IBM Integration Bus

Developing an application to demonstrate workload flexibility using Callable Flows

Featuring:

- Addition of new operation in HR_Service REST API to access a callable application
- Using the CallableFlowInvoke node in a REST API
- Create a callable application using CallableInput and CallableReply nodes

September 2016

Hands-on lab built at product Version 10.0.0.6

1. Introduction

IIB V10.0.0.4 introduced Callable Flows. This enables a message flow (or integration service, or REST API) to invoke a separate message flow in a call/return (blocked wait) programming model.

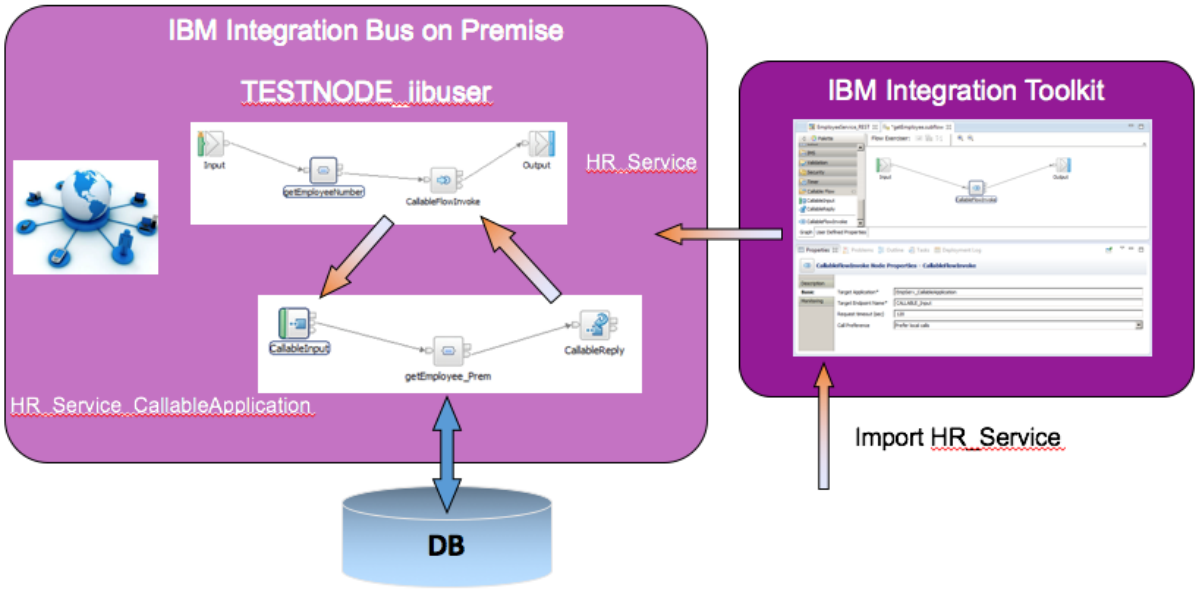
The calling and called message flows can operate on remote Integration Nodes, including IIB on Cloud and IIB On Premise.

1.1 Scenario Overview

This is the Development lab for Callable Flows in which you will create the applications and run a simple local test. The artefacts are then used for two of the following labs – IIB on Cloud Callable Flows (16L12) and Docker Callable Flows (16L11), which provide additional runtime opportunities for the same developed applications.

In this lab, you will use the HR_Service REST API that was developed in 16L02 (Create a REST API with a swagger document) in this series of labs.

You will add a new operation. The implementation of this operation invokes a 'Callable' application. The callable application will access a database and retrieve employee details.



1.2 Outline of tasks

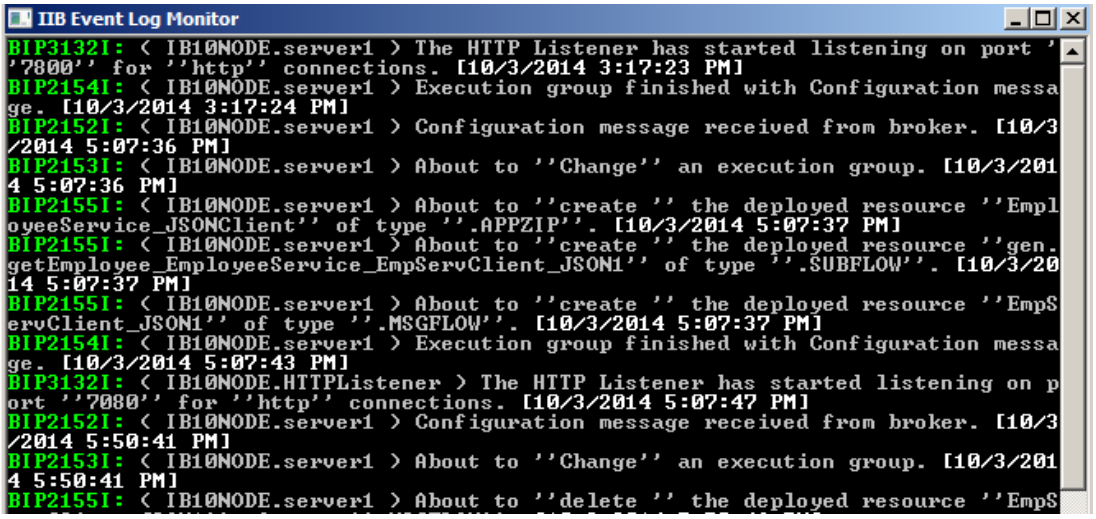
The tasks to complete in this lab are the following:

1. Import a REST API solution
2. Add an operation to the REST API to call a 'Callable' application
3. Import a map to access employees data
4. Create the Callable Application

1.3 Open the Windows Log Monitor for IIB

A useful tool for IIB development on Windows is the IIB Log Viewer. This tool continuously monitors the Windows Event Log, and all messages from the log are displayed immediately.

From the Start menu, click IIB Event Log Monitor. The Monitor will open; it is useful to have this always open in the background.



This tool is not shipped as part of the IIB product; please contact us directly if you would like a copy.

Important note

The instructions in this lab guide are based on a Windows implementation, with a user named "iibuser".
The Windows VMWare image on which this lab is based is not available outside IBM, so you will need to provide your own software product installations where necessary.

This lab prepares the artefacts for deploying a REST API and a Callable Flow application, running on different Integration Nodes. It is assumed that one of the key scenarios is running the REST API on IIB on Cloud and hence the reference in this lab as 'Cloud Application'.

You should use the Windows user "iibuser". This user is a member of mqbrkrs and mqm, but is not a member of Administrators. The user "iibuser" can create new IIB nodes and do all required IIB development work. However, installation of the IIB product requires Administrator privileges (not required in this lab).

The lab uses DB2 HRDB database. HRDB contains two tables, EMPLOYEE and DEPARTMENT. These tables have been populated with data required for this lab. (The DDL for the HRDB is available in the student10 folder; The corresponding DDL for Microsoft SQL/Server is available in Integration service _SQLServer, and we intend to provide the DDL for Oracle over time).

For the scenario, a single JSON Schema is used, which the built resources reference.

1.4 Configure Integration Bus node to work with DB2

Note 1

Login to Windows VM Image as the user "iibuser", password = "passw0rd".

Start the IIB Toolkit from the Start menu.

Note 2

If you have already done Lab 1 or Lab2 in this series (create a REST API), you can skip straight to Import the REST API.

The HRDB database, and the EMPLOYEE and DEPARTMENT tables have already been created on the supplied VMWare image. If you wish to recreate your own instance of this database, the commands

1_Create_HRDB_database.cmd and
2_Create_HRDB_Tables.cmd

are provided for this. If used in conjunction with the VM image, these commands must be run under the user "iibadmin". Appropriate database permissions are included in the scripts to GRANT access to the user iibuser.

To run this lab, the Integration Bus node must be enabled to allow a JDBC connection to the HRDB database.

1. Open an IIB Command Console (from the Start menu), and navigate to

   c:\student10\Create_HR_database

2. Run the command

   3_Create_JDBC_for_HRDB

   Accept the defaults presented in the script. This will create the required JDBC configurable service for the HRDB database.

3. Run the command

   4_Create_HRDB_SecurityID

4. Stop and restart the node to enable the above definitions to be activated

   mqsistop TESTNODE_iibuser

   mqsistart TESTNODE_iibuser

This will create the necessary security credentials enabling TESTNODE_iibuser to connect to the database.

1.5 Configure TESTNODE_iibuser for REST APIs

The IIB support for the REST API requires some special configuration for the IIB node and server. Cross-Origin Resource Scripting (CORS) must be enabled for the IIB node to execute REST applications. This is also required when testing with the SwaggerUI test tool. See http://www.w3.org/TR/cors/?cm_mc_uid=09173639950214518562833&cm_mc_sid_50200000=1452177651 for further information.

