

IBM z/OS Connect EE V3.0

# Developing RESTful APIs for IMS Database Services



**IBM Z**  
**Wildfire Team –**  
**Washington System Center**

## Table of Contents

<b>Overview .....</b>	<b>3</b>
<b><i>IMS Enterprise Suite Explorer for Development</i> .....</b>	<b>4</b>
<b><i>Connect to a z/OS Connect EE Server</i> .....</b>	<b>17</b>
<b><i>z/OS Connect EE APIs and IMS</i> .....</b>	<b>21</b>
<i>Create the IMS Service Projects</i> .....	21
<i>Test the IMS Database services with Postman</i> .....	29
<i>Summary</i> .....	33
<b><i>Create the IMS API Project</i> .....</b>	<b>34</b>
<i>Import the SAR files</i> .....	36
<i>Compose an API for IMS Database Services</i> .....	38
<b><i>Deploy the API to a z/OS Connect EE Server</i> .....</b>	<b>45</b>
<b><i>Test the IMS APIs</i> .....</b>	<b>47</b>

**Important:** On the desktop there is a file named *Developing APIs CopyPaste.txt*. This file contains commands and other text used in this workshop. Locate that file and open it. Use the copy-and-paste function (**Ctrl-C** and **Ctrl-V**) to enter commands or text. It will save time and help avoid typo errors. As a reminder text that appears in this file will be highlighted in yellow.

## Overview

**Important – You do not need any skills with IMS to perform this exercise. Even if IMS is not relevant to your current plans performing this exercise will give additional experience using the API toolkit with developing APIs.**

The objective of these exercises is to gain experience with working with z/OS Connect EE and the API toolkit. These two products allow the exposure of z/OS resources to JSON clients. More in-depth information about the customization of z/OS Connect EE, z/OS Connect EE security, the use of the API toolkit and other topics is provided by the 1-day *ZCONNEE – z/OS Connect Workshop*. For information about scheduling this workshop in your area contact your IBM representative.

## *General Exercise Information and Guidelines*

- ✓ This exercise requires using z/OS user identity *USER1*. The password for this user will be provided by the lab instructor.
- ✓ Any time you have any questions about the use of IBM z/OS Explorer, 3270 screens, features or tools do not hesitate to ask the instructor for assistance.
- ✓ Text in **bold** and highlighted in **yellow** in this document should be available for copying and pasting in a file named *Development APIs CopyPaste* file on the desktop
- ✓ Please note that there may be minor differences between the screen shots in this exercise versus what you see when performing this exercise. These differences should not impact the completion of this exercise.

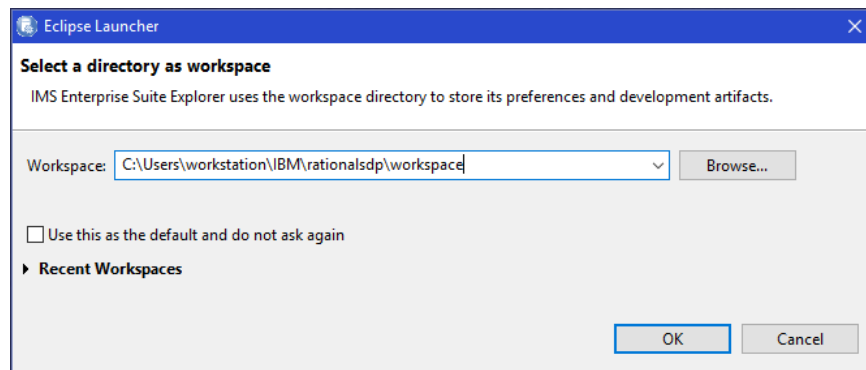
## *IMS Enterprise Suite Explorer for Development*

Start the *IMS Explorer Suite* by opening the *IMS Enterprise Suite Explorer for Development* icon on the desktop. Begin by establishing a database connection to IMS Connect running on z/OS. from IBM z/OS Explorer. If you have performed one of the other exercises in this series of exercises this step may not be required.

**Tech-Tip:** Windows desktop tools can be opened either by double clicking the icon or by selecting the icon and right mouse button clicking and then selecting the *Open* option.

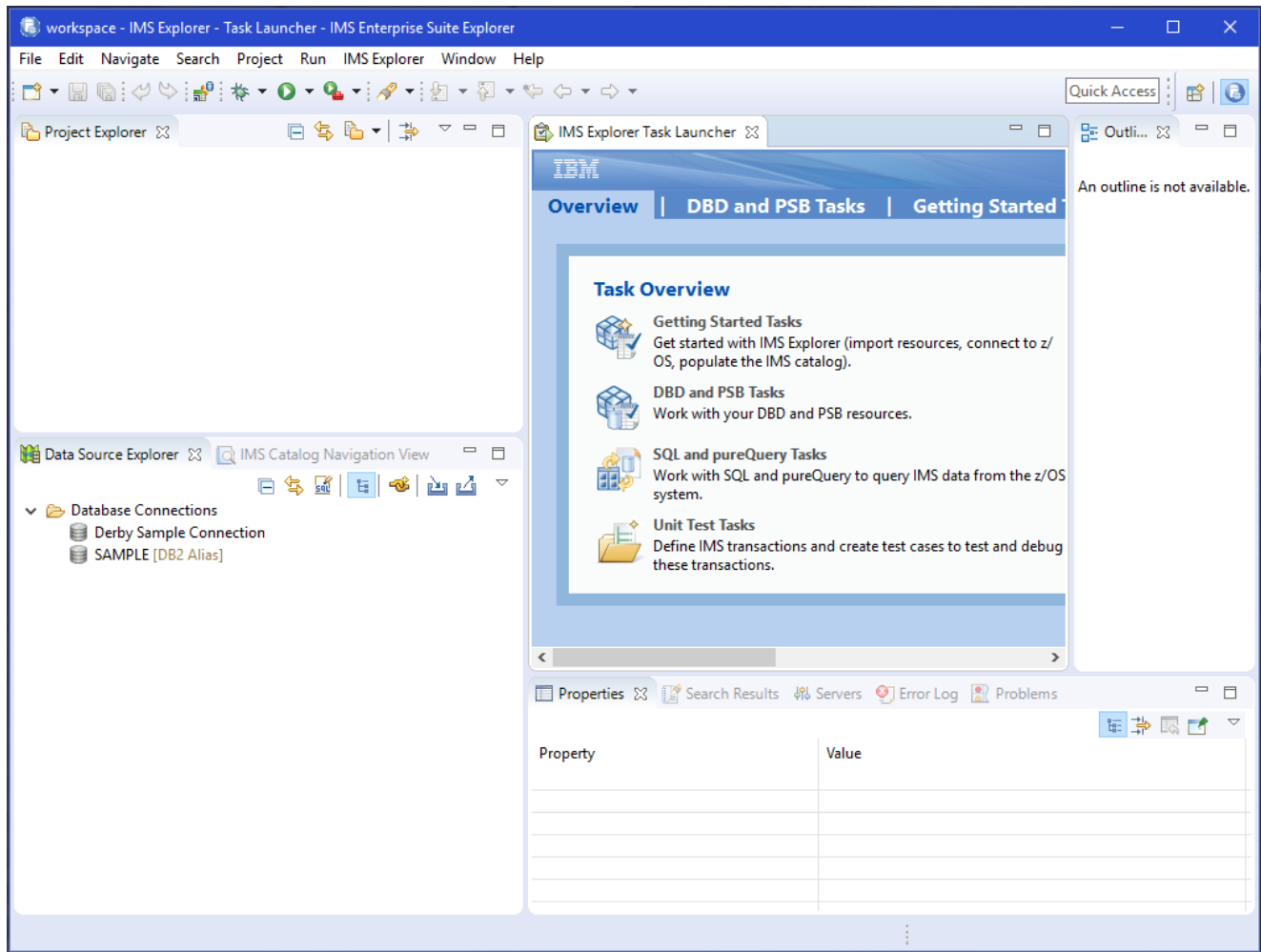
\_\_\_1. On the workstation desktop, locate the *IMS Enterprise Suite Explorer for Development* icon and double click on it to open the IMS Explorer.

\_\_\_2. You will be prompted for a workspace:



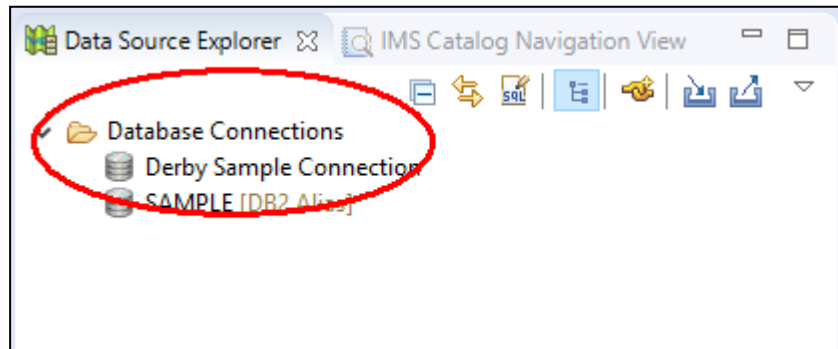
Take the default value by clicking **OK**.

3. When the IMS Explorer is initialized you should see:

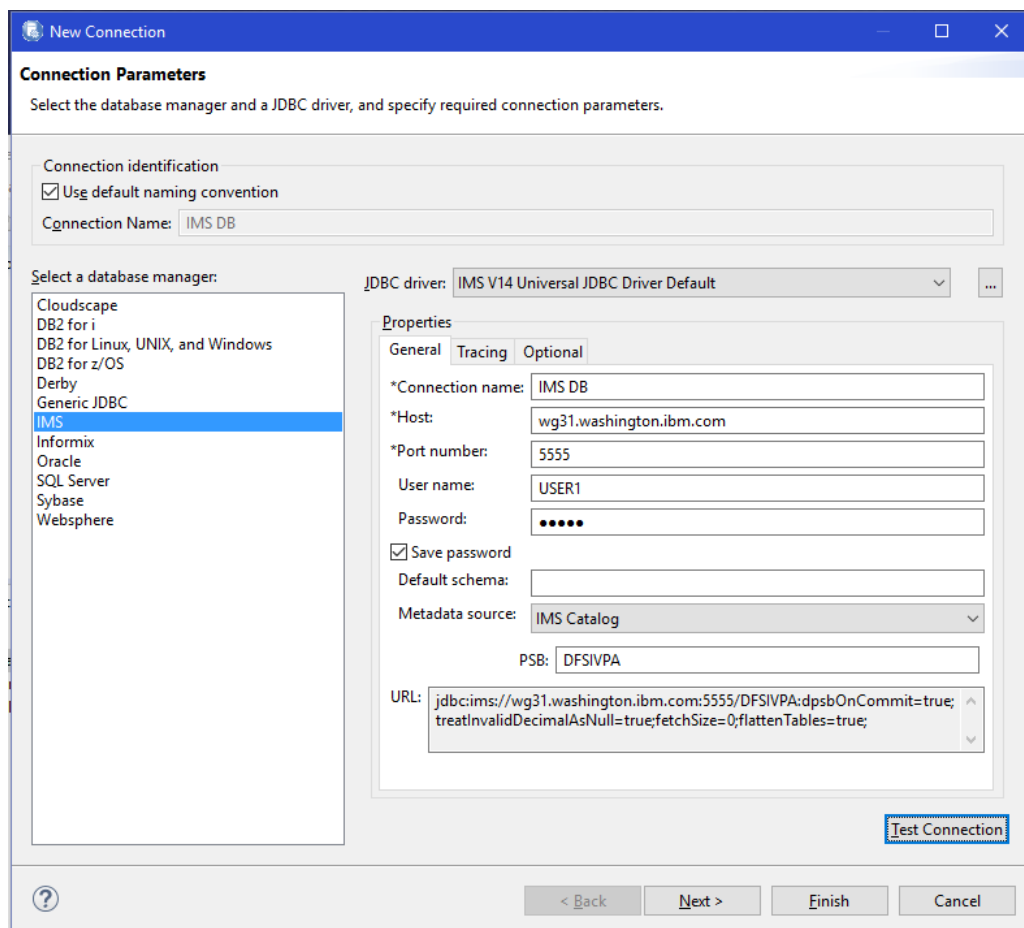


Next we need connect to the IMS Catalog to obtain details about the database that is to accessed.

4. To add a connection to the database, select *Database Connections* in the *Data Source Explorer* in the lower left hand view and then click the right mouse button.



5. Select the *New* option to open the *New Connection – Connection Parameters* window. Select *IMS* on the left-hand side under *Select a database manager* options. On the *JDBC driver* properties window enter ***wg31.washington.ibm.com*** as the *Host*, ***5555*** as the *Port number*, ***USER1*** as the *User name*, ***USER1***'s password as the *Password* and ***PSB DFSIVPA*** as the *PSB name*.



**Tech-Tip:** The value for the port number was derived from the IMS Connect for ODA access.

```
ODACCESS=(ODBMAUTOCONN=Y,
DRDAPORT=(ID=5555,PORTTMOT=6000),ODBMTMOT=6000)
```

**Tech-Tip:** PSB DFSIVPA is a sample program specification block provided with the IMS IVP. The sample PSB was modified to change the *PROCOPT* attribute from *LS* to *A*

```
ATSIVPA    PCB      TYPE=DB,DBDNAME=IVPDB1,PROCOPT=LS,KEYLEN=10
           SENSEG   NAME=A1111111,PARENT=0
           PSBGEN   LANG=ASSEM,PSBNAME=DFSIVPA
           END
```

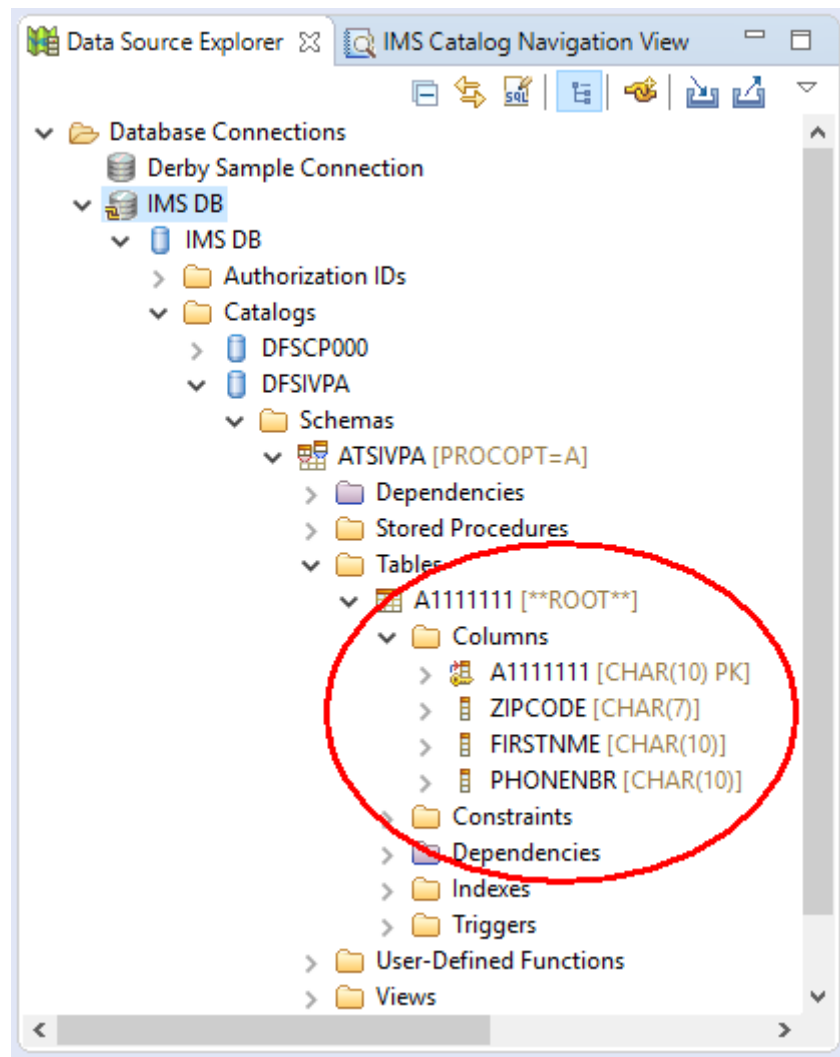
The IVPDB1 data base description (DBD) referenced by PSB DFSIVPA was also modified to by adding FIELD macros for FIRSTNME,PHONENBR and ZIPCODE.

```
DBD      NAME=IVPDB1,ACCESS=(HIDAM,OSAM)
DATASET  DD1=DFSIVD1,DEVICE=3380,SIZE=2048
SEGM     NAME=A1111111,PARENT=0,BYTES=40,RULES=(LLV,LAST),      x
        PTR=(TB,CTR)
FIELD    NAME=(A1111111,SEQ,U),BYTES=010,START=00001,TYPE=C
FIELD    NAME=FIRSTNME,BYTES=010,START=00011,TYPE=C
FIELD    NAME=PHONENBR,BYTES=010,START=00021,TYPE=C
FIELD    NAME=ZIPCODE,BYTES=7,START=00031,TYPE=C
LCHILD   NAME=(A1,IVPDB1I),POINTER=INDX,RULES=LAST
DBDGEN
FINISH
END
```

PSBGENs and DBDGENs were performed and utility DFS3UACB was used to update the IMS Catalog with these updates.

\_\_\_6. Click the **Finish** button to continue.

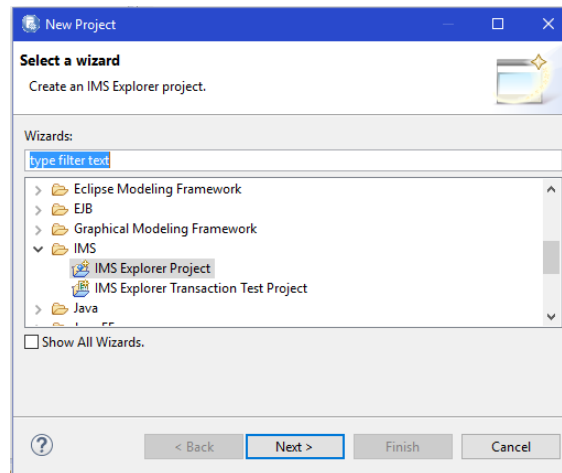
7. This adds an *IMS DB* entry to the list of *Database Connections*. Expand *Catalogs*, then expand *PSB DFSIVPA*, then expand *Tables*, then expand table *A111111* and finally expand *Columns* under table *A111111* to display the column names that can be used in SQL queries (see below).



8. Switch to the *IMS Catalog Navigation View* and select *IMS Catalog Connections*. Right mouse button click and select the *Import an IMS Catalog connection* option. On the *Import an IMS Catalog Connection* window select *IMS DB* as the *Connection name* and click **Finish** to continue.

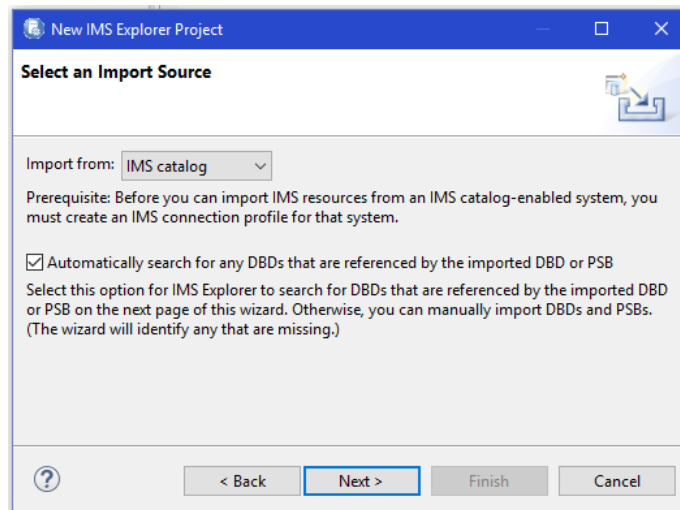


9. In the Project Explorer view right mouse button click and select *New* → *Project*. Expand the *IMS* folder and select *IMS Explorer Project*. Click **Next** to continue.

