

IBM Partner Engagement Manager create a new activity, IBM Cloud Node-RED and CloudbantDB integration

Authors:	Mehmet Cambaz
Version:	1.0
Date Created:	13.10.2017
Last Updated:	21.11.2017

Disclaimer:

All of the information is given “as-is” basis and author do not give any kind of guarantee of any sort, use it as your own risk.

Document Revision History

Revision	Date	Description	Author
1.0	21.11.2017	First Draft	Mehmet Cambaz

Document Approval History

Name	Signature	Date	Version Approved/Comments

Reference Documents

Document Name	Description	Owner	Location

Table of Contents

1. Introduction
2. Content

1. Introduction

This document's purpose is to give helpful information on how to integrate IBM Partner Engagement Manager with IBM Bluemix (renamed as IBM Cloud) CloudantDB through Node-Red to show capabilities of PEM so that it can be used on trading partner information gathering activities to end system integrations.

The know-how shared on this document:

- Create your own activities and tasks from ground zero with IBM PEM
- Integrate through REST APIs with IBM PEM
- Use IBM Cloud Node-Red to have business and data access layer
- Use IBM Cloud Node-Red components to store and gather data from IBM CloudantDB
- Show that database integration with IBM PEM through REST APIs

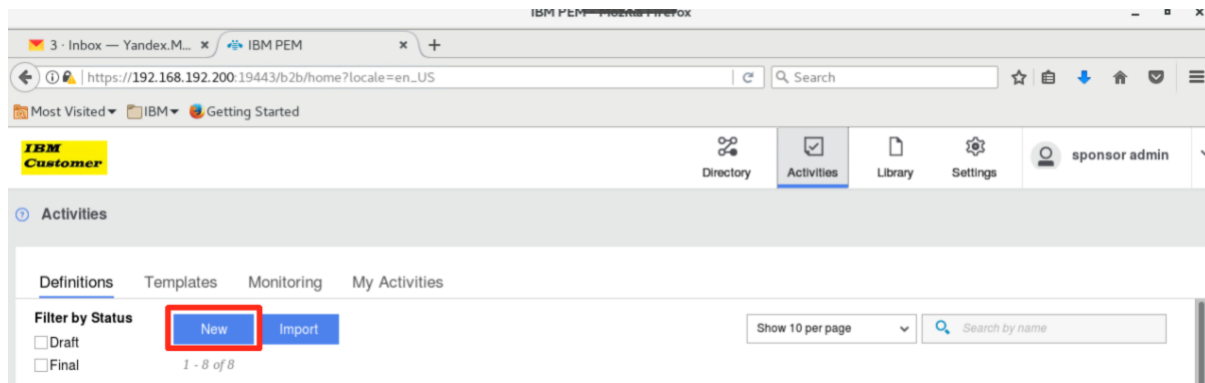
Useful documents around the technologies and solutions used:

For IBM Cloud Node-RED configurations see the addendum file: ***"Node-RED-configuration-for-PEM.pdf"***

For IBM Partner Engagement Manager Node-RED API and certificate adding configurations see the addendum file: ***"PEM-APIs-Certificates-configuration.pdf"***

2. Content

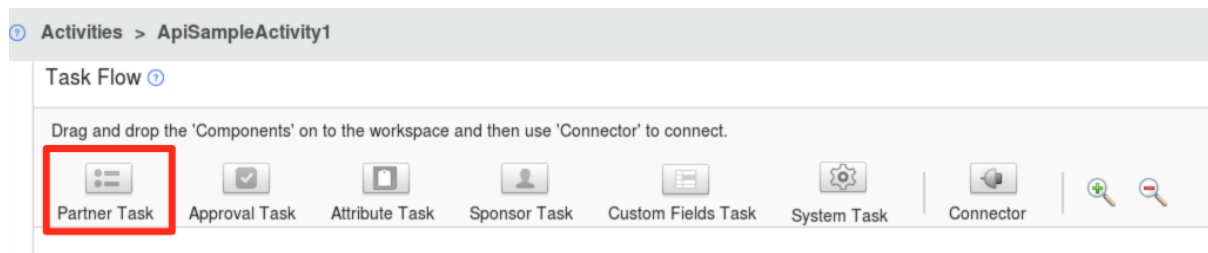
IBM Partner Engagement Manager has activity design capabilities out of the box and can integrate to other systems which has RESTful API endpoints. In order to create new activity we need to login as Sponsor Admin at PEM and click to Activities menu and New at Definitions tab.