1. Ensure that TESTNODE_iibuser is started.

2. Check that CORS has been enabled on the IIB node by running the following command in an Integration Console:

```
mqsireportproperties TESTNODE_iibuser
      -e default
      -o HTTPConnector
      -r
```

3. If CORS is enabled, you will see the following lines (amongst others):

```
corsEnabled='true'
 corsAllowOrigins='*'
 corsAllowCredentials='false'
 corsExposeHeaders='Content-Type'
 corsMaxAge='-1'
 corsAllowMethods='GET,HEAD,POST,PUT,PATCH,DELETE,OPTIONS'
 corsAllowHeaders='Accept,Accept-Language,Content-Language,Content-
   Type'
```

4. If CORS has not been enabled, run the following commands:

```
mqsichangeproperties TESTNODE_iibuser
      -e default
      -o HTTPConnector
      -n corsEnabled -v true

mqsistop TESTNODE_iibuser

mqsistart TESTNODE_iibuser
```

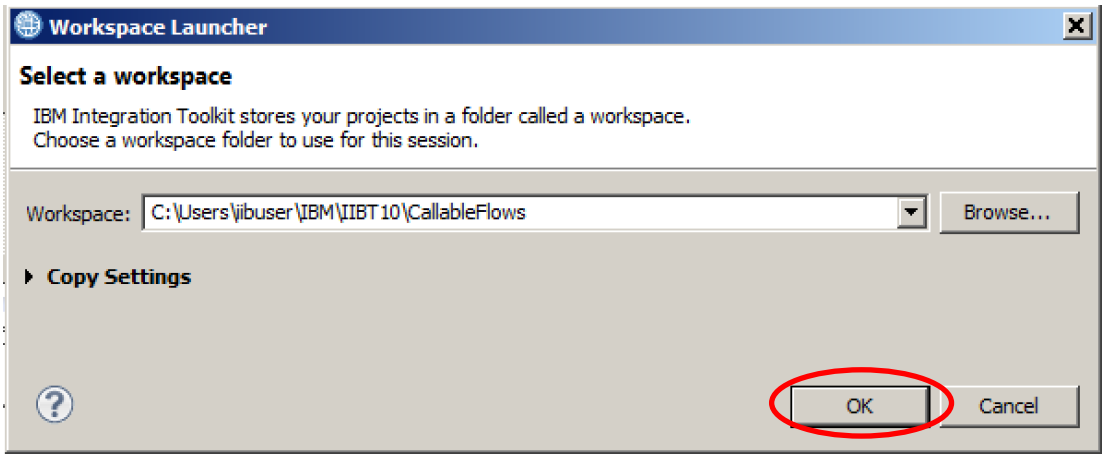
2. Import IIB REST API and create new resource

In this section you will import a partially-built solution and add a new resource to the REST API to invoke a 'Callable' application. The scenario is based on the HR_Service example that you may have used in other labs in this series.

2.1 Import REST API and Shared Library

1. Open the Integration Toolkit from the Start menu with a new Workspace, for example ***C:\Users\iibuser\IBM\IIBT10\CallableFlows***

   If you have the Integration Toolkit already open, you can switch to a new Workspace by click on *File -> Switch Workspace -> Other*



2. When the Integration Toolkit restarts, right-click in the **Application Development** view and select Import. Click **Next**, navigate to the folder

   C:\student10\CallableFlows\Resources

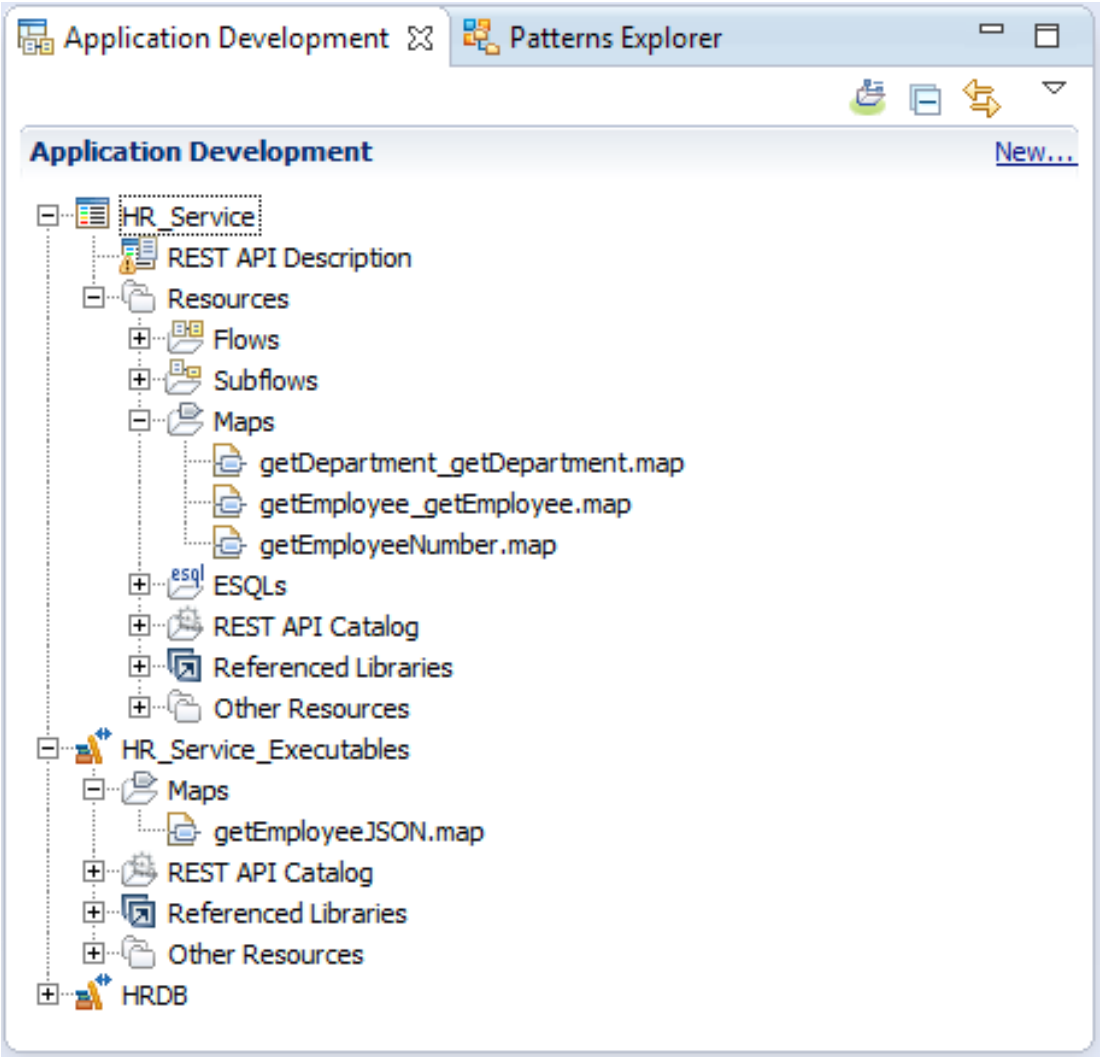
   Select the file **HR_Service_CallableFlows.zip** and click Open. Confirm the import in the next window.

   Similarly, click Import again, then Next and navigate to

   C:\student10\CallableFlows\Solution

   Select the file **HR_Service_Executables.zip**

   Once the projects are imported, you should see the REST API in the **`Application Development`** view:



2.3 Add new resource to the REST API

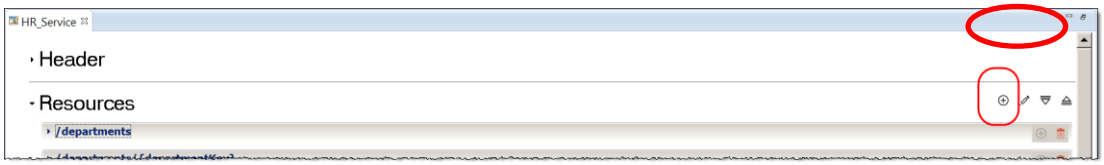
1. Double-Click on REST API Description.

   You will see all the resources available.

   The imported solution has an operation in the **`/employees/{employeeNumber}`** resource implemented.

   For this lab you will add a new resource, which will be used for the Callable Flow implementation.

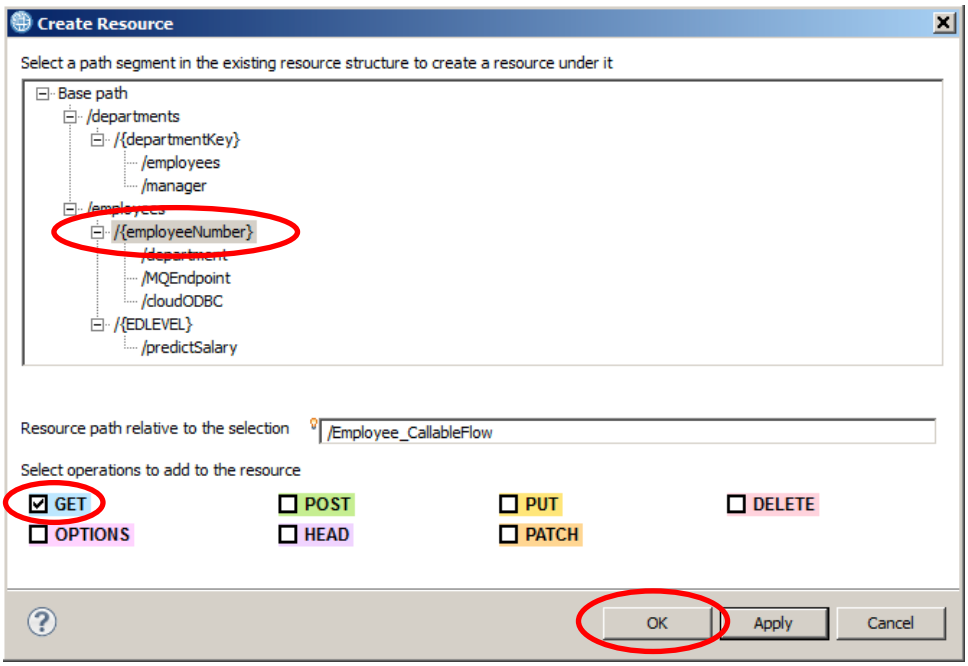
   Click on the icon ⊕ as shown.



