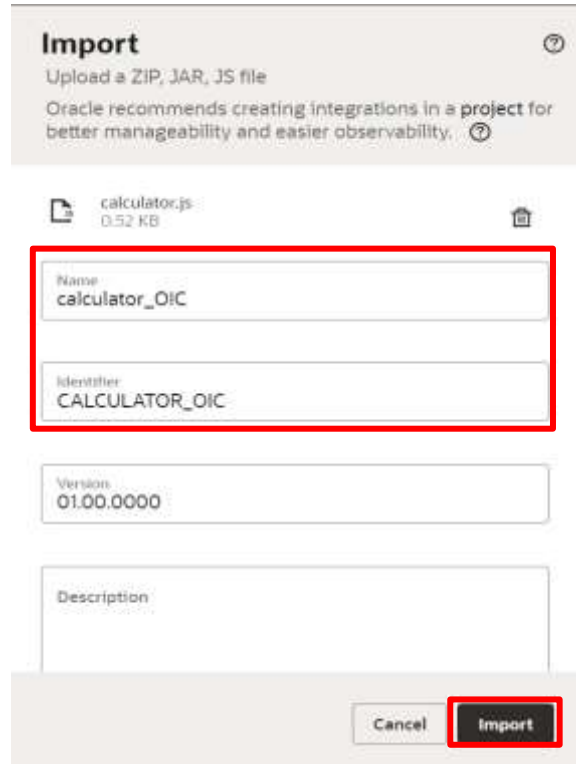
Step 9: Now Click on the Libraries tab and click on the Import button to import a library.



Step 10: After that, we can see the Import page, here we need to drag and drop our calculator.js file.



Step 11: After Drag and Drop we can see that kind of page, here we need to fill in the required details and click on the Import button library.



Import ⓘ
Upload a ZIP, JAR, JS file
Oracle recommends creating integrations in a project for better manageability and easier observability. ⓘ

calculator.js
0.52 KB

Name
calculator_OIC

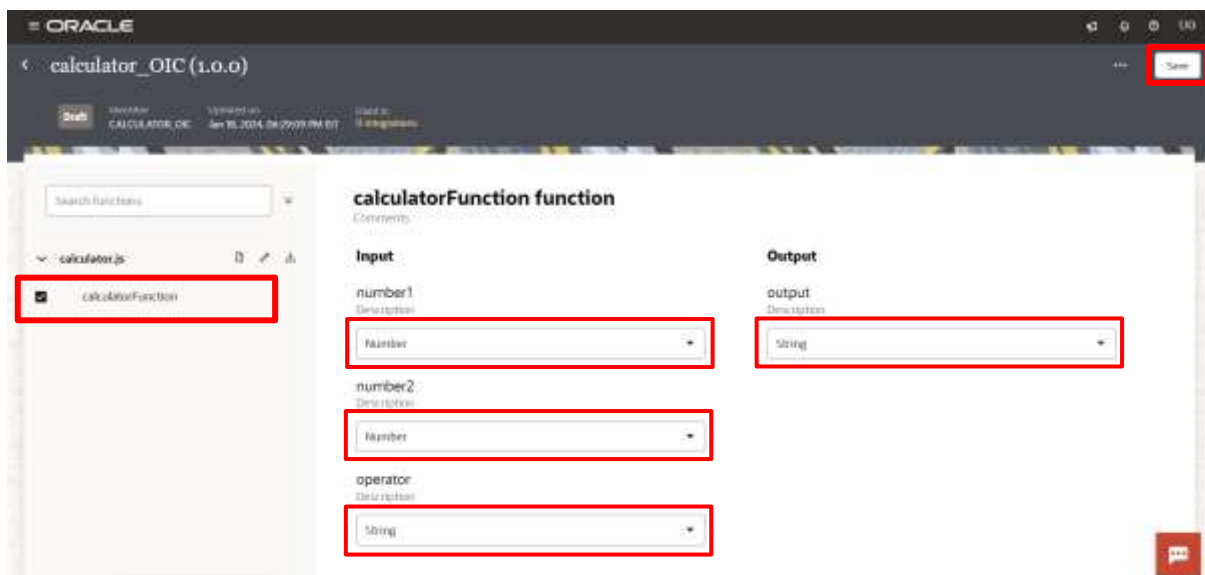
Identifier
CALCULATOR_OIC

Version
01.00.0000

Description

Cancel Import

Step 12: Now we can see the Function Name on the left side, and we need to Click on the check box and we need to change the input and output data Type, and click on the Save button.



ORACLE
calculator_OIC (1.0.0) Save

calculator_OIC
Draft
CALCULATOR_OIC
Jan 19, 2024, 04:09:09 PM EDT
Used by: 0 integrations