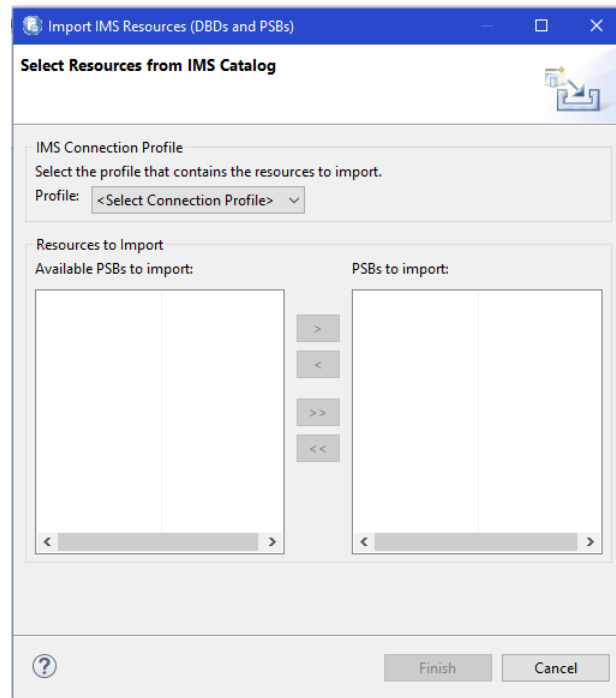


10. Enter *myIMSProject* as the Project name on the *New IMS Explorer Project* window and click **Next** to continue.

11. On the *Select an Import Source* window use the pull-down arrow to select *IMS catalog* and click **Next** to continue.

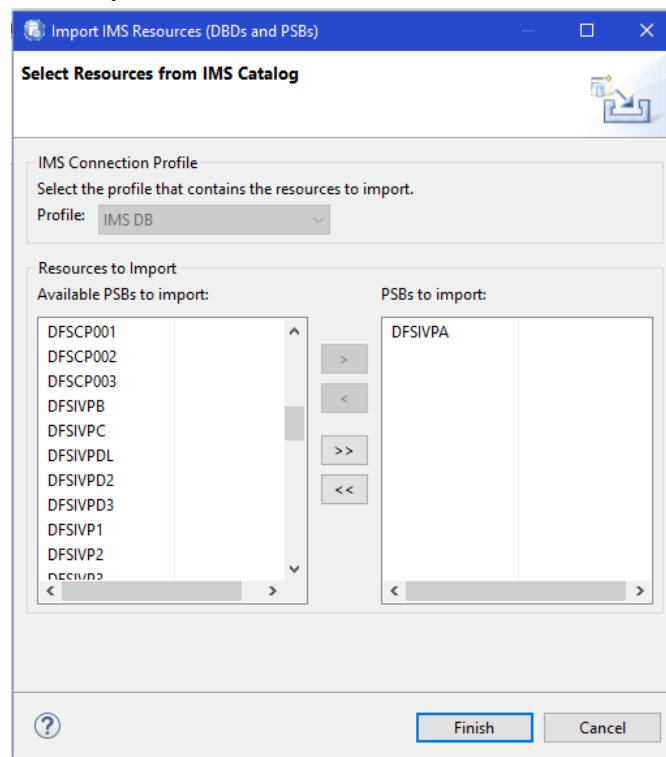


12. On the *Select Resources from IMS Catalog* window use the **Add PSB** button to display this window.

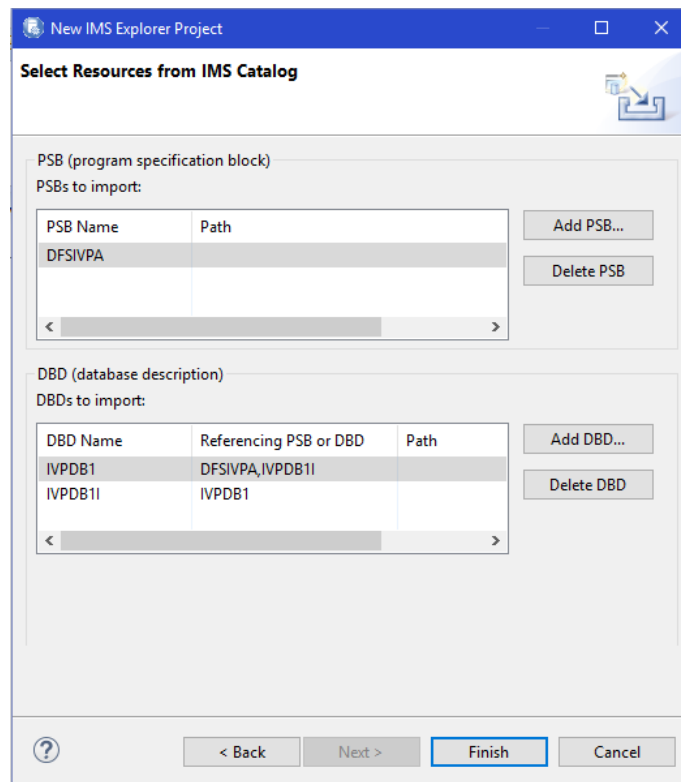


13. Use the pull-down arrow to select *IMS DB* as the *Profile*. This will access the IMS Catalog and populate a list of PSB under *Available PSBs to import*.

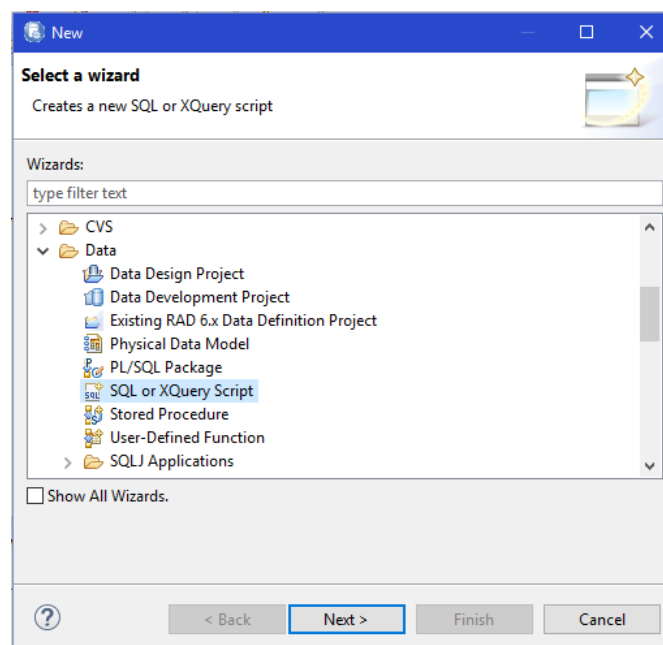
14. Locate PSB *DFSIVPA* and use the **>** symbol to move PSB *DFSIVPA* to the list of PSB to be imported.



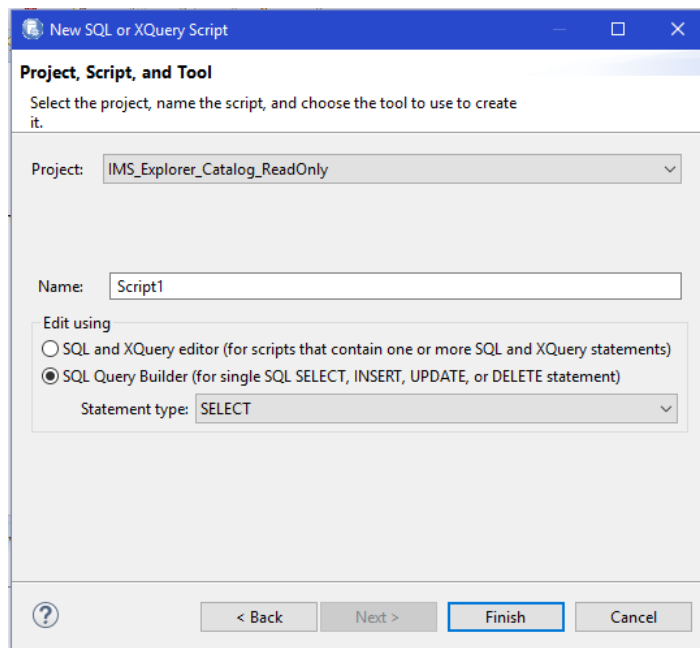
15. Click **Finish** to continue. This will automatically add the DBDs references by DFSIVPA to the list of DBDs to be imported. Click **Finish** to continue.



16. Select the new *myIMSProject* under *Project Explorer* and right mouse button click and select *New* → *Project* → *Other* → *Data* and select the *SQL or xQuery Script* wizard.

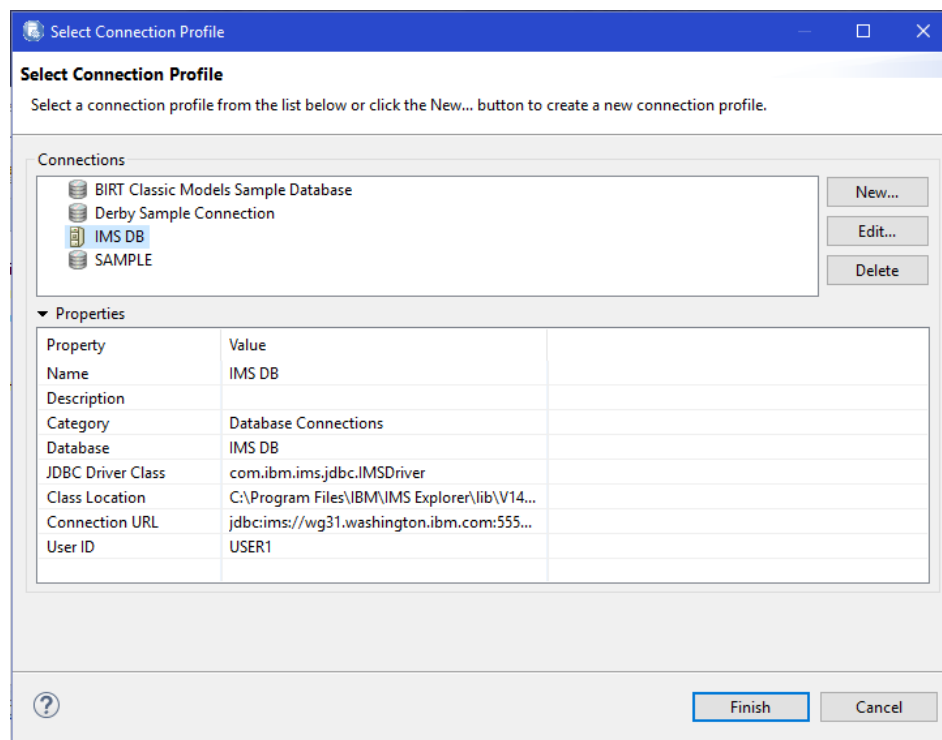


17. Click **Next** to continue and on the *Project, Script, and Tool* window select the radio button beside *SQL Query Builder* (for a single *SQL SELECT, INSERT, UPDATE, or DELETE* statement). Click **Finish** to continue.

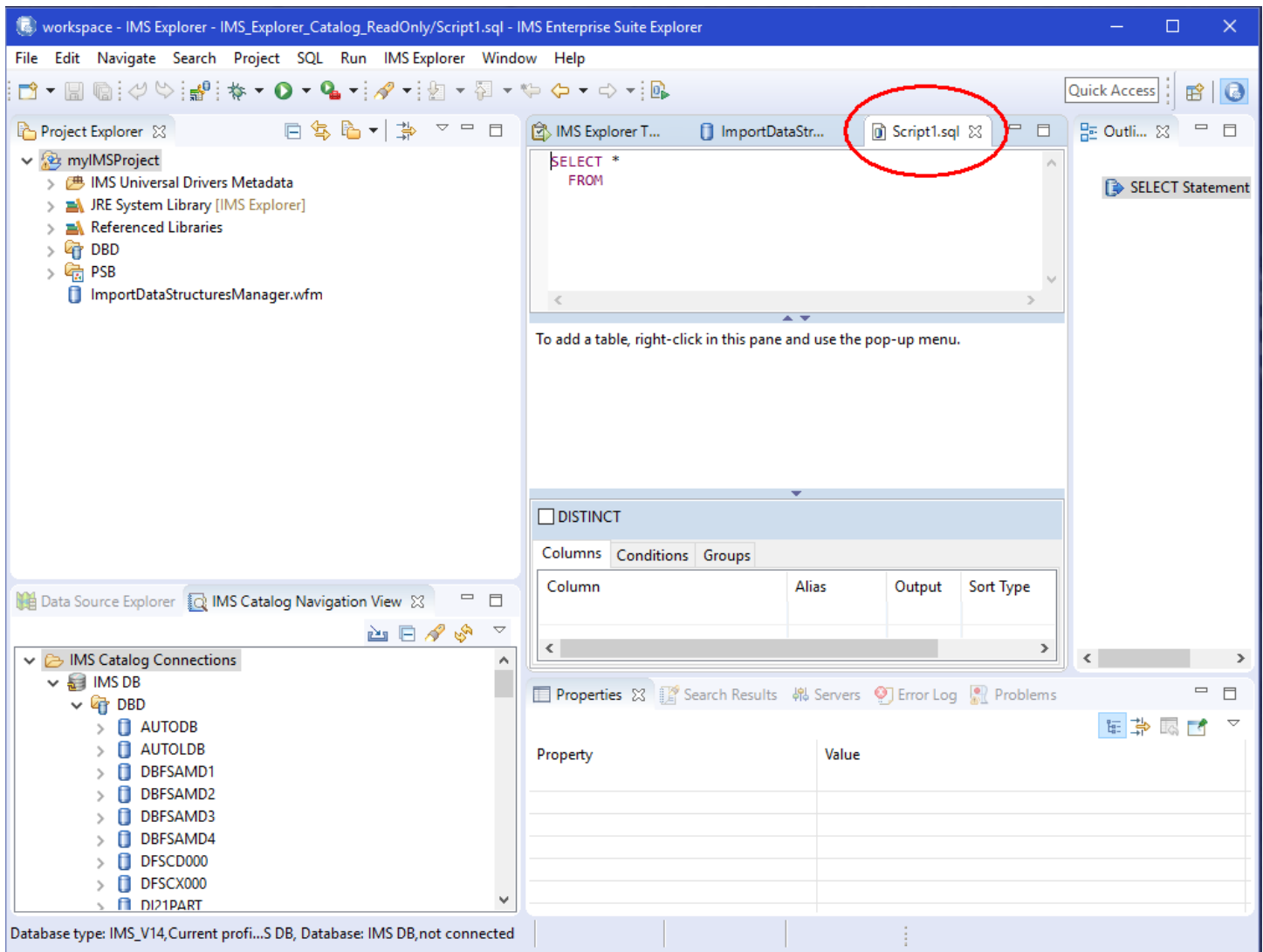


**Tech-Tip:** The IMS database provider only supports SELECT statements at this point.

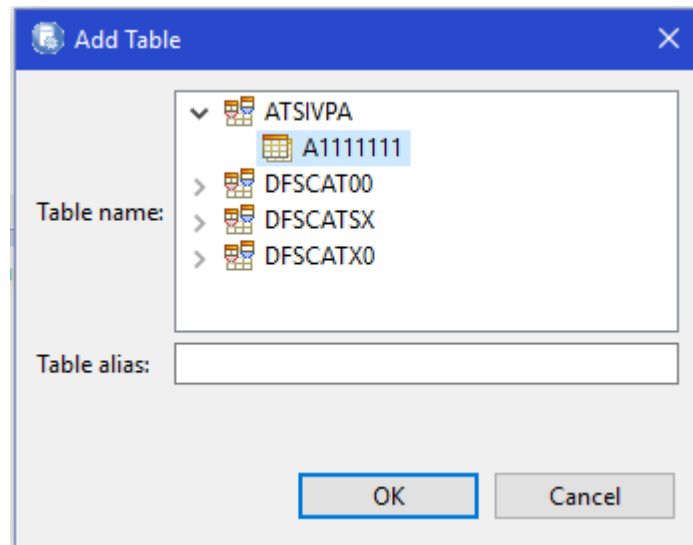
18. On the *Select Connection Profile* window select *IMS DB* and click **Finish** to continue.



19. This will open a new tab entitled *Script1.sql*. Double click on this tab's title so this tab will become the focus of the workbench.



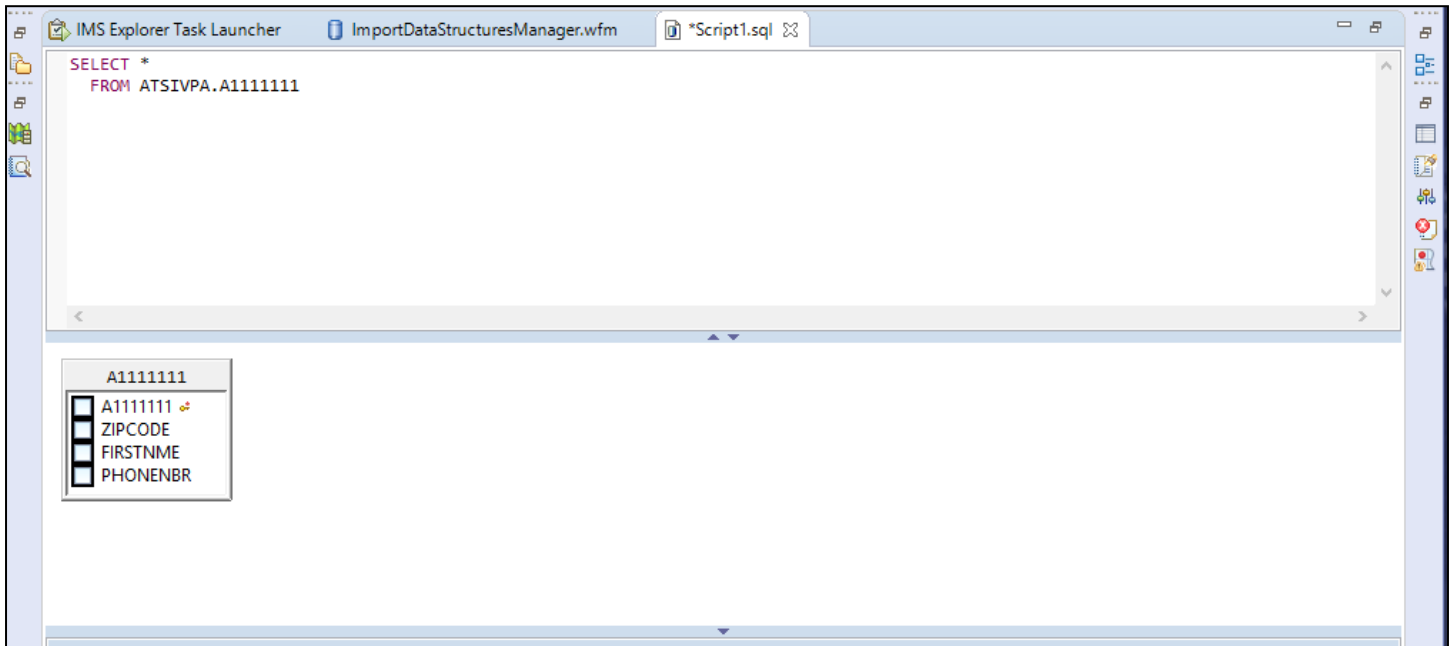
20. Right mouse button click in the pane with the text *To add a table, right-click in this pane and use the pop-up menu*. Select the *Add Table* option. In the *Add Table* window, expand *ATSIVPA* and select table *A1111111*. Click **OK** to continue.



**Tech-Tip:** The values are derived from the IMS Catalog configuration perform earlier. This will form the table name used in any SQL SELECT statement. The full table name consists of a schema name (which is the PCB label (ATSIVPA) in the PSB DFSIVPA) and the table name (which the segment name in the DBD IVPDB1).

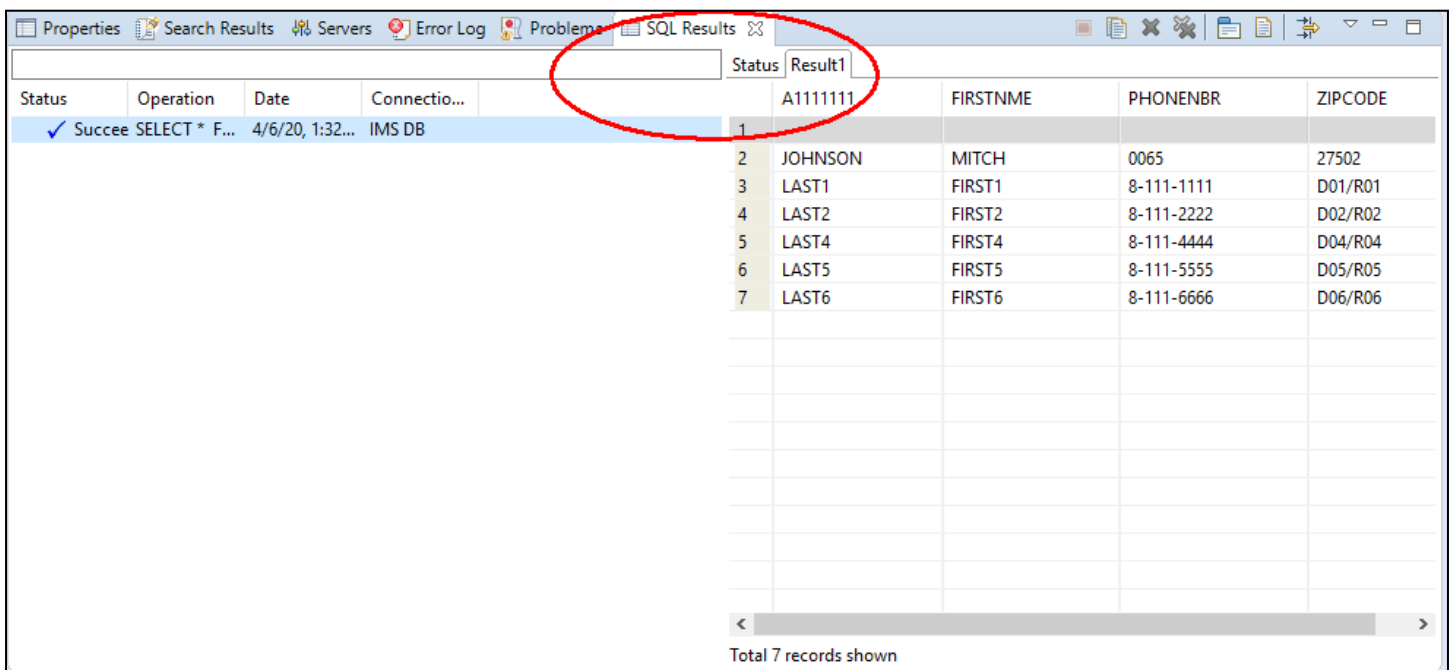
In this example the segment and the key field have the same name.

21. This populates the upper pane with a simple select statement *SELECT \* FROM ATSIVPA.A1111111*



22. The SQL command can be executed by right mouse button clicking in the upper pane and then selecting the *Run SQL* option.

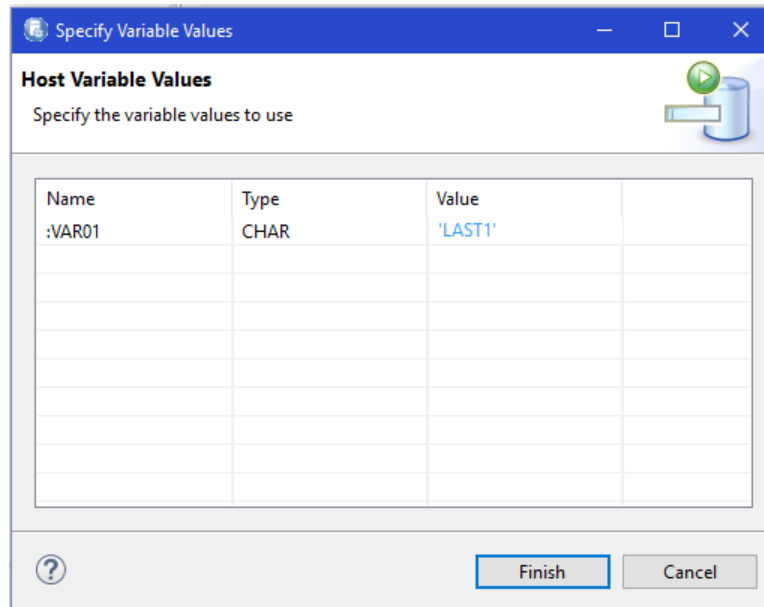
23. The results of the query can be displayed by double clicking the SQL tab title, e.g. Script1.sql and viewing the *SQL Results* pane in the lower right-hand corner with stacked tabs *Status* and *Result1*.



In the SQL script pane change the *SELECT \* FROM ATSIVPA.A1111111* to

***SELECT \* FROM ATSIVPA.A1111111 WHERE A1111111 = ?***

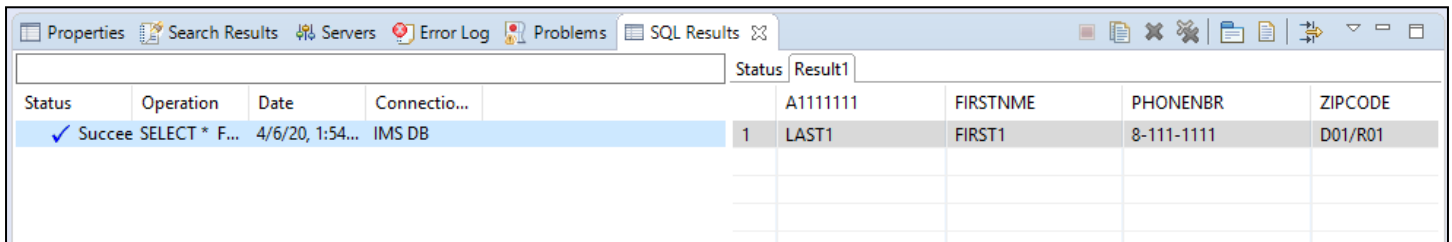
24. Run the SQL again but time a pop-window will appear asking for the value of the host variable represented by the question mark. Use the pull-down in the *Value* column and enter **LAST1** as the value and then click **Finish** to continue.



The dialog box titled "Specify Variable Values" has a subtitle "Host Variable Values" and a description "Specify the variable values to use". It contains a table with three columns: Name, Type, and Value. The first row shows Name as ":VAR01", Type as "CHAR", and Value as "'LAST1'". There are several empty rows below. At the bottom, there are "Finish" and "Cancel" buttons.

Name	Type	Value
:VAR01	CHAR	'LAST1'

25. The results should be:



The SQL Results window shows a table with columns: Status, Operation, Date, Connection, Status, and Result1. The first row indicates a successful query: "✓ Success SELECT \* F..." on 4/6/20 at 1:54... connected to IMS DB. The second row shows the results of the query.

Status	Operation	Date	Connection	Status	Result1
✓ Success	SELECT * F...	4/6/20, 1:54...	IMS DB	A1111111	FIRSTNAME
1	LAST1	FIRST1	8-111-1111	ZIPCODE	D01/R01

26. Repeat invoking SQL commands using these SELECT examples:

```

SELECT * FROM ATSIKPA.A1111111 WHERE ZIPCODE IN ('D04/R04', 'D05/R05')
SELECT * FROM ATSIKPA.A1111111 WHERE PHONENBR < '8-111-3333'
SELECT * FROM ATSIKPA.A1111111 WHERE PHONENBR > '8-111-3333'
SELECT * FROM ATSIKPA.A1111111 WHERE PHONENBR = '0065'

```

This completes the use of the IMS Explorer.



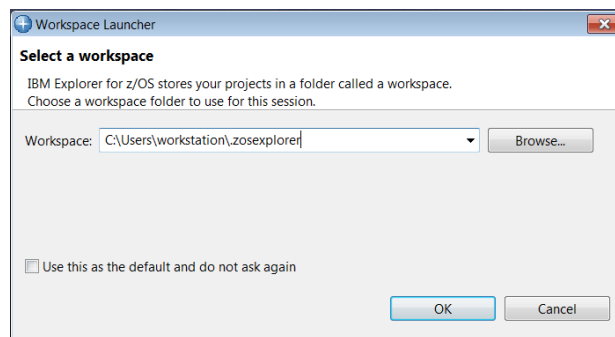
## Connect to a z/OS Connect EE Server

Next establish connections to your z/OS Connect and IMS Connect servers from IBM z/OS Explorer. If you have performed one of the other exercises in this series of exercises connecting to the z/OS Connect server many not be required.

**Tech-Tip:** Desktop tools can be opened either by double clicking the icon or by selecting the icon and right mouse button clicking and then selecting the *Open* option.

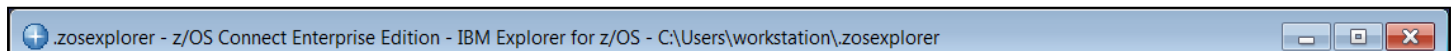
1. On the workstation desktop, locate the *z/OS Explorer* icon and double click on it to open the IBM Explorer.

2. You will be prompted for a workspace:



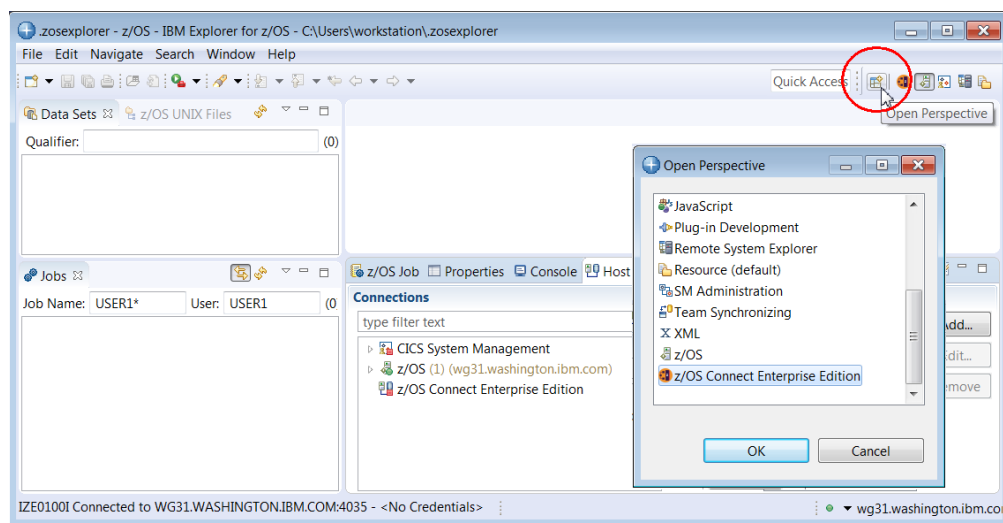
Take the default value by clicking **OK**.

3. The Explorer should open in the *z/OS Connect Enterprise Edition* perspective. Verify this by looking in the upper left corner. You should see:

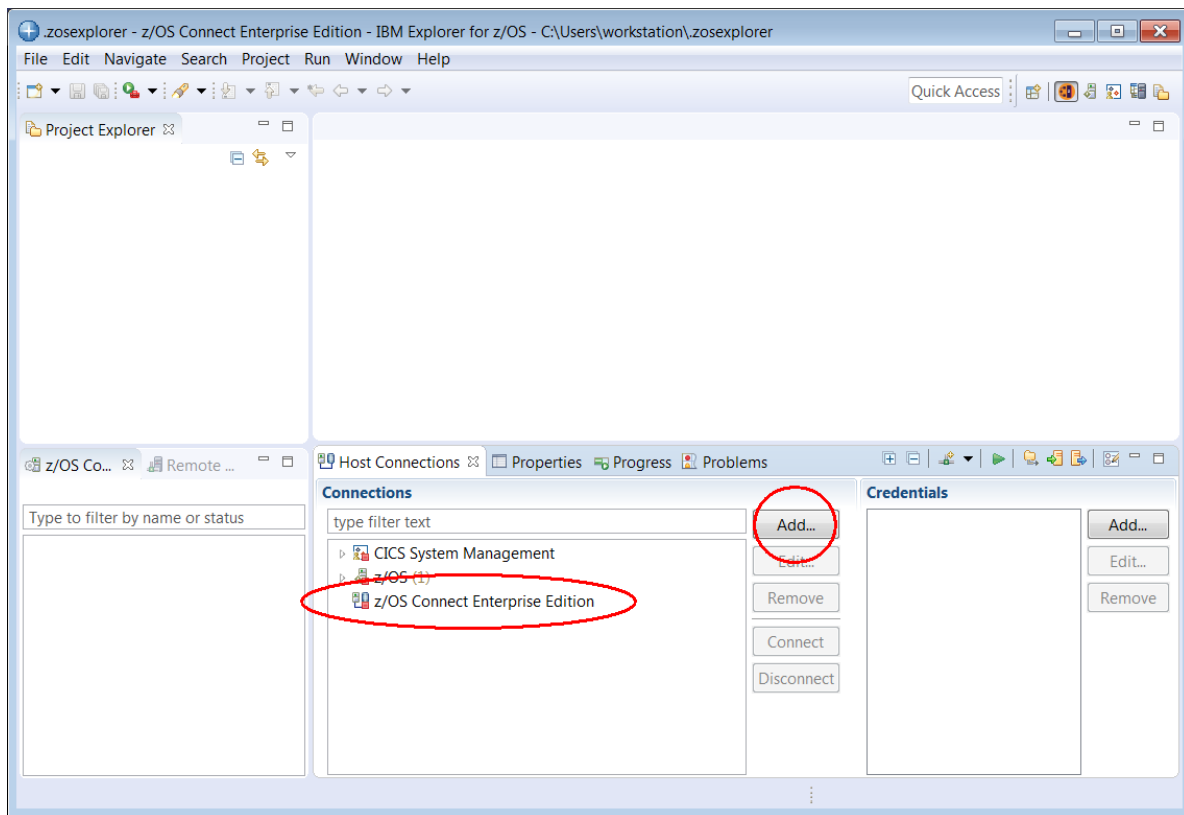


N.B. If a *Welcome* screen is displayed then click the white X beside *Welcome* to close this view.

4. If the current perspective is not *z/OS Connect Enterprise Edition*, select the *Open Perspective* icon on the top right side to display the list of available perspectives, see below. Select **z/OS Connect Enterprise Edition** and click the **OK** button to switch to this perspective.



5. To add a connection to the z/OS Connect Server select *z/OS Connect Enterprise Edition* connection in the *Host connections* tab in the lower view and then click the **Add** button.



**Tech-Tip:** Eclipse based development tools like z/OS Explorer; provide a graphical interface consisting of multiple views within a single window.

A view is an area in the window dedicated to providing a specific tool or function. For example, in the window above, *Host Connections* and *Project Explorer* are views that use different areas of the window for displaying information. At bottom on the right there is a single area for displaying the contents of four views stacked together (commonly called a *stacked views*), *z/OS Host Connections*, *Properties*, *Progress* and *Problems*. In a stacked view, the contents of each view can be displayed by clicking on the view tab (the name of the view).

At any time, a specific view can be enlarged to fill the entire window by double clicking in the view's title bar. Double clicking in the view's title bar will be restored the original arrangement. If a z/OS Explorer view is closed or otherwise disappears, the original arrangement can be restored by selecting Windows → Reset Perspective in the window's tool bar.

Eclipse based tools also can display multiple views based on the current role of the user. In this context, a window is known as a perspective. The contents (or views) of a perspective are based on the role the user, i.e., developer or administrator.

6. In the pop-up list displayed select *z/OS Connect Enterprise Edition* and on the *Add z/OS Connect Enterprise Edition Connection* screen enter **wg31.washington.ibm.com** for the *Host name*, **9453** for the *Port Number*, check the box for *Secure connection (TLS/SSL)* and then click the **Save and Connect** button.

7. On the *z/OS Connect Enterprise Edition – User ID* required screen create new credentials for a *User ID* of **Fred** and a *Password or Passphrase* of **fredpwd** (case matters). Remember the server is configured to use basic security. If SAF security had been enabled, then a valid RACF User ID and password will have to be used instead. Click **OK** to continue.
8. Click the **Accept** button on the *Server certificate alert – Accept this certificate* screen. You may be presented with another prompt for a userid and password, enter **Fred** and **fredpwd** again.

9. The status icon beside **wg31:9453** should now be a green circle with a lock. This shows that a secure connection has been established between the z/OS Explorer and the z/OS Connect server. A red box indicates that no connection exists.
10. Next add a connection to the IMS Database subsystem. In the *Host connections* tab in the lower view click the **Add** button.

11. In the pop-up list displayed select *IMS Database* and on the *Add IMS Database Connection* screen enter **wg31.washington.ibm.com** for the *Host name*, 5555 for the *Port Number*, be sure the box for *Secure connection (TLS/SSL)* is unchecked and then click the **Save and Connect** button.

12. On the *IMS Database* required screen create new credentials for a *User ID* of **USER1** and then USER1's password for *Password or Passphrase*. Remember IMS uses RACF security so a valid RACF User ID and password is required. Click **OK** to continue.
13. A connection to the remote z/OS system was previously added. In the *Host Connection* view expand *z/OS Remote System* under *z/OS* and select *wg31.washington.ibm.com*. If the connection is not active the **Connect** button will be enabled. Click the **Connect** button and this will establish a session to the z/OS system. .

## Summary

The next step is the creation of the service and the composing and deployment of the API and then the testing of the API functions.

## ***z/OS Connect EE APIs and IMS***

This section of the exercise provides an opportunity to compose and deploy an API that accesses IMS.

Two z/OS Connect services will be created with each corresponding to two of the file SQL commands explored in the IMS Explorer earlier in this exercise. Adding z/OS Connect services for the other SQL commands tested earlier will be left as an optional activity.

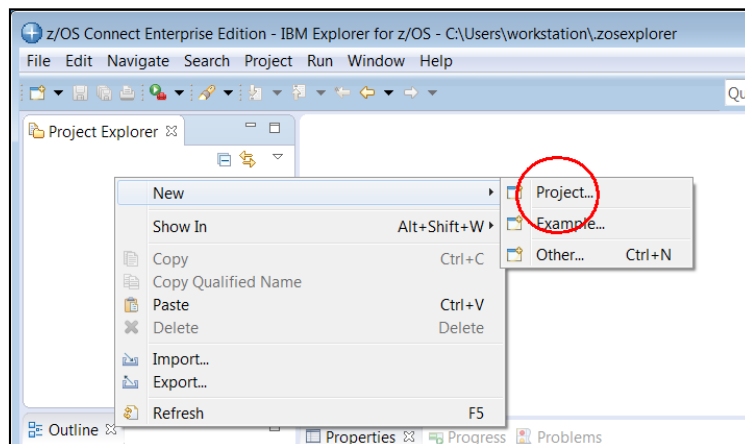
These 2 services will then be integrated into a RESTful API.

### ***Create the IMS Service Projects***

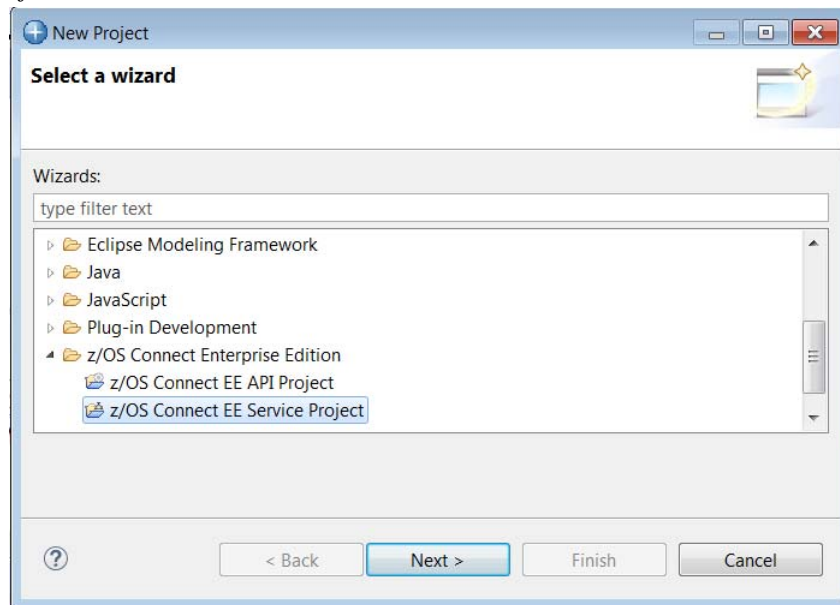
The first step is to create the 2 z/OS Connect services which provide the interaction with 2 of the SQL commands tested earlier. One service will select a phone book entry based on last name and the other will select phone book entries based on phone number.

Switch to the *z/OS Connect Enterprise Edition* perspective. Start by creating the *selectByName* service.

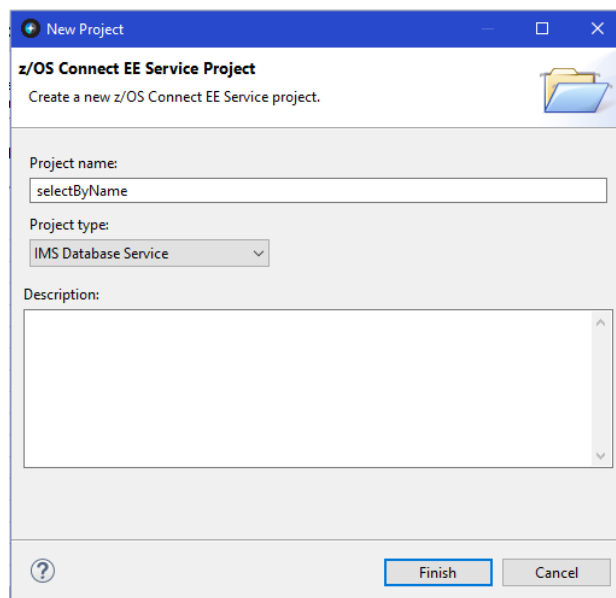
1. Select **File** on the tool bar and then on the pop up select **New → Project**. Expand the *General* folder and select *Project* to create a target project for exporting the Service Archive (SAR) files. Click **Next** to continue.
2. On the *New Project* window enter **Services** as the *Project name*. Click **Finish** to continue. This action will add a new project in the *Project Explorer* named *Services*. If this project already exists continue with Step 3.
3. In the upper left, position your mouse anywhere in the *Project Explorer* view and right mouse click, then select **New → Project**:



4. In the *New Project* window, scroll down and open the *z/OS Connect Enterprise Edition* folder and select *z/OS Connect EE Service Project* and then click the **Next** button.



5. On the new *New Project* window enter *selectByName* Project name and use the pull-down arrow to select *IMS Database Service* as the *Project type*. Click **Finish** to continue



6. This will open the *Service Project Editor:Definition* window for the *selectByName* service. For now, disregard the message about the 4 errors detected, they will be addressed shortly.

**selectByName Service** 4 errors detected

**General Information**  
 Edit or update the general information of the service.  
 Type: IMS Database Service  
 Version: 1.0.0  
 Description:

**Actions**  
 Steps to create a service:  
 1. Input service version.  
 2. Specify the SQL command for the service.  
 3. Specify Database Connection Properties and Generate Service Interface.  
 4. [Complete the configuration for the service.](#)  
 5. [Deploy the service.](#)  
 6. [Export the service.](#)

**Enter or import SQL Command**  
 SQL Command:  Import...

**Specify Database Connection Properties and Generate Service Interface**  
 Database Connection:  Database Name:  Generate Service Interface...

Definition Configuration

**Tech-Tip:** If this view is closed it can be reopened by double clicking the *service.properties* file in the service project.

7. Next use the pull-down arrow in the *Database Connection* area to select *wg31:5555*. Enter the PSB name *DFSIVPA* in the *Database Name* area. Finally enter this SQL command in the SQL commands area.

```
SELECT FIRSTNME, ZIPCODE, PHONENBR, A1111111  
FROM ATSIIPA.A1111111 WHERE A1111111=?
```

**Service Project Editor: Definition**

**General Information**  
Edit or update the general information of the service.  
Type: IMS Database Service  
Version: 1.0.0  
Description:

**Enter or import SQL Command**  
SQL Command: SELECT FIRSTNME, ZIPCODE, PHONENBR, A1111111  
FROM ATSIIPA.A1111111 WHERE A1111111=?

**Specify Database Connection Properties and Generate Service Interface**  
Database Connection: wg31:5555 Database Name: DFSIVPA Generate Service Interface...

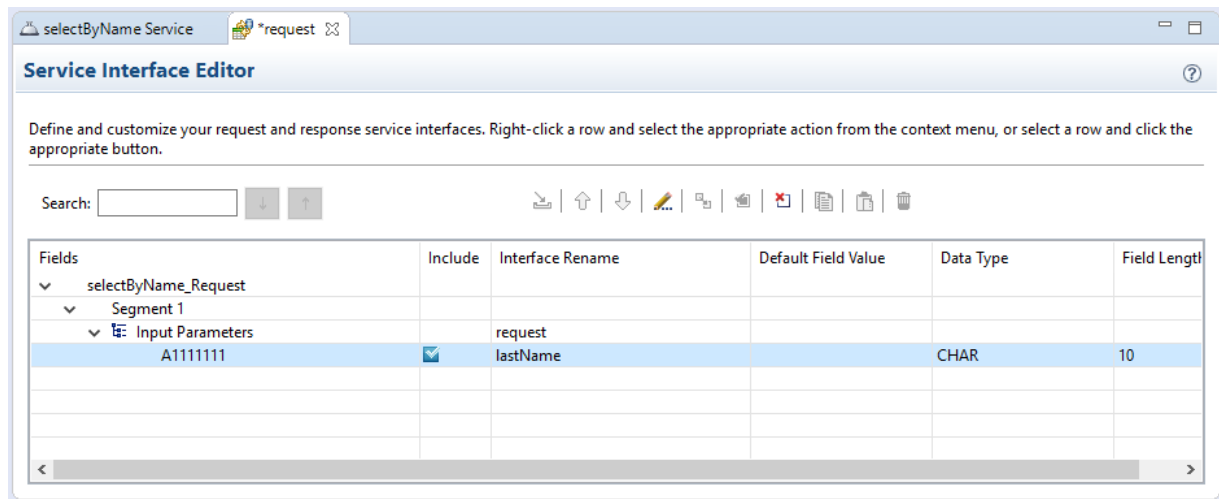
**Actions**  
Steps to create a service:  
1. Input service version.  
2. Specify the SQL command for the service.  
3. Specify Database Connection Properties and Generate Service Interface.  
4. Complete the configuration for the service.  
5. Deploy the service.  
6. Export the service.

8. Click the **Generate Service Interface** button to validate the SQL command and to generate the request and response service interface files.
9. Click on [4. Complete the configuration for the service](#) and enter the value *DFSIVPAConn* in the area beside *Connection reference*. The connection reference binds this service to the *zosConnectionRestClientConnection* configuraton element with the same name (or ID) in the *server.xml* file (see below);

```
<connectionFactory id="DFSIVPAConn">  
  <properties.imsudbjLocal  
    databaseName="DFSIVPA"  
    dataStoreName="IVP1"  
    dataStoreServer="wg31.washington.ibm.com"  
    driverType="4"  
    portNumber="5555"  
    user="USER1"  
    password="USER1"  
    flattenTables="True" />  
</connectionFactory>
```

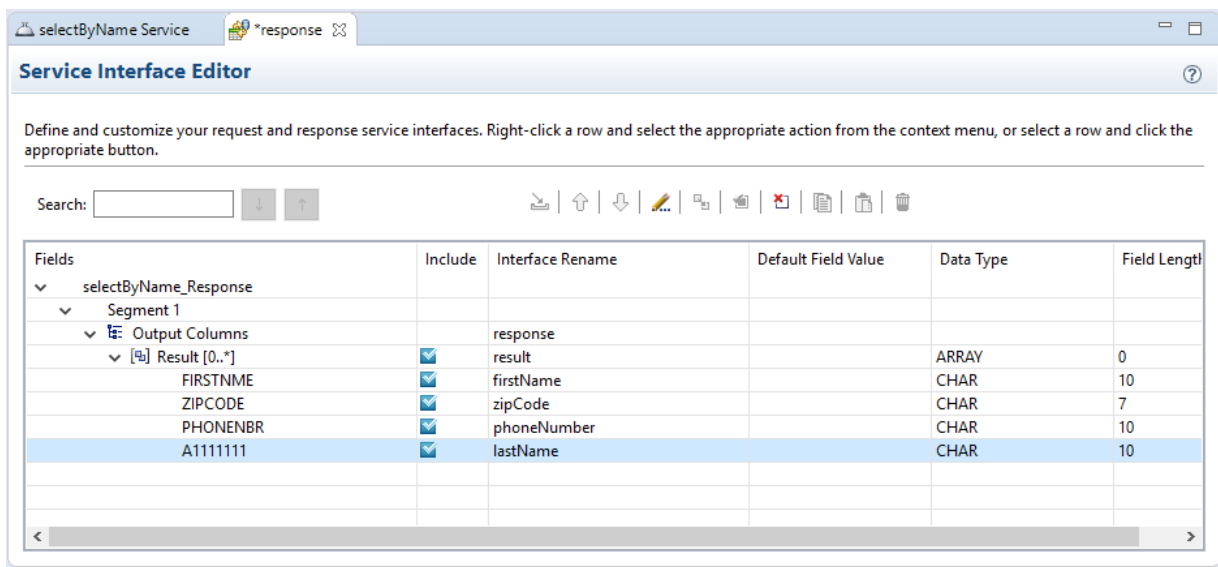


10. Switch back to the *Definition* view by clicking on the *Definition* tab at the bottom of the view.
11. Next expand the *selectByName* project in the *Project Explorer* view and then expand the *service-interfaces* folder. This folder contains the request and response service interfaces created when the **Generate Service Interface** button was clicked.
12. Double click *request.si* to open the *Service Interface Editor*. Change the interface name for field *A1111111* to *lastName*.

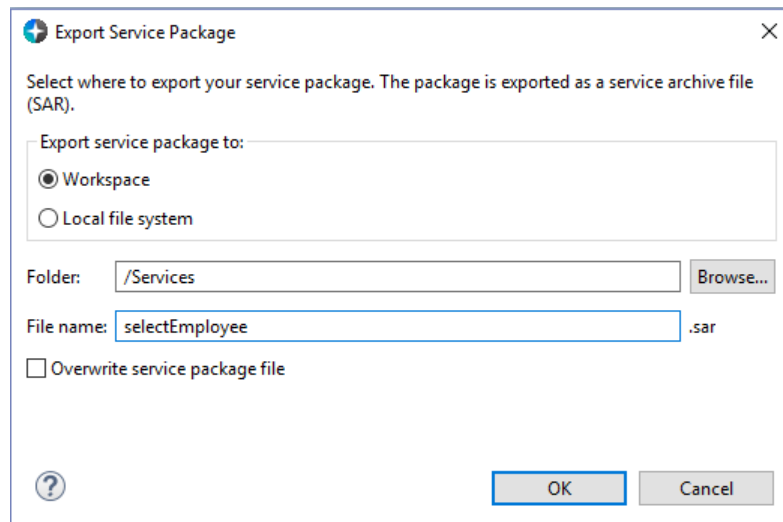


**Tech-Tip:** Field property (e.g. *Interface name*) or exclude specific fields entirely from the interface. Either can be done by selecting a field and right mouse button clicking or by selecting a field and using the desired tool icon in the Service Interface toolbar.

13. Double click *response.si* to open the *Service Interface Editor*. Change the interface name for field *FIRSTNME* to *firstName*, field *ZIPCODE* to *zipCode*, field *PHONENBR* to *phoneNumber* and field *A1111111* to *lastName*.



14. Save the changes made so far by using the key sequence **Ctrl-S** and close any open Service Interface editor views.
15. Next click on **5. Deploy the service** on right hand side of the view under *Actions*. This will open a *Deploy Service* window. On the *Deploy Service* window select the target server (wg31:9453) and click **OK** twice to have the service installed in the server.
16. Next click on **6. Export the service** on right hand side of the view under *Actions*. This will open a *Export Service* window. On the *Export Service* window select the radio button beside *Workspace* and use the **Browse** button to select the *Services* folder. Click **OK** to continue

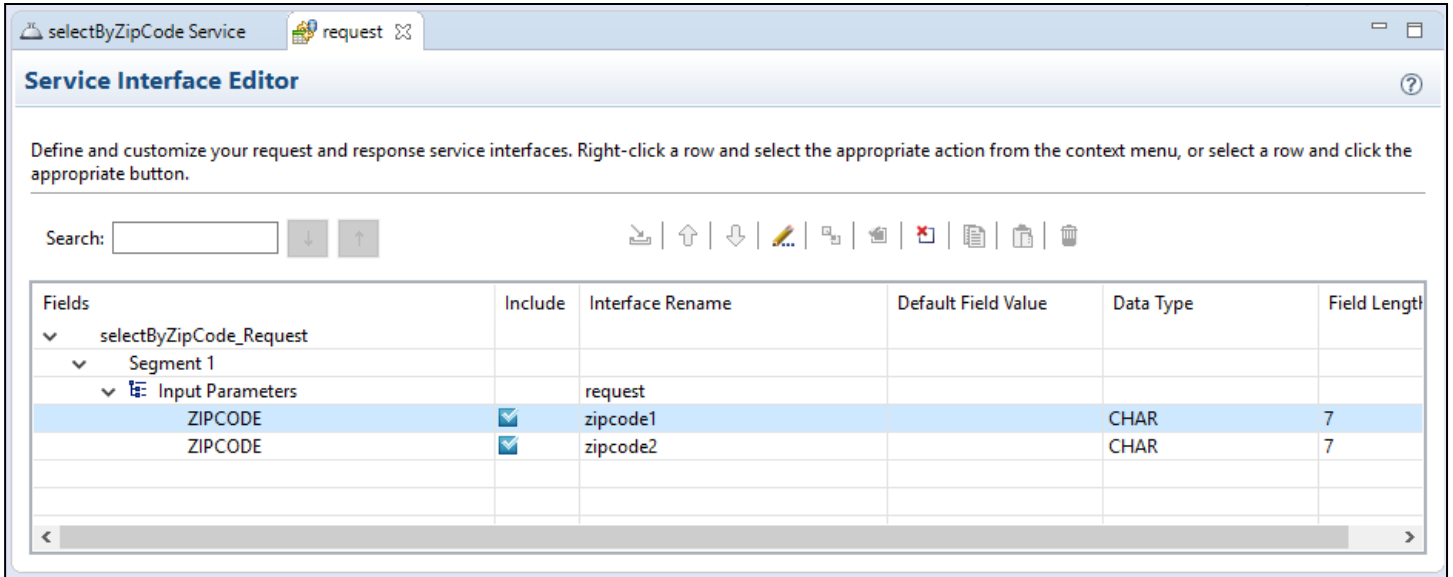


**Tech-Tip:** To re-access the *Service Project Editor* just double click on the *service.properties* file in the Project Explorer view.

17. Repeat steps 3 through 16 to create a service for z/OS Connect service *selectByZipCode* where the SQL SELECT command is



```
SELECT FIRSTNME, ZIPCODE, PHONENBR, A1111111 FROM ATSVPA.A1111111  
WHERE ZIPCODE IN (?,?)
```









In this case use the *Service Interface Editor* to rename the interface fields in the request message to the names shown below.



**Service Interface Editor**

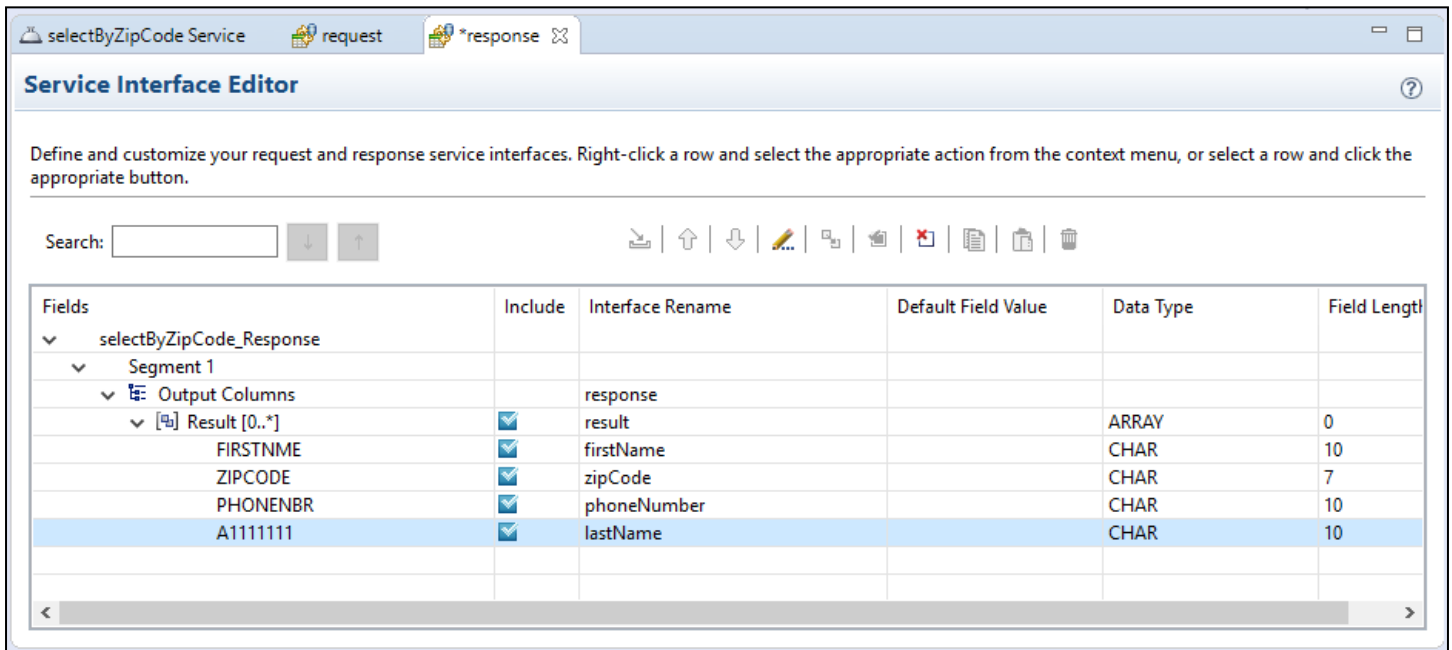
Define and customize your request and response service interfaces. Right-click a row and select the appropriate action from the context menu, or select a row and click the appropriate button.

Search:   



Fields	Include	Interface Rename	Default Field Value	Data Type	Field Length
selectByZipCode_Request					
Segment 1					
Input Parameters		request			
ZIPCODE	<input checked="" type="checkbox"/>	zipcode1		CHAR	7
ZIPCODE	<input checked="" type="checkbox"/>	zipcode2		CHAR	7









Use the *Service Interface Editor* to rename the interface fields in the response message to the names shown below.



**Service Interface Editor**

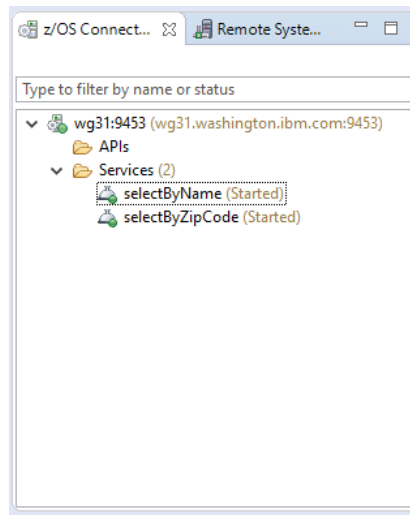
Define and customize your request and response service interfaces. Right-click a row and select the appropriate action from the context menu, or select a row and click the appropriate button.

Search:   

Fields	Include	Interface Rename	Default Field Value	Data Type	Field Length
selectByZipCode_Response					
Segment 1					
Output Columns		response			
Result [0..*]	<input checked="" type="checkbox"/>	result		ARRAY	0
FIRSTNME	<input checked="" type="checkbox"/>	firstName		CHAR	10
ZIPCODE	<input checked="" type="checkbox"/>	zipCode		CHAR	7
PHONENBR	<input checked="" type="checkbox"/>	phoneNumber		CHAR	10
A1111111	<input checked="" type="checkbox"/>	lastName		CHAR	10

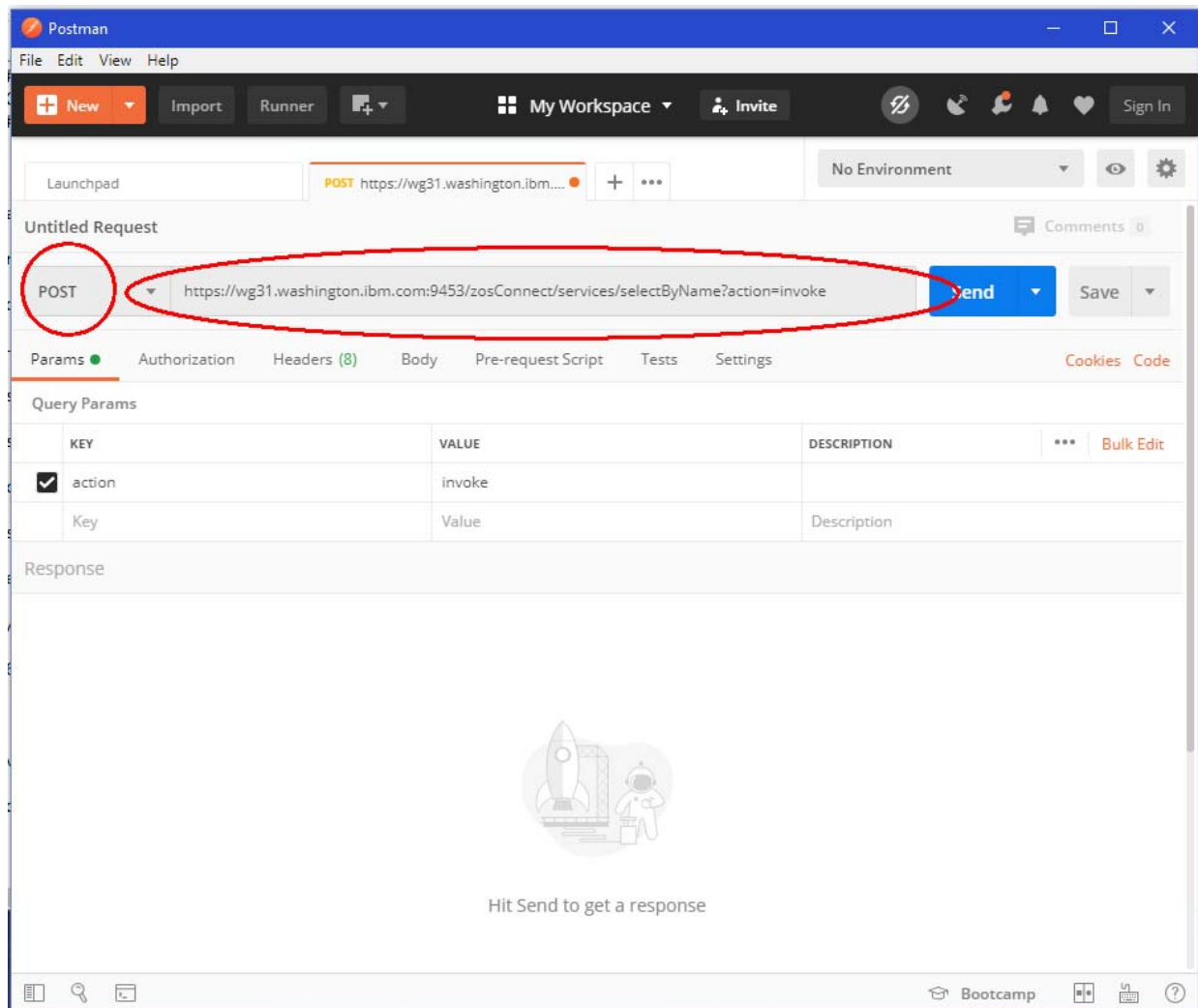
18. When finished the target server (*wg31:9453*) should have these 2 services installed and started.