2. In the Create Resource dialog, do the following:

   - Highlight **/{employeeNumber}**
   - Give a new 'Resource path' - **/Employee_CallableFlow**
   - Select operation **GET**

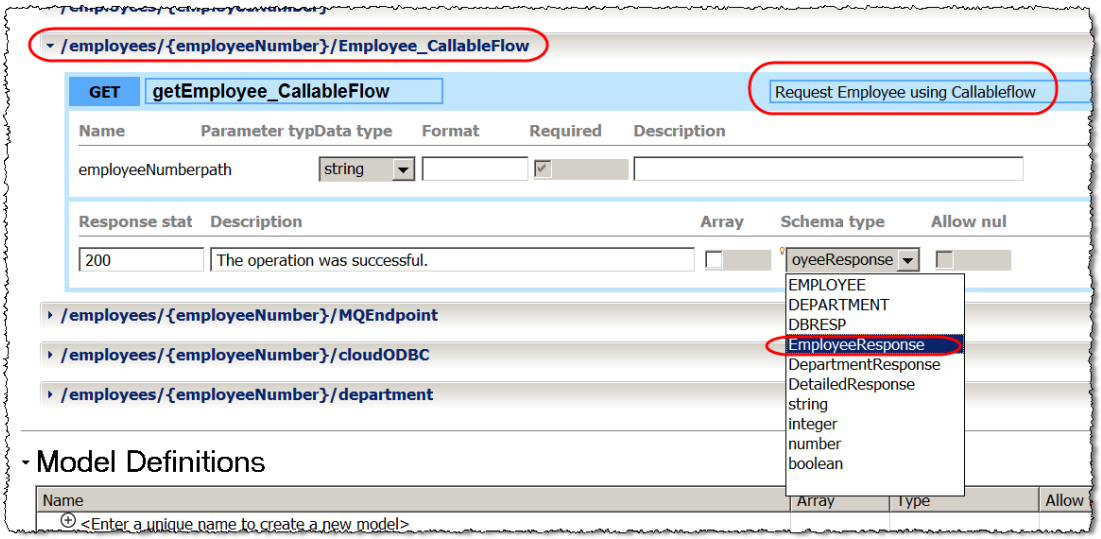
   Click OK.



3. The new resource **`/employees/{employeeNumber}/Employee_CallableFlow`** has been created.

   Expand the resource to see its operations.

   Click on the drop down menu for available 'Schema Types' and select 'EmployeeResponse'.

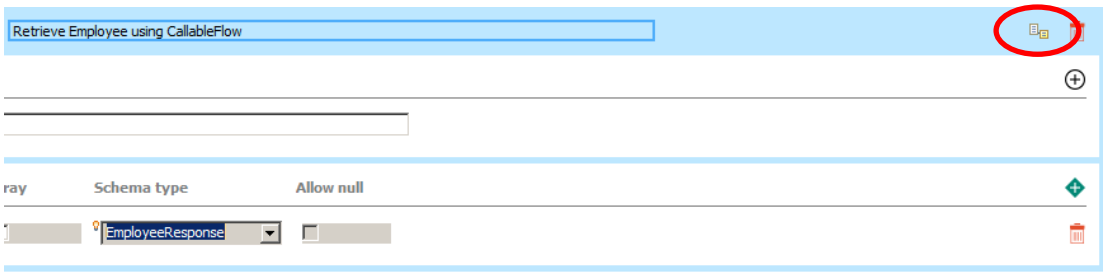


   You may have also noticed that you can provide your own comment for the operation. Add '**Retrieve Employee using CallableFlow**'.

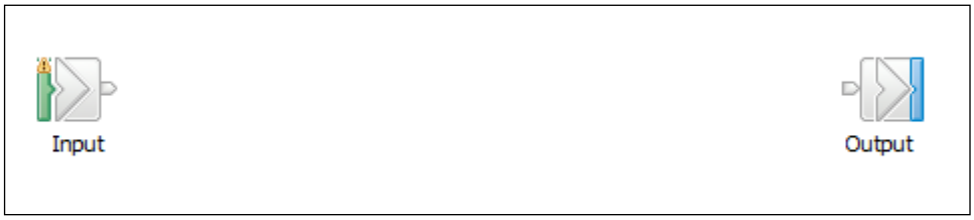
   Save the updated HR_Service.

4. Scroll to the right of **`/employees/{employeeNumber}/Employee_CallableFlow`**.

   You will be modifying the **`getEmployee_CallableFlow`** operation so to implement it, click on the icon  as shown



5. This will open the subflow editor with a skeleton subflow.

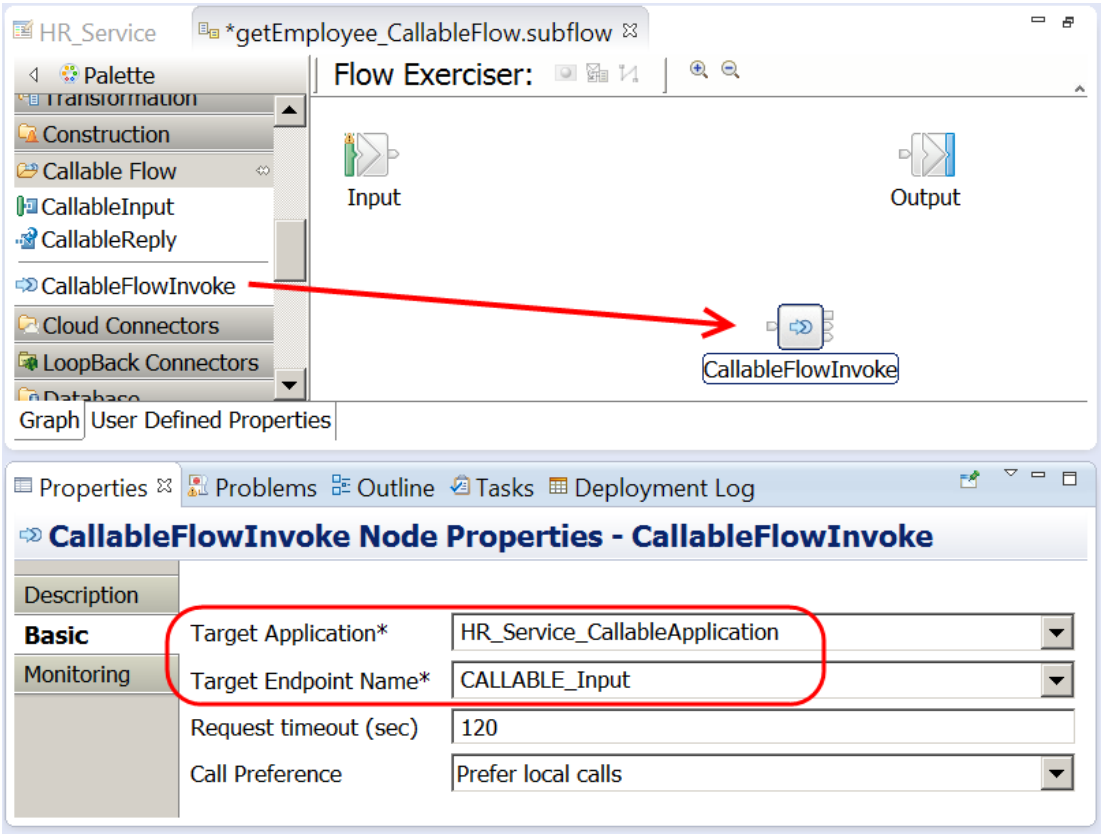


6. In the Node Palette on the left, expand the 'Callable Flow' group. This contains three nodes, which were introduced in IIB V10.0.0.4.

   Drag and drop a new **`CallableFlowInvoke`** node onto the flow editor as shown.

   In the **Properties** of the CallableFlowInvoke node, set the following properties:

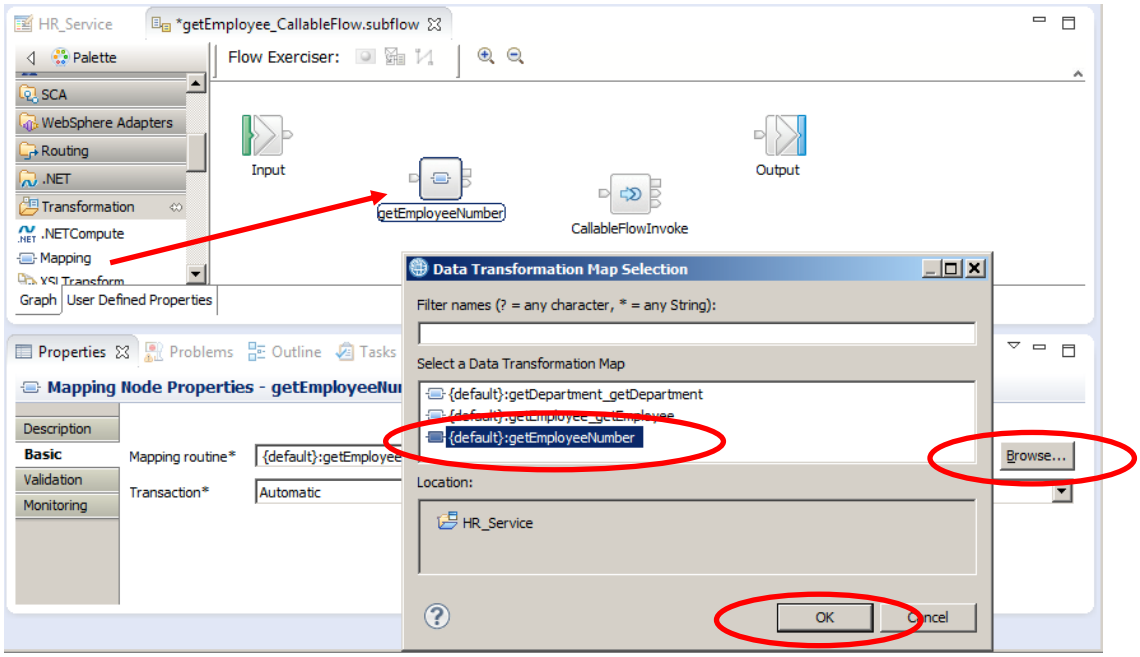
   - **Target Application**:
     - **HR_Service_CallableApplication** – this is the IIB application which will be deployed On-Premise and will retrieve the employee data from the database.
   - **Target Endpoint Name**:
     - **CALLABLE_Input** – Callable Input node in the Callable Application.