Search functions

calculator.js
calculatorFunction

calculatorFunction function
Comments

Input

number1
Description
Number

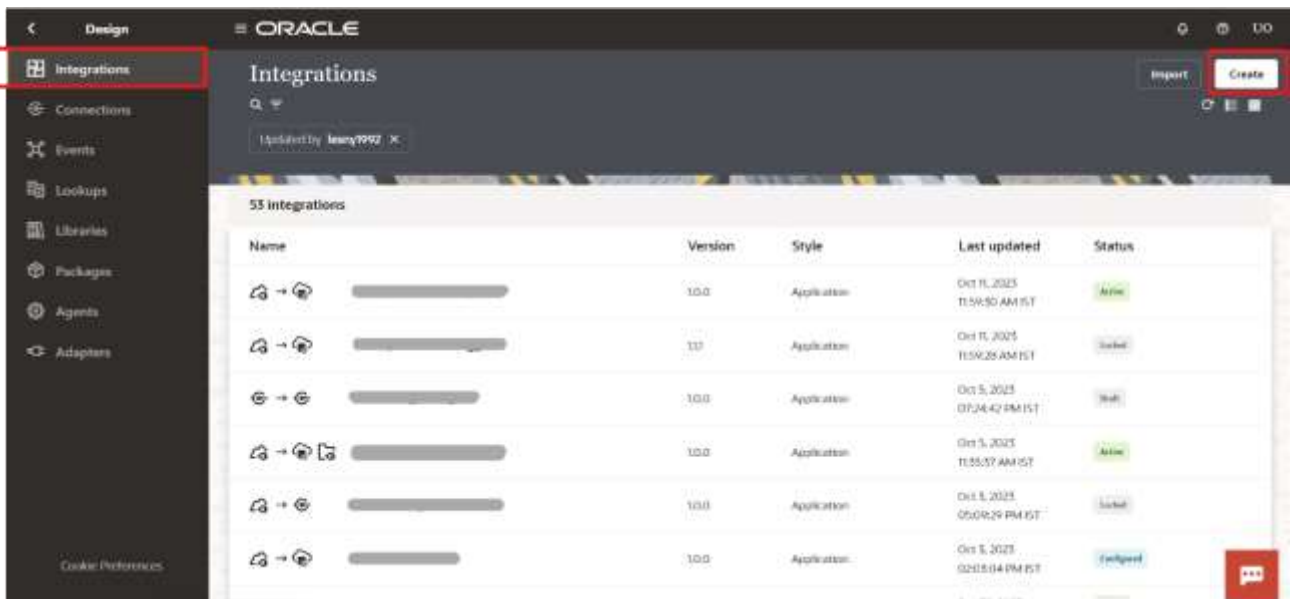
number2
Description
Number

operator
Description
String

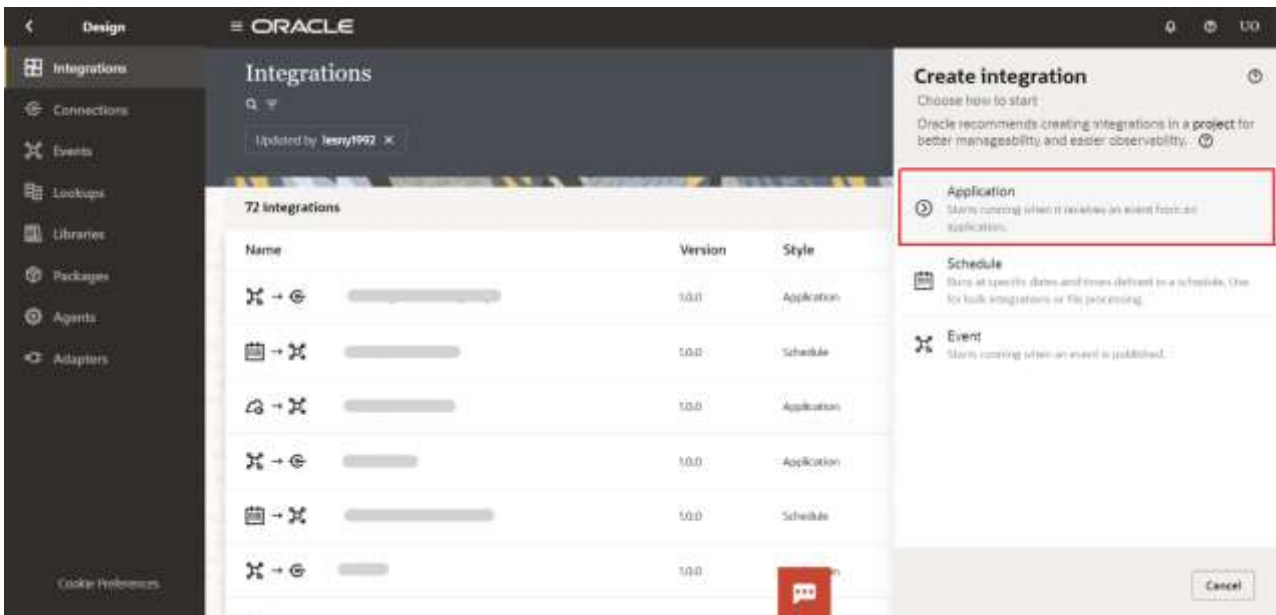
Output

output
Description
String

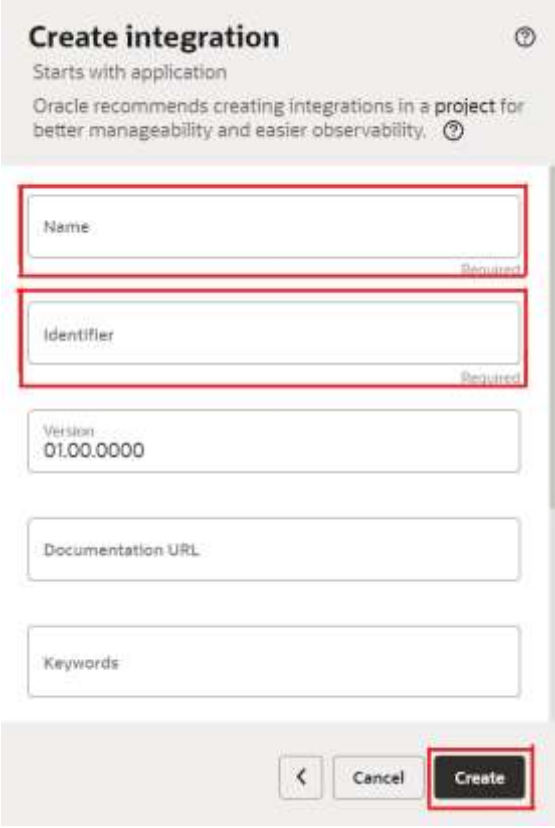
Step 13: Click on the Integration tab and click on Create.



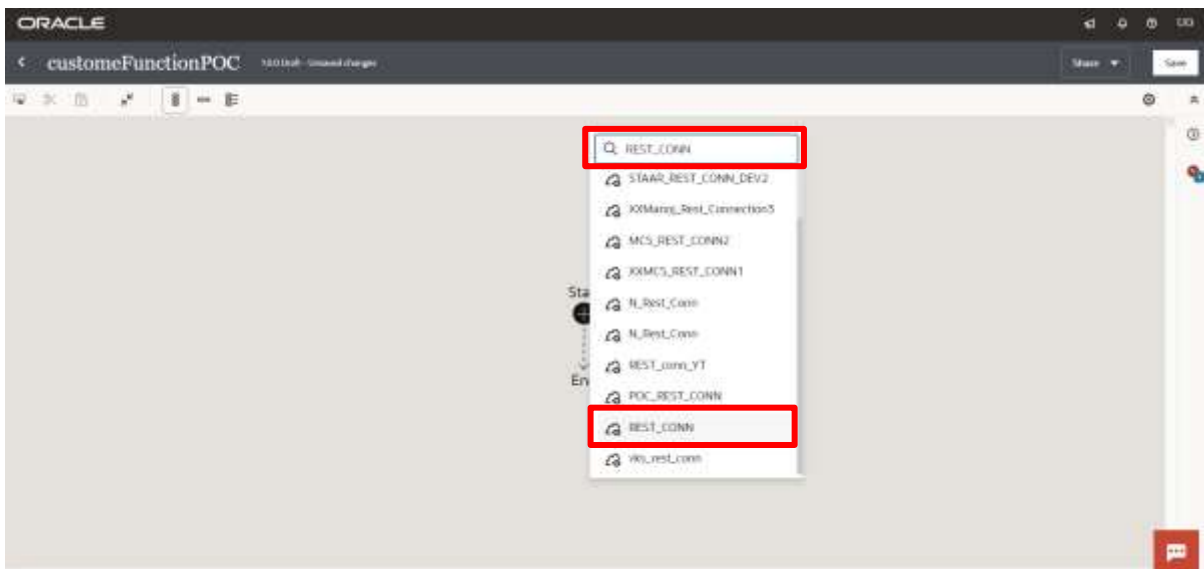
Step 14: Now Click on the Application tab to create a real-time Integration.



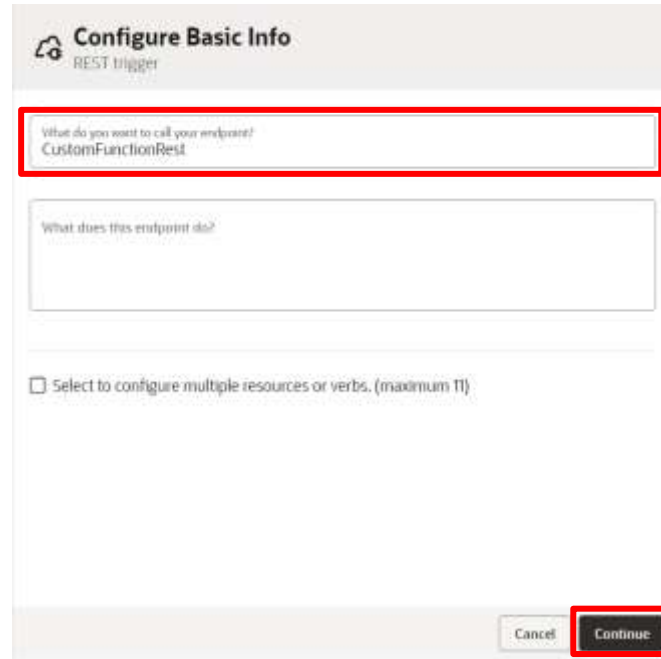
Step 15: In this create integration page give any suitable name and unique Identifier name and click on create button.