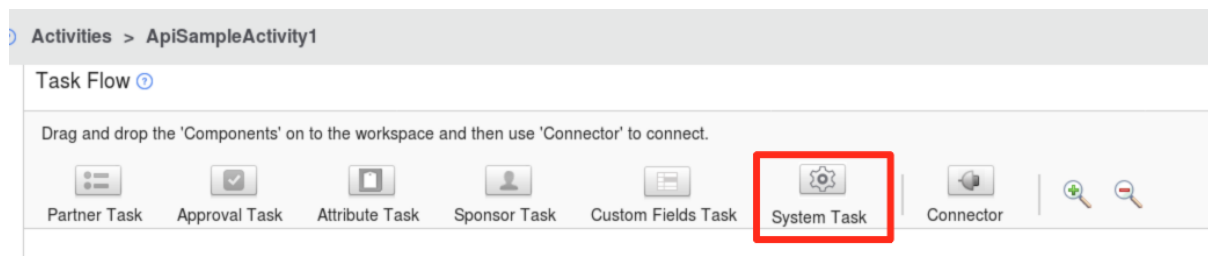
After that you will give a name to your activity, we will create an activity flow which would be:

1. Get Info from partner via IBM PEM
2. Integrate with IBM Cloud Node-RED REST API to store data in CloudantDB
3. Show the stored data via PEM to partner and sponsor

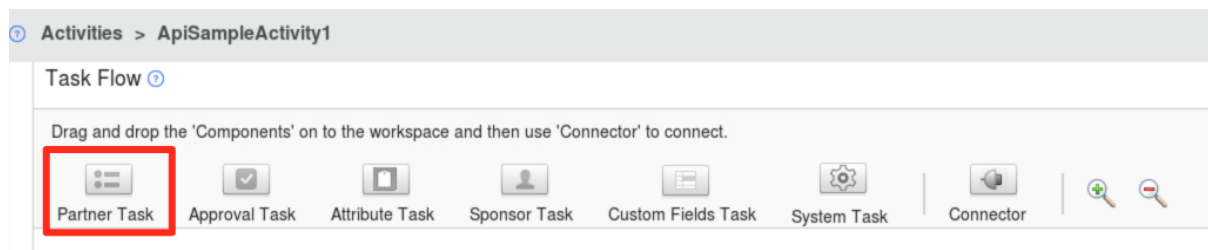
Step1: Trading Partner will enter the related information at this task (this will be a **Partner Task**)



Step2: PEM will gather the entered information, build up the json data and integrate to end system via REST API *POST* action (this will be a **System Task**)



Step3: Pem will show the id of the record since it has been inserted to end system task (this will be a **Partner Task**)



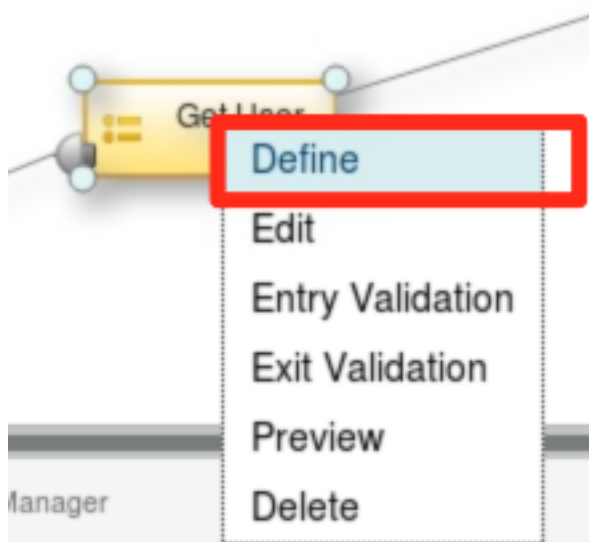
With the 3 tasks dragged and dropped down to activity, we will connect them with the Connector in order to complete the flow:



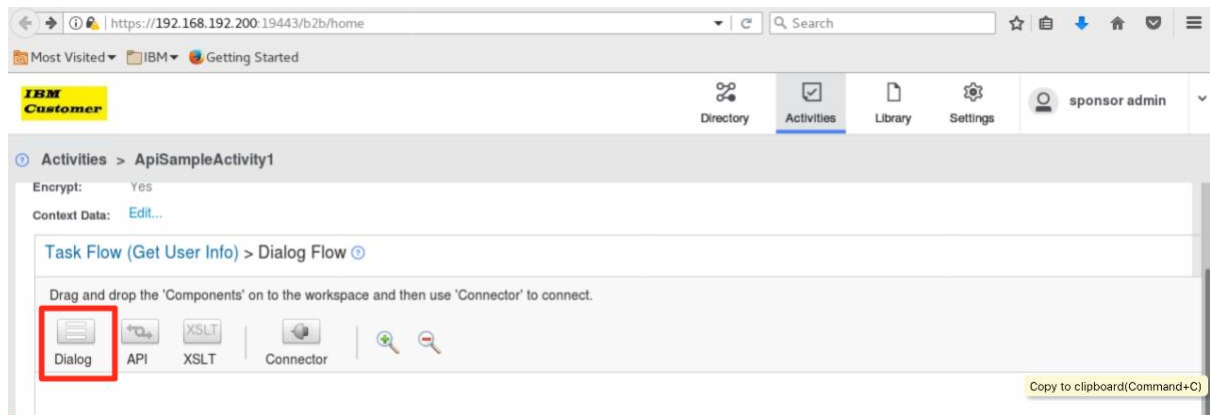
The final flow would be like below:



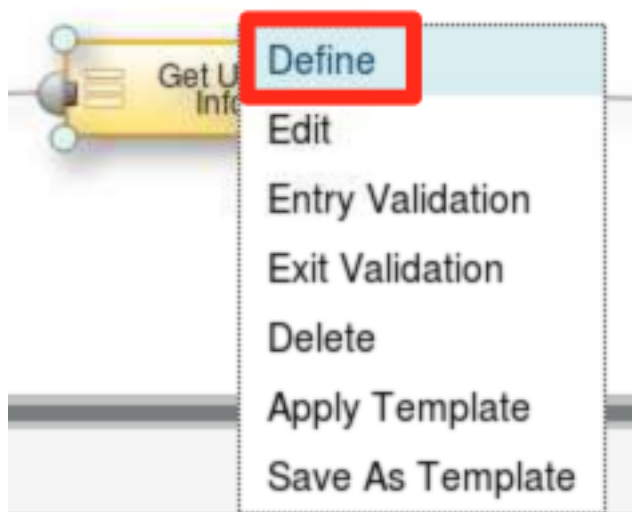
The first task is “**Get User Info**” which would be modified by right clicking and clicking Define



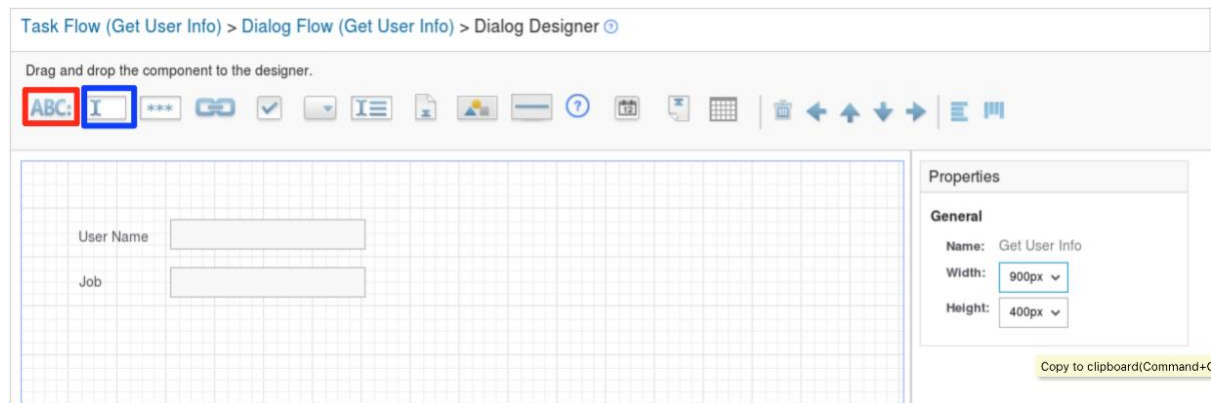
Since this is a task in the activity, we will add the relevant task here, the aim here to get the information from trading partner so we will add **Dialog** we will add connectors from Start to Our Dialog and then to End.