7. From the 'Transformation' group in the palette, drag and drop a Mapping node before the CallableFlowInvoke node and name it **getEmployeeNumber**.

   Highlight the mapping node and click **Browse**.

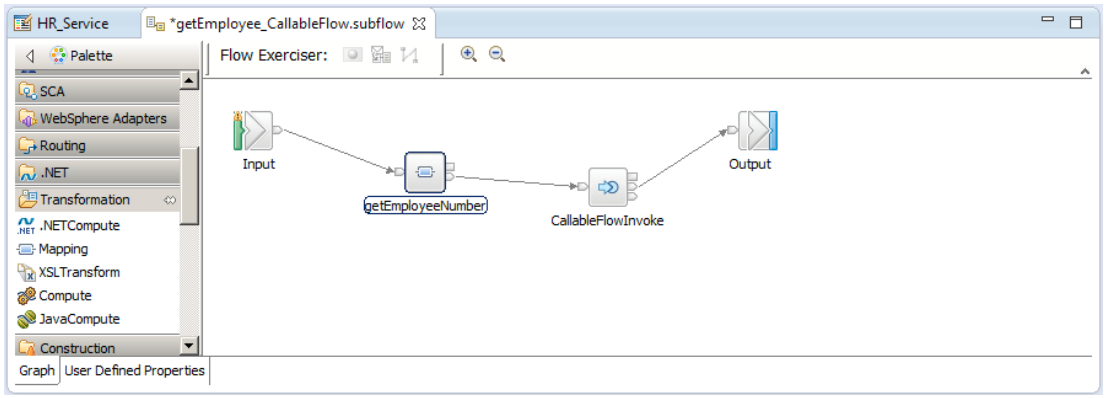
   Select **getEmployeeNumber** map and click **OK**.



   The map **getEmployeeNumber** retrieves the employeeNumber element from the Local Environment, passed on by the REST API call. The map was created as part of the starting solution for Lab 16L09 Endpoint connectors.

8. Connect the nodes as shown.

   Save the flow with **`Ctrl-S`**.



3. Create the Callable application

In this part of the lab you will create the Callable application which will be used in IIB on Cloud Callable Flows (16L12) and Docker Callable Flows (16L11) labs.

3.1 Generate the new application and add artefacts

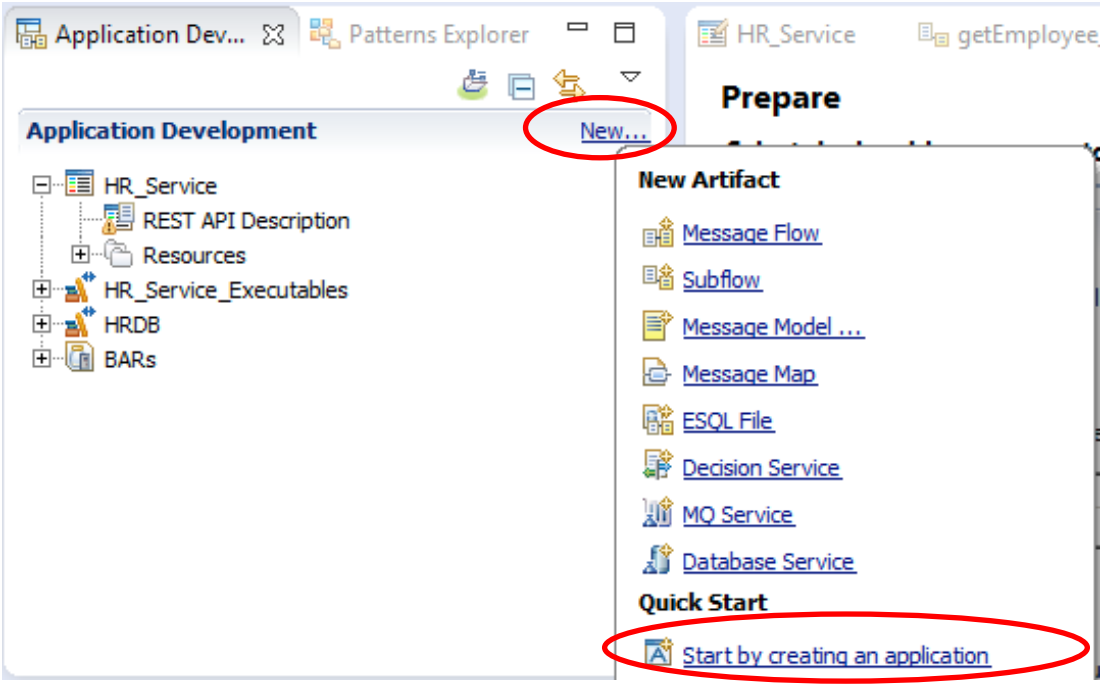
The Application that you will develop will use the EMPLOYEE table from the HRDB database. This requires the HRDB Database Definition project, which represents the tables' schemas. This is used by the Mapping nodes that access the EMPLOYEE table.

The HRDB project will not be developed in this lab - see Lab 16L02 for details of how to do this. In this lab the project is included in a shared library and is already in your Workspace from the import of the solution earlier.

In addition, the map getEmployeeJSON – in Shared Library "HR_Service_Executables" uses an existing JSON document (HR-Employee-and-Department-Services.json) which has defined schemas to represent the input and output messages. The map getEmployeeJSON retrieves rows from the EMPLOYEE table.

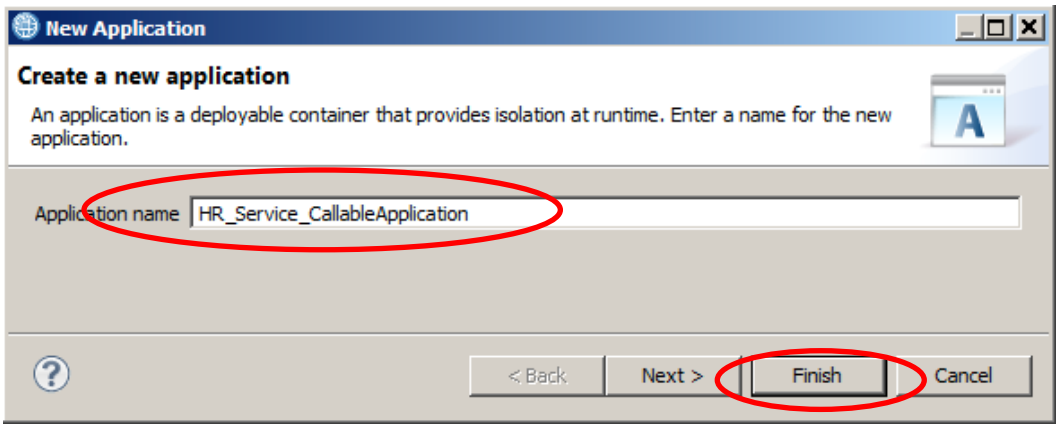
The generation of these artefacts was done in earlier labs in this series and have been imported into the workspace. If you wish to see how they were created, please review Lab 16L09 Endpoint Connectors lab.

1. In the **`Application Development`** view click '**`New`**…' and then '**`Start by creating an application`**'.

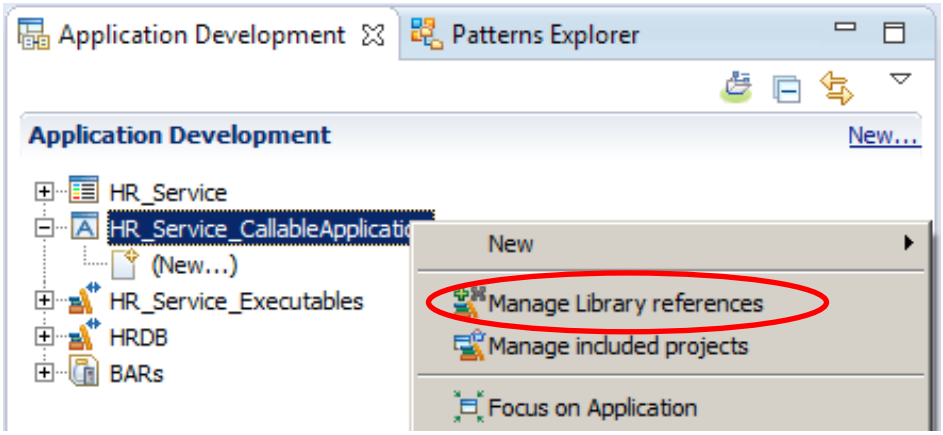


2. Name the application '**`HR_Service_CallableApplication`**'.

   Click '**`Finish`**'.