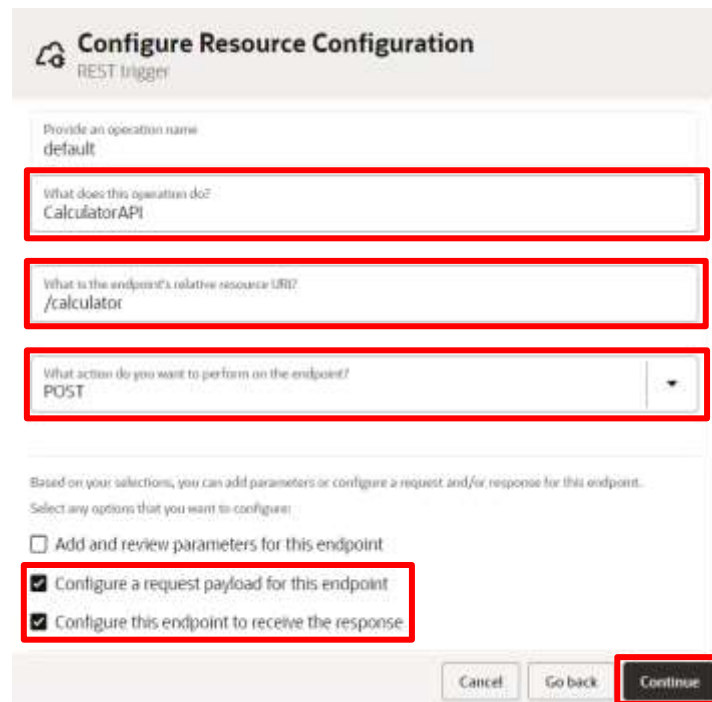
Step 16: On this page need to search the Rest connector that we created previously, we need to select that.



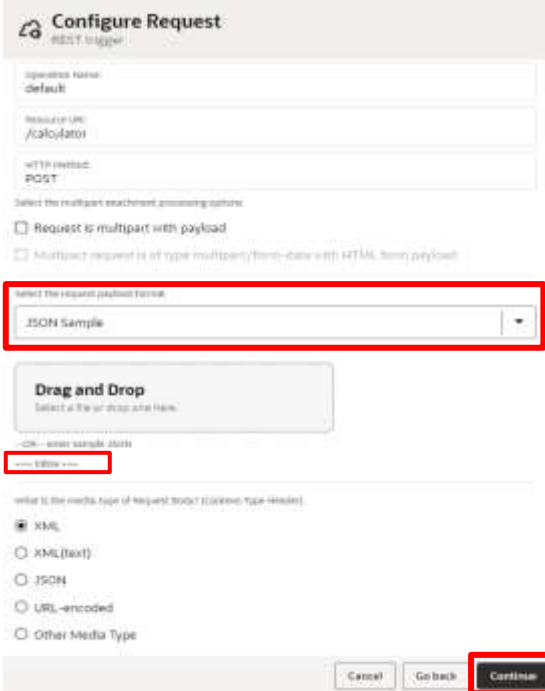
Step 17: After that, it will display the Basic Info page, where we need to fill in the required field and click on the Continue button.



Step 18: After that, it will display the Configure Resource page, where we need to fill in the required fields, and check 2 check boxes for request and response profile, and click on the Continue button.



Step 19: After that, it will display the Configure Request page, where we need to select the Request payload format as JSON Sample and click on the inline link to configure the JSON request profile.



Configure Request
REST Client

Operation Name: Default

Resource URI: /calculator

HTTP Method: POST

Select the request payload format:

Request is multipart with payload

Multipart request is of type multipart/form-data with HTML form payload

Select the request payload format:

JSON Sample

Drag and Drop
Select a file or drop one here.

OK - JSON Sample (JSON)

JSON Sample

Select the media type of request body (Default: application/json)

XML

XML(text)

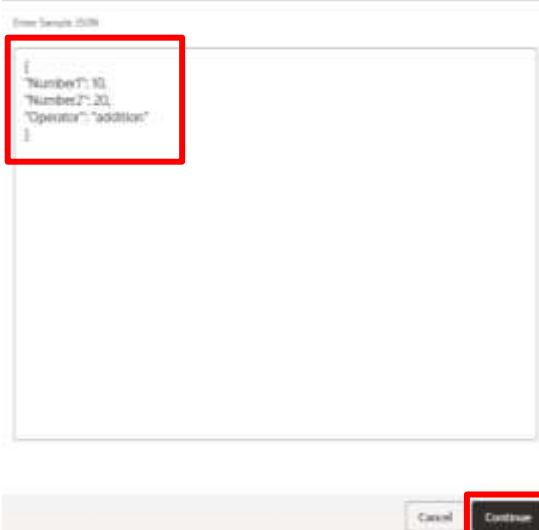
JSON

URL-encoded

Other Media Type

Cancel Go back Continue

Step 20: After that, we will get a pop-up, where we need to define our request JSON format and click on the Continue button.



Enter Sample JSON

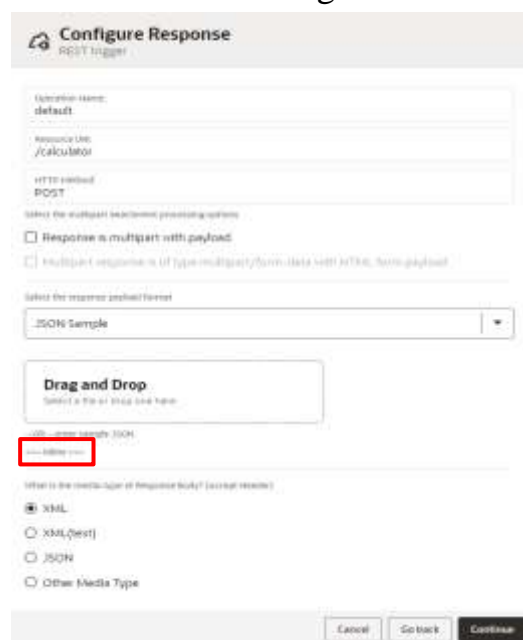
```
{  
  "Number1": 10,  
  "Number2": 20,  
  "Operator": "addition"  
}
```

Cancel Continue

Step 21: After that, we will get the Configure Request page, where we need to click on the Continue button.



Step 22: After clicking on the continue button it will display the Configure Response page, where we need to select the Response payload format as JSON Sample and click on the inline link to configure the JSON response profile.



©TGH Software Solutions Pvt. Ltd.

No part of this document may be copied, reproduced, republished, uploaded, posted, publicly displayed, encoded, translated, transmitted or distributed in any way to any other computer, server, website or other medium for publication or distribution, without TGH's prior written consent



Step 23: After that, we will get a pop-up, where we need to define our response JSON format and click on the Continue button.

Enter Sample JSON

```
{  
  "Result": ""  
}
```

Cancel Continue

Step 24: After that, we will get the Configure Response page, where we need to click on the Continue button.

Configure Response
REST API

Operation name: default

Response URL: /COMUBOX

HTTP Method: POST

Select the default response processing options:

response is multipart with payload

multipart response is of type multipart/form-data with HTML form payload

Select the response payload format:

JSON Sample

Drag and Drop
Select a file to drag or paste.

or enter sample JSON

Response wrapper: response-wrapper

Select the media type of "Response Body" (except Header):

XML

XML(text)