We will right click Dialog task we had added and click Define so that we can implement inside



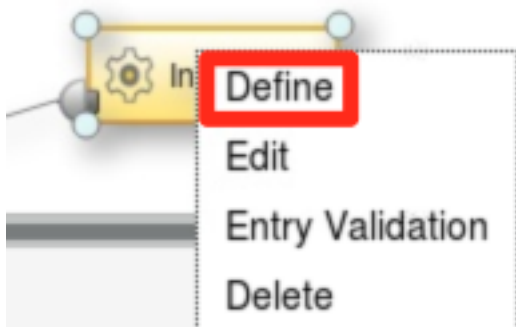
It will open us a Dialog Designer which is the UI/form we are designing to the trading partner to get info. We will add 2 labels and 2 textboxes (Username and Job)



In order to find the entered data in activity context we will update the names of textboxes to *userText* and *jobText*



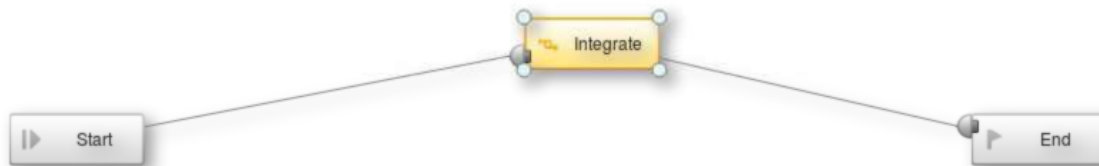
The second task is the “**IntegrateWithRestApi**” which would be modified by right clicking and clicking Define



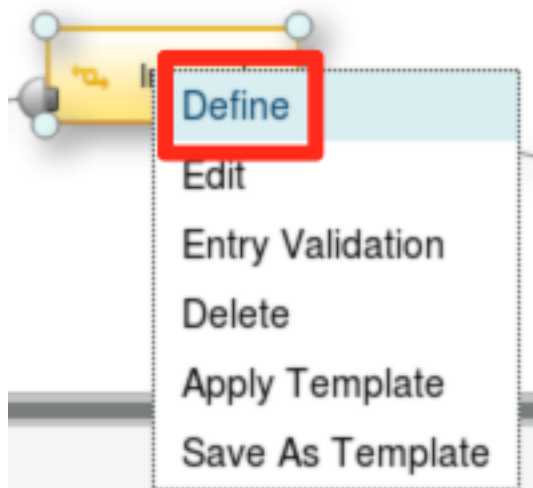
It will open us a Dialog Designer and we will add and **API** task here and connect with connectors from start to our added task and from that task to end

Task Flow (IntegrateWithRestApi) > Dialog Flow ?

Drag and drop the 'Component' on to the workspace and then use 'Connector' to connect.



In order to configure the REST API integration with the gathered information we right click to added API task and click Define



Here we select API we had created,
add the relevant api call url: here it is **add** its request method is **POST**
the input/output format is **JSON**
the request will be mapped to the textboxes that has the filled input from partner
add the sample response which REST API returns

Define (Integrate)

* API Configuration:

lbmCloudNodeRedApi

...

?

Host Prefix:

☒

* URL:

https://mc34.eu-...443/

add

...

?

* Request Method:

POST

...

Headers:

Edit...

* Input/Output Format:

JSON

...

Escape Request:

☐

?

Request:

{
 "name": "\${/Parameters/ActivityContext/Task[@Name='Get User Info']/Dialog[@Name='Get User Info']/TextBox[@Name='userText']}",
 "job": "\${/Parameters/ActivityContext/Task[@Name='Get User Info']/Dialog[@Name='Get User Info']/TextBox[@Name='jobText']}"
}

?

...

File Attachment:

...

Select

?

Sample Response:

{
 "name": "morpheus",
 "job": "leader"
}

?