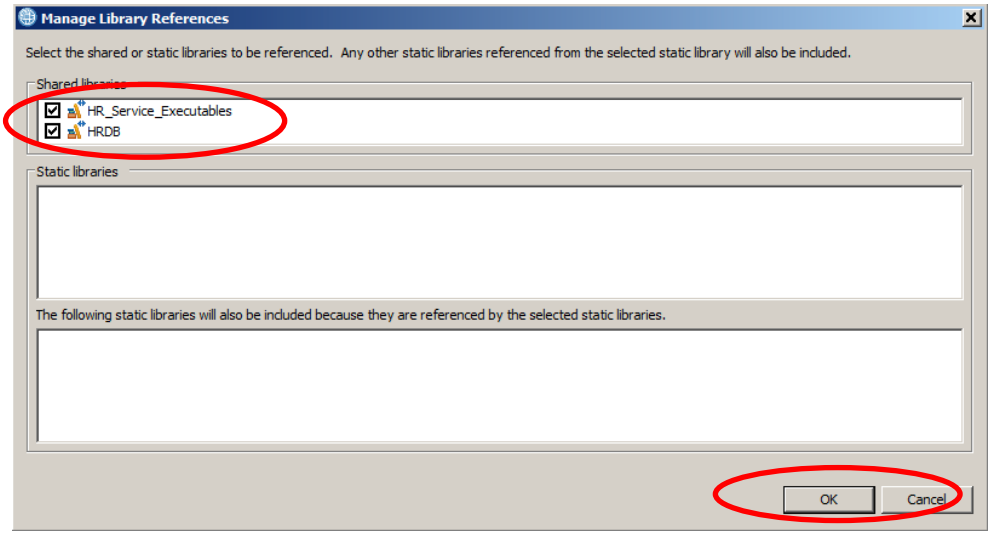


3. Right-click the new application and then 'Manage Library references'



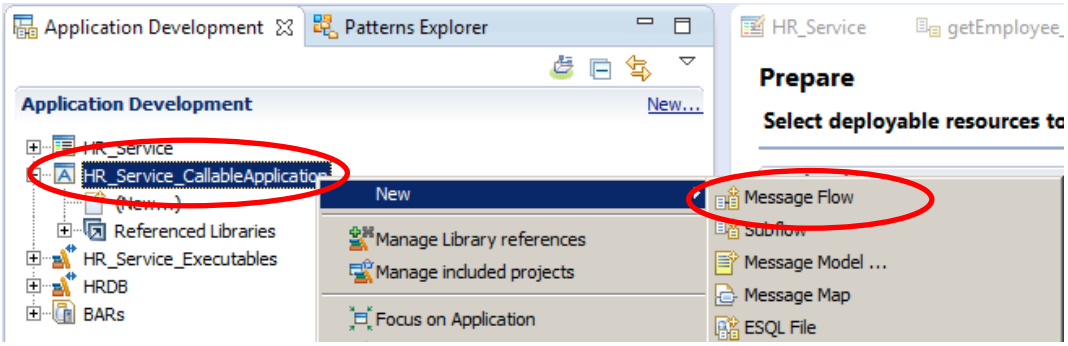
4. In the opened dialog, tick the box for HR_Service_Executables and HRDB, and then OK.

   This will allow the HR_Service_CallableApplication to reference this shared library.

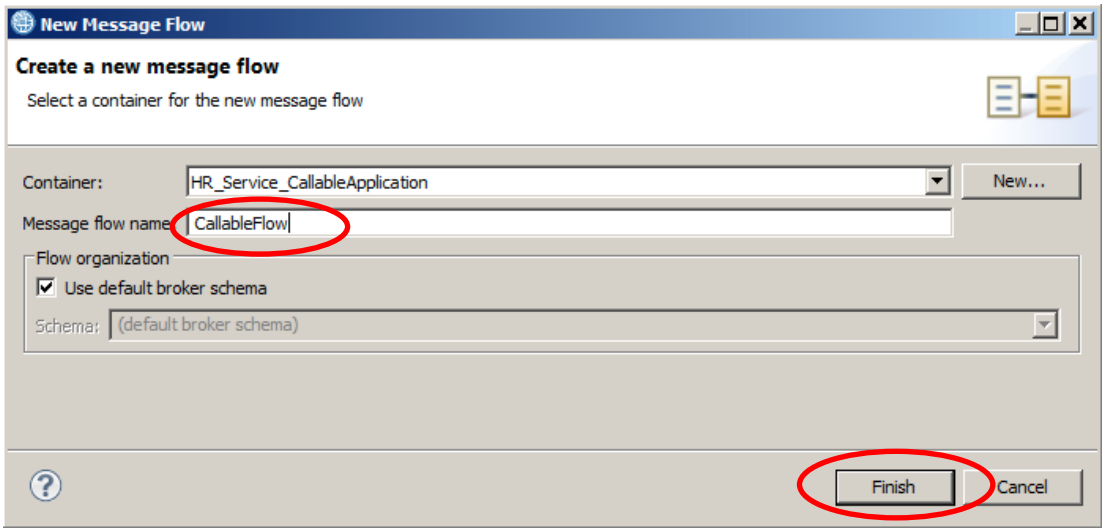


3.2 Create the Callable Application's message flow

1. Right-click **`HR_Service_CallableApplication`** then **`New`** -> **`Message Flow.`**



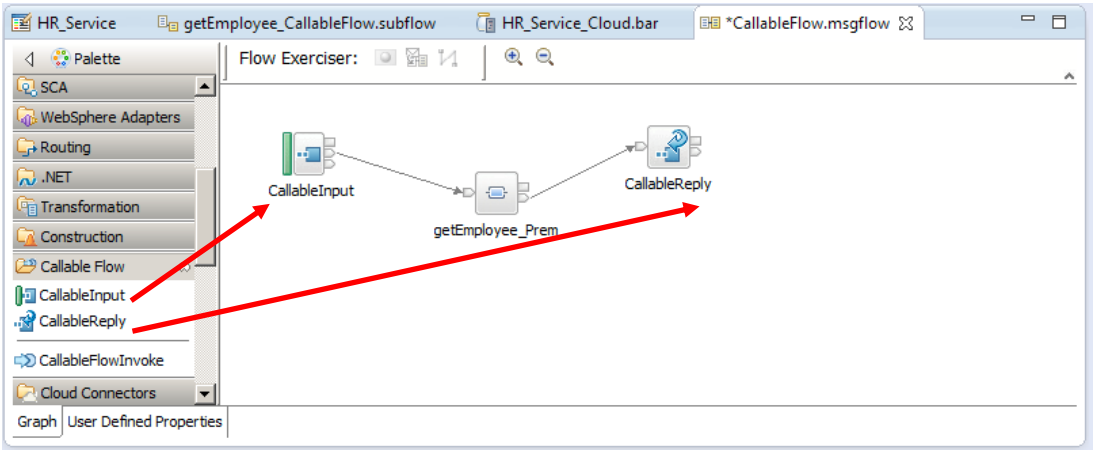
2. Name the new message flow '**`CallableFlow`**' and click **`Finish`**.



3. Expand the **`CallableFlow`** folder in the Palette and drop a **`CallableInput`** and **`CallableReply`** node onto the flow editor.

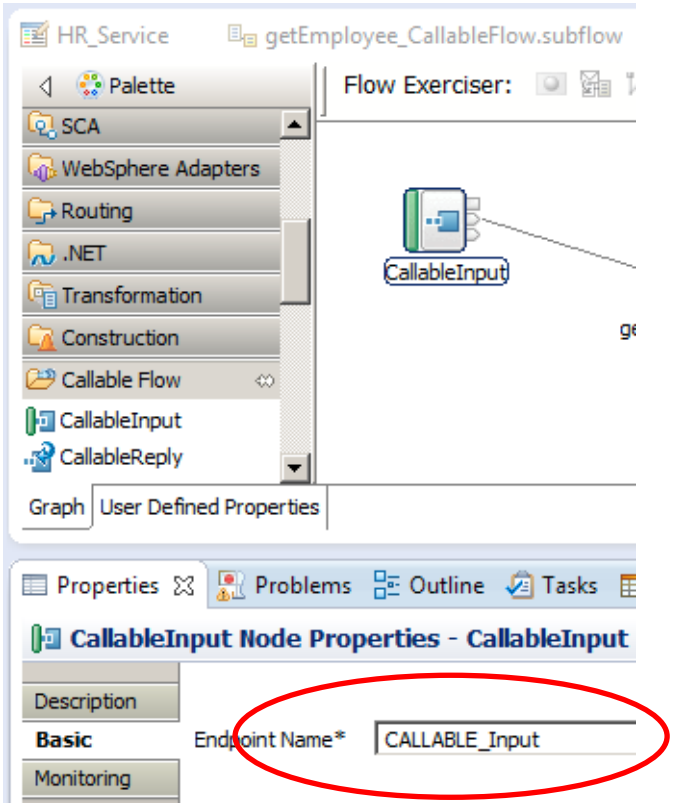
   From the **`Transformation`** folder, drop a **`Mapping`** node onto the editor, and name it **`getEmployee_Prem`**.

   Connect the nodes as shown below.