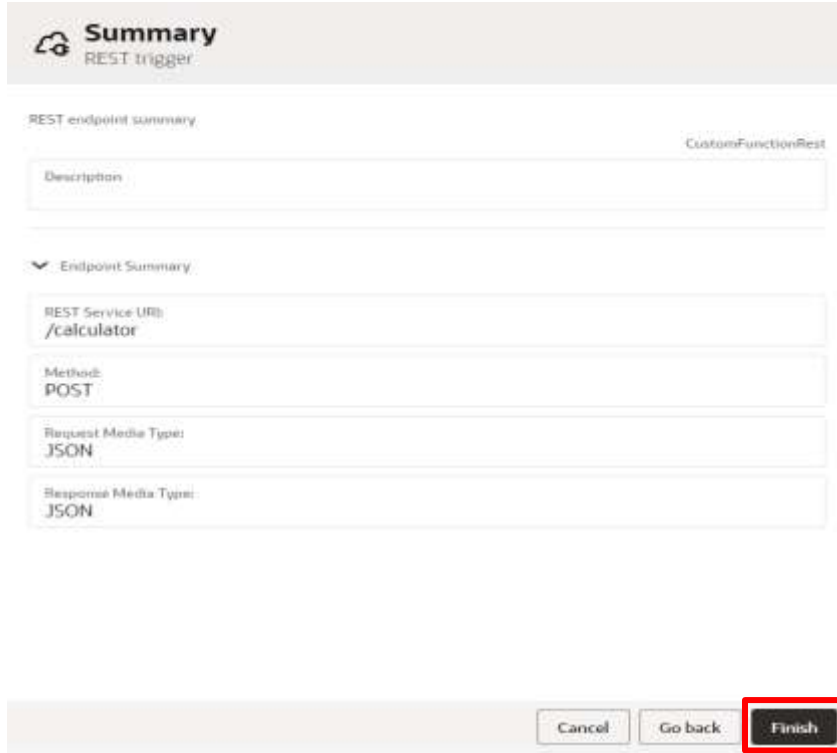
JSON

Other Media Type

Media Type

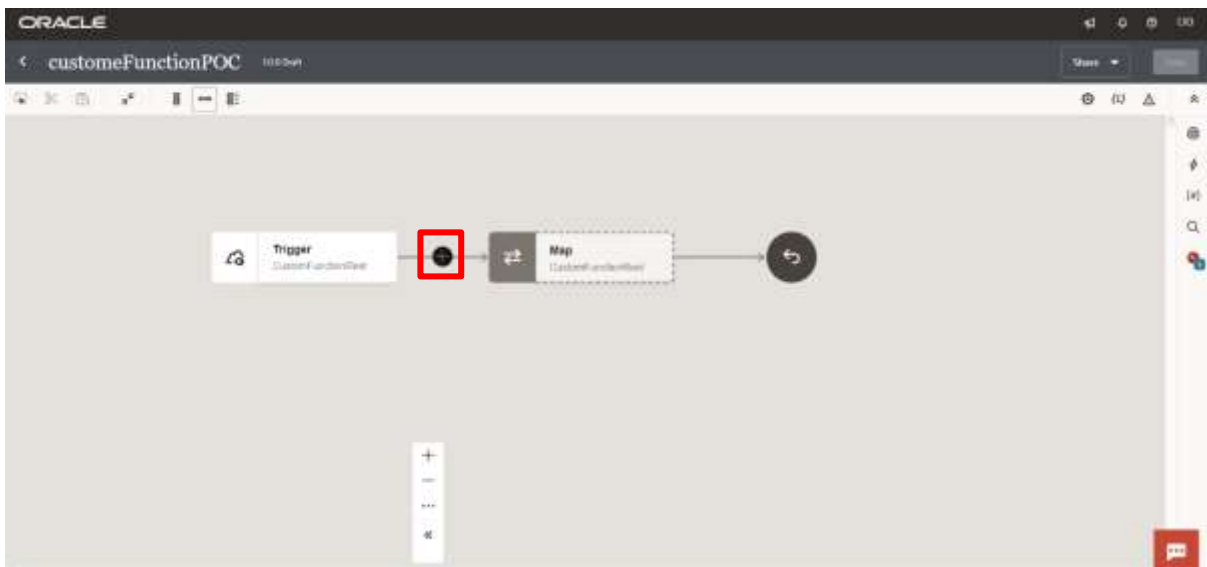
Cancel Go back Continue

Step 25: After that, we will get the Summary page, where we need to check every detail and click on the Finish button.

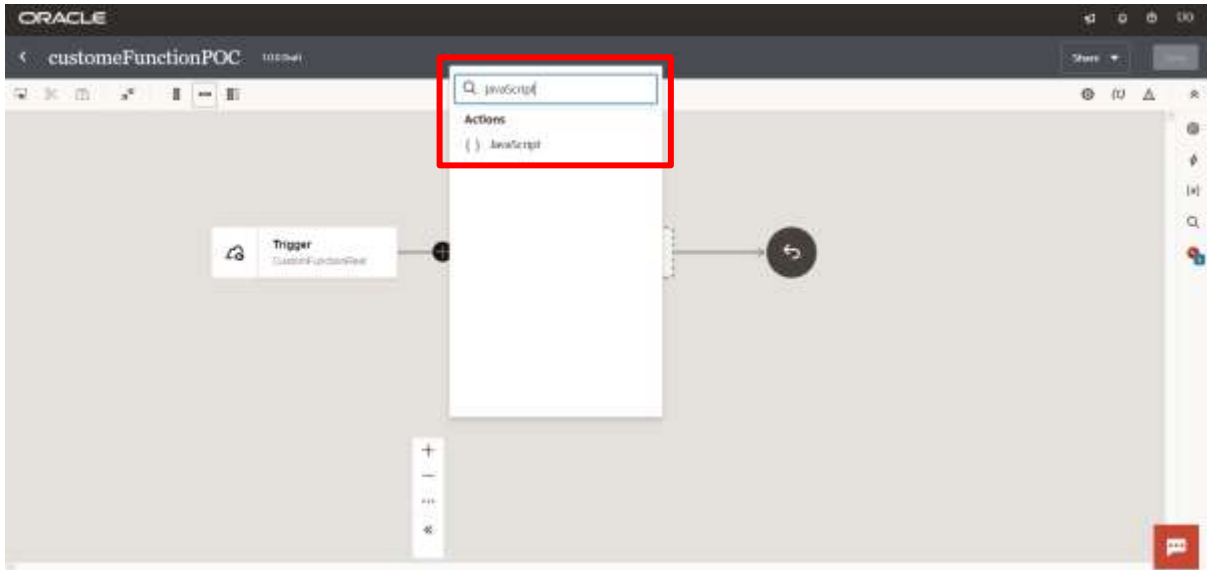


The image shows a 'Summary' page for a REST trigger. At the top, it says 'Summary REST trigger'. Below that, there's a section for 'REST endpoint summary' with a 'Description' field. To the right, it says 'CustomFunctionRest'. Underneath, there's a 'Endpoint Summary' section with a dropdown arrow. It contains four fields: 'REST Service URL: /calculator', 'Method: POST', 'Request Media Type: JSON', and 'Response Media Type: JSON'. At the bottom right, there are three buttons: 'Cancel', 'Go back', and 'Finish'. The 'Finish' button is highlighted with a red box.

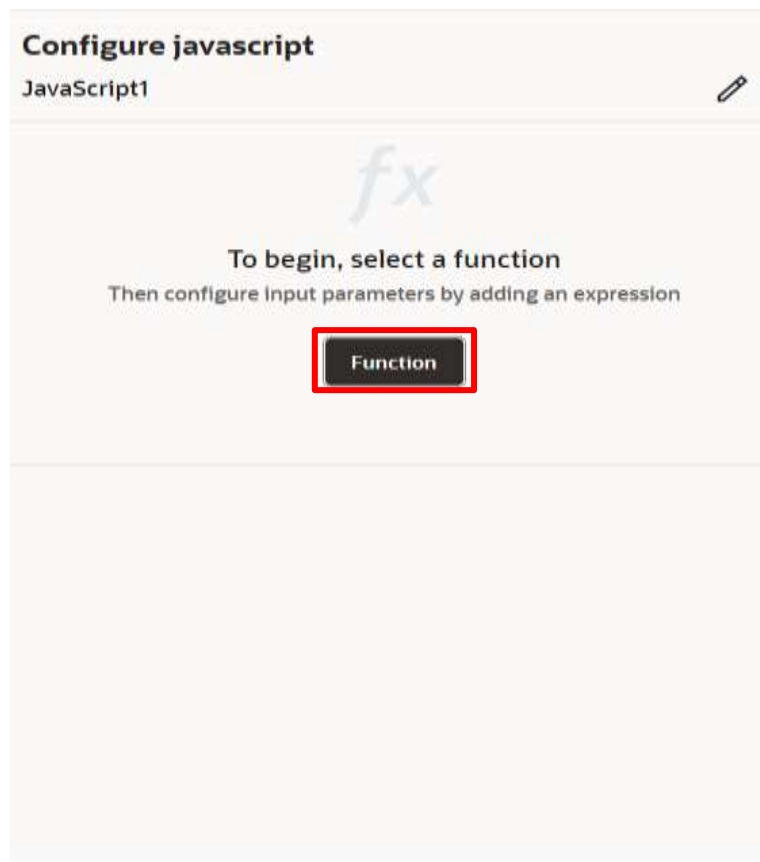
Step 26: After that, we will go to the Process canvas and need to click on the Plus button to add the JavaScript shape.