Call in Testmode:

☒

Cancel

OK

API Request Mapping (Integrate)

Request

▼ Root

name [\${/Parameters/ActivityContext/Task[@Name='Get User Info']/Dialog[@Name='Get User Info']/TextBox[@Name='userText']}]

job [\${/Parameters/ActivityContext/Task[@Name='Get User Info']/Dialog[@Name='Get User Info']/TextBox[@Name='jobText']}]

Context

▼ Parameters

▼ ActivityContext [ActivityContext]

ParticipantActivityKey [ParticipantActivityKey]

▼ Task [Get User Info]

▼ DialogUI [Get User Info]

Label [C-1]

Label [C-2]

TextBox [userText]

TextBox [jobText]

Task [IntegrateWithRestApi]

Task [Show User Info]

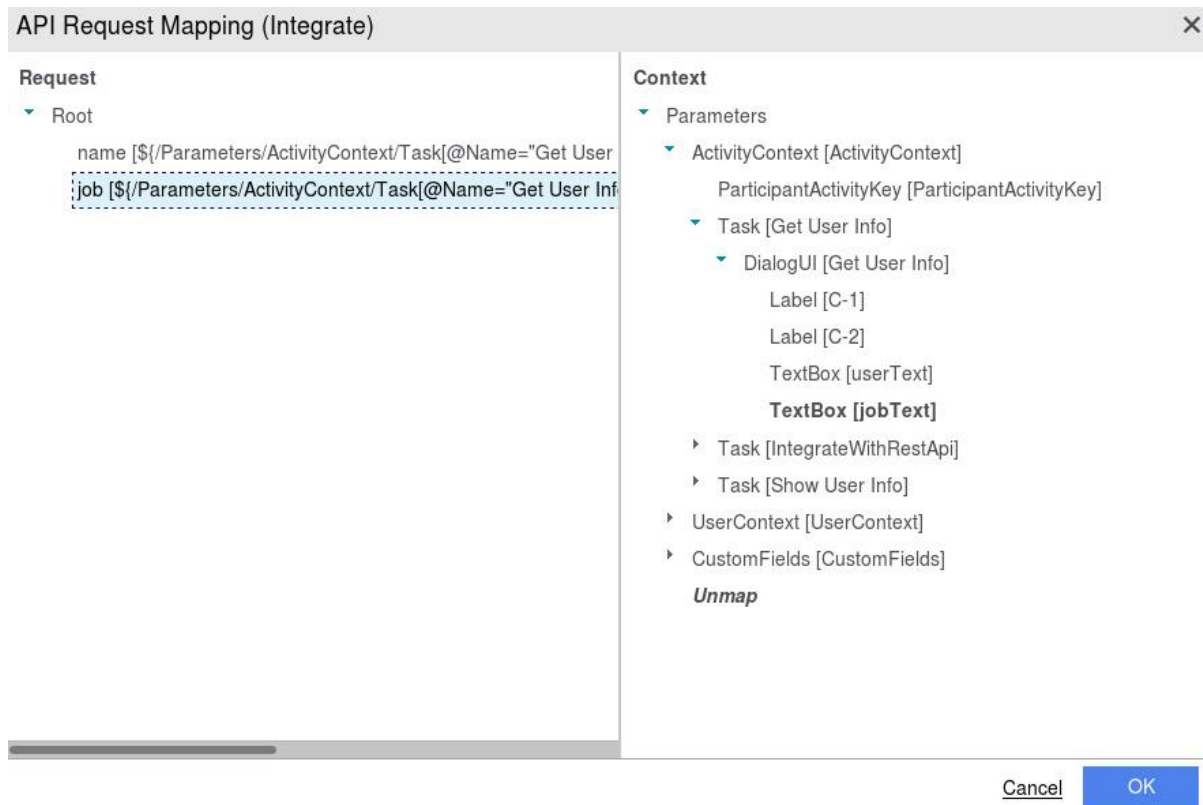
UserContext [UserContext]

CustomFields [CustomFields]