## Test the IMS Database services with Postman

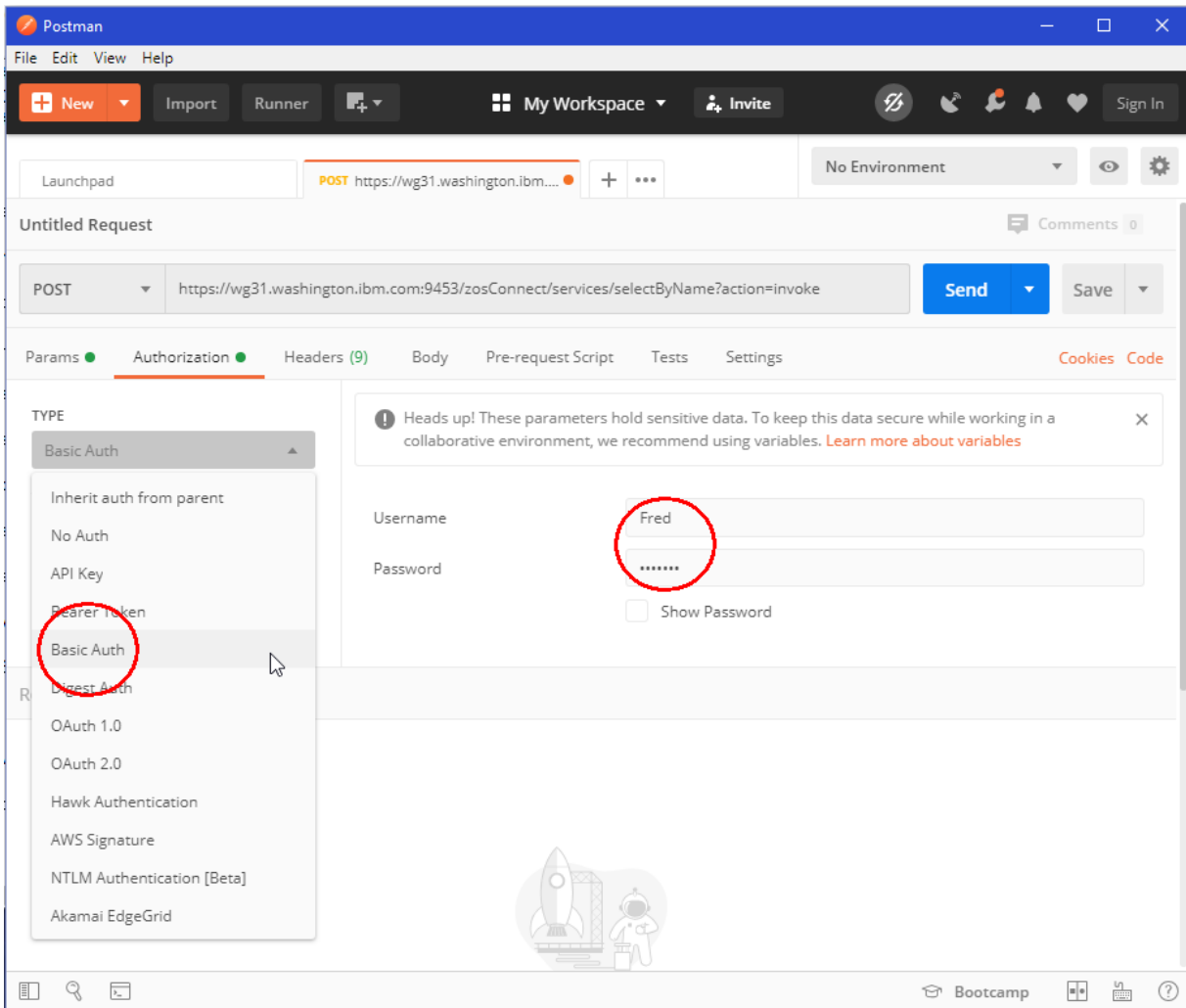
1. Open the *Postman* tool icon on the desktop and if necessary reply to any prompts and close any welcome messages, use the down arrow to select **POST** and enter the URL below in the URL area (see below).

**<https://wg31.washington.ibm.com:9453/zosConnect/services/selectByName?action=invoke>**

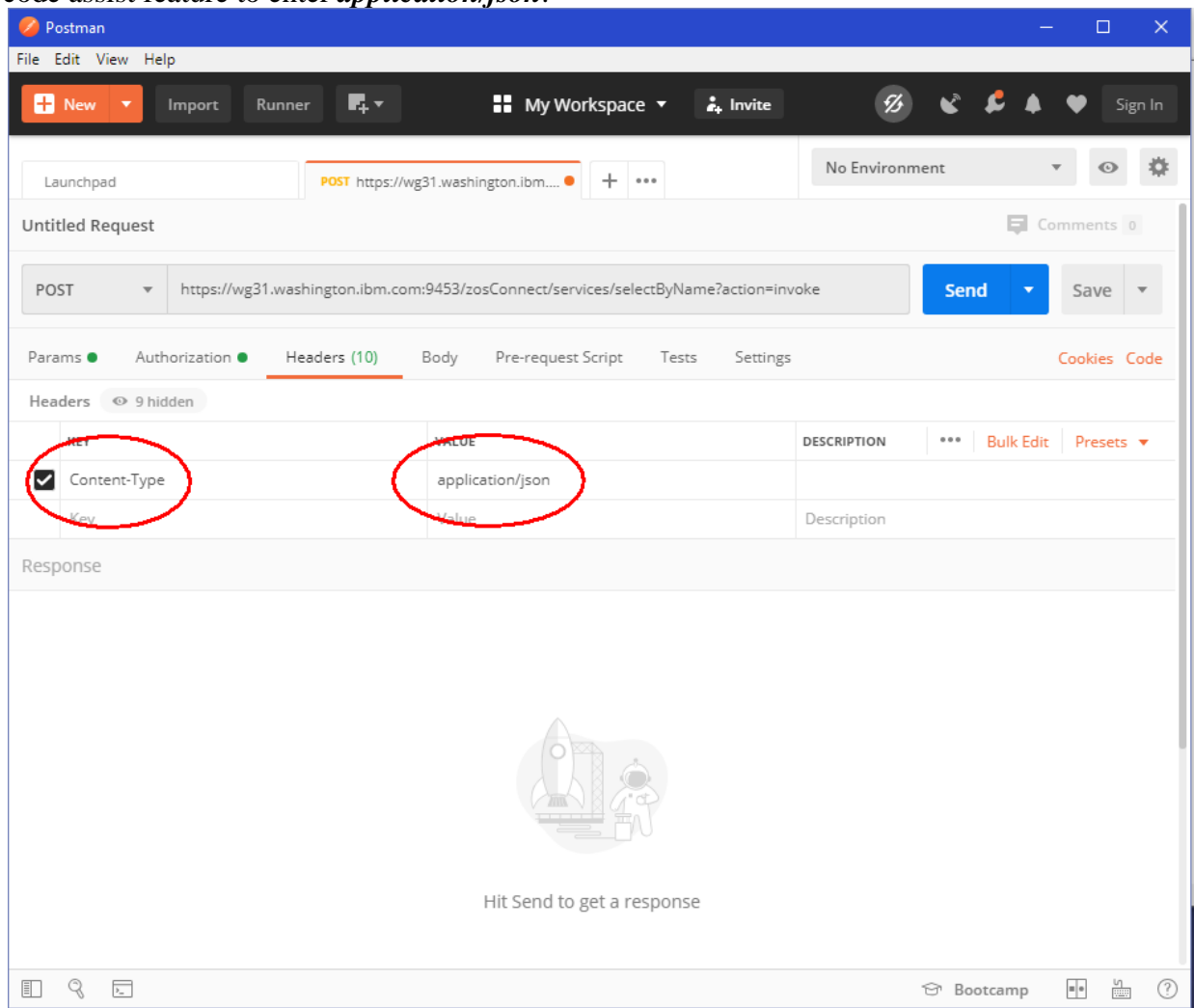


**Tech-Tip:** If the above Postman view is not displayed select *File* on the toolbar and then choose *New Tab* on the pull down. Alternatively, if the *Launchpad* view is displayed, click on the *Create a request* option.

2. Next select the *Authorization* tab to enter an authorization identity and password. Use the pull down arrow to select *Basic Auth* and enter **Fred** as the username and **fredpwd** as the Password.



3. Next select the *Headers* tab and under *KEY* use the code assist feature to enter *Content-Type* and under *VALUE* use the code assist feature to enter *application/json*.



**Tech-Tip:** Code assist simply means that when text is entered in field, all the valid values for that field that match the typed text will be displayed. You can select the desired value for the field from the list displayed and that value will populate that field.

4. Next select the *Body* tab and select the *raw* radio button and enter the JSON message below in the *Body* area and press the **Send** button.

```
{ "request": {  
  "lastName": "LAST4"  
}}
```

5. Pressing the **Send** button invokes the service. The Status of request should be 200 OK and pressing the *Pretty* tab will display the response message in an easy to read format, see below.

The screenshot displays the REST Client application interface. At the top, a tab shows the request method as **POST** and the URL as `https://wg31.washington.ibm.com:9453/zosConnect/services/selectByName?action=invoke`. Below this, the **Body** tab is selected, and the **raw** radio button is chosen. The request body contains the JSON: 

```
{ "request": {  
  "lastName": "LAST4"  
}}
```

. The **Send** button is visible. Below the request area, the response is shown with a status of **200 OK**, a time of **389ms**, and a size of **822 B**. The **Pretty** tab is selected, displaying the response JSON in a formatted manner: 

```
{  
  "response": {  
    "result": [  
      {  
        "firstName": "FIRST4",  
        "zipCode": "D04/R04",  
        "lastName": "LAST4",  
        "phoneNumber": "8-111-4444"  
      }  
    ]  
  }  
}
```



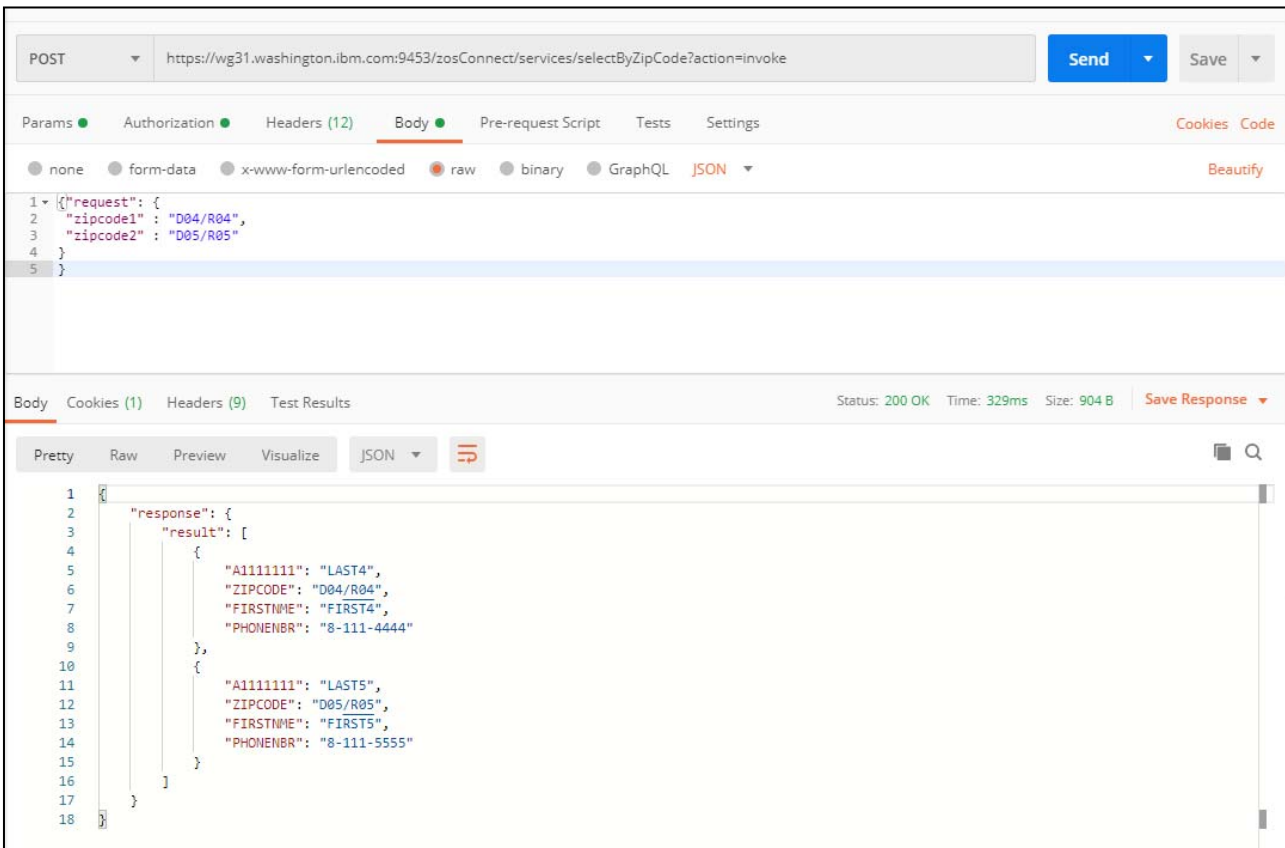
6. Replace the URL with this URL

<https://wg31.washington.ibm.com:9453/zosConnect/services/selectByZipCode?action=invoke>

7. Next replace the JSON request message in the *Body* tab with the JSON message below and press the **Send** button.

```
{
  "request": {
    "zipcode1": "D04/R04",
    "zipcode2": "D05/R05"
  }
}
```

8. Pressing the **Send** button invokes the service. The Status of request should be 200 OK and pressing the *Pretty* tab will display the response message in an easy to read format, see below

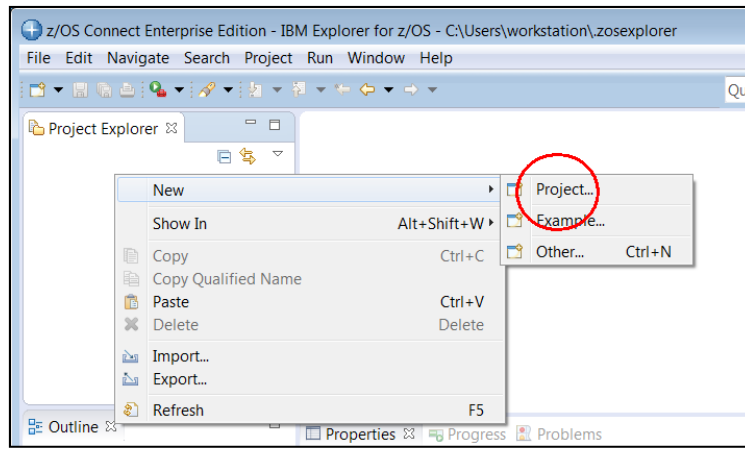


## Summary

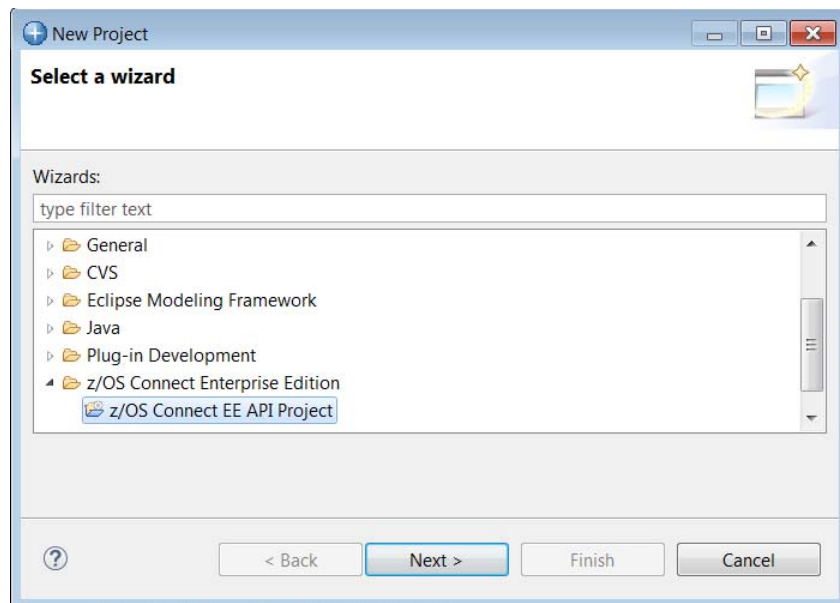
The contents of the IMS Catalog was used to develop SQL SELECT statements for accessing an IMS database. The SQL SELECT statement was used to generate the z/OS Connect artifacts for a z/OS Connect services. These services were deployed to the runtime and tested and then exported for use developing APIs.

## Create the IMS API Project

1. In the *z/OS Connect Enterprise Edition* perspective of the z/OS Explorer create a new API project by clicking the right mouse button and selecting *New* → *Project*:



2. In the *New Project* screen, scroll down and open the *z/OS Connect Enterprise Edition* folder and select *z/OS Connect EE API Project* and then click the **Next** button.



3. Enter **IMSAPI** for the *Project name*. Set the *API name* is set to **imsPhonebook** and the *Base path* is set to **/imdb**. Click **Finish** to continue.

**New Project**

**z/OS Connect EE API Project**  
Create a new z/OS Connect EE API project.

Project name:

API name:

Base path:

Description:

**Important:** The values are somewhat arbitrary, but they do relate to later tasks. If you use the values and cases as supplied, then the subsequent commands and the use of subsequent URLs will work seamlessly.

4. You should now see something like the view below. The view may need to be adjusted by dragging the view boundary lines.

**API Editor**

**Describe your API**

Name:  Description:

Base path:

Version:

**Path**

**Methods (4)**

Method	Service...	Mapping...	Icons
POST	<input type="text"/>	<input type="text"/>	+ - X
GET	<input type="text"/>	<input type="text"/>	+ - X
PUT	<input type="text"/>	<input type="text"/>	+ - X
DELETE	<input type="text"/>	<input type="text"/>	+ - X

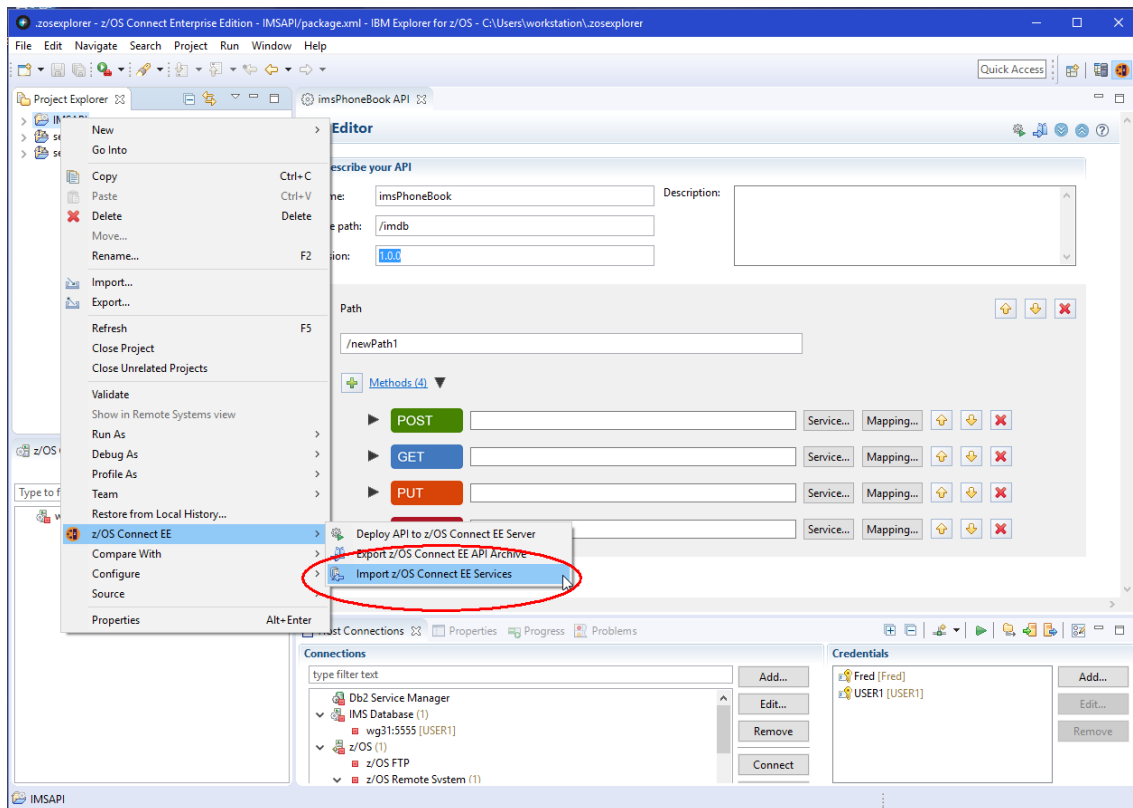
**Tech-Tip:** The *API Editor* view above can be reopened if closed by double clicking on *package.xml*.

## Summary

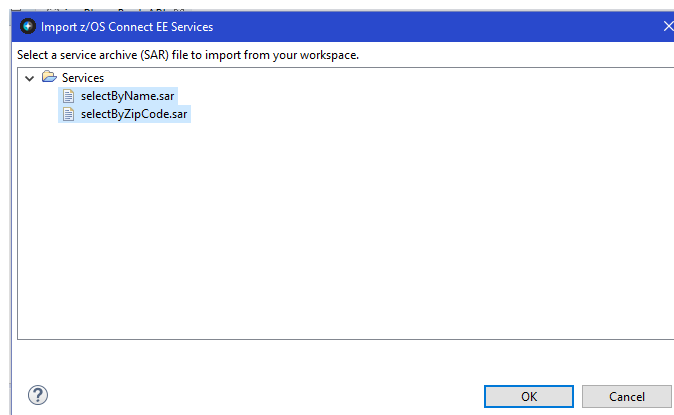
This created the basic framework for the API project in the API editor

### Import the SAR files

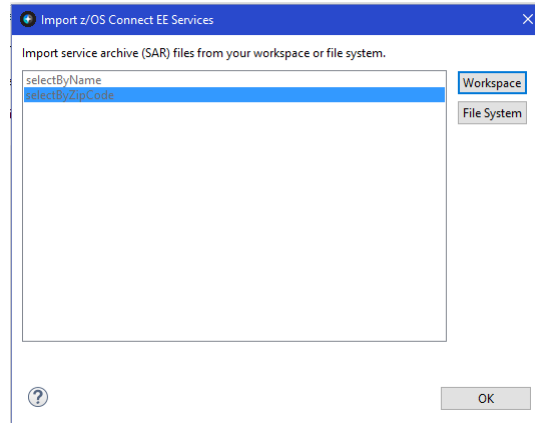
1. In the z/OS Explorer in the *z/OS Connect Enterprise Edition* perspective in the *Project Explorer* view (upper left), right-click on the *IMSAPI* project, then select *z/OS Connect EE* and then *Import z/OS Connect EE Services* (see below):



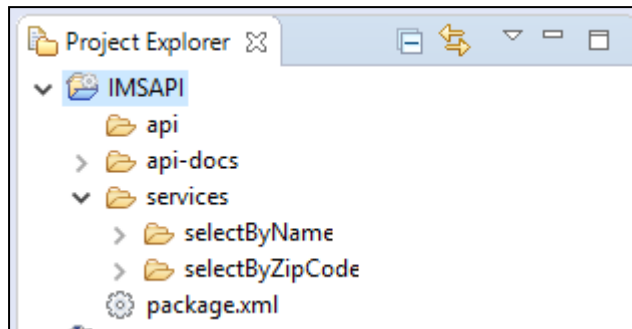
2. In the *Import z/OS Connect EE Services* window click on the **Workspace** button and expand the *Services* folder. Select the 2 SAR files and click on the **OK** button twice.



3. The two service archive files should appear in the *Import z/OS Connect EE Services* screen. Click the **OK** button to import them into the workspace.



4. In the *Project Explorer* view (upper left), expand the *services* folder to see the the imported service:



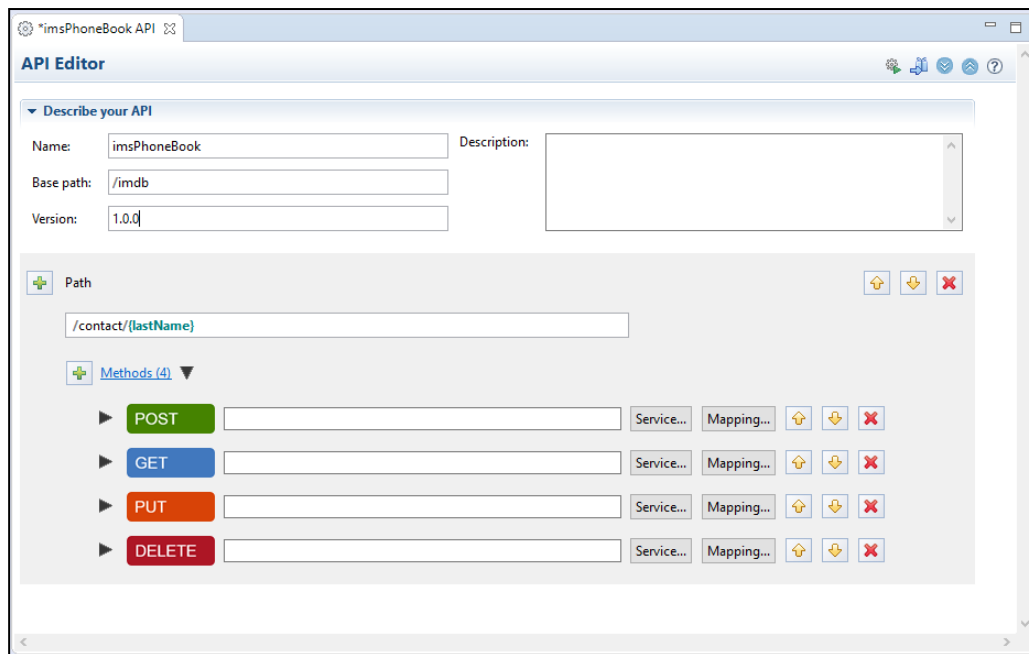
**Tech-Tip:** To re-access the *API Editor* just double click on the *package.xml* file in the Project Explorer view.

## Summary

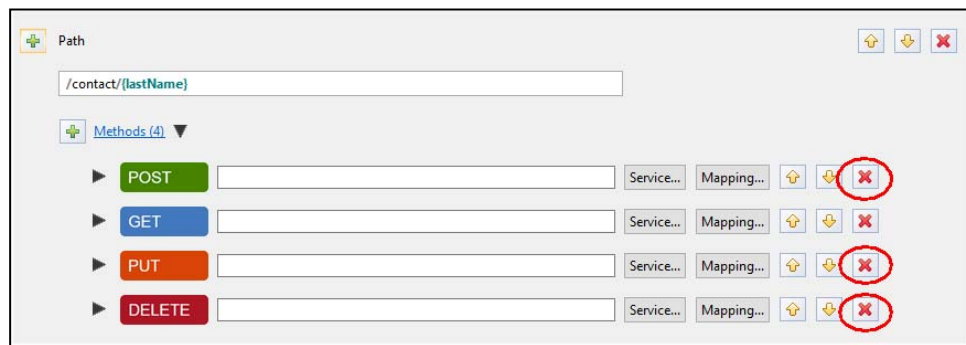
The SAR file created earlier have been imported into the API editor. That provides the editor with information about the underlying services and the JSON schemas.

## Compose an API for IMS Database Services

- \_\_\_1. Start by entering a *Path* of **/contact/{lastName}** in the z/OS Connect EE API Editor view as shown below:



- \_\_\_2. For the *IMS API* when the *contact* path parameter is present the supported HTTP methods will be the **GET** method, Remove the **DELETE**, **POST** and **PUT** methods by clicking the X icon to the right of each method.



**Note:** The */contact* path element again is somewhat arbitrary, but is used to distinguish this request from other requests that may be configured in the same API.

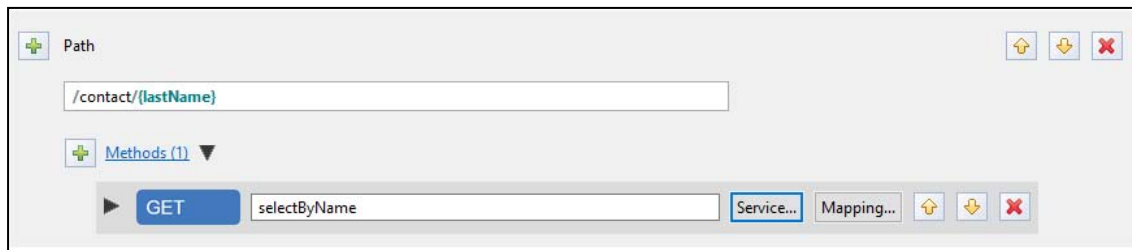
The *{lastName}* element is a path parameter in the URL that will be used to provide the key of the record for get REST requests.

The full URL to invoke the methods for this particular path will be

<https://hostname:port/imddb/contact/#####>

where ##### is the last name of the record in the phone book data base.

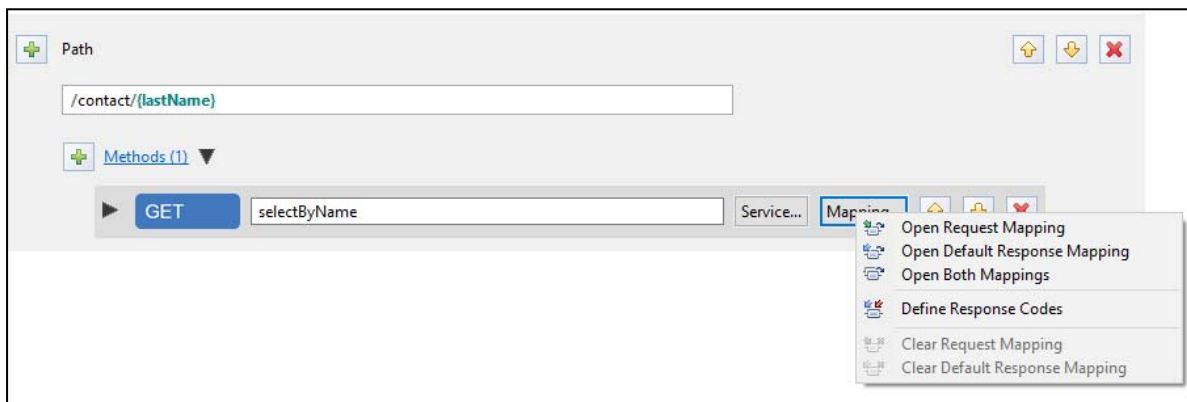
- \_\_\_3. Click on the **Service** button to the right of the **GET** method. Then select the *selectByName* service from the list of service archive files and click **OK**. This will populate the field to the right of the method.



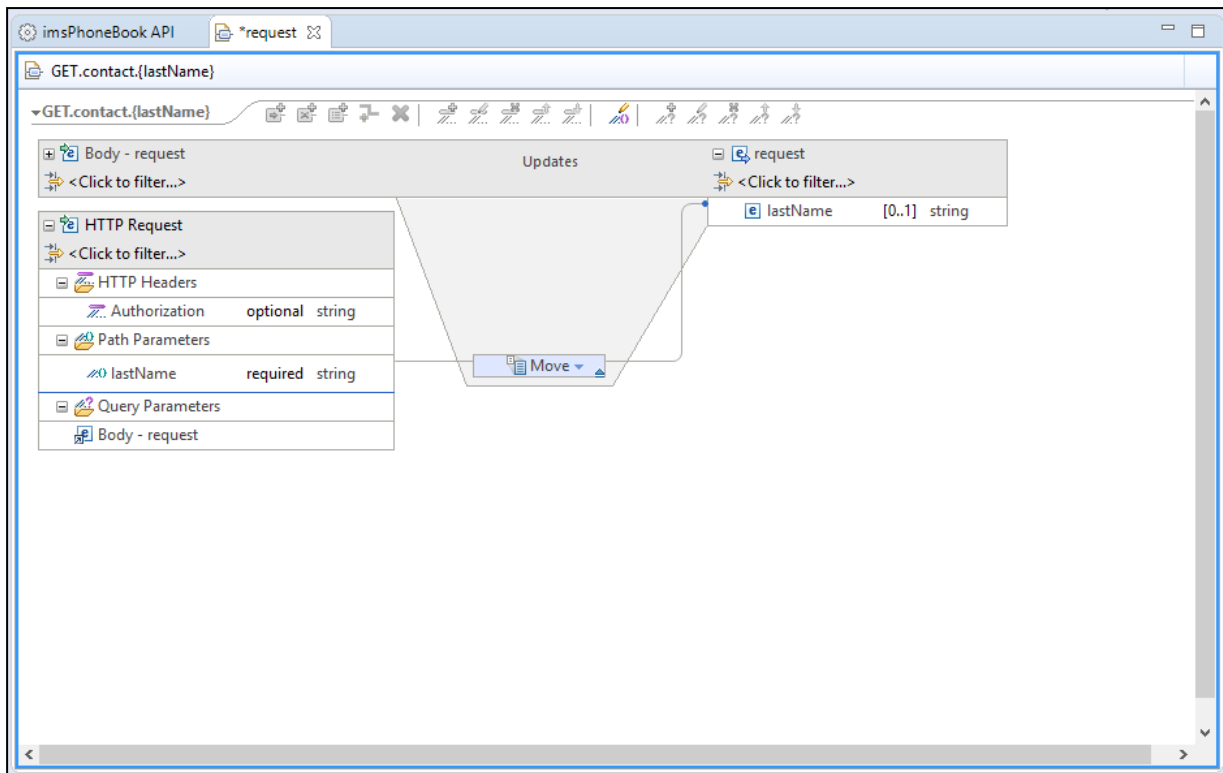
- \_\_\_4. Save the changes made so far by using the key sequence **Ctrl-S**.

**Tech-Tip:** If any change is made in any edit view an asterisk (\*) will appear before the name of the artifact in the view tab, e.g. *\*package.xml*. Changes can be saved at any time by using the **Ctrl-S** key sequence.

- \_\_\_5. Next, click on the **Mapping** button beside the **GET** method and then select *Open Request Mapping*:



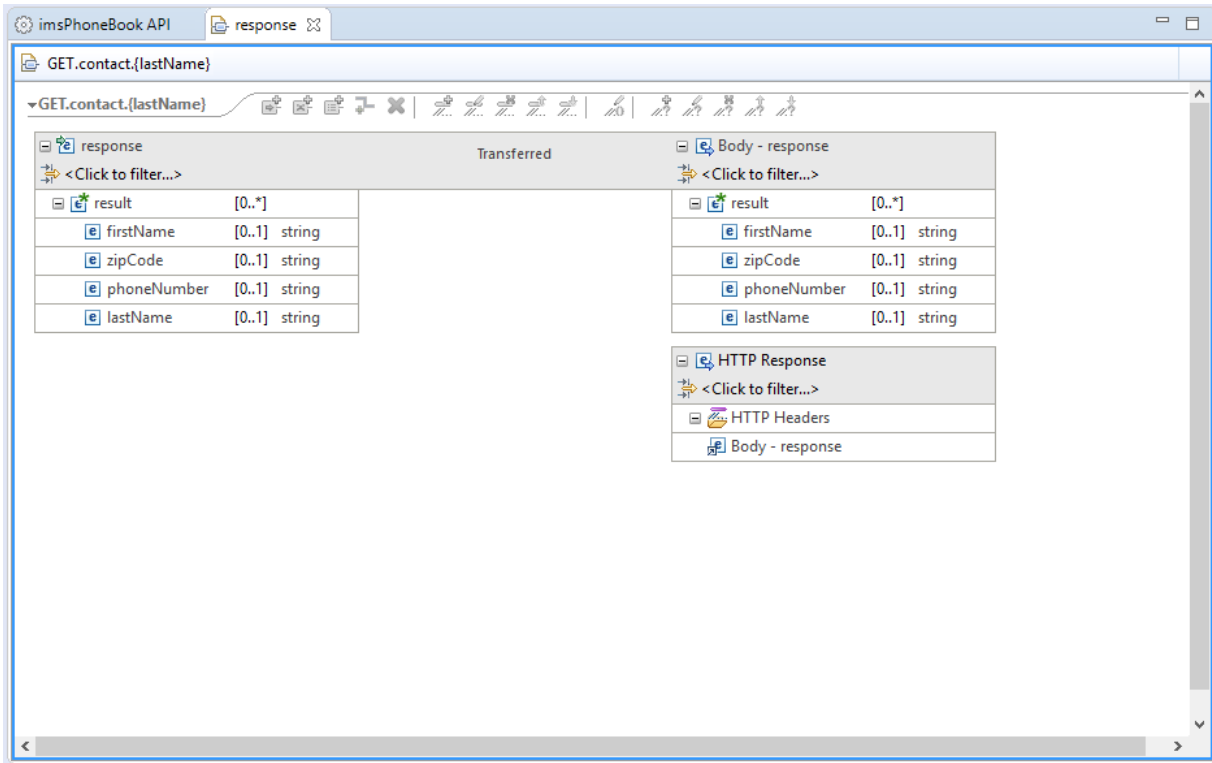
- \_\_\_6. In the mapping view that opens, use your mouse to select the *lastName* field under *Path Parameter* and drag it over to the *lastName* field on the right hand side. The result is a line that maps a move of the value of *lastName* from the URL to the field *lastName*. This means the value of *{lastName}* parameter specified in a URL will be moved to the *lastName* field.



- \_\_\_7. Use the **Ctrl-S** key sequence to save all changes and close the *GET.contact.{lastName}* view.

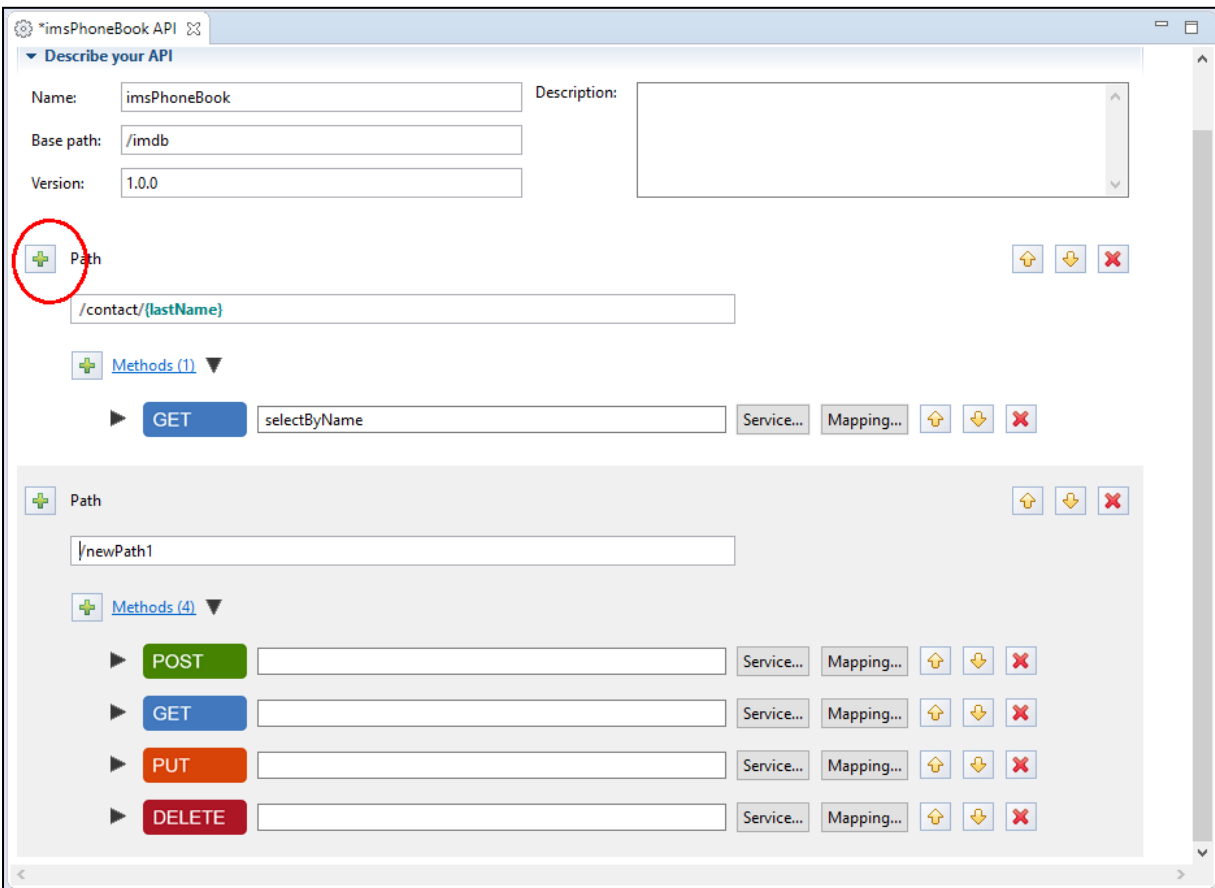


- \_\_\_8. For the **GET** method the default response mapping will return the columns names exposed in the IMS SELECT SQL statment service to the REST client. Click the **Mapping** button beside the **GET** method and select the *Open Default Response Mapping* option.
- \_\_\_9. Use the slider bar to fully expose the *ResultSet Output* structure. You will see the interface names as provided by the services developer.

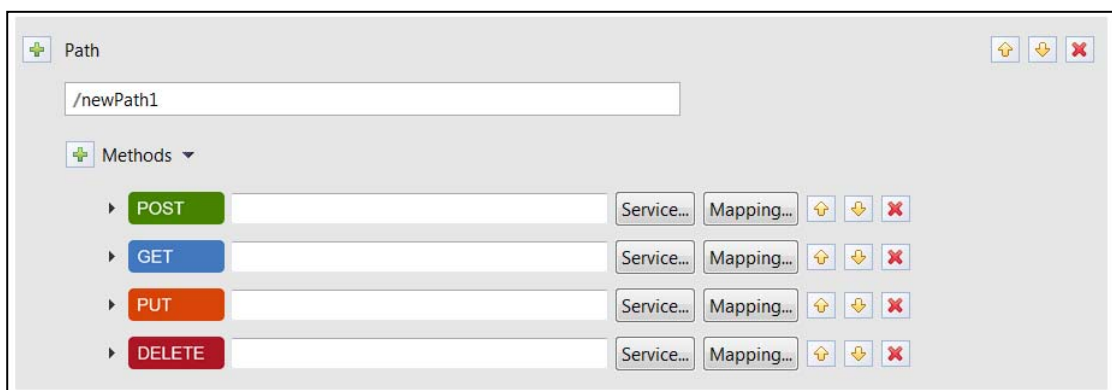


- \_\_\_10. Save all changes with the **Ctrl-S** key sequence and close the response view.

- \_\_\_11. Next, we want to add a Path for a **GET** method for the *selectByZipCode* service. The values for the zip codes are to be included in the URL so no JSON request message is required. Click the plus icon beside Path on the z/OS Connect EE API Editor view to add another path to the API.



The result is another full set of methods for the new *PATH*.



\_\_\_12. Enter a path value of **/zipcodes/{zipcode1}?zipcode2** and remove the **POST**, **PUT** and **DELETE** methods.

The screenshot shows a configuration window with two main sections: 'Path' and 'Methods (1)'. The 'Path' section has a text input field containing '/zipcodes/{zipcod1}?zipcode2'. The 'Methods (1)' section has a dropdown menu showing 'GET' and a 'Service...' button. There are also 'Mapping...' and 'Mapping...' buttons at the bottom right of the 'Methods' section. The interface includes standard UI elements like plus, minus, and close icons for both sections.

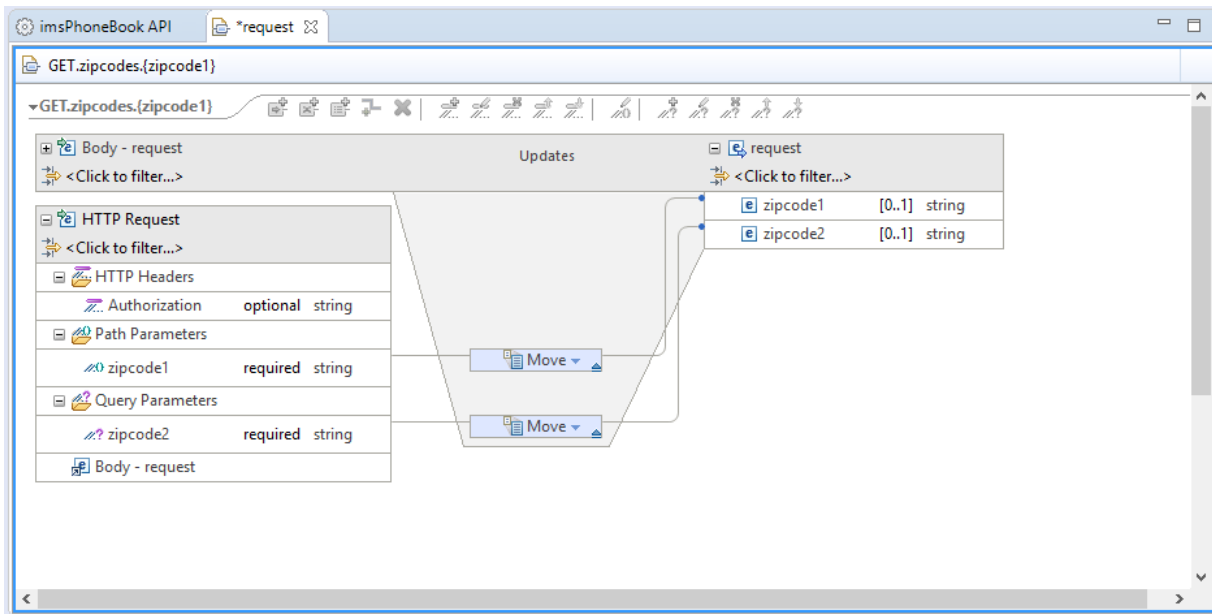
**Tech-Tip:** Additional *Paths* can be added by clicking the + icon beside *Path* and additional *Methods* can be added by clicking the + icon beside *Methods*.

\_\_\_16. Click the **Service** button beside **GET** and select the *selectByZipCode* service:

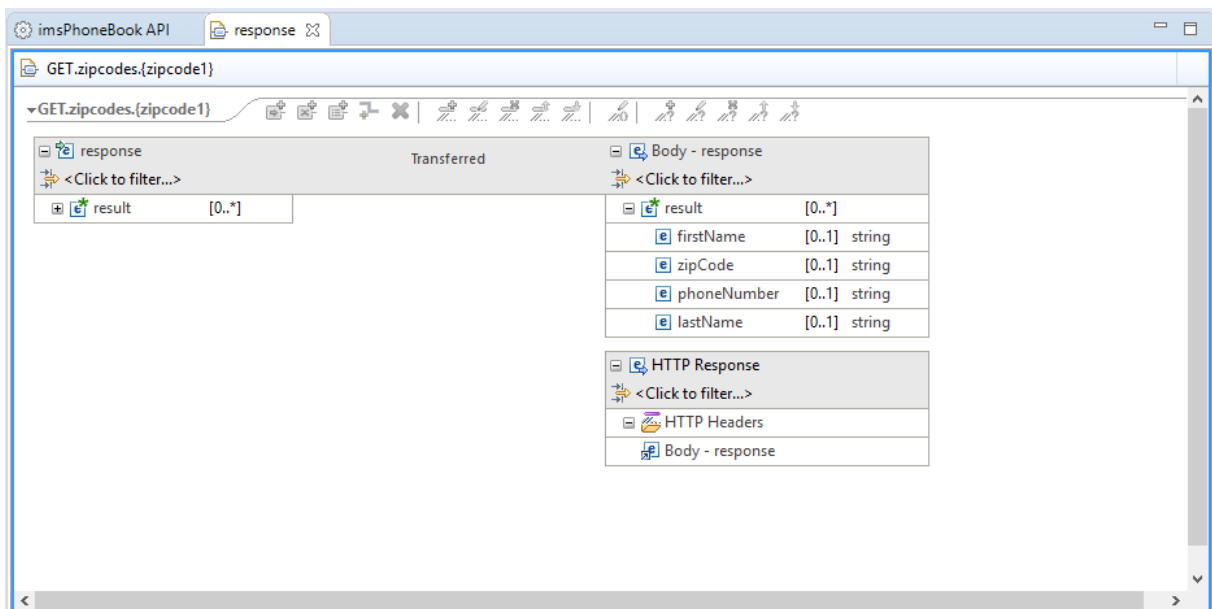
\_\_\_17. Save the changes by using the key sequence **Ctrl-S**.

\_\_\_18. Click on *Mapping* → *Open request mapping*.

- \_\_\_19. Use the left mouse button to drag the *zipcode1 Path Parameters* from the left-hand side to the *zipcode1* field on the right side. Use the left mouse button to drag the *zipcode2 Query Parameter* from the left-hand side to the *zipcode2* field on the right side.



- \_\_\_20. For the **GET** method the default response mapping will return all columns to the REST client.



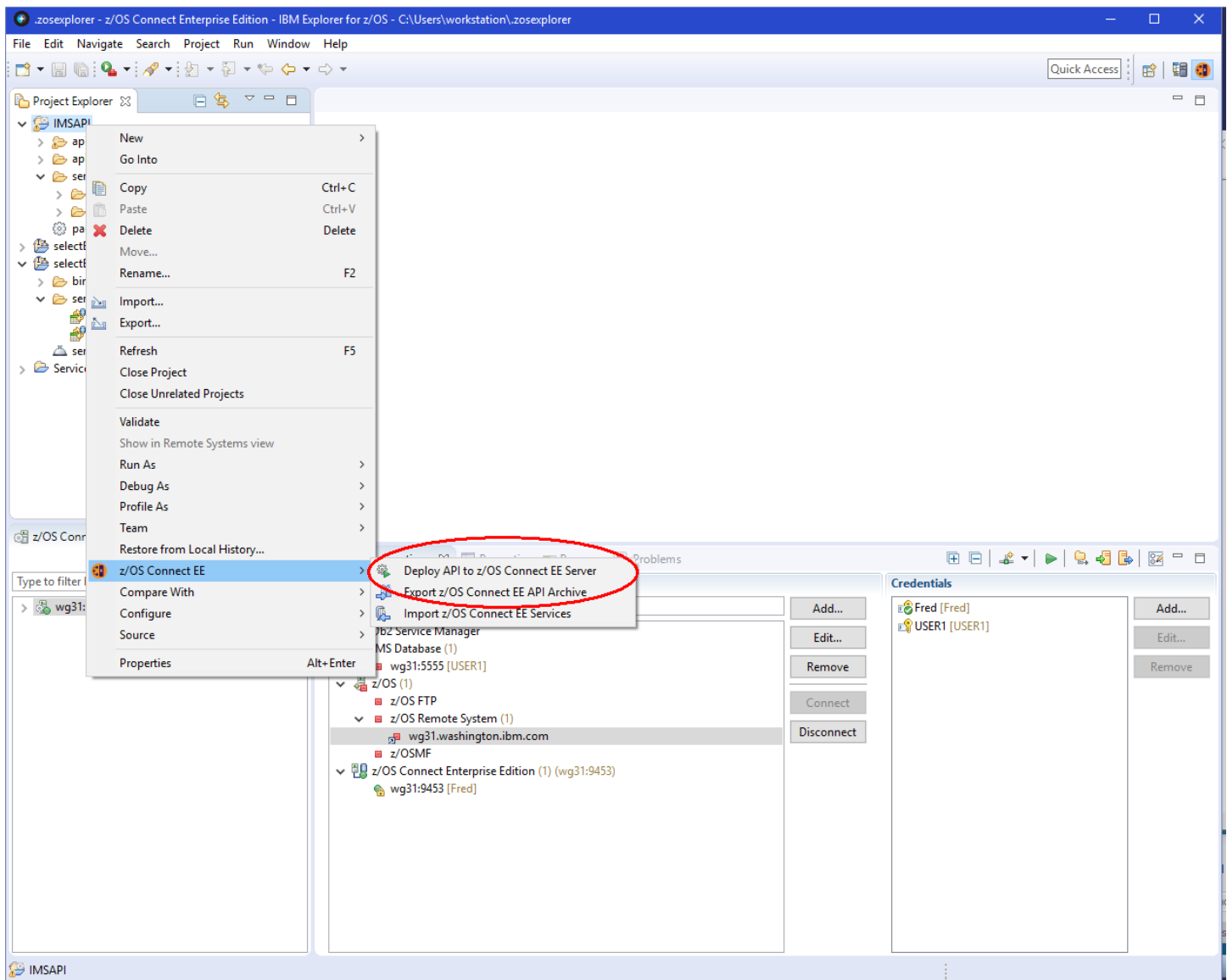
- \_\_\_21. Close and save all open *request* or *response* mapping tabs.

## Summary

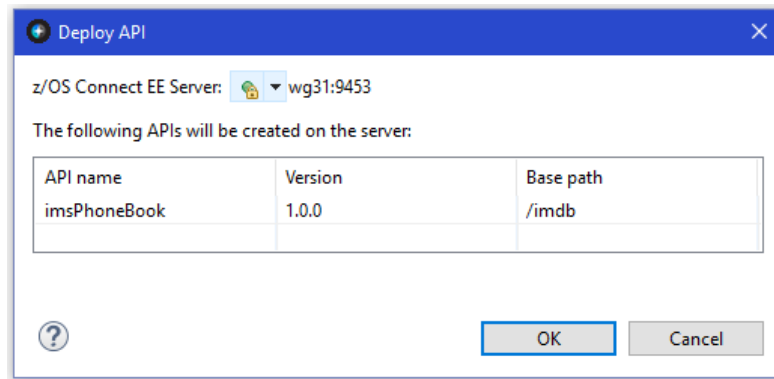
You created the API, which consists of two paths and the request and response mapping associated with each. That API will now be deployed into z/OS Connect EE V3.0.

## Deploy the API to a z/OS Connect EE Server

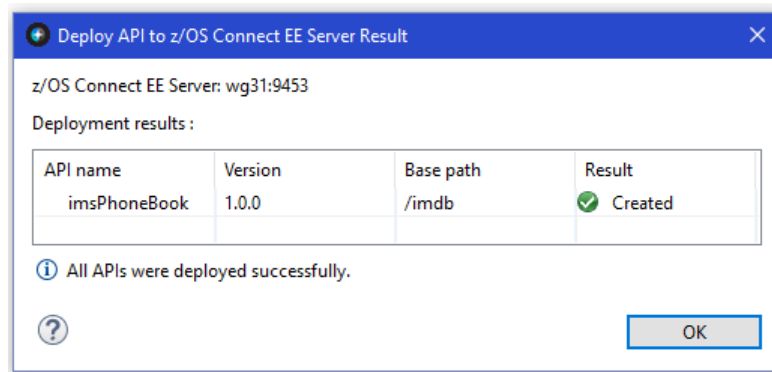
1. In the *Project Explorer* view (upper left), right-mouse click on the *IMSAPI* folder, then select *z/OS Connect EE* → *Deploy API to z/OS Connect EE Server*.



2. If the z/OS Explorer is connected to only one z/OS Connect server there is only one choice (*wg31:9453*). If z/OS Explorer had multiple connections to z/OS Connect servers then the pull-down arrow would allow a selection to which server to deploy, select *wg31:9453* from the list. Click **OK** on this screen to continue.



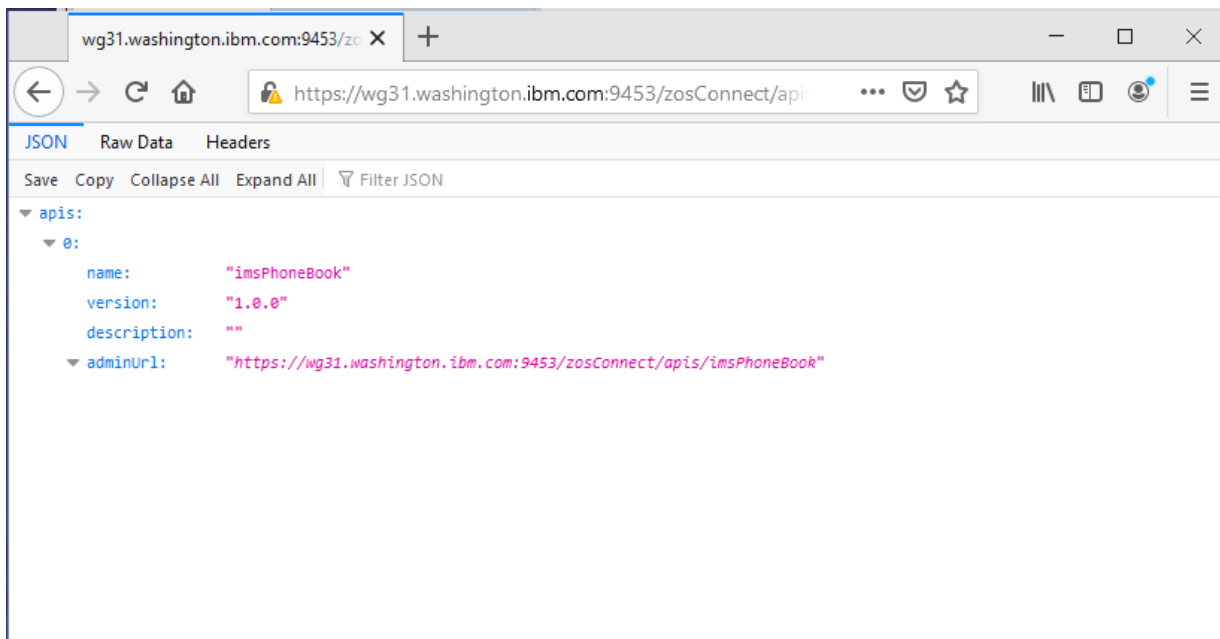
3. The API artifacts will be transferred to z/OS in an API archive (AAR) file and copied into the */var/ats/zosconnect/servers/server1/resources/zosconnect/apis* directory.



## Test the IMS APIs

1. Next enter URL <https://wg31.washington.ibm.com:9453/zosConnect/apis> in the Firefox browser and you should see the window below. The *imsPhoneBook* API now shows as being available.

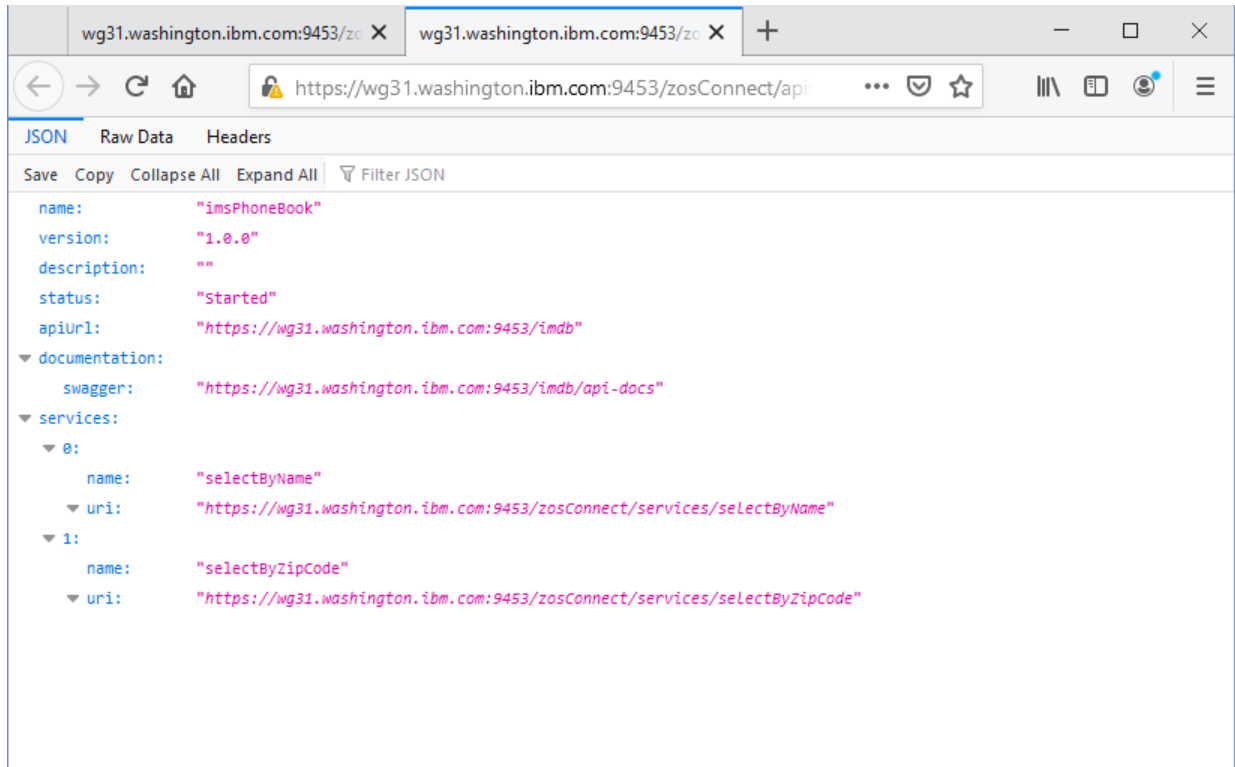
**Tech Tip:** You may be challenged by Firefox because the digital certificate used by the Liberty z/OS server is self-signed. Click the **Advanced** button to continue. Scroll down and then click on the **Accept the Risk and Continue** button. Next you may see a prompt you for a userid and password. If you do see the prompt, enter the username **Fred** and password **fredpwd** (case matters) and click **OK**. Remember we are using basic security, and this is the user identity and password defined in the server.xml file.