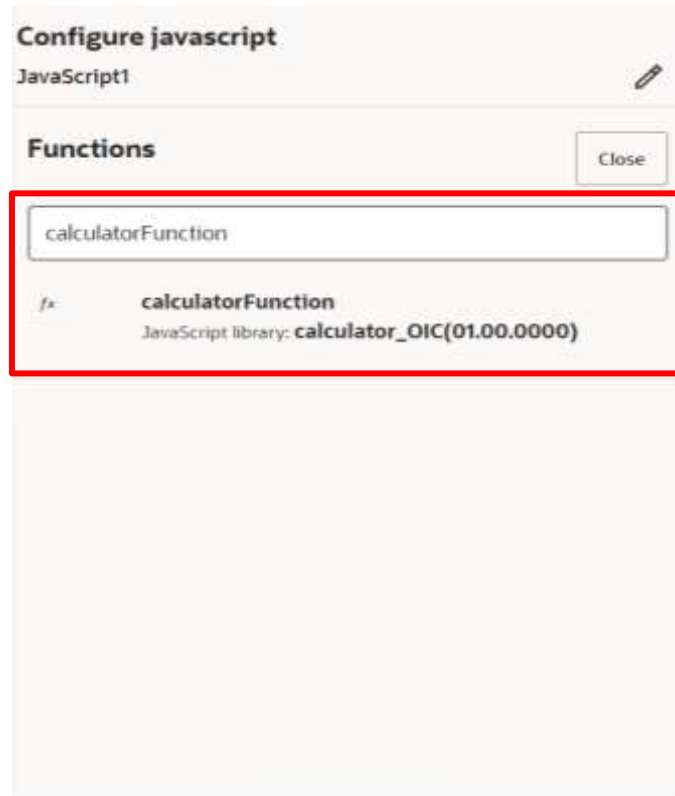
Step 27: After that, we need to search JavaScript in the search box and select that shape.



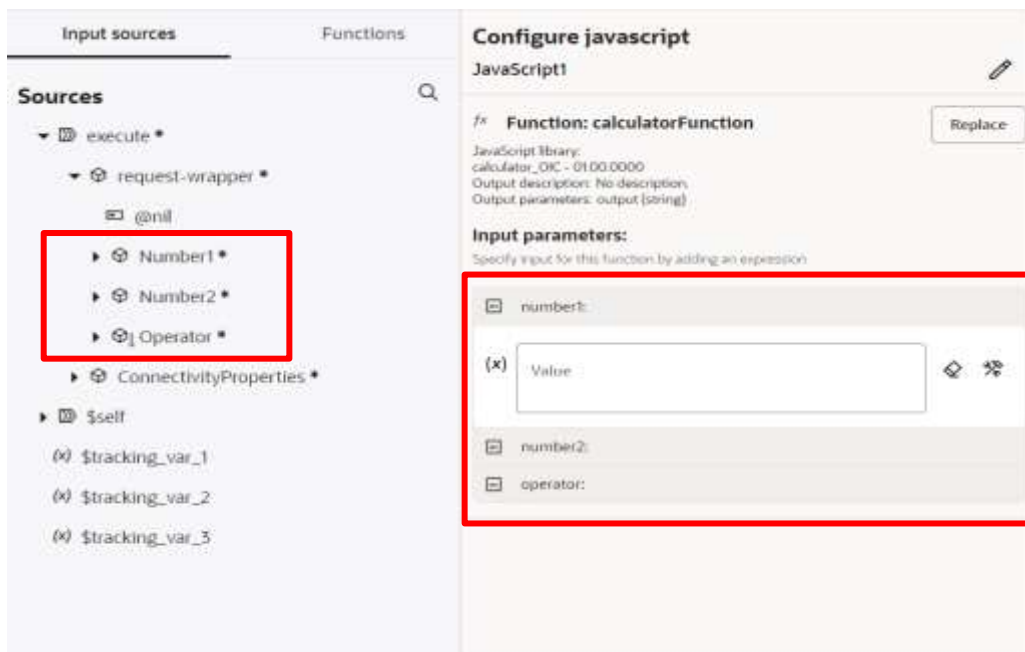
Step 28: After that, we will get the Configure javascript page, where we need to click on the Function button.



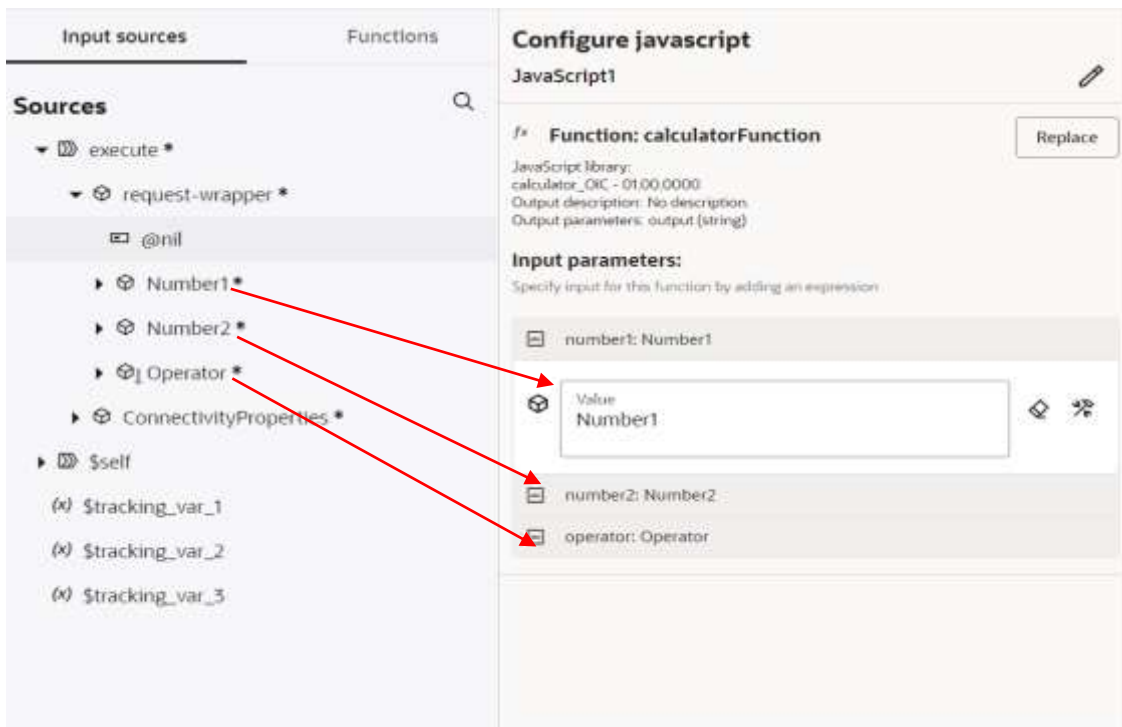
Step 29: After that, we need to search the function name in the search box and select the required function.