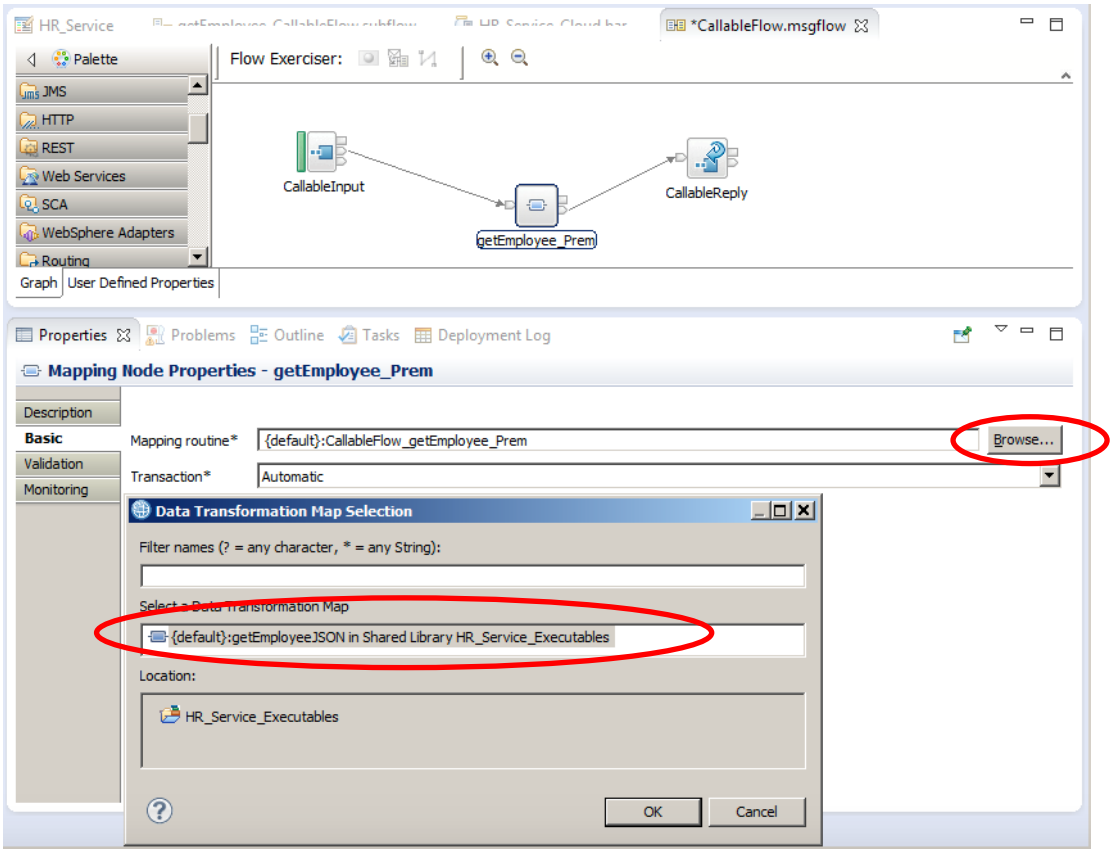
4. Highlight the **`CallableInput`** node and on the **`Basic Properties`**, set the Endpoint Name to **CALLABLE_Input**.

   This Endpoint name needs to match the Target Endpoint Name that was specified in the CallableFlowInvoke node in the implementation for /employees/{employeeNumber}/Employee_CallableFlow in HR_Service (step 2.3.6).

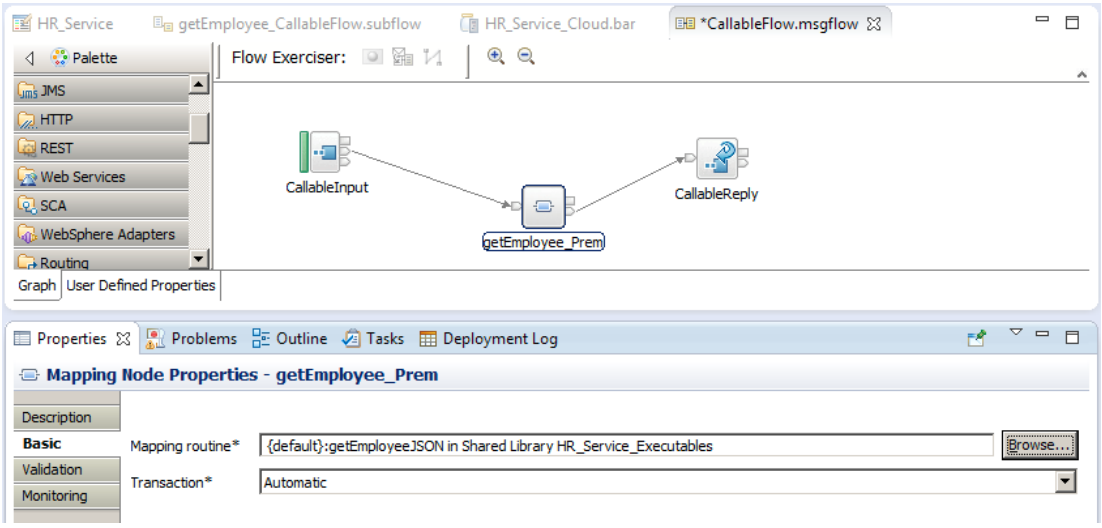


5. Highlight the Mapping node **`getEmployee_Prem`**.

   In the properties of the map node, use the **`Browse`** button to set the name of the map file to **`getEmployee_JSON`**. (This is a map in the HR_Service_Executables shared library that you imported and referenced earlier in this lab).



6. The flow is now complete, so **Save** and close the flow.

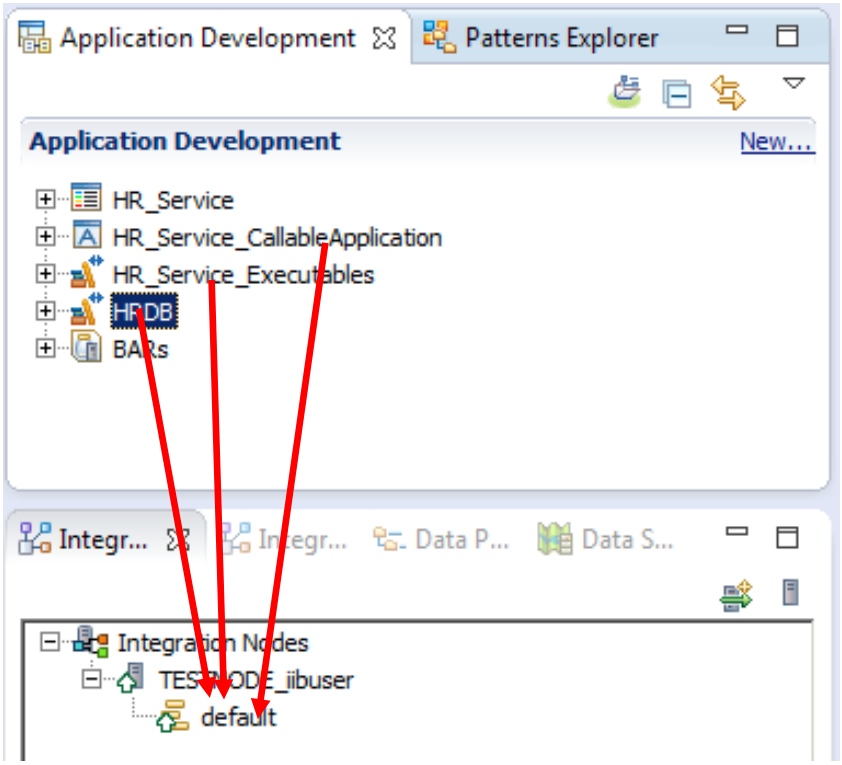


4. Test the new REST API Resource and Callable Application

In this part of the lab a quick test will be done to make sure that your developed solution works. Please note that in this test you will be deploying all the resources to the same Integration server. In IIB on Cloud Callable Flows (16L12) and Docker Callable Flows (16L11) labs you will see how you can use Callable Flows between different Nodes/ environments.

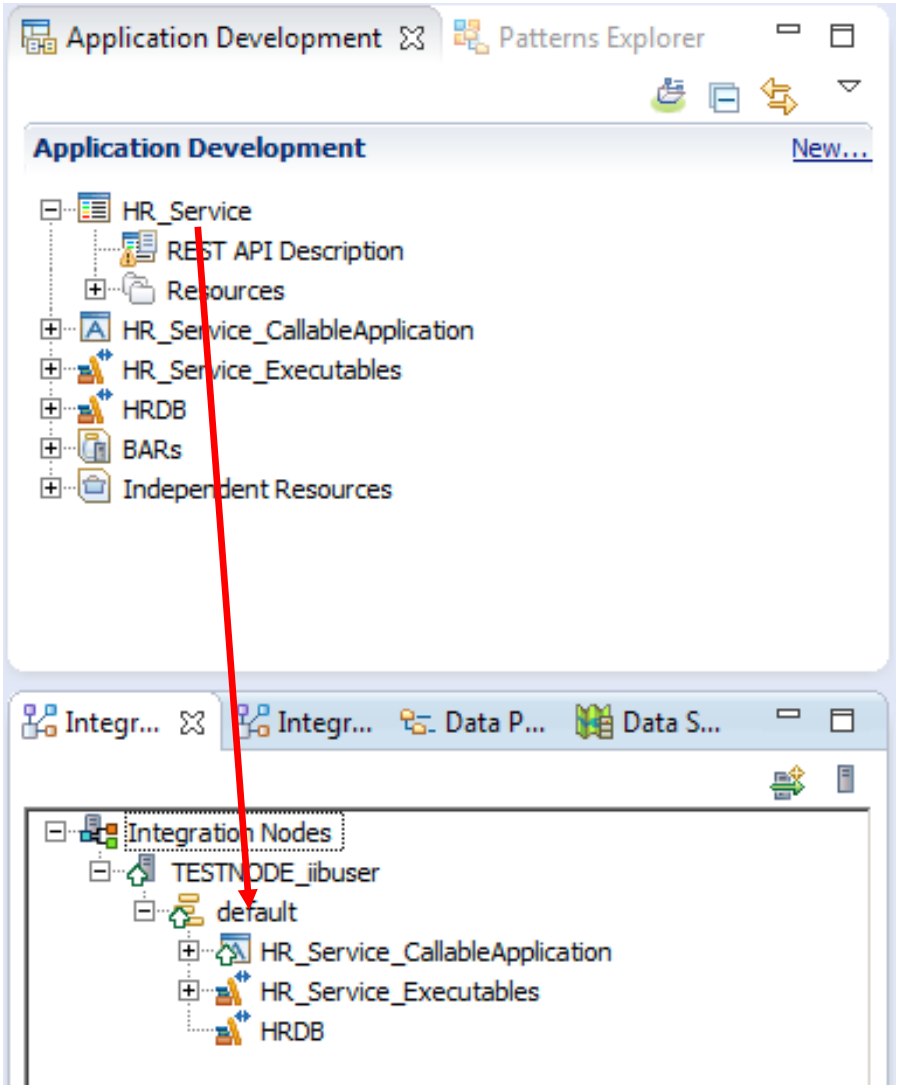
4.1 Deploy the Shared Library and the Callable Application