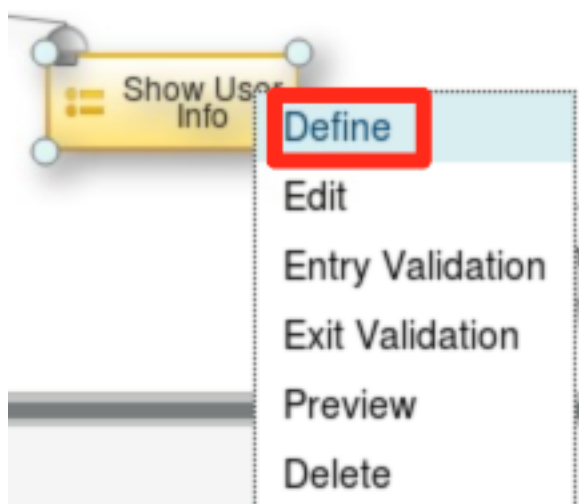
Unmap

Cancel

OK



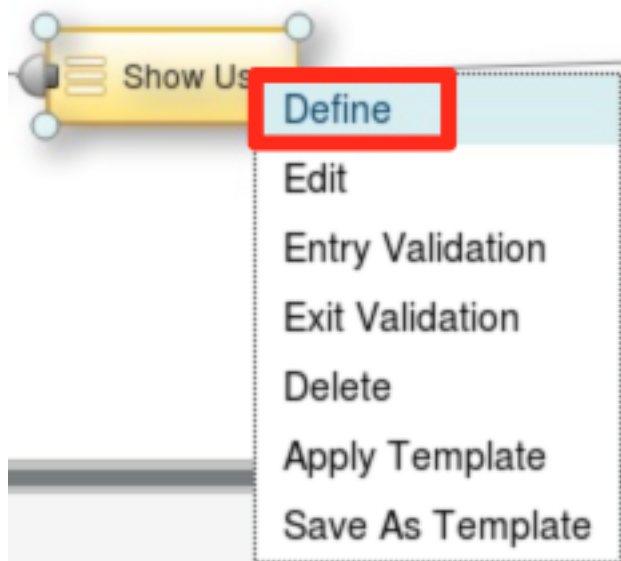
The third task is the “**Show User Info**” which would be modified by right clicking and clicking Define



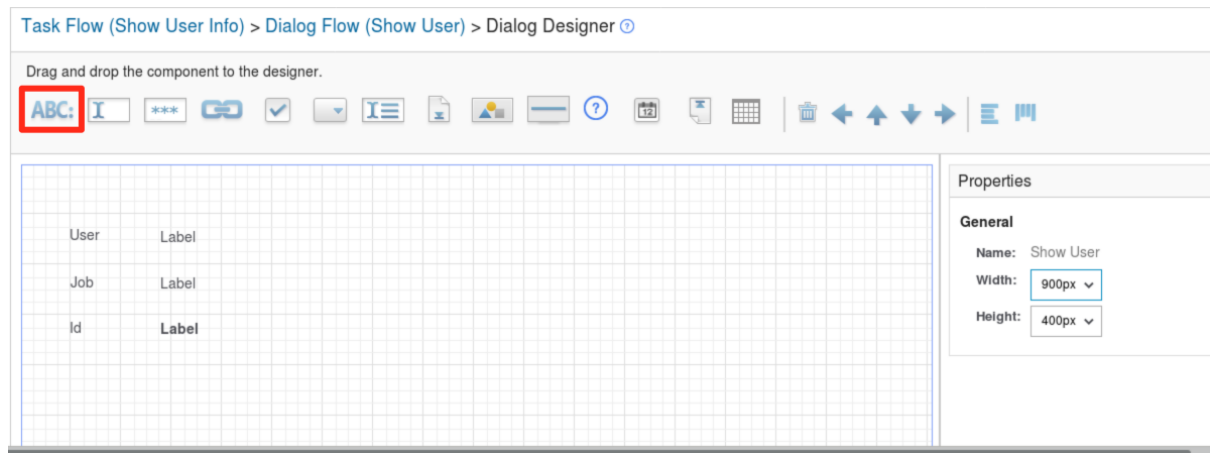
So we will add **Dialog** we will add connectors from Start to Our Dialog and then to End.



We will right click Dialog task we had added and click Define so that we can implement inside