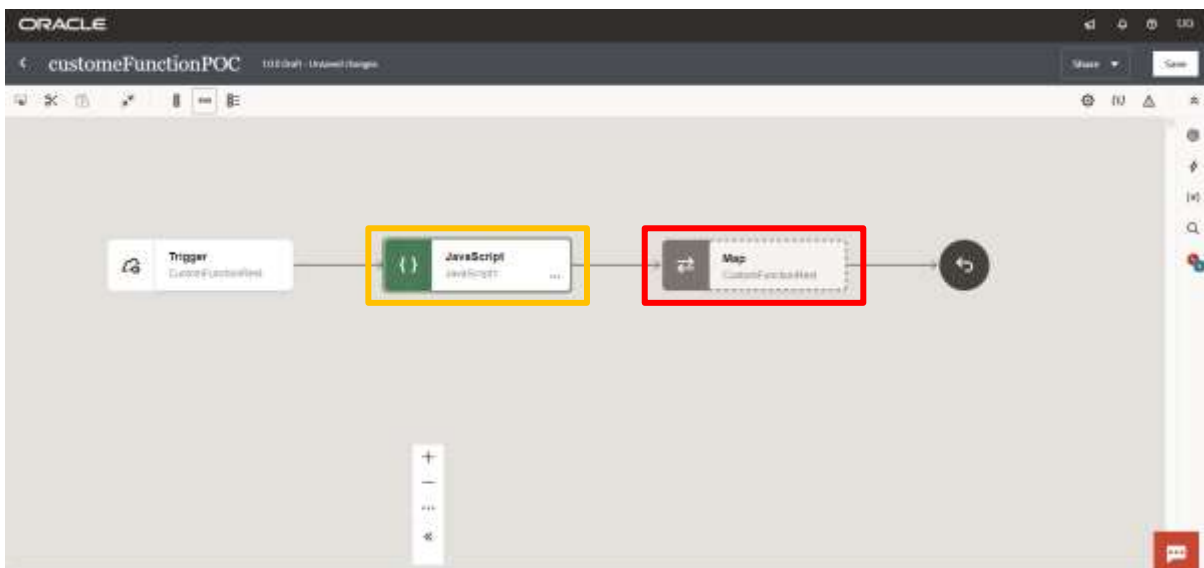
Step 30: After that, we need to drag Number 1, Number 2, and Operator from the source side and drop in the input parameters section accordingly.



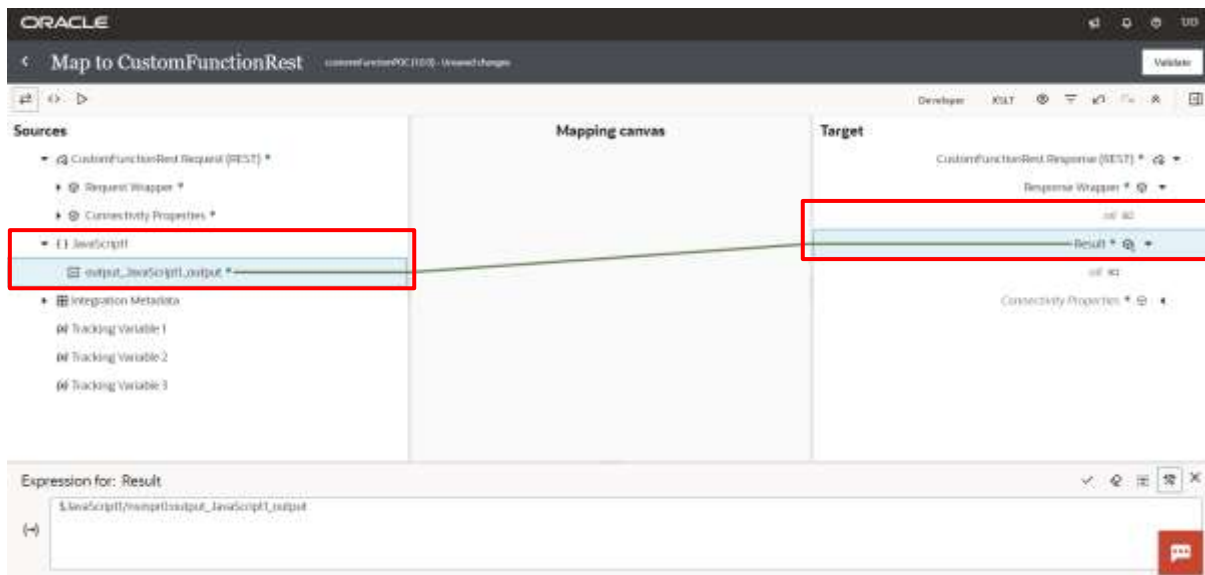
Step 31: After that, we can see the values are mapped successfully.



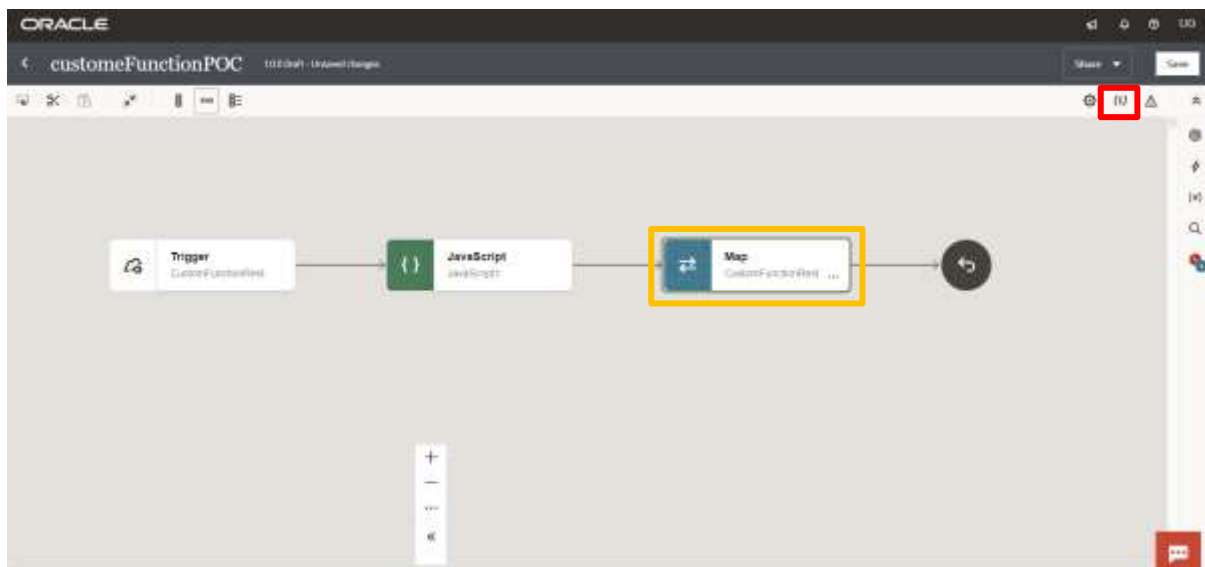
Step 32: After that, we need to click outside the Configure javascript tab and we can see the JavaScript shape is successfully created and we need to double-click on the Map shape.



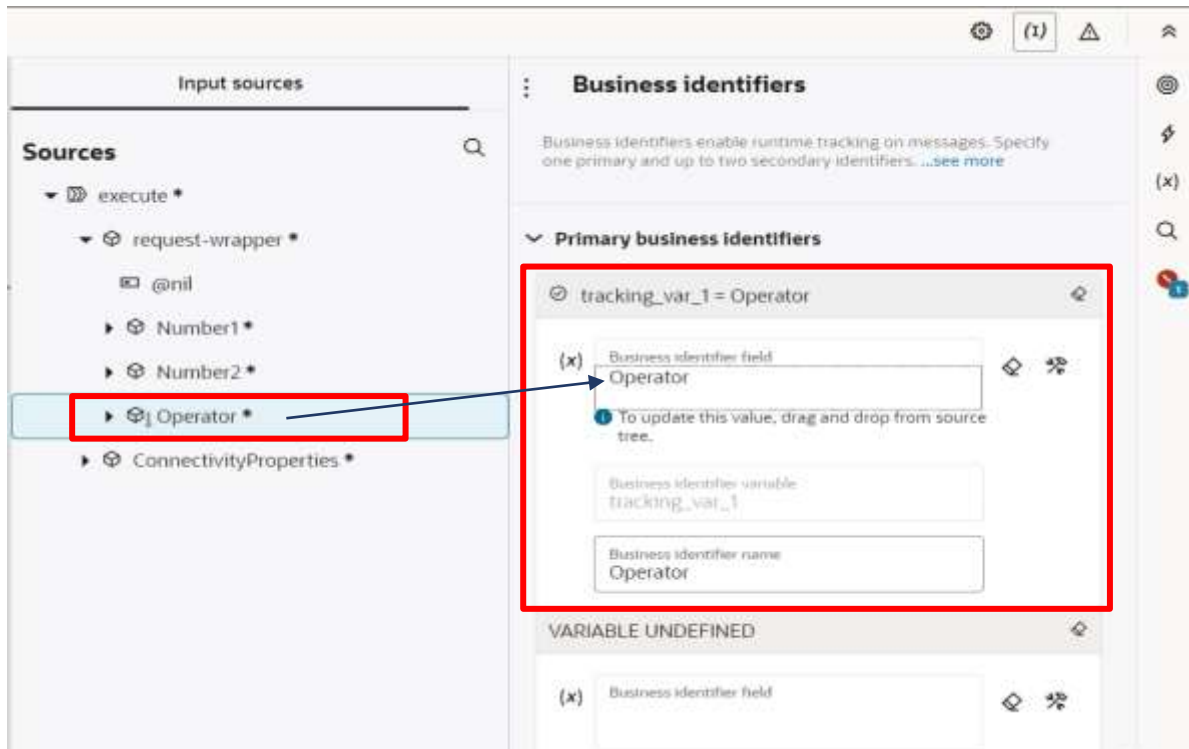
Step 33: After that, it will open the map shape and we need to configure that for that we need to drag the JavaScript_output from the source side and drop on the Result field on the target side and click on the Validate button.



Step 34: Our Integration is completed now we need to set the Business Identifier, for that we need to click on the (I) button.



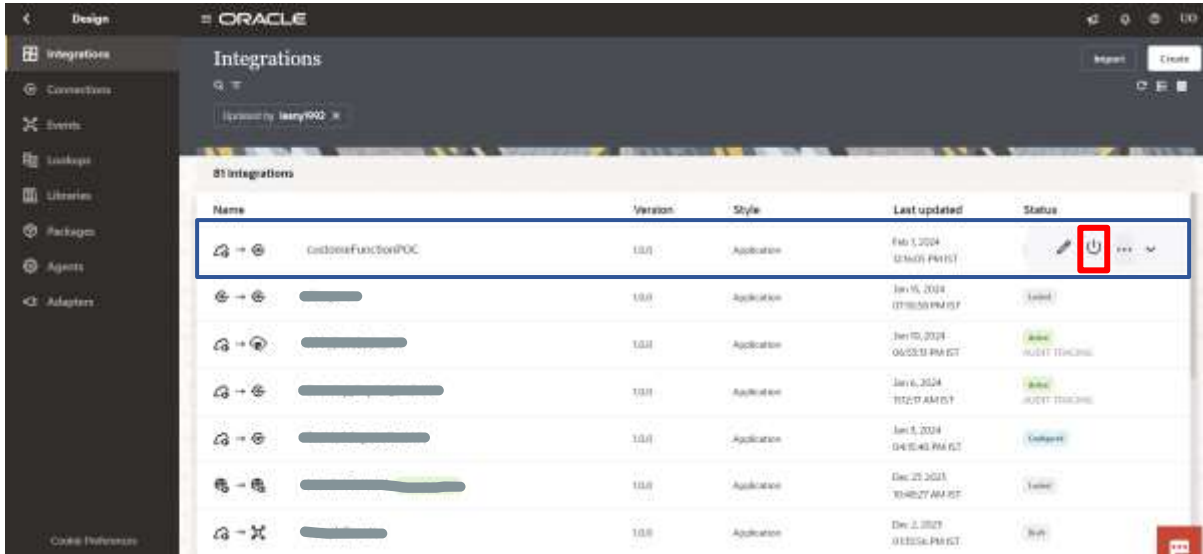
Step 35: After that, we need to drag the “Operator” from the source side and paste it in the Business Identifier field, and click on the save button.



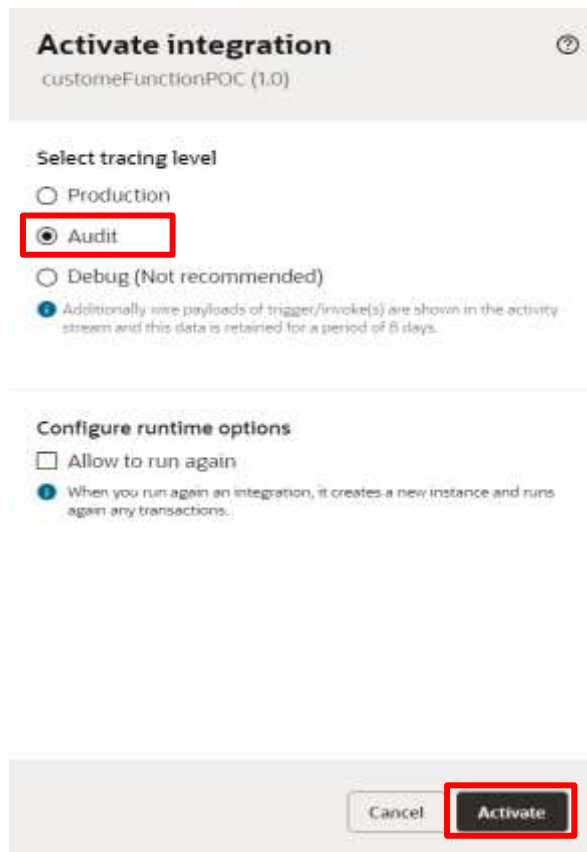
Step 36: After that, we need to click on the Save button first and the left arrow to close our Integration.



Step 37: To activate our Integration we need to hover over the Integration and click on the power button.



Step 38: In this Activate integration page we need to select the Audit and click on the Activate button.



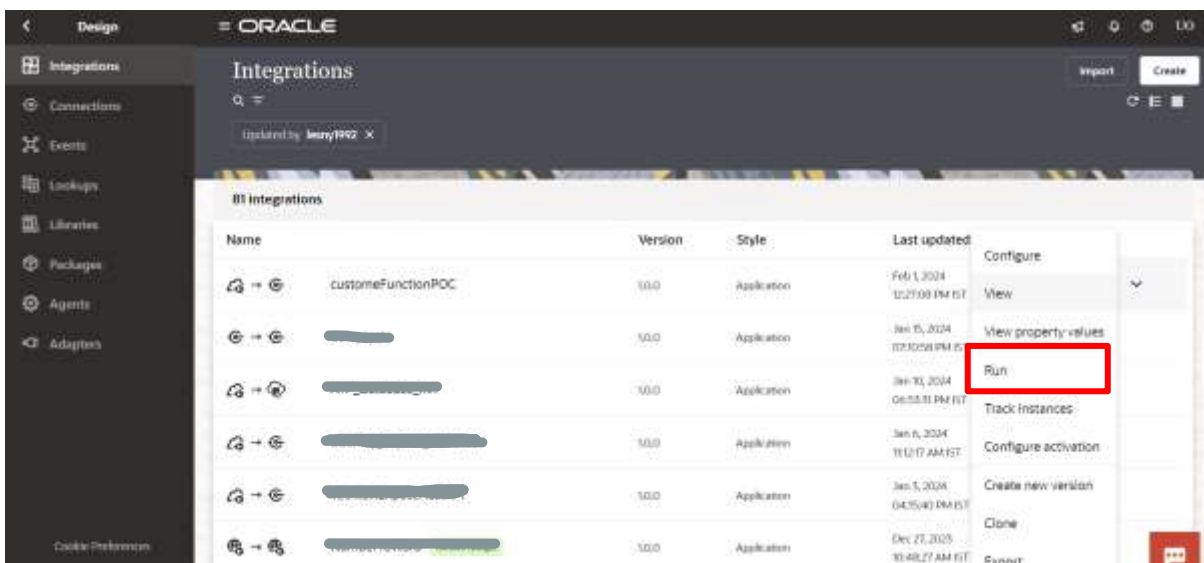
Step 39: After that, we can see our Integration is in the Active stage.