1. In the IIB Toolkit navigator, deploy **first** the HRDB and HR_Service_Executables shared libraries to the default server and then the HR_Service_CallableApplication.



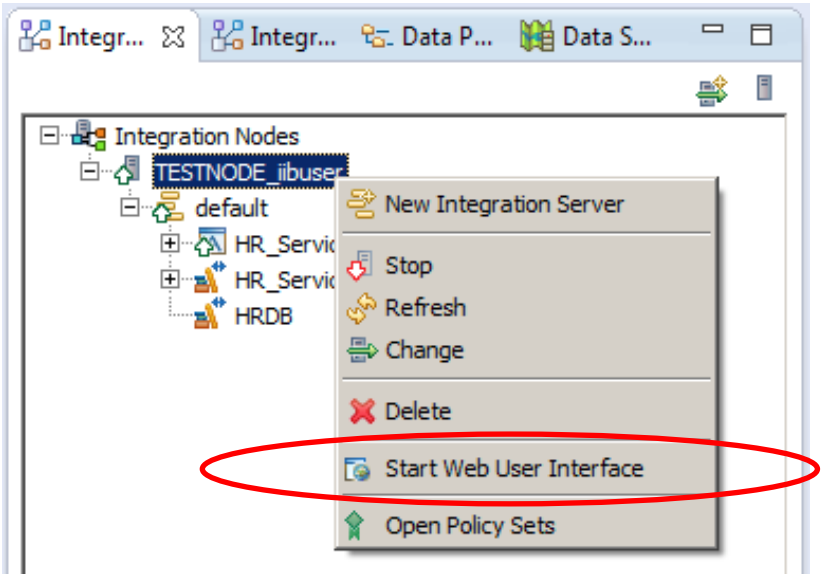
4.2 Deploy the service

1. In the IIB Toolkit navigator, deploy HR_Service REST API to the default server.



4.3 Test the service

1. Open the IIB web UI by right-clicking TESTNODE_iibuser and selecting Start Web User Interface.

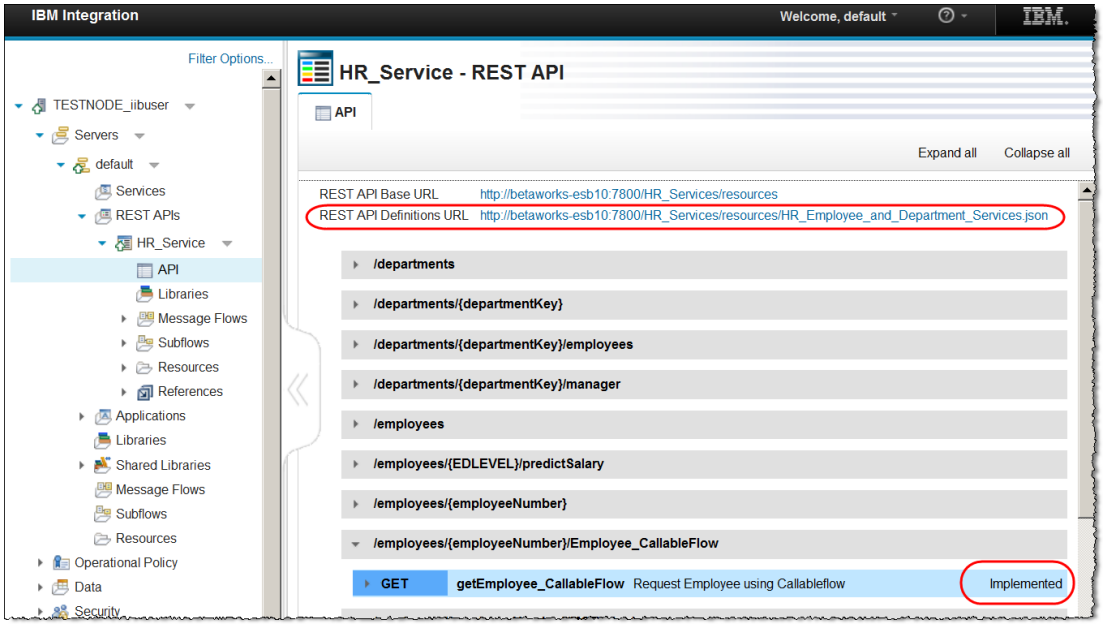


2. You will be switched to the default browser. Fully expand TESTNODE_iibuser, down to the HR_Service REST API, as shown below.

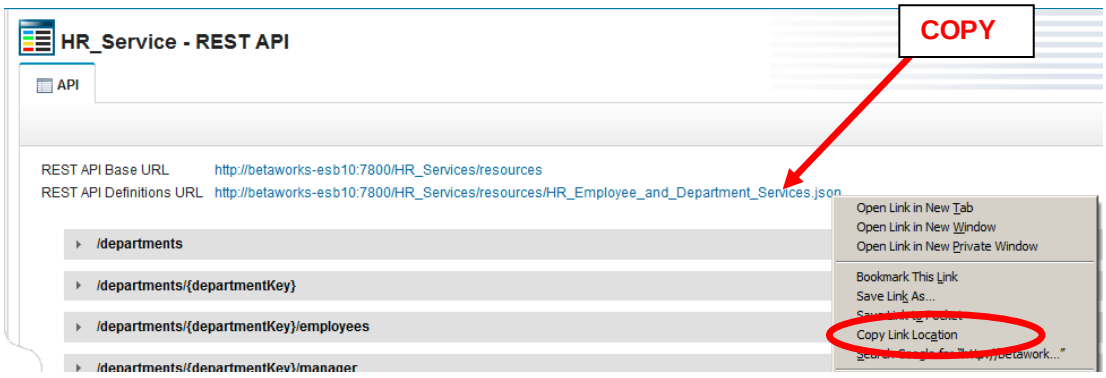
   Under HR_Service, click "API", which will show you the available operations in the REST API, and whether they have been implemented. You will be able to see a few operations that had been implemented.

   For this lab we will use **Employee_CallableFlow** operation, so check that you see the operation has been implemented.

   It will also show you the URLs for the REST API and the definitions (the json file).



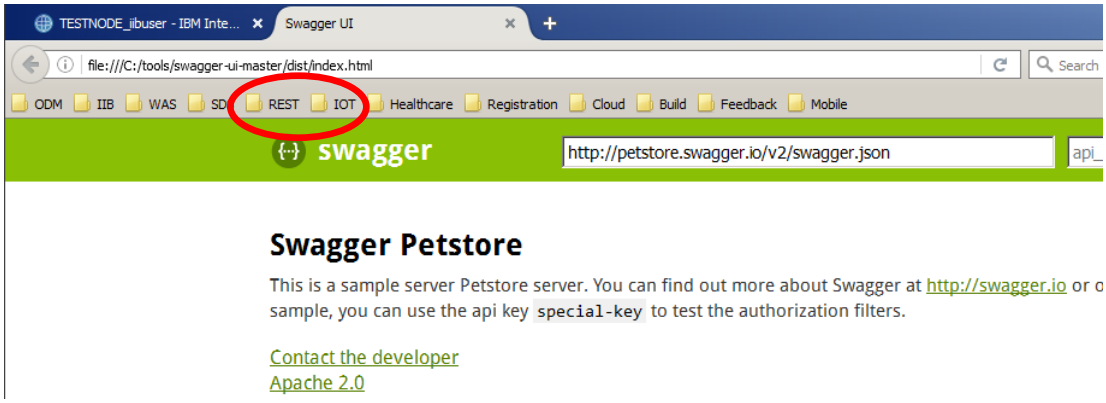
3. On the "REST API Definitions URL", right-click and select "Copy Link Location".