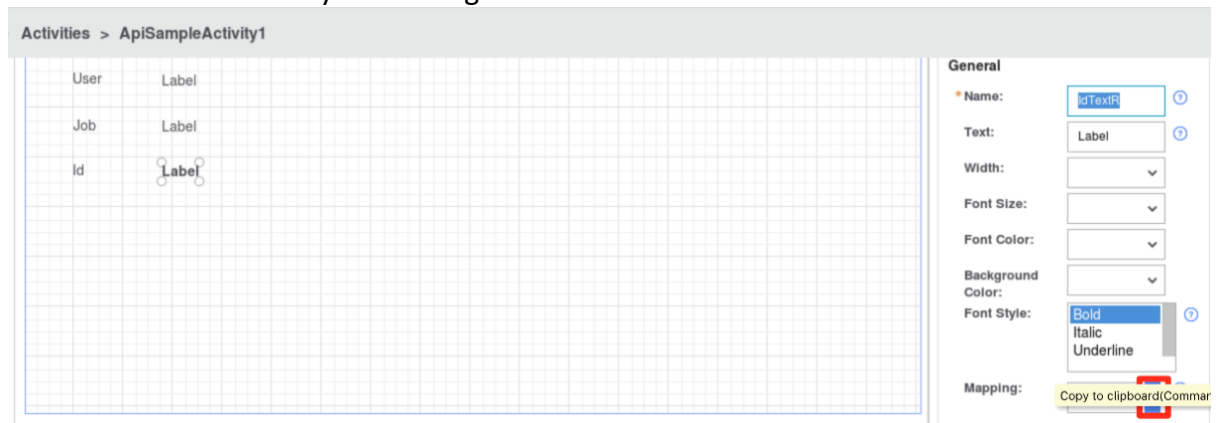


It will open us a Dialog Designer which is the UI/form we are designing to the trading partner to show the info read-only (Username, Job and Id)



We will give second column labels names to find them easily at the activity context (UserTextR , JobTextR R means read-only)

In order to map the information to be shown at the UI we will click the ... button and select what value in the activity to be assigned here



It will pop up a window about the possible mapping values:

Select a node

Context

Parameters

ActivityContext [ActivityContext]

ParticipantActivityKey [ParticipantActivityKey]

Task [Get User Info]

Task [IntegrateWithRestApi]

DialogAPI [Integrate]

Response [Response]

ObjectNode

name

job

Task [Show User Info]

UserContext [UserContext]

CustomFields [CustomFields]

Unmap

Select a node, to display text in the **Label** component.

Close

Select a node

Context

Parameters

ActivityContext [ActivityContext]

ParticipantActivityKey [ParticipantActivityKey]

Task [Get User Info]

Task [IntegrateWithRestApi]

DialogAPI [Integrate]

Response [Response]

ObjectNode

name

job

Task [Show User Info]

UserContext [UserContext]

CustomFields [CustomFields]

Unmap

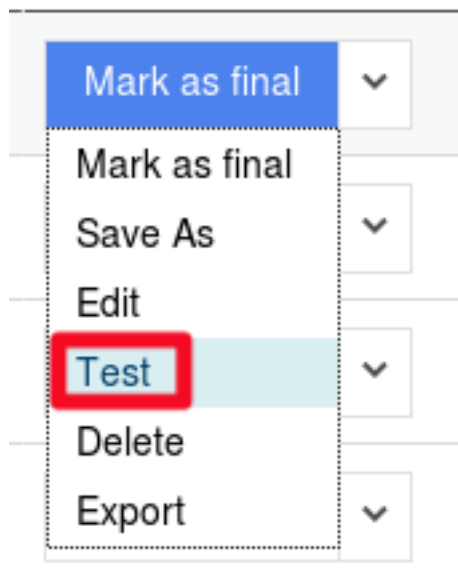
Select a node, to display text in the **Label** component.

Close

What we will assign to the labels to the information which we had gathered from the REST API which was a JSON response. In the json response of the Task **IntegrateWithRestApi** there is **DialogAPI** and its API call response is shown as **Response** and its child class is **ObjectNode** and it has the id information which was returned from the other system.


Lets test the activity before making it final:

Actions



Activities > Task Execution (ApiSampleActivity1)





Activity Overview (Test Mode)

Activity Name : ApiSampleActivity1 Status : 

Activity Description : None Start Date : Nov 21, 2017, 12:00:36 PM

Partner View Sponsor View

1 - 2 of 2

Task	Status/Action	Owner
 Get User Info	 Start task	Unassigned Assign Owner
 Show User Info	 Start task	Unassigned Assign Owner

1 - 2 of 2

Get User Info

Get User Info

User Name

mehmet

Job

csp

Cancel

Finish

Summary:

We had used IBM PEM's activity design capabilities to design our own UIs, REST API capabilities to integrate with databases or any application provides REST API. We also leveraged IBM Cloud's vast capabilities to set up an environment to create REST APIs and infrastructure easily creating a hybrid cloud environment.