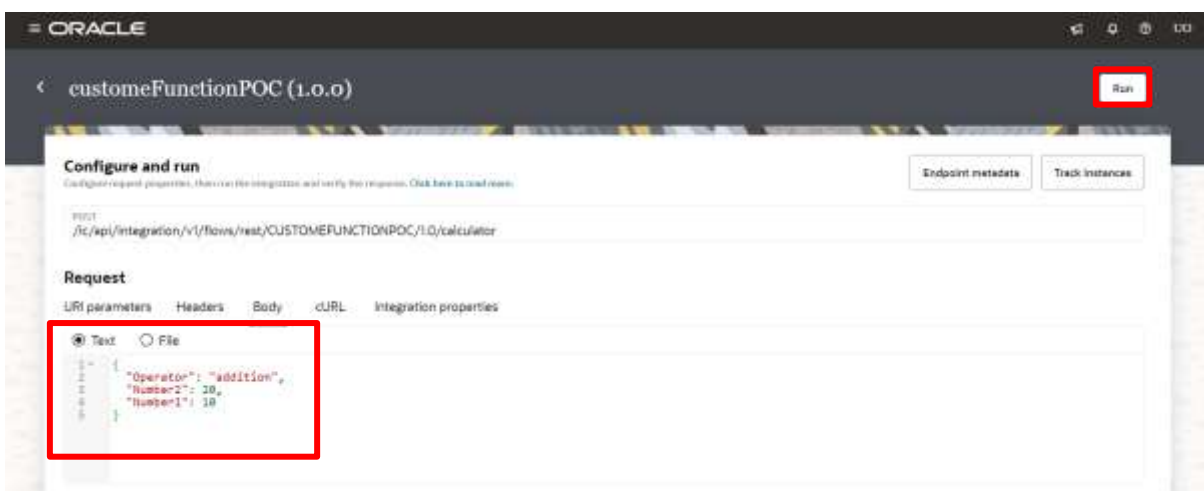
Step 40: After that, we need to hover over the Integration and click on the three dots.



Step 41: After that, we need to click on the “Run” button.



Step 42: In this Configure and run page we need to provide a body to our API and click on the Run button.

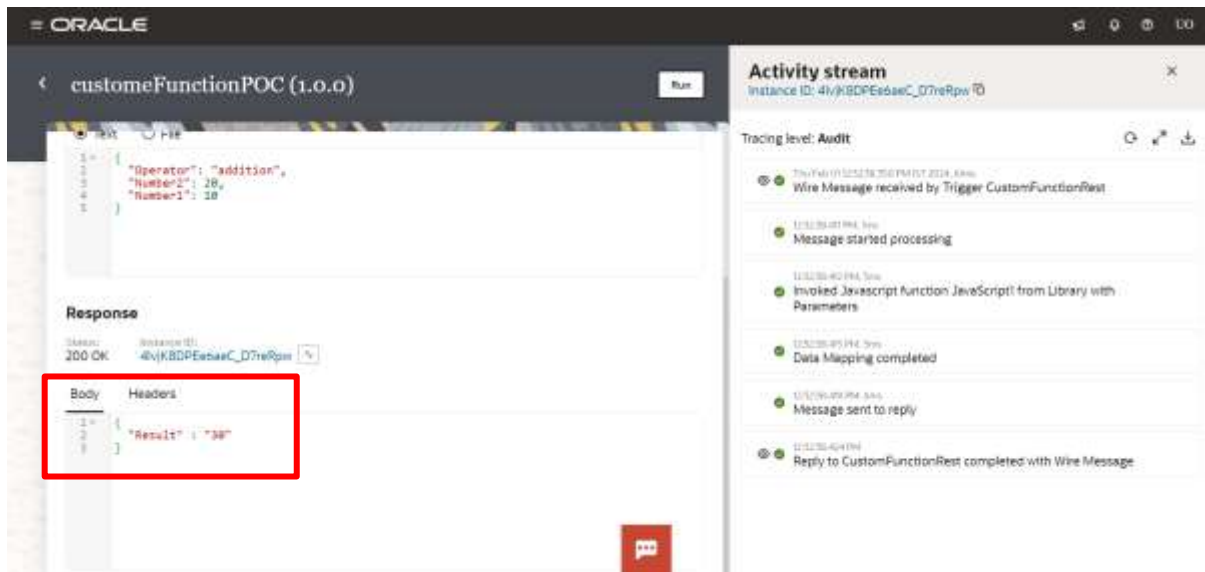


©TGH Software Solutions Pvt. Ltd.

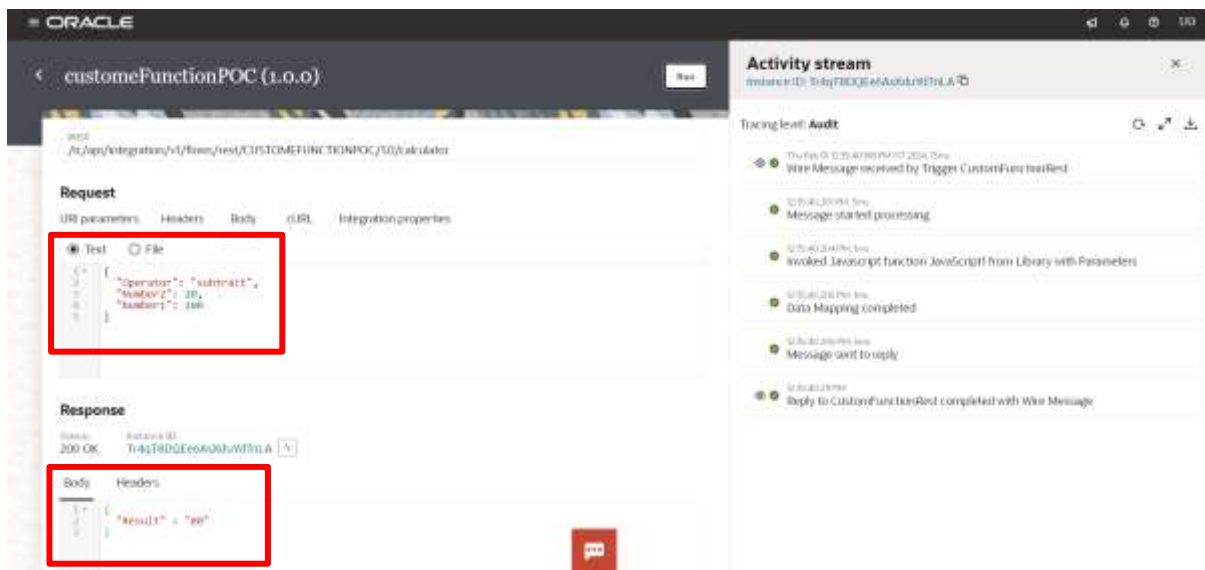
No part of this document may be copied, reproduced, republished, uploaded, posted, publicly displayed, encoded, translated, transmitted or distributed in any way to any other computer, server, website or other medium for publication or distribution, without TGH's prior written consent



Step 43: Now in the Response section we can see the output.



Step 44: Now if we change the request and click on the run button, we can see a different output.



Thank You



TGH

Making Integrations Simpler

TGH Software Solutions Pvt. Ltd.

www.techygeekhub.com

At TGH, we specialize in driving digital transformation through seamless Integration Technologies.

Operating as an INTEGRATION FACTORY, we serve as a one-stop shop for all your integration needs. Our expert team is well-versed in enterprise software and legacy system integration, along with leading iPaaS technologies like Boomi, MuleSoft, Workato, OIC, and more.

We're committed to enhancing business processes and solving problems through our integration expertise.



Email address

connect@techygeekhub.com



Phone number

+ 011-40071137
+ 91-8810610395



Our offices

Noida Office

iThum
Plot No -40, Tower A,
Office No: 712,
Sector-62, Noida,
Uttar Pradesh, 201301

Hyderabad Office

Plot no: 6/3, 5th Floor,
Techno Pearl Building,
HUDA Techno Enclave,
HITEC City, Hyderabad,
Telangana 500081