4. In Firefox, open a new tab, and open the SwaggerUI tool (using the bookmark in the REST folder).

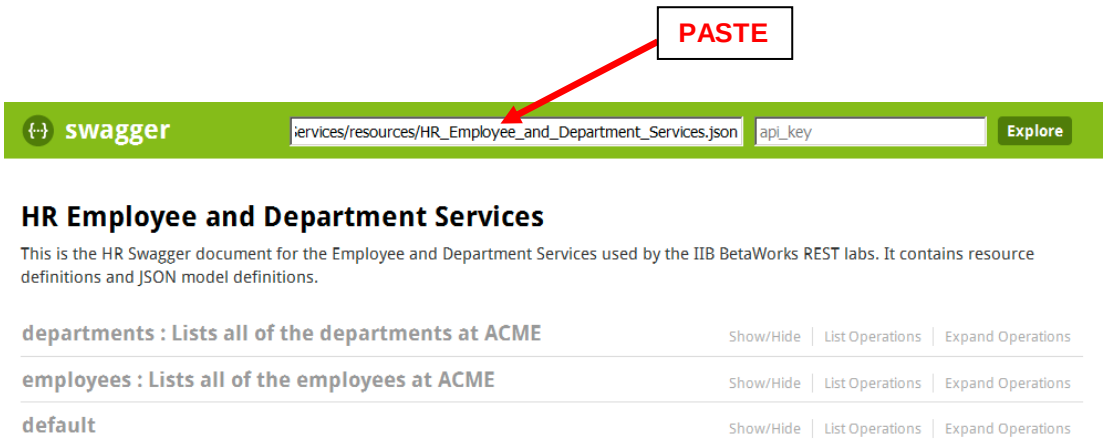
   If you are using your own environment, open Swagger UI.

   By default, this will open the Petstore Swagger document.



5. In the entry field (not the browser address field), paste the contents of the clipboard and click Explore.

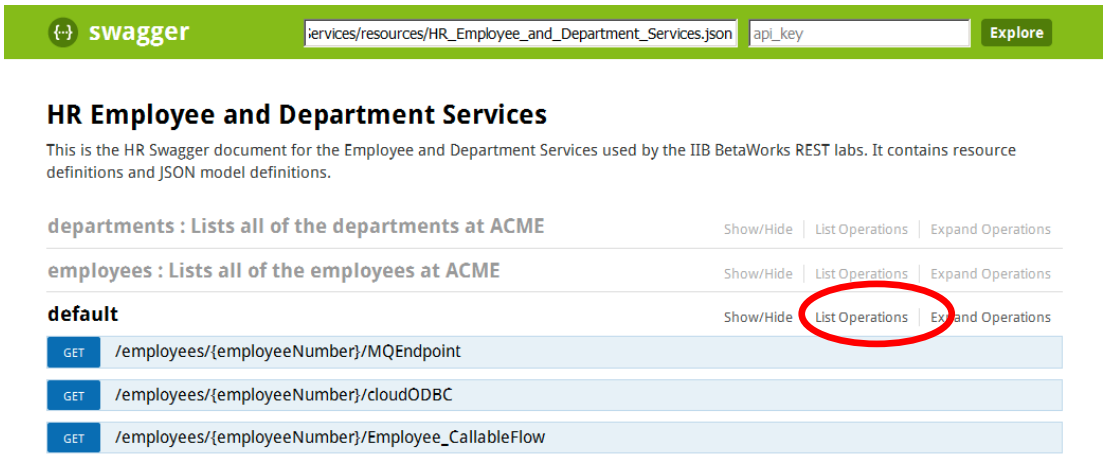
   The two high-level tags departments and employees, will be shown.



6. We are concerned with the getEmployee_CallableFlow, which is in the 'default' section because the IIB Toolkit does not offer the option to provide a "tags" parameter on the new definition.

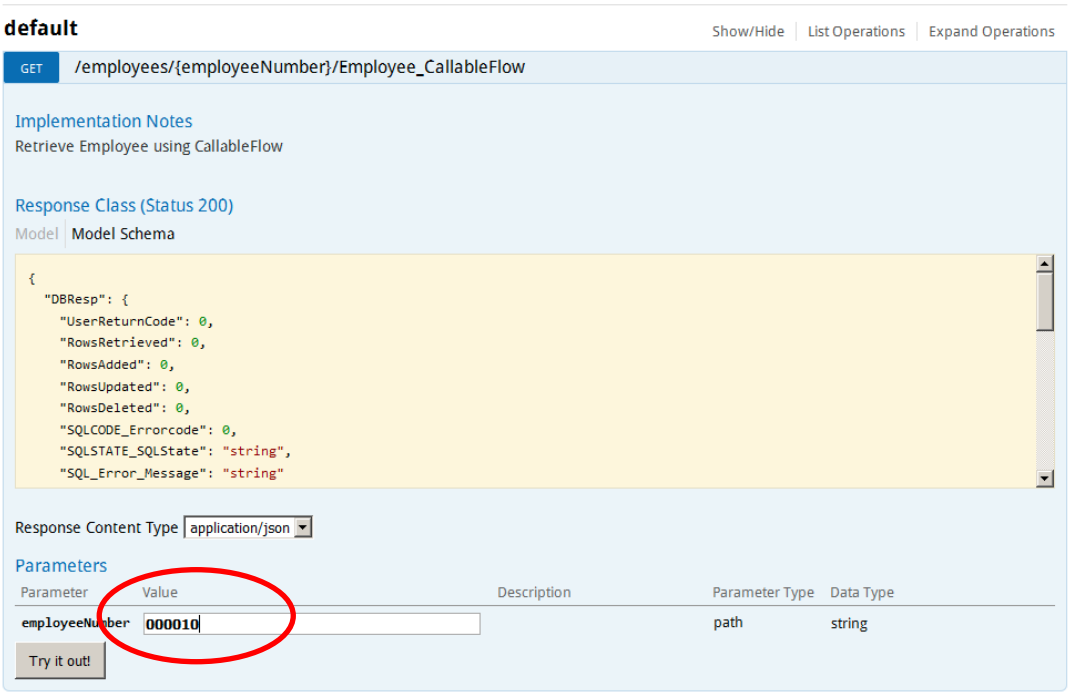
   Click "List Operations" to show the operations related to employees.

   Note that SwaggerUI does not have any knowledge at this point of whether the operation has been implemented.



7. Expand the GET **employees/{employeeNumber}/Employee_CallableFlow** operation by clicking it.

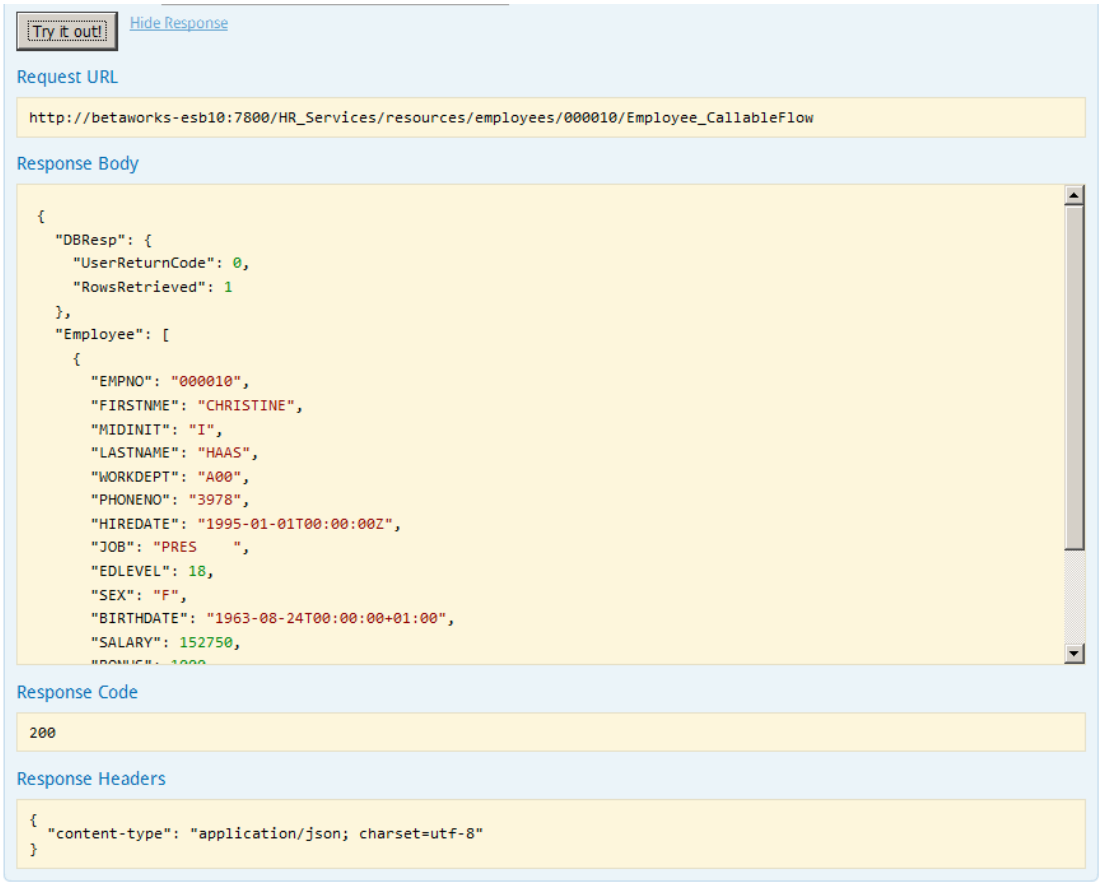
   The input parameter is employeeNumber. Provide a suitable value, say 000010.



8. When you have provided an employeeNumber, click Try it out!

   If successful, the returned data will look something like this. Note the database response information (user return code, number of rows returned), as well as the user data.

   Note that you can copy the Request URL below, and paste directly into a browser.



   Now that you have completed this lab for developing Callable Flows artefacts you can proceed to IIB on Cloud Callable Flows (16L12) and Docker Callable Flows (16L11) labs. In these labs you will see the value of using Callable Flows in distributed environments such as IIB on Cloud or Docker.

END OF LAB GUIDE