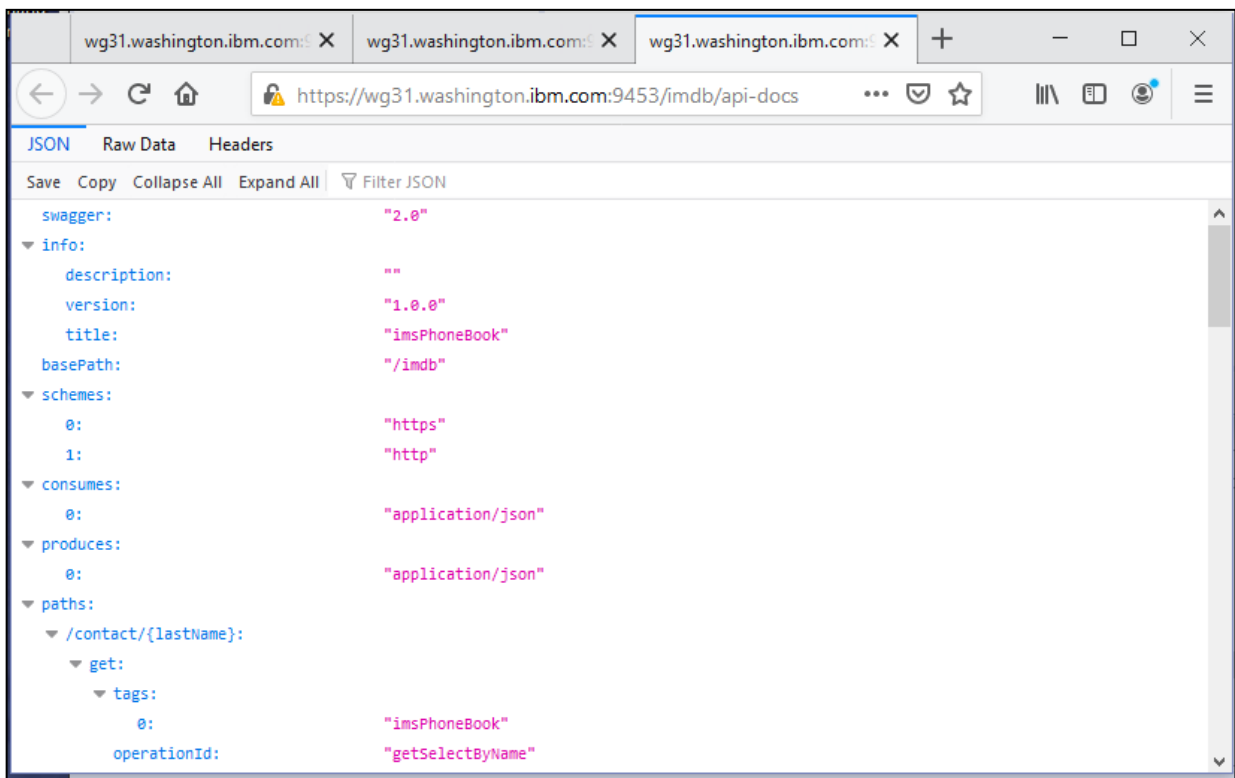
Note that other APIs may also be displayed.

**Tech-Tip:** It is very important to access the z/OS Connect server from a browser prior to any testing using the Swagger UI or Firefox RESTClient extension. Accessing a z/OS Connect URL from a browser starts an SSL handshake between the browser and the server. If this handshake has not performed prior to performing any test the test will fail with no message in the browser and no explanation. Ensuring this handshake has been performed is why you may be directed to access a z/OS Connect URL prior to using the Swagger UI or RESTClient extension during this exercise.

2. If you click on *adminUrl* URL the window below should be displayed:



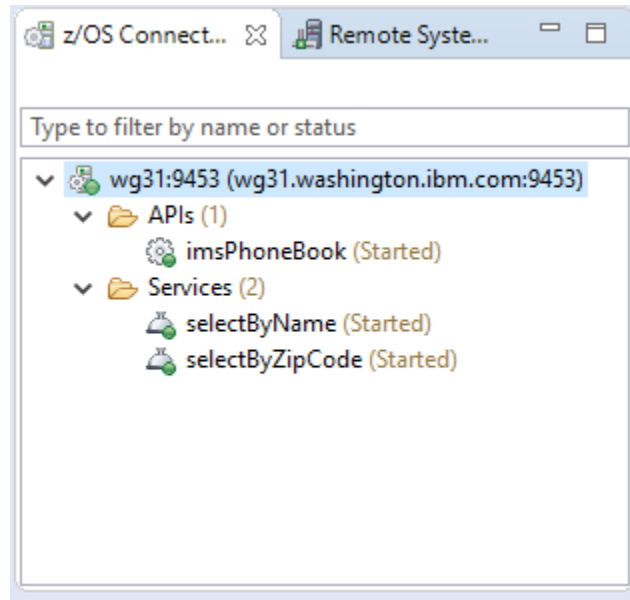
3. Finally click on the *swagger* URL and you should see the Swagger document associated with this API.



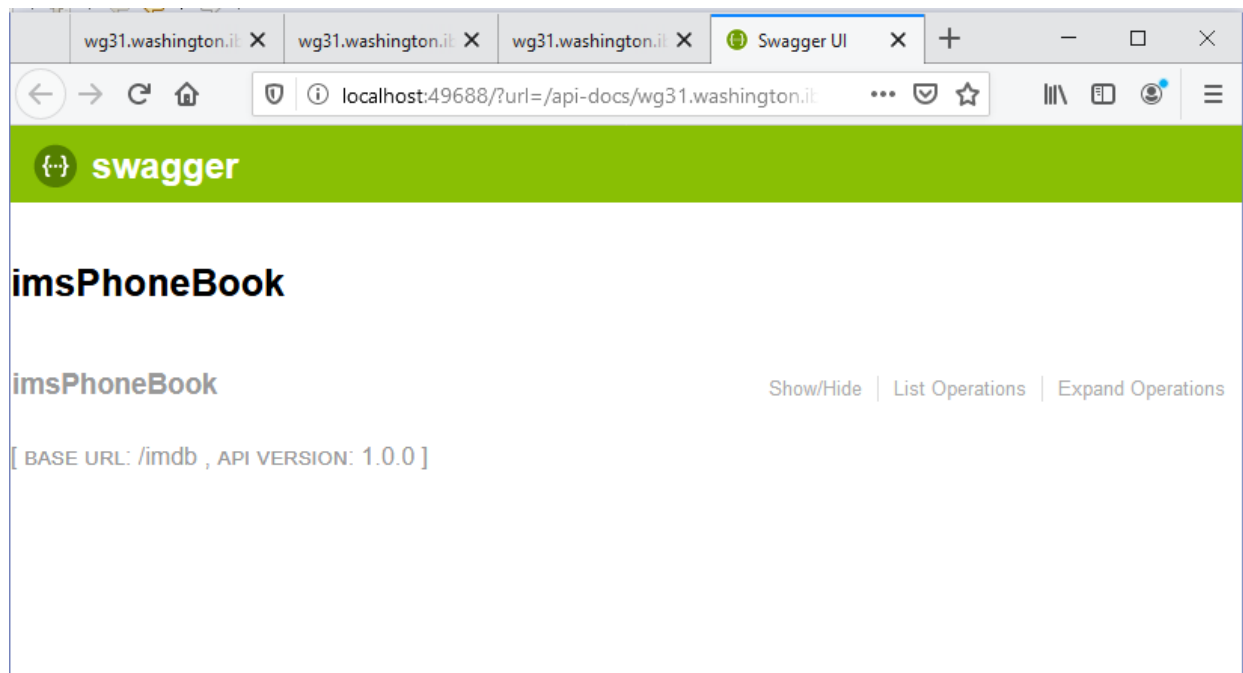


Explore this Swagger document and you will see the results of the request and response mapping performed earlier. This Swagger document can be used by a developer or other tooling to develop REST clients for this specific API.

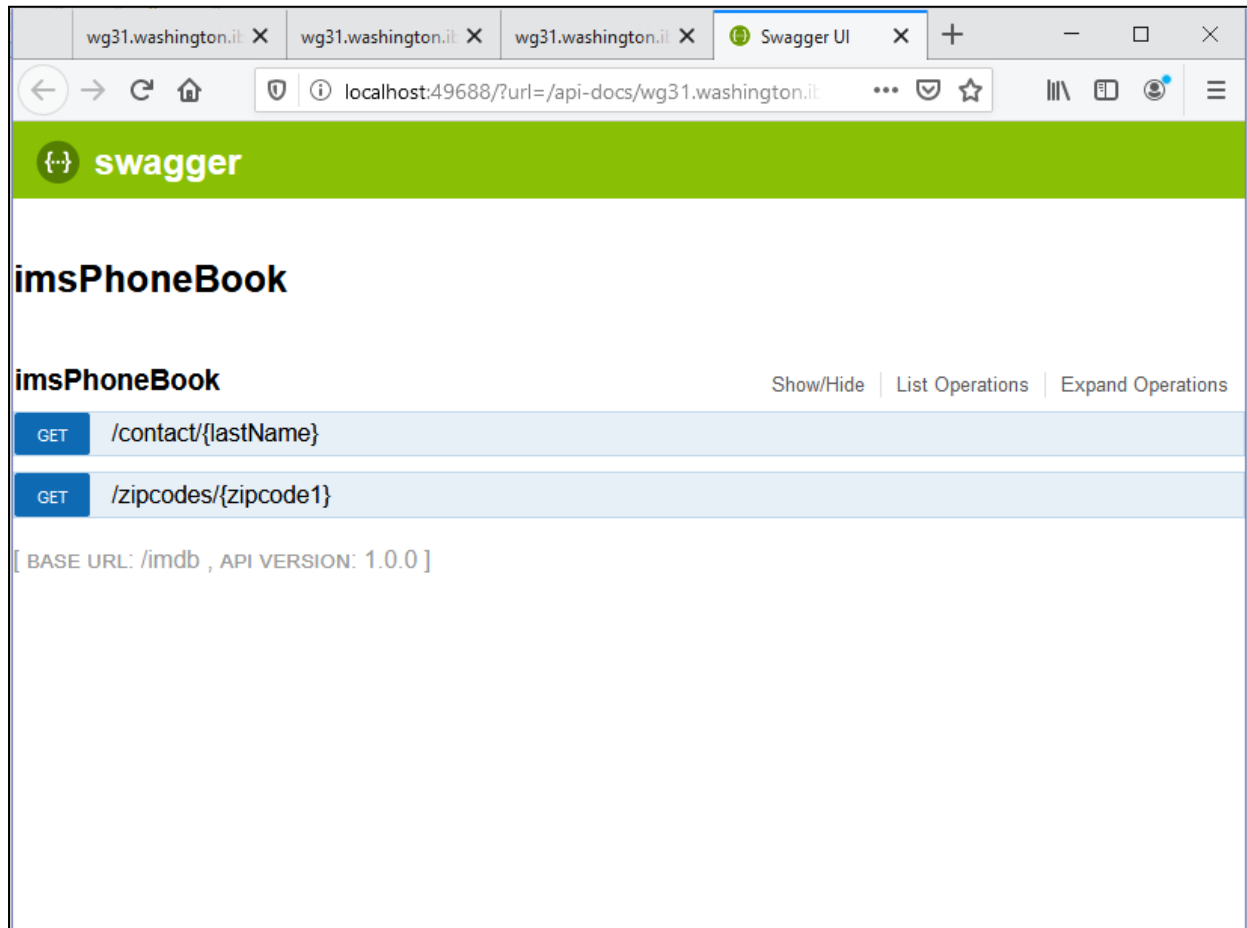
4. In the lower left-hand side of the *z/OS Connect Explorer* perspective there is view entitled *z/OS Connect EE Servers*. Expand *wg31:9453* and then expand the *APIs* folder. the *RESTClient* icon. You should see a list of the APIs installed in the server.



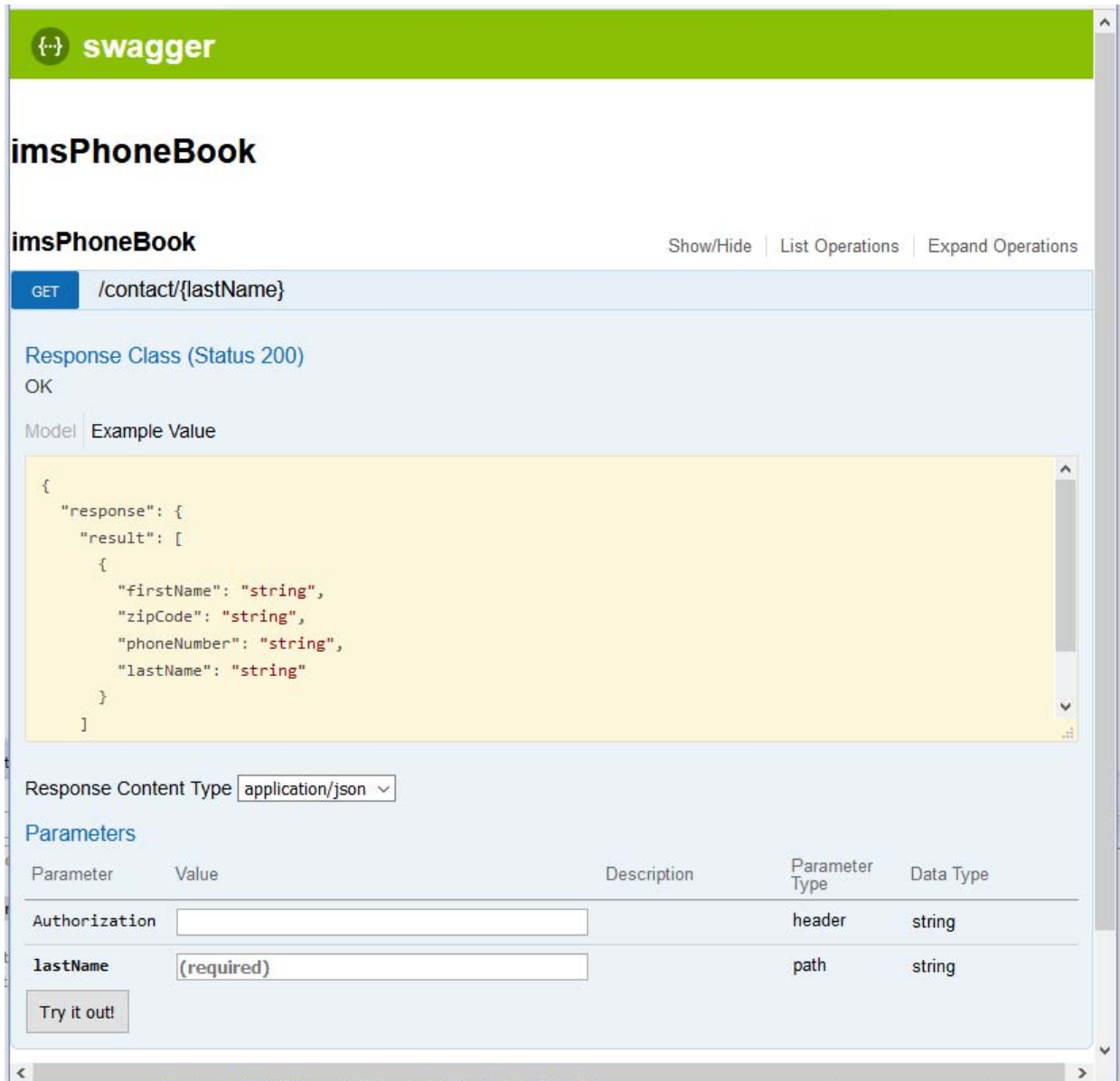
5. Right mouse button click on *imsPhoneBook* and select *Open in Swagger UI*. Click OK if an informational prompt appears. This will open a new view showing a *Swagger* test client (see below).



6. Click on *List Operations* option in this view and this will display a list of available HTTP methods in this API.



7. Select the *GET* method for selecting a phone book entry from the data base by the last name by clicking on the `/contact/{lastName}` URI string. Remember this was the *Path* specified for the *GET* method for the *selectByName* service when the API was defined. This action will expand this method in this view and provides a Swagger UI test client (you may have to use the slider bar and adjust the perspective to see the entire client).



8. Enter **LAST2** in the box beside *lastName* and **Basic RnJlZDpmcmVkcHdk** for *Authorization* and press the **Try it out!** button. You may see a Security Alert pop-up warning about the self-signed certificate being used by the z/OS Connect EE server. Click **Yes** on this pop-up.

9. Scroll down the view and you should see the Request URL and Response Body which contains the results of the GET method (see below). Note that the columns removed from the interface in an earlier steps are not present.

#### Request URL

```
https://wg31.washington.ibm.com:9453/imdb/contact/LAST1
```

#### Request Headers

```
{  
  "Accept": "application/json"  
}
```

#### Response Body

```
{  
  "response": {  
    "result": [  
      {  
        "firstName": "FIRST1",  
        "zipCode": "D01/R01",  
        "lastName": "LAST1",  
        "phoneNumber": "8-111-1111"  
      }  
    ]  
  }  
}
```

#### Response Code

```
200
```

#### Response Headers

```
{  
  "content-language": "en-US",  
  "content-type": "application/json"  
}
```

Repeat the **GET** methods with other records (see table below) and verify the results are as expected.

Last Name	First Name	Phone Number	Zip Code
JOHNSON	MITCH	0065	27502
LAST1	FIRST1	8-111-1111	D01/R01
LAST2	FIRST2	8-111-2222	D02/R02
LAST4	FIRST4	8-111-4444	D04/R04
LAST5	FIRST5	8-111-5555	D05/R05
LAST6	FIRST6	8-111-6666	D06/R06

10. Collapse the Swagger UI test areas for the **GET** methods for `/contact/{lastName}` clicking on the URIs.

11. Click on the URI for the **GET** method for `/zipcodes/{zipcode1}` to open its Swagger Test user interface and scroll down to the *Response Content Type* area.

Response Content Type application/json ▾

Parameters

Parameter	Value	Description	Parameter Type	Data Type
Authorization	<input type="text"/>		header	string
zipcode1	<input type="text" value="(required)"/>		path	string
zipcode2	<input type="text" value="(required)"/>		query	string

Try it out! [Hide Response](#)

Note that this API requires two parameters, a path parameter *zipcode1* and a query parameter *zipod2*. These are present because the path `/zipcode/{zipcode1}?zipcode2` was specified when the API was developed. Enter **D01/R01** for the *zipcode1* and **D02/R02** for *zipcode2* and **Basic RnJlZDpmcmVkcHdk** for *Authorization* and press the **Try it Out!** button. Scroll down and you should see the following information in the *Response Body*.

```
Response Body

{
  "response": {
    "result": [
      {
        "firstName": "FIRST1",
        "zipCode": "D01/R01",
        "lastName": "LAST1",
        "phoneNumber": "8-111-1111"
      },
      {
        "firstName": "FIRST2",
        "zipCode": "D02/R02",
        "lastName": "LAST2",
        "phoneNumber": "8-111-2222"
      }
    ]
  }
}
```

## Summary

You have verified the API. The API layer operates above the service layer you defined and tested earlier. The API layer provides a further level of abstraction and allows a more flexible use of HTTP verbs, and better mapping of data via the API editor function.

As an optional activity you might want to add services for the other SQL commands tested in IMS Explorer and then add them to the API.

**Congratulations, you have completed this exercise.**