IBM z/OS Connect EE V3.0

Developing RESTful APIs for VSAMCICS DVM Services

IBM Z
Wildfire Team –
Washington System Center

Lab Version Date: November 19, 2021

Table of Contents

Overview **3**
Use the DVM Studio to create VSAMCICS services **4**
Create a DVM virtual table *4*
Create a DVM web service *13*
Deploy the DVM services *31*
Export the DVM services as z/OS Connect service archive files *34*
Using DVM services in a z/OS Connect EE API **35**
Connect to a z/OS Connect EE Server *35*
Create the VSAMCICS DVM API Project *38*
Import the SAR files generated by the DVM Studio *40*
Compose an API for the DVM Rest Services *43*
Deploy the API to a z/OS Connect EE Server *59*
Test the VSAM CICS APIs using Swagger UI *61*
Test the VSAM CICS APIs using Postman *71*

Mitch Johnson (mitchj@us.ibm.com)

Overview

The objective of these exercises is to gain experience with working with the Data Virtualization Manager (DVM) Studio and the z/OS Connect EE API Toolkit. These two products allow the exposure of z/OS resources to JSON clients using DVM services rather than z/OS Connect services.

The RESTful APIs created in this exercise will be accessing a VSAM defined to CICS. Using CICS as a resource manager provides support for inserting new records and changing or deleting existing records in the VSAM data set.

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

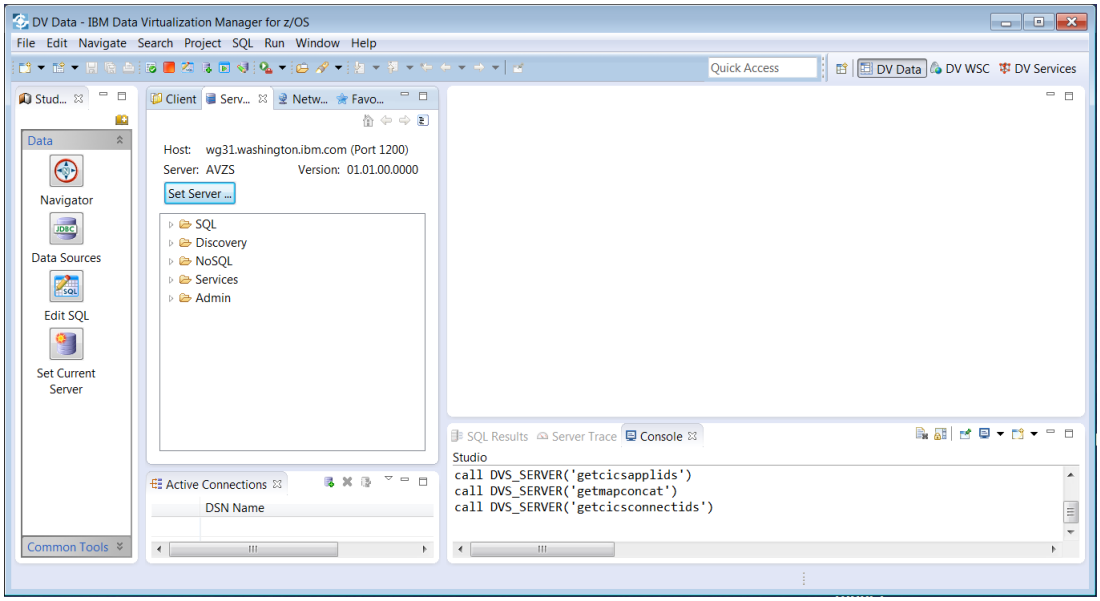
- ✓ This exercise requires using z/OS user identity *USER1*. The password for this user will be provided by the lab instructor.
- ✓ dd not hesitate to request assistance any time you have any questions about the use of the Data Virtualization Manager Studio, IBM z/OS Explorer, z/OS Connect EE Toolkit features or other tools.
- ✓ The VSAM data set being used for this exercise is the VSAM data set provided by the CICS Catalog Manager sample application. For details on this VSAM data set see member DFH$ECAT in the CICS SDFHINST target data set.
- ✓ This exercise accessed a VSAM data set through a CICS region. In this scenario, the contents of the VSAM data set can be modified since CICS is a resource manager controlling concurrent access. Using DVM to access a VSAM data set directly can only read the data set unless VSAM RLS is enabled (another resource manager).
- ✓ Please note that there may be minor differences between the screen shots in this exercise versus what you see on your desktop. These differences should not impact the completion of this exercise.
- ✓ Text in **bold** and highlighted in **yellow** in this document should be available for copying and pasting in a file named *Development APIs for DVM CopyPaste* file on the desktop.

Use the DVM Studio to create VSAMCICS services

Create a DVM virtual table

Access to a VSAM or non VSAM dataset using the DVM SQL commands requires the creation of virtual table. The virtual table represents the layout or contents of the records in the data sets.

___1. On the workstation desktop, locate the Data Virtualizaiton Manager Studio icon and double click on it to open the tool. You should automatically be connected to the DVM server running on z/OS, see below.



Tech-Tip: Eclipse based development tools like DVM Studio; provide a graphical interface consisting of multiple views within a single window.

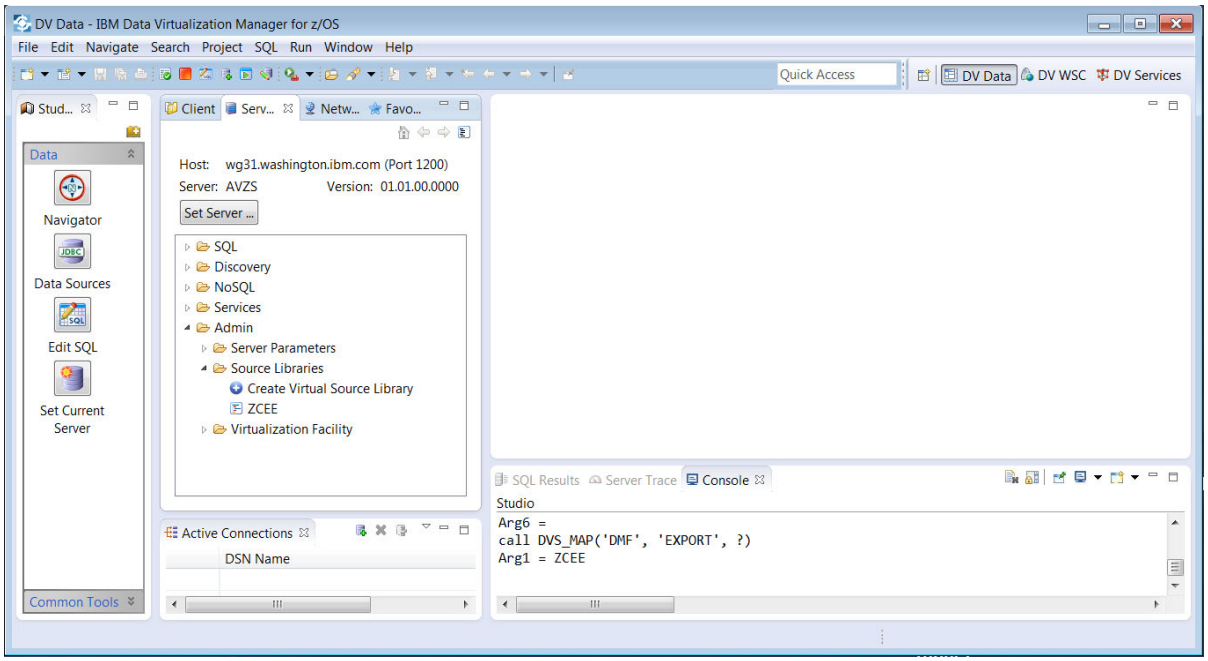
A view is an area in the window dedicated to providing a specific tool or function. For example, in the window above, *Console, Studio Navigator and Server,* are views that use different areas of the window for displaying information. At bottom on the right there is a single area for displaying the contents of three views stacked together (commonly called a *stacked views*), *Console, SQL Results and Server Trace,* 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 DVM Studio 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.

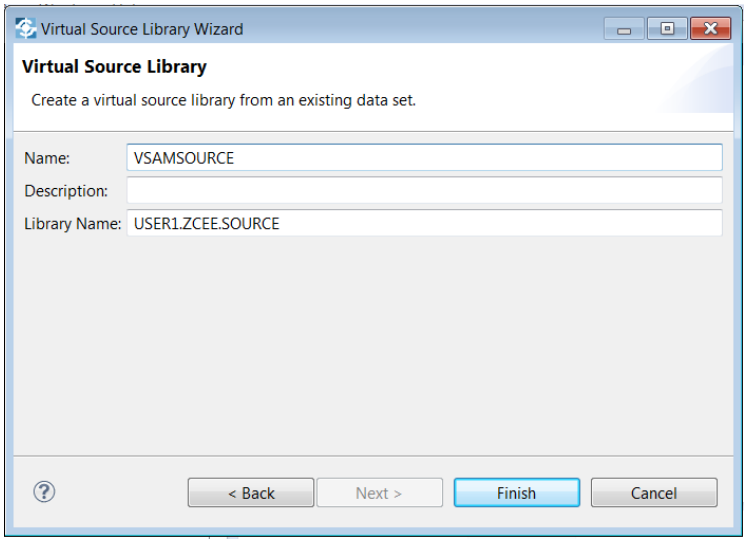
Mitch Johnson (mitchj@us.ibm.com)

___2. In the *Server* view, expand *Admin* then expand *Source Libraries* to display the *Create Virtual Source Library* wizard, see below.



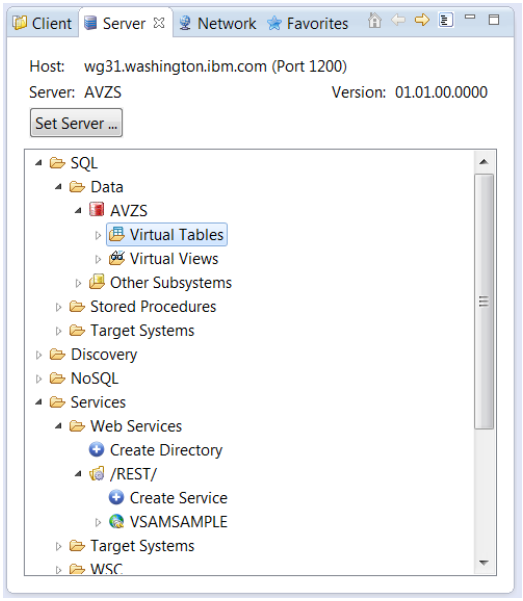
___3. Double click the *Create Virtual Source Library* wizard and on the *New - Select a wizard* window, select *Data Set* and click **Next** to continue.

___4. On the *Virtual Source Library,* enter ***VSAMSOURCE*** as the *Name* of the virtual source library and ***USER1.ZCEE.SOURCE*** as the *Library Name* and press **Finish** to continue.

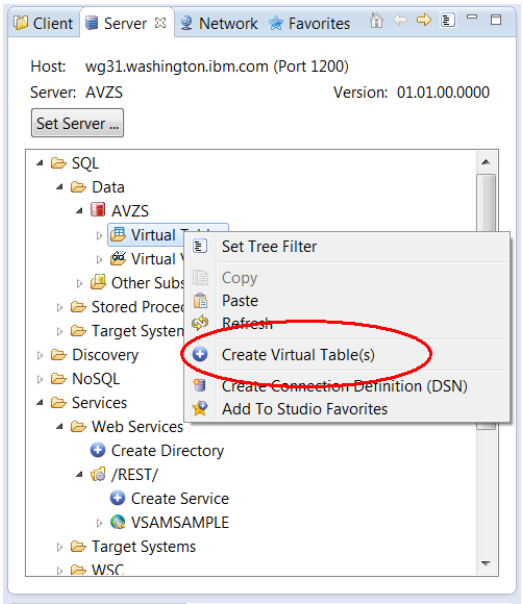


Important: The values used for names, e.g., VSAMSOURCE, are somewhat arbitrary, but they do relate to later tasks. If you use the values and cases as supplied then in subsequent windows, results, etc. will be consistent with what is shown in this document.

___5. Next expand the *SQL* folder, then the *Data* folder and then the *AVZS* folder to display the *Virtual Tables* and *Virtual Views*, see below.

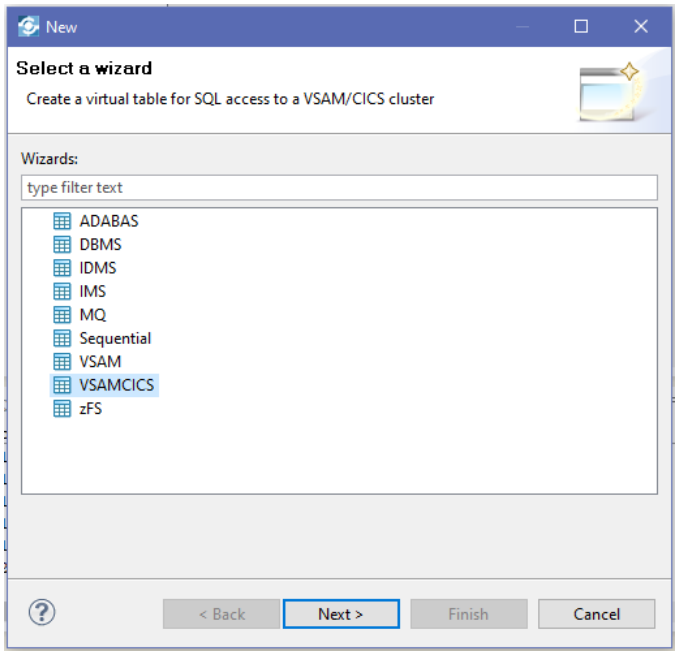


___6. Select the *Virtual Tables* folder and right mouse button click and then select *Create Virtual Table(s)* property, see below.



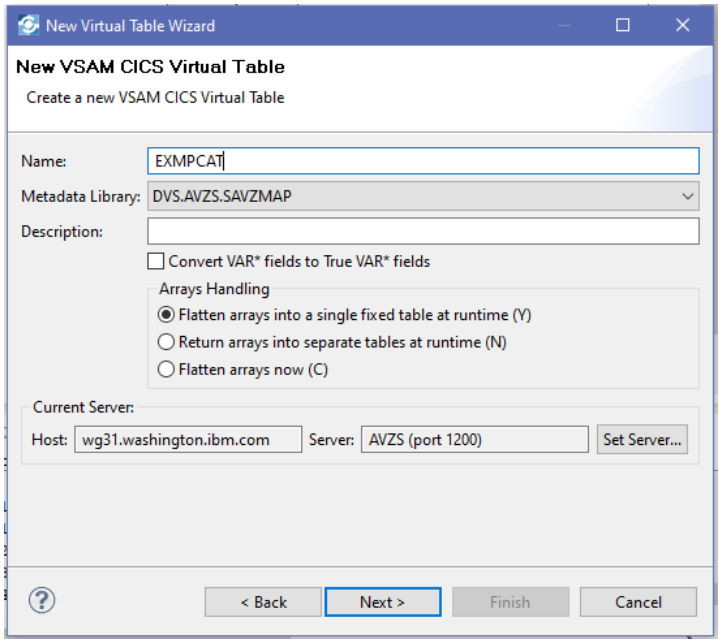
Mitch Johnson (mitchj@us.ibm.com)

___7. On the *Select a wizard* window select *VSAMCICS* and press **Next** to continue.



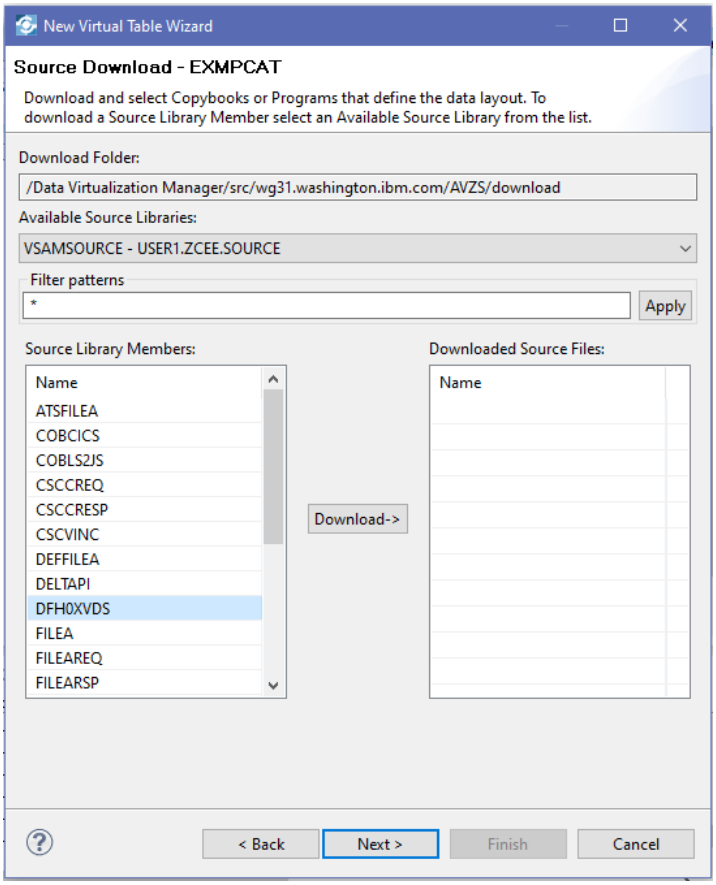
Tech-Tip: The processes for accessing Sequential, native VSAM and ZFS files using the DVM Studio are fundamentally the same as the process described in this exercise. These instructions could be used as the starting point for developing APIs for these other types of MVS data sets and OMVS files.

___8. On the *New VSAM Virtual Table* window, enter ***EXMPCAT*** as the name and press **Next** to continue.



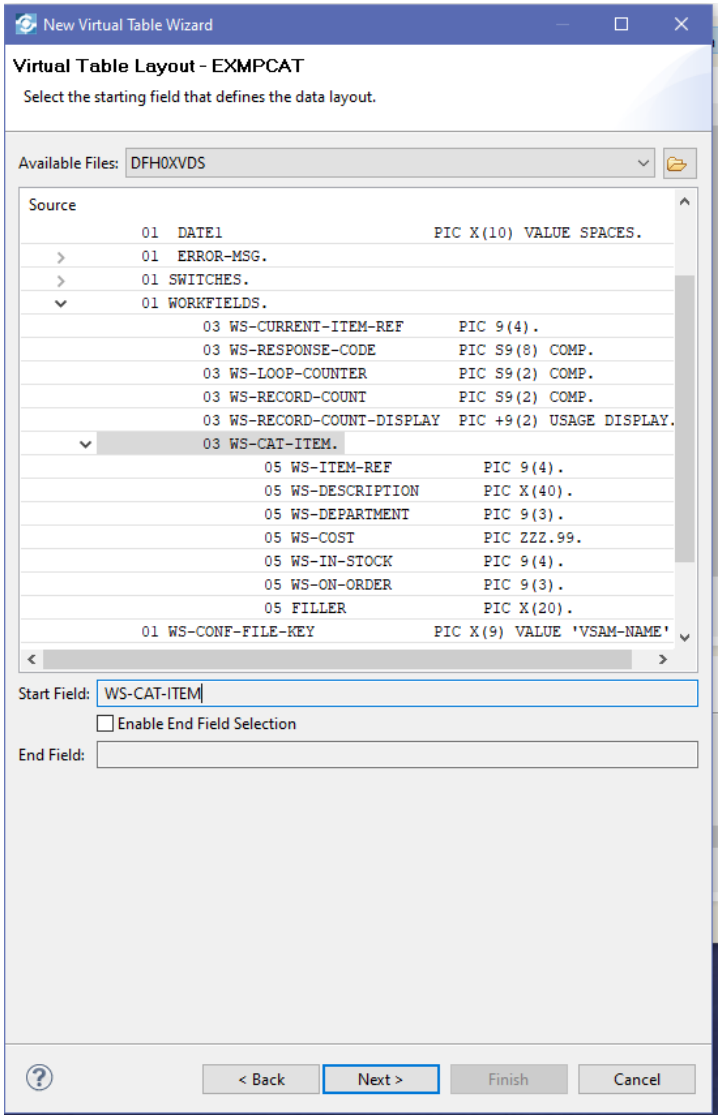
Mitch Johnson (mitchj@us.ibm.com)

___9. On the *Source Download – EXMPCAT* window, use the pull-down arrow to select the *VSAMSOURCE-USER1.ZCEE.SOURCE* source library (created earlier). This will download a list of the members in this partitioned data set. Select member *DFH0XVDS* and use the **Download** button to have this source downloaded to the workstation.



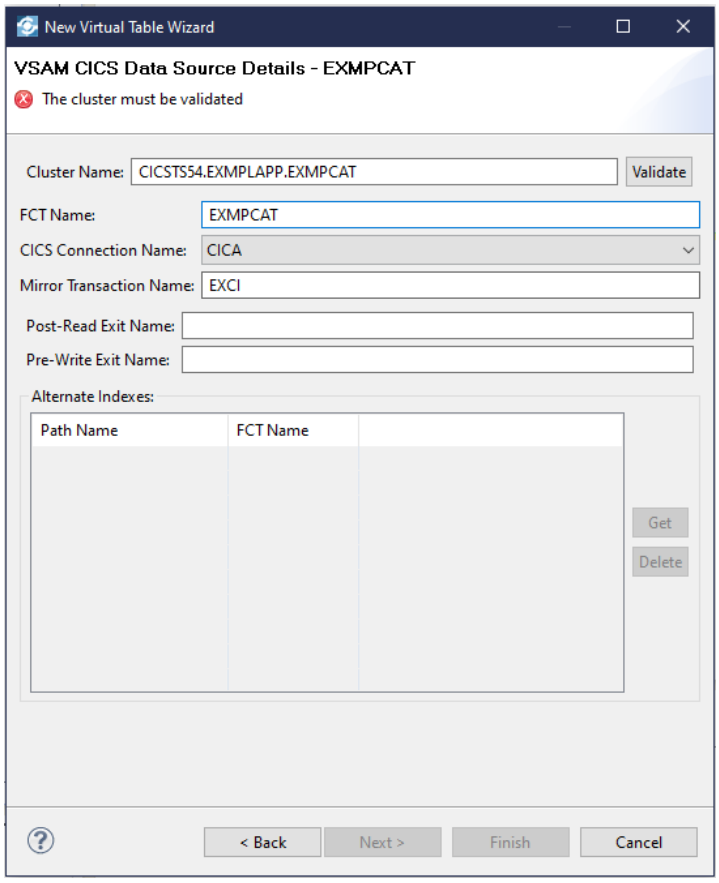
Mitch Johnson (mitchj@us.ibm.com)

___10. Click **Next** to continue. This will display the *Virtual Table Layout – EXMPCAT* window. Scroll down and expand the COBOL *WORKFIELDS* structure until the *WS-CAT-ITEM* structure is displayed (see below). This structure represents the actual layout of a record in the VSAM file. Select *WS-CAT-ITEM* and click **Next** to continue.

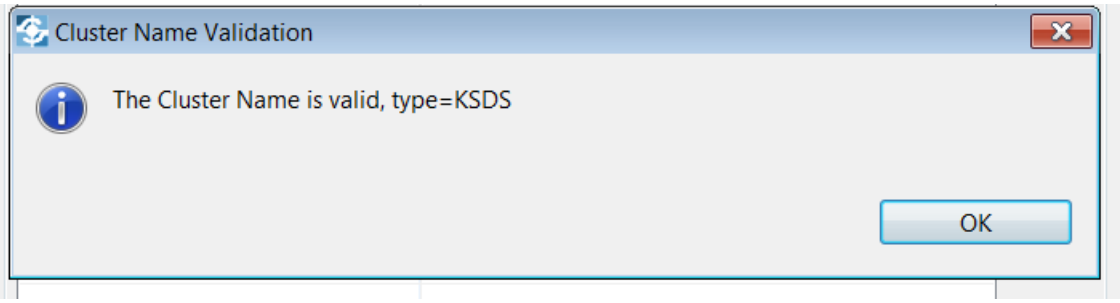


Mitch Johnson (mitchj@us.ibm.com)

___11. On the *VSAM Data Source Details – EXMPCAT* window, enter ***CICSTS54.EXMPLAPP.EXMPCAT*** as the *Cluster Name*, ***EXMPCAT*** as the *FCT Name,* ***CICA*** as the *CICS Connection Name* and ***CSMI*** as the *Mirror Transaction Name*. Press the **Validate** button.



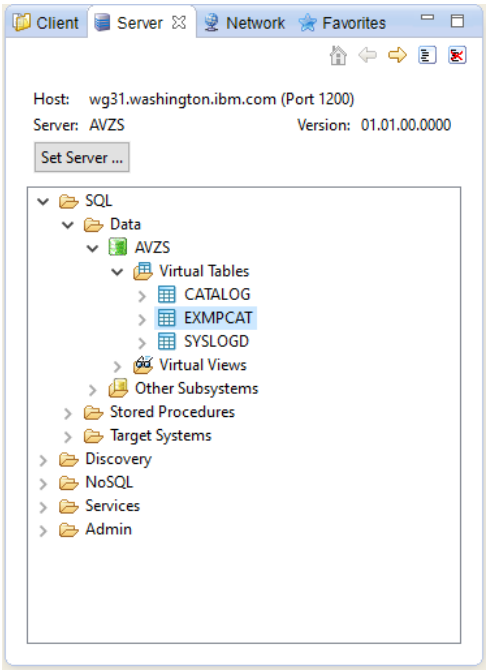
___12. This will validate that the VSAM data set does exist and will display the type of VSAM access allowed, e.g., KSDS, ESDS, RRDS.



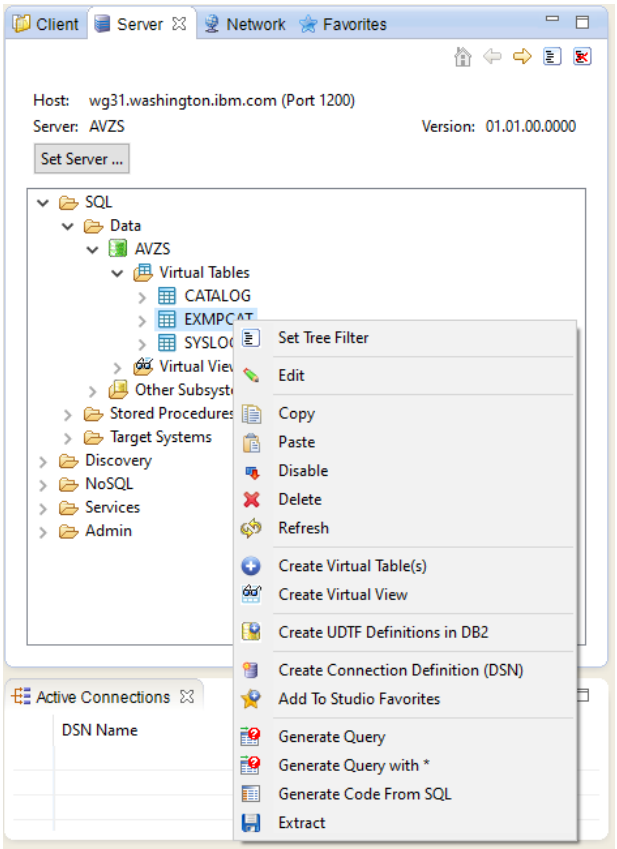
___13. Click the **OK** button and then the **Finish** button to continue.

Mitch Johnson (mitchj@us.ibm.com)

___14. In the list of *Virtual Tables* an entry for *EXMPCAT* should now appear. Select *EXMPCAT* and right mouse button click.

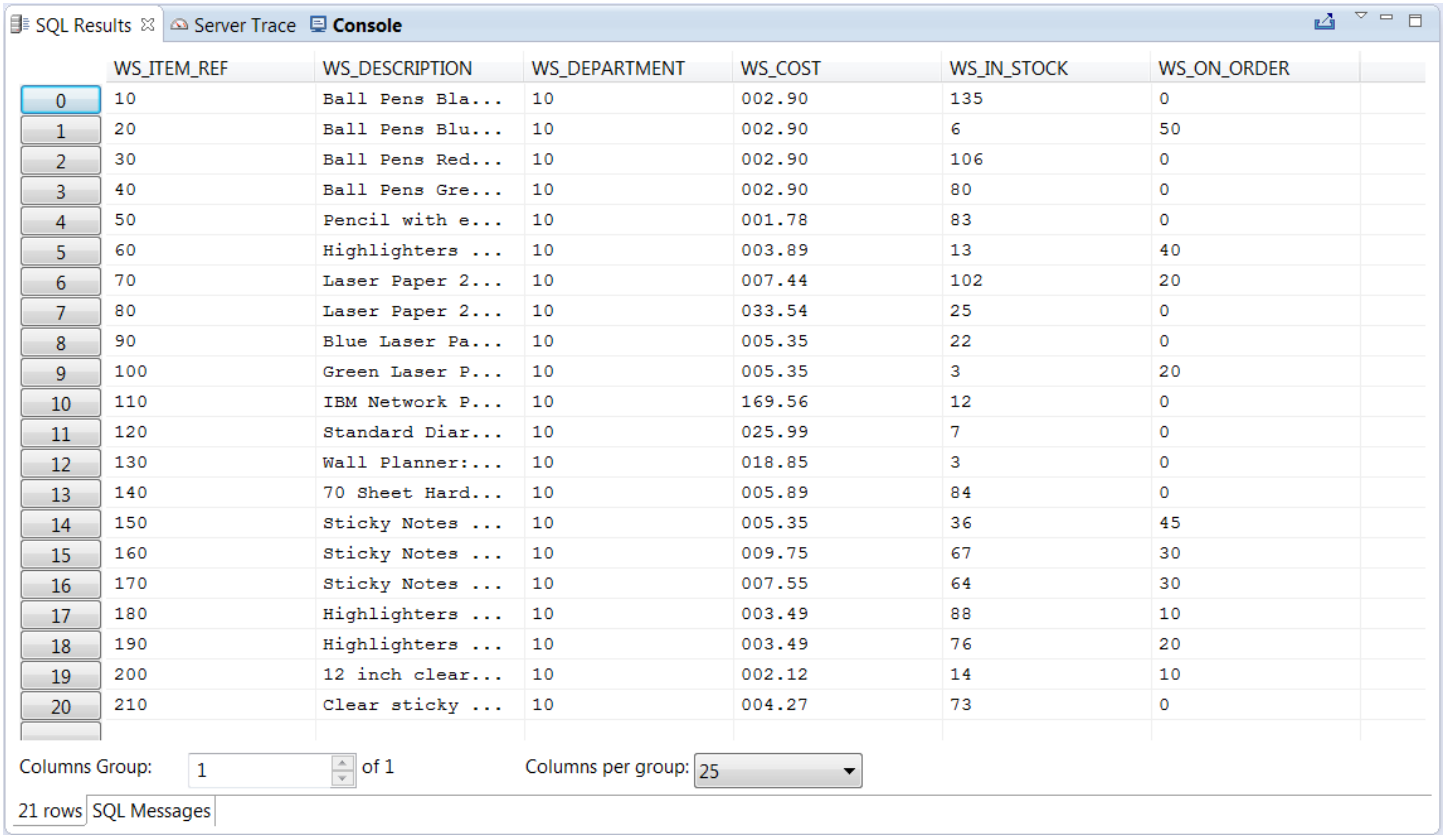


___15. Select option *Generate Query* and click **Yes** on the *Execute Query*? pop up window.



Mitch Johnson (mitchj@us.ibm.com)

___16. This will access the VSAM data set and display the contents of this VSAM data set in the view on the lower right-hand side

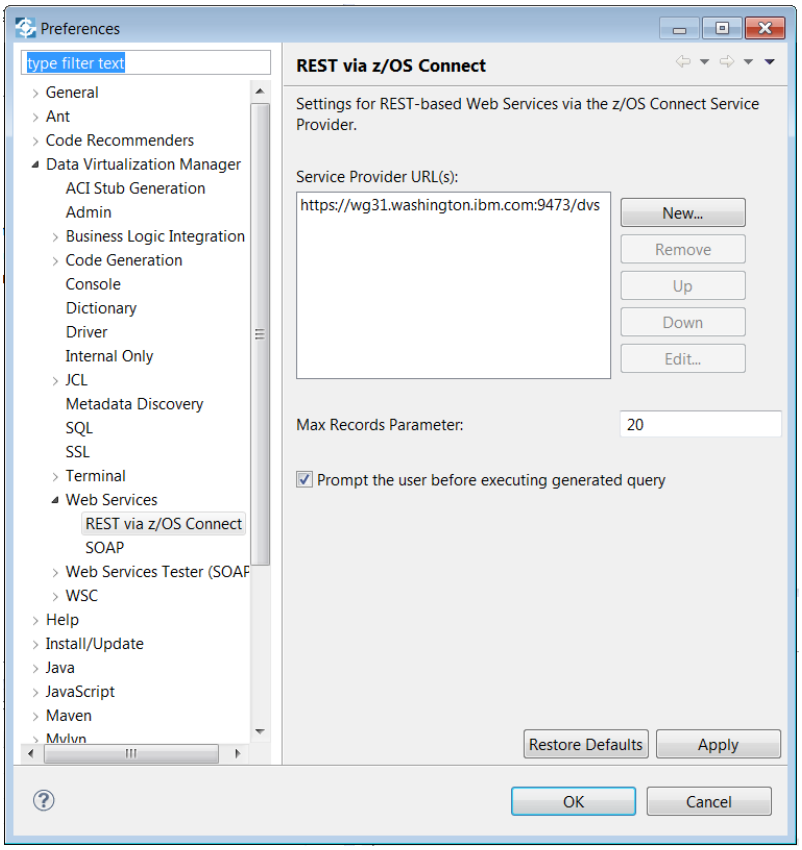


	WS_ITEM_REF	WS_DESCRIPTION	WS_DEPARTMENT	WS_COST	WS_IN_STOCK	WS_ON_ORDER
0	10	Ball Pens Bla...	10	002.90	135	0
1	20	Ball Pens Blu...	10	002.90	6	50
2	30	Ball Pens Red...	10	002.90	106	0
3	40	Ball Pens Gre...	10	002.90	80	0
4	50	Pencil with e...	10	001.78	83	0
5	60	Highlighters ...	10	003.89	13	40
6	70	Laser Paper 2...	10	007.44	102	20
7	80	Laser Paper 2...	10	033.54	25	0
8	90	Blue Laser Pa...	10	005.35	22	0
9	100	Green Laser P...	10	005.35	3	20
10	110	IBM Network P...	10	169.56	12	0
11	120	Standard Diar...	10	025.99	7	0
12	130	Wall Planner:...	10	018.85	3	0
13	140	70 Sheet Hard...	10	005.89	84	0
14	150	Sticky Notes ...	10	005.35	36	45
15	160	Sticky Notes ...	10	009.75	67	30
16	170	Sticky Notes ...	10	007.55	64	30
17	180	Highlighters ...	10	003.49	88	10
18	190	Highlighters ...	10	003.49	76	20
19	200	12 inch clear...	10	002.12	14	10
20	210	Clear sticky ...	10	004.27	73	0

Verification of the virtual table completes the creation of the virtual tables for this VSAM data set.

Create a DVM web service

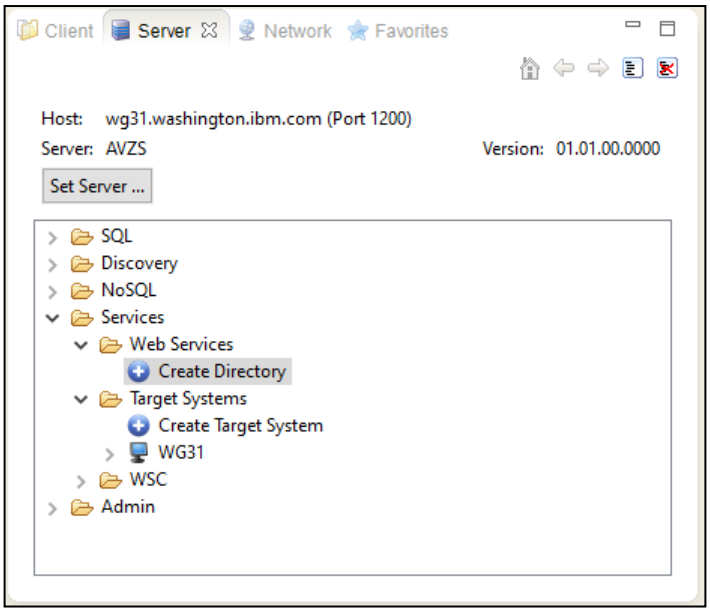
___1. Confirm that the DVM studio is properly configured to communicate with the z/OS Connect EE server with the DVM service provider installed. On the DVM Studio toolbar, click on *Windows* then *Preferences* to display the Eclipse *Preferences* window. Expand *Data Virtualization Manager* and then *Web Services* to select *REST via z/OS Connect,* see below:



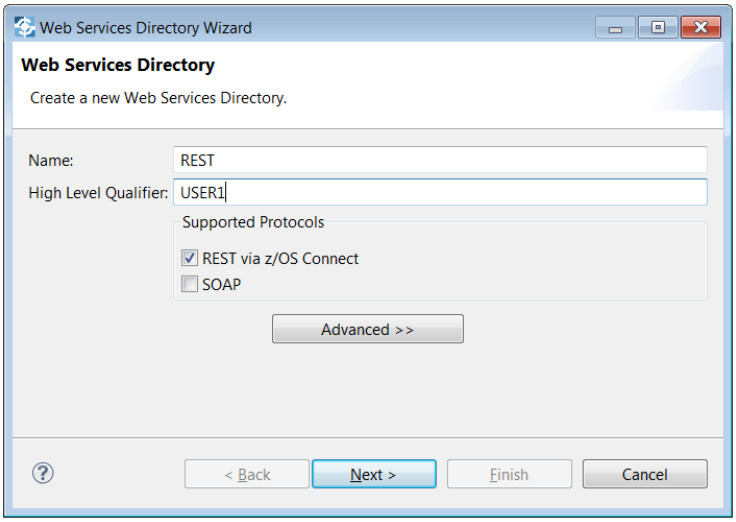
___2. Ensure the *Service Provider URL(s)* is set to ***https://wg31.washington.ibm.com:9473/dvs***. If not, select the provider and use the **Edit** button to set it correctly.

Mitch Johnson (mitchj@us.ibm.com)

___3. Back in the *Server* view, expand the *Services* and then the *Web Services* folders.



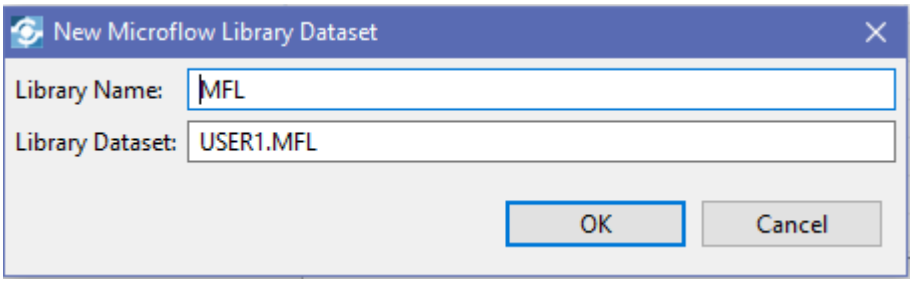
___4. If the */REST/* Web Services directory does not exist, double click on *Create Directory* to open the *Web Services Directory Wizard*. Enter ***REST*** and ***USER1*** as shown below. Click **Next** to continue. Otherwise continue with Step 7.



Tech-Tip: The *High-Level Qualifier* will be used to create a micro flow partitioned data set with a data set name of USER1.MFL.

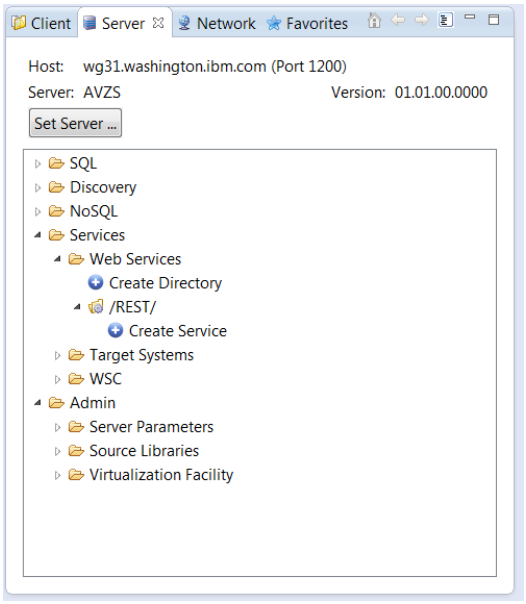
Mitch Johnson (mitchj@us.ibm.com)

___5. On the *Web Service Directory Wizard – Microflow Library* window, if there is a current Microflow liberty (e.g., USER1.MFL), select it. Otherwise, click the **Create New Microflow Library** button and accept the defaults on the *New Microflow Library Dataset* pop-up window. Click **OK** to continue.



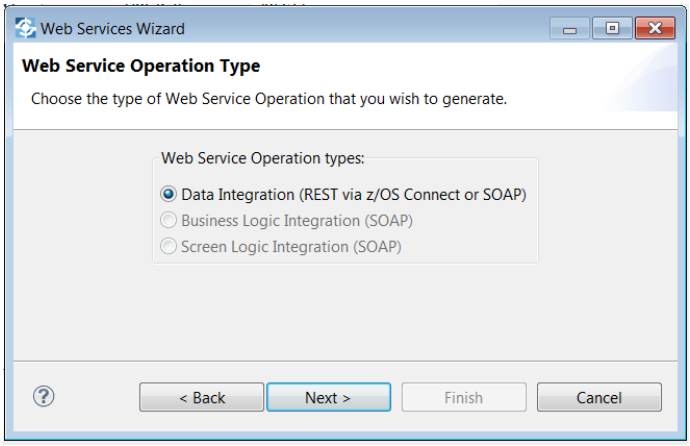
___6. On the *Microflow Library* window, select *MFL* under *Current Microflow Libraries* and click **Finish** to continue.

___7. Expand */REST/* and use the *Create Service* wizard to create a new web service.

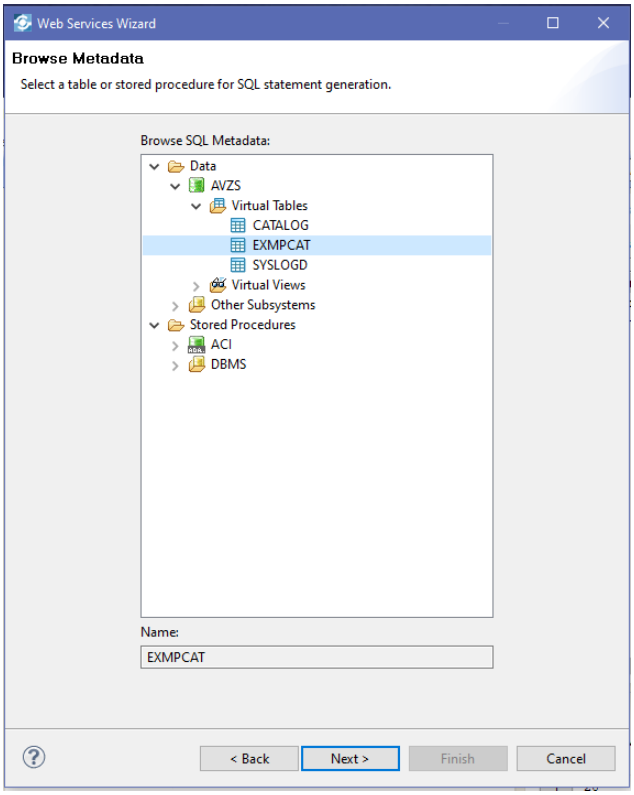


___8. On the *Web Service – Create a new Web Service* window, enter ***EXMPCAT*** as the *Name* and press **Next** to continue.

___9. On the *Web Service Operation Type* window ensure the radio button beside *Data Integration (REST via z/OS Connect or SOAP)* is selected and click **Next** to continue.

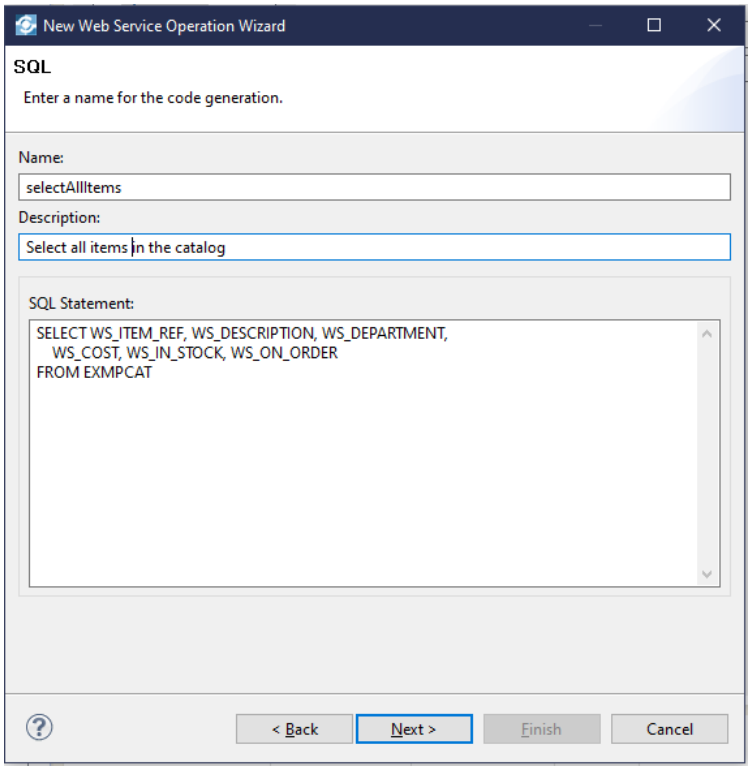


___10. On the *Browse Metadata* window, expand the *Data* folder then the *AVZS* folder and then the *Virtual Tables* folder to display the virtual table created in the previous section. Select *EXMPCAT* and press **Next** to continue.

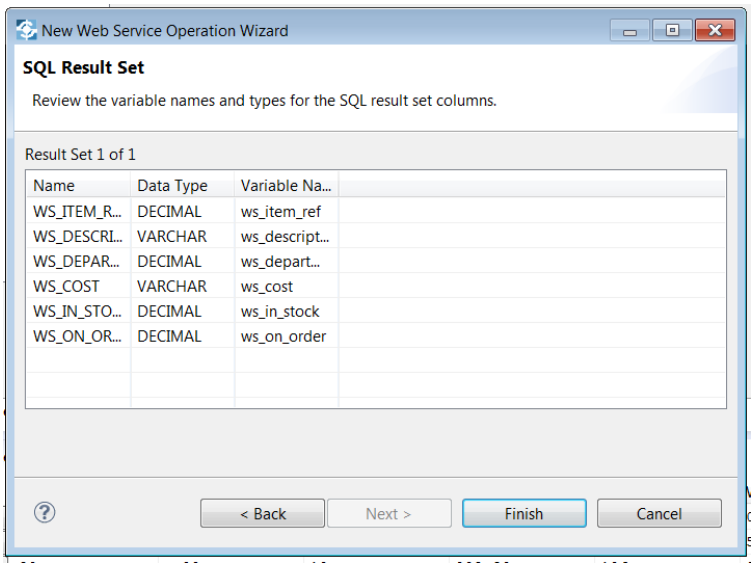


Mitch Johnson (mitchj@us.ibm.com)

___11. The default SQL statement will be displayed on the *SQL* window. Change the name of the operation to ***selectAllItems*** to indicate that this operation will retrieve all records from the VSAM data set. Click **Next** to continue.



___12. The next window to be displayed will show the results that will be returned when the operation is executed.

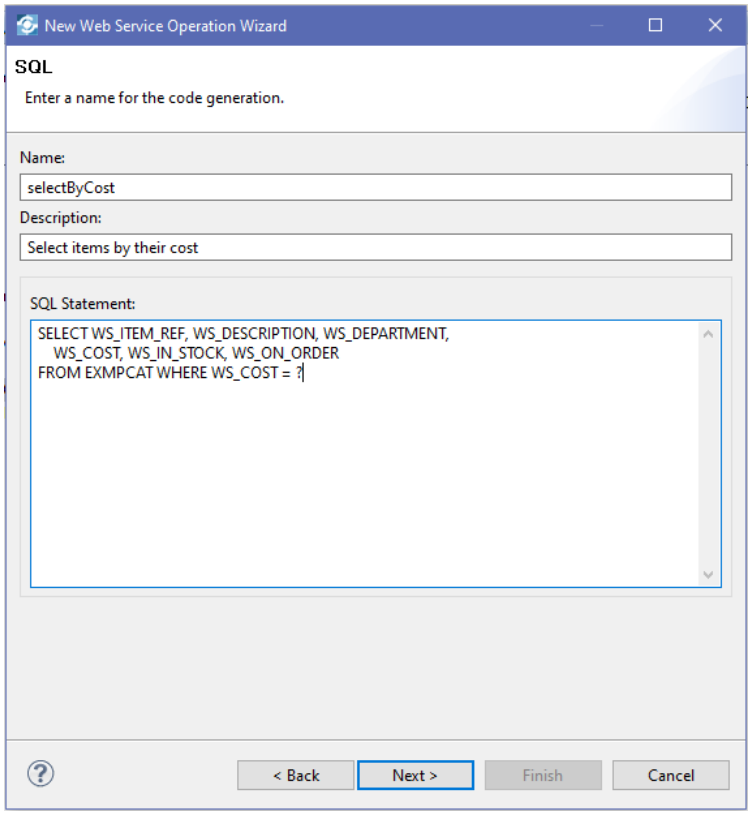


Tech-Tip: This is useful because we could have removed some of the columns on the previous windows. This display simply confirms which columns will be returned.

Mitch Johnson (mitchj@us.ibm.com)

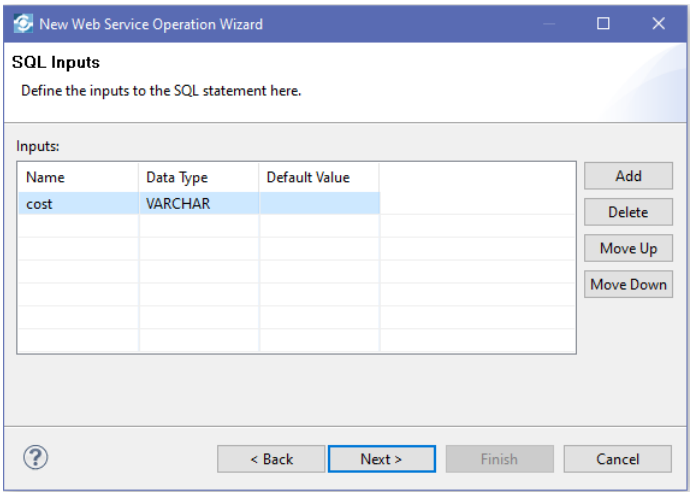
___13. Click **Finish** to continue.

___14. Use the *Create Operation* wizard under *EXMPCAT* to create a new operation. This operation should be named ***selectByCost*** and the *SQL Statement* should be changed by adding ***WHERE WS_COST = ?*** , see below:



Mitch Johnson (mitchj@us.ibm.com)

___15. Click **Next** to continue. Since a WHERE cause has been added with a variable, providing a value for this variable will be required. The next window to be displayed, *SQL Inputs*, will give us an opportunity to give meaningful names to this and other variables. On this window click on the value of variable name in the *Name* column and change the contents to ***cost***. Click **Next** to continue.

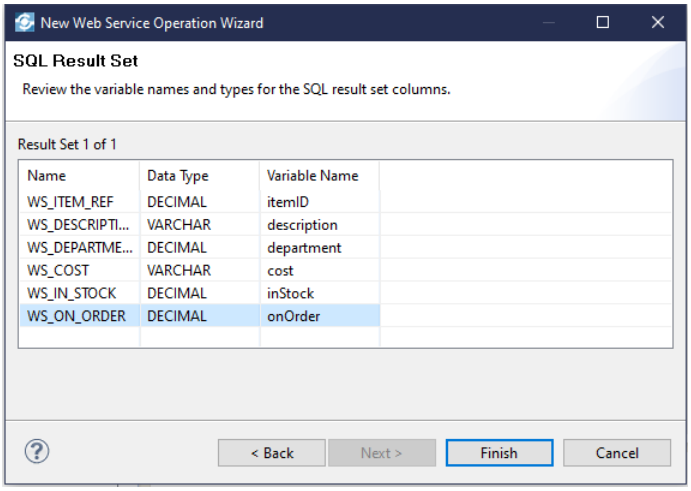


Tech-Tip: The numeric portions of the default SQL input field names, e.g. Input2, etc. will vary from what you may see. The number increments while using the tooling.

___16. The columns that will be returned are displayed on the SQL Result Set window. Change the *Variable Names* as shown below and then click **Finish** to continue.

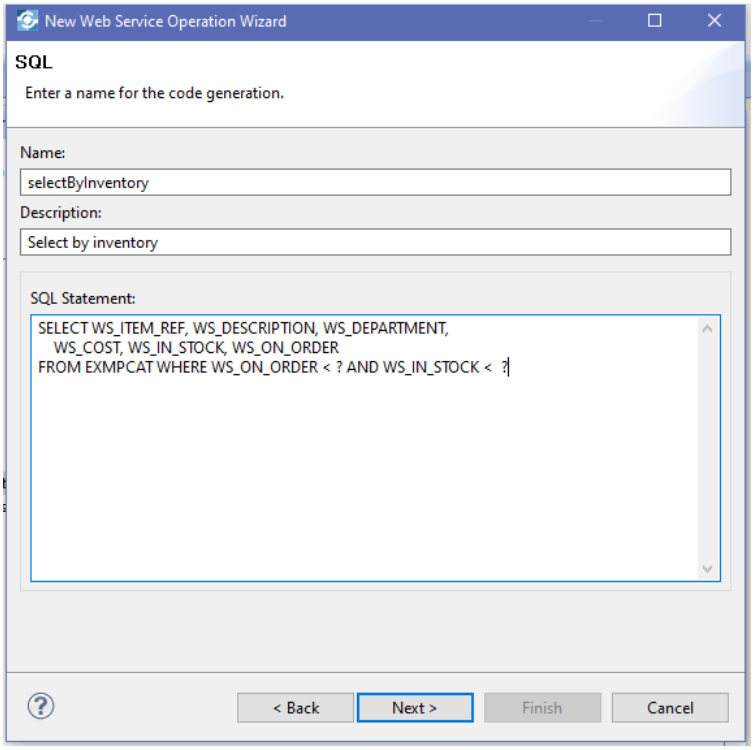
Change:

- *ws_item_ref* to *itemID*
- *ws_description* to *description*
- *ws_department* to *department*
- *ws-cost* to *cost*
- *ws_in_stock* to *inStock*
- *ws_on_order* to *onOrder*

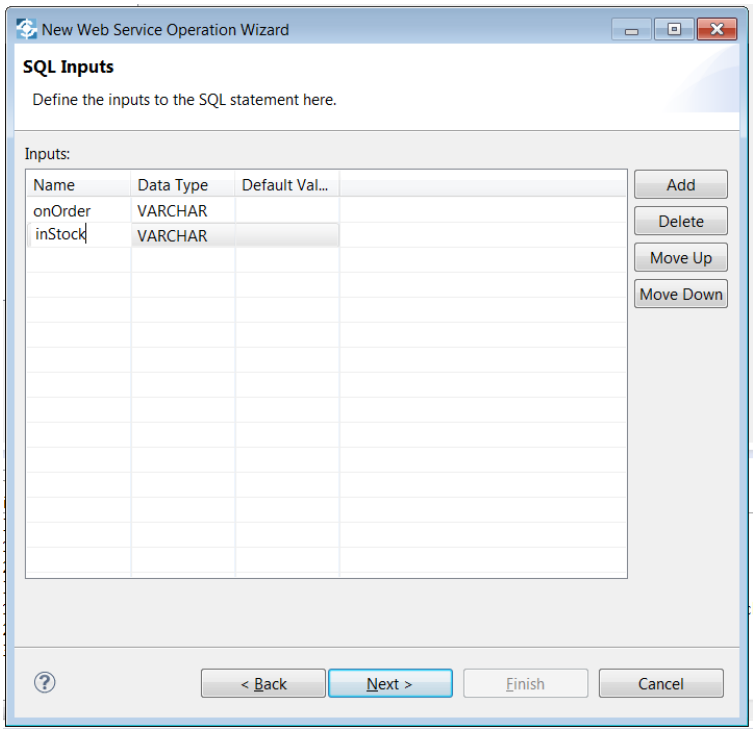


Mitch Johnson (mitchj@us.ibm.com)

___17. Use the *Create Operation* wizard under *EXMPCAT* to create a new operation. This operation should be named ***selectByInventory*** and the *SQL Statement* should be changed by adding ***WHERE WS_ON_ORDER < ? AND WS_IN_STOCK < ?*** , see below:



___18. Click **Next** to continue. Since a *WHERE* cause has been added with variables, providing meaningful names for these variables will be required. The next window to be displayed, *SQL Inputs*, will give us a chance to give meaningful names to these variables. On this window click on the values of variable name in the *Name* column and change the contents to ***onOrder*** and ***instock***. Click **Next** to continue.

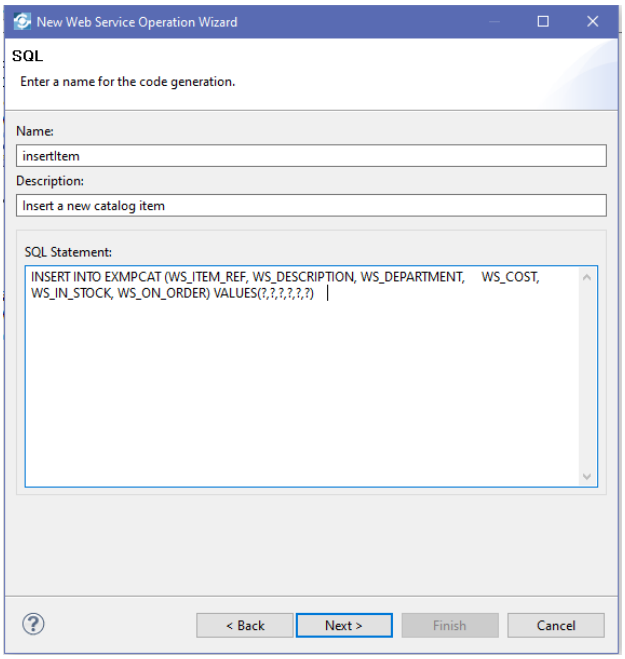


___19. The columns that will be returned are displayed on the *SQL Result Set* window. Change the Variable Names as shown below and then click **Finish** to continue.

Change:

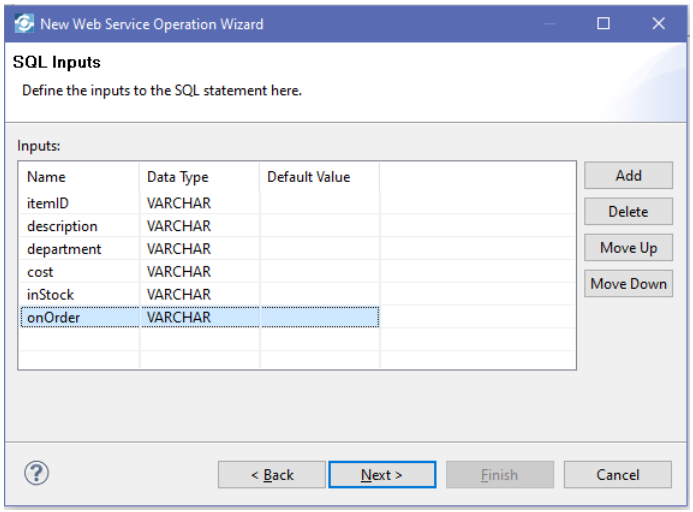
- *ws_item_ref* to *itemID*
- *ws_description* to *description*
- *ws_department* to *department*
- *ws-cost* to *cost*
- *ws_in_stock* to *inStock*
- *ws_on_order* to *onOrder*

___20. Use the *Create Operation* wizard under *EXMPCAT* to create a new operation. This operation should be named ***insertItem*** and the *SQL Statement* should be changed to ***INSERT INTO EXMPCAT (WS_ITEM_REF, WS_DESCRIPTION, WS_DEPARTMENT, WS_COST, WS_IN_STOCK, WS_ON_ORDER) VALUES(?,?,?,?,?,?)*** , see below:



___21. Click **Next** to continue. The next window to be displayed, *SQL Inputs*, will give us a chance to give meaningful names to these variables. On this window click on the value of variable name in the *Name* column and change the input field names to more meaningful values. For example, change:

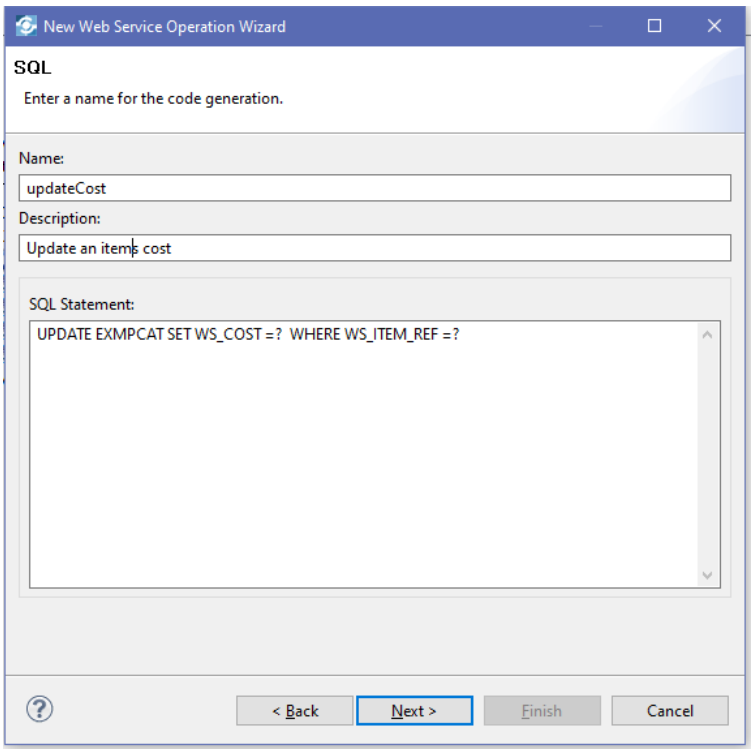
- *Input26* to *itemID*
- *Input27* to *description*
- *Input28* to *department*
- *Input29* to *cost*
- *Input30* to *inStock*
- *Input31* to *onOrder*



Mitch Johnson (mitchj@us.ibm.com)

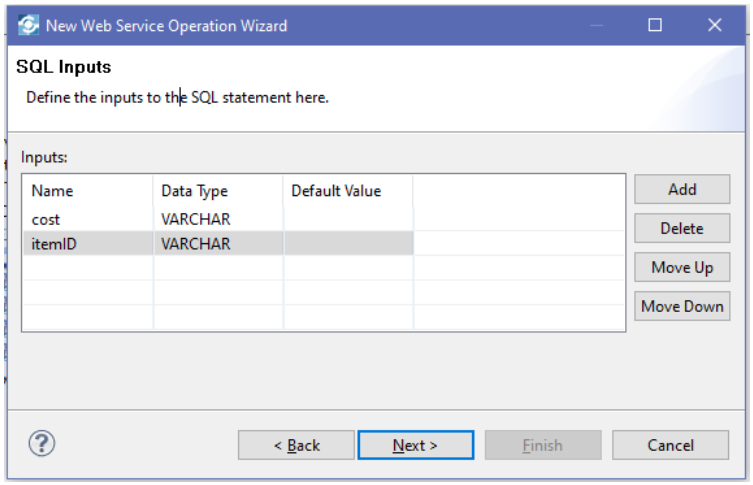
___22. Click **Next** and then **Finish** to continue.

___23. Use the *Create Operation* wizard under *EXMPCAT* to create a new operation. This operation should be named ***updateCost*** and the *SQL Statement* should be changed to ***UPDATE EXMPCAT SET WS_COST =? WHERE WS_ITEM_REF =?*** , see below:



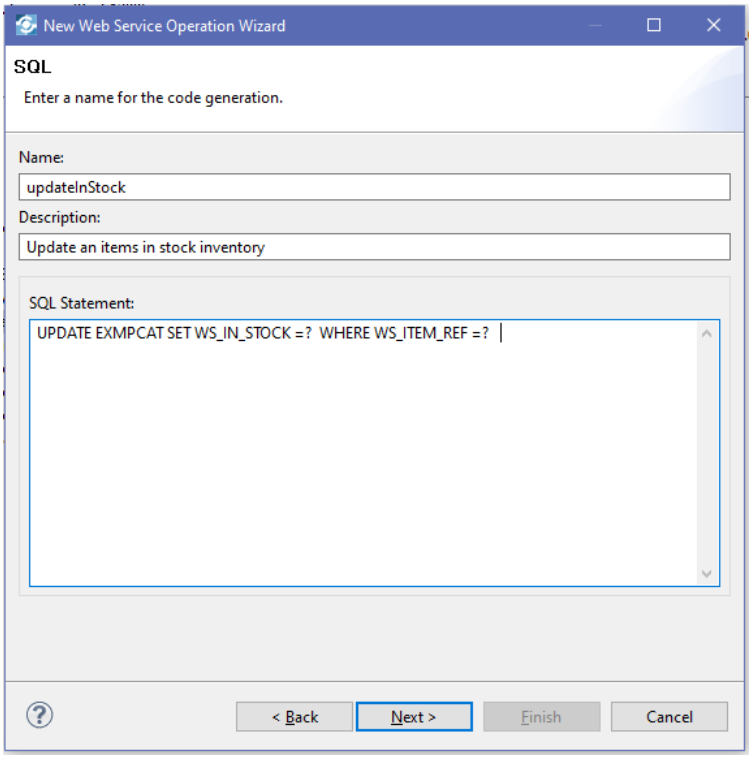
___24. Click **Next** to continue. The next window to be displayed, *SQL Inputs*, will give us a chance to give meaningful names to these variables. On this window click on the value of variable name in the *Name* column and change the input field names to more meaningful values. For example, change:

- *Input34* to *cost*
- *Input35* to *itemID*



___25. Click **Next** and then **Finish** to continue.

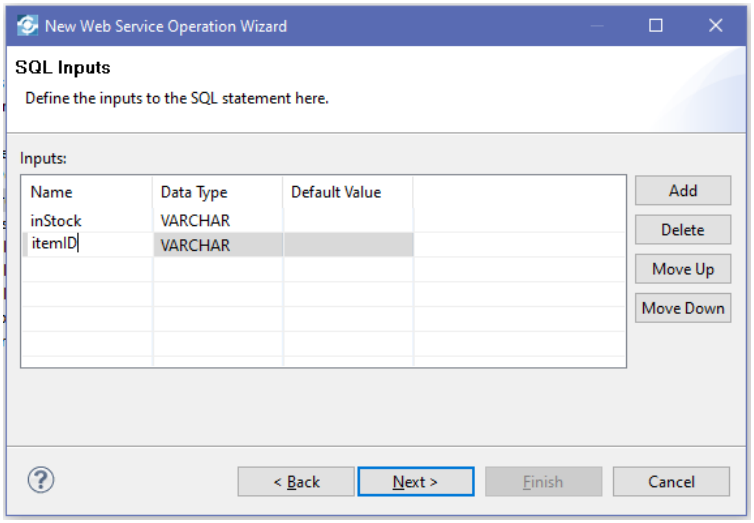
___26. Use the *Create Operation* wizard under *EXMPCAT* to create a new operation. This operation should be named ***updateInStock*** and the *SQL Statement* should be changed to ***UPDATE EXMPCAT SET WS_IN_STOCK =? WHERE WS_ITEM_REF =?*** , see below:



Mitch Johnson (mitchj@us.ibm.com)

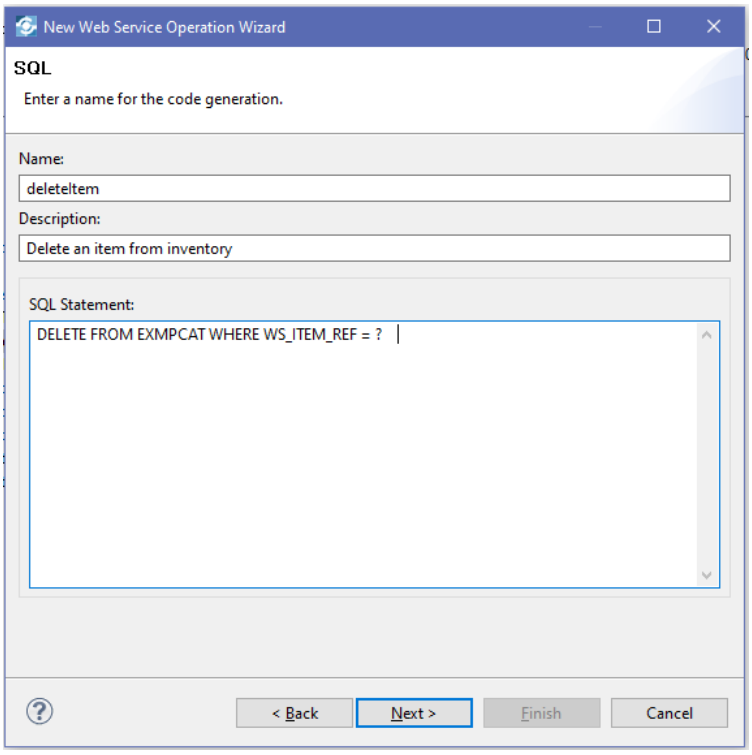
___27. Click **Next** to continue. The next window to be displayed, *SQL Inputs*, will give us a chance to give meaningful names to these variables. On this window click on the value of variable name in the *Name* column and change the input field names to more meaningful values. For example, change:

- *Input42* to *inStock*
- *Input43* to *itemID*



___28. Click **Next** and then **Finish** to continue.

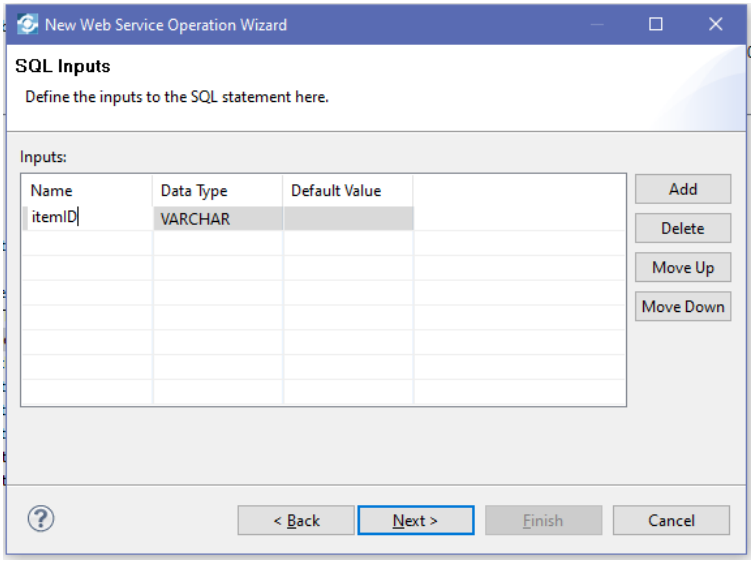
___29. Use the *Create Operation* wizard under *EXMPCAT* to create a new operation. This operation should be named ***deleteItem*** and the *SQL Statement* should be changed to ***DELETE FROM EXMPCAT WHERE WS_ITEM_REF = ?*** , see below:



Mitch Johnson (mitchj@us.ibm.com)

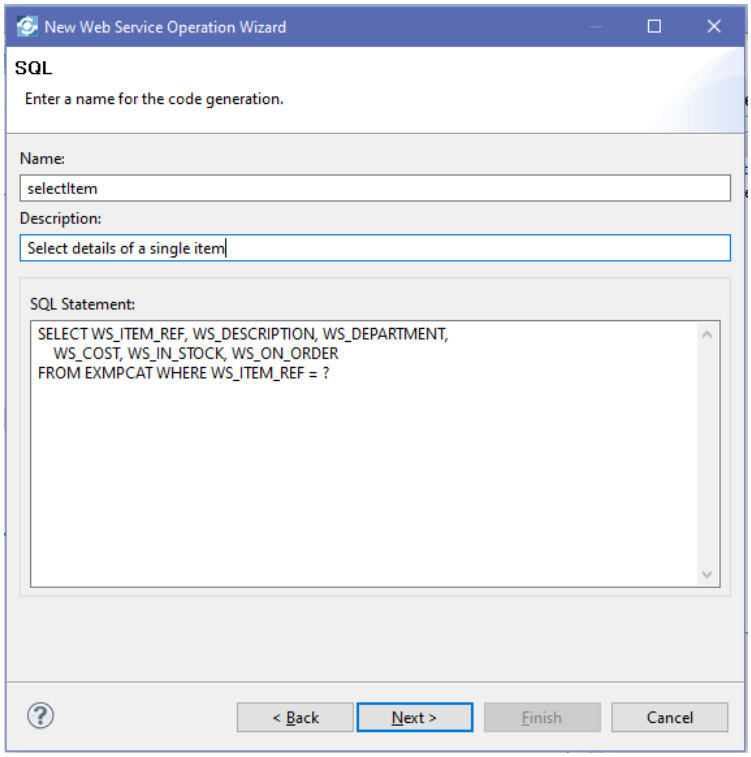
___30. Click **Next** to continue. The next window to be displayed, *SQL Inputs*, will give us a chance to give meaningful names to these variables. On this window click on the value of variable name in the *Name* column and change the input field names to more meaningful values. For example, change:

- *Input45* to *itemID*



___31. Click **Next** and then **Finish** to continue.

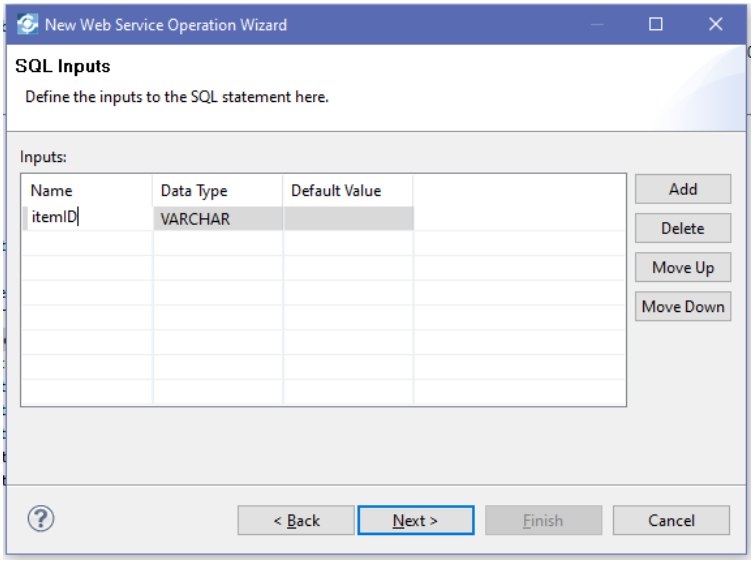
___32. Use the *Create Operation* wizard under *EXMPCAT* to create a new operation. This operation should be named ***selectItem*** and the *SQL Statement* should be adding ***WHERE WS_ITEM_REF = ?***, see below.



Mitch Johnson (mitchj@us.ibm.com)

___33. Click **Next** to continue. The next window to be displayed, *SQL Inputs*, will give us an opportunity to give meaningful names to these variables. On this window click on the value of variable name in the *Name* column and change the input field name to a more meaningful value. For example, change:

- *Input47* to *itemID*



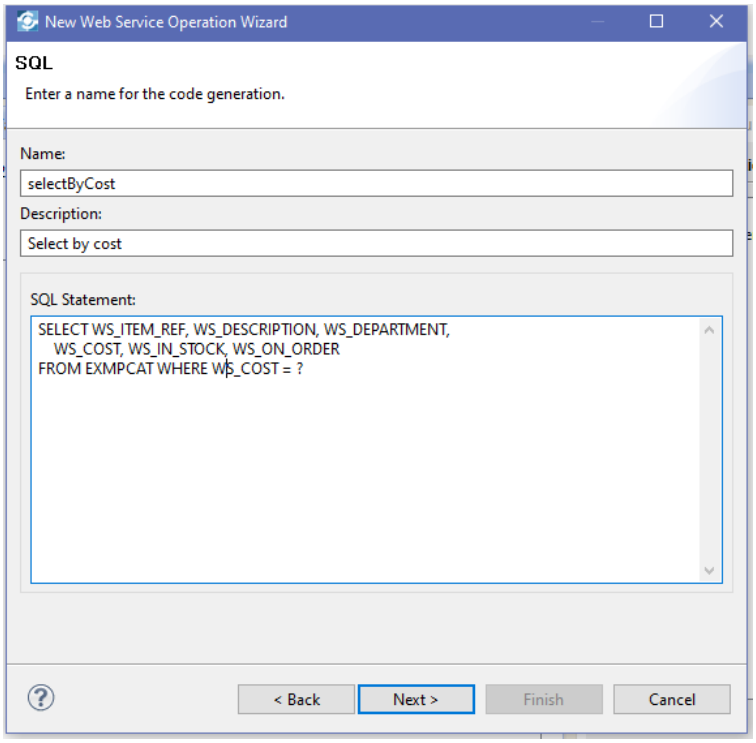
___34. Click **Next** to continue. The columns that will be returned are displayed on the SQL Result Set window. Change the Variable Names as shown below.

Change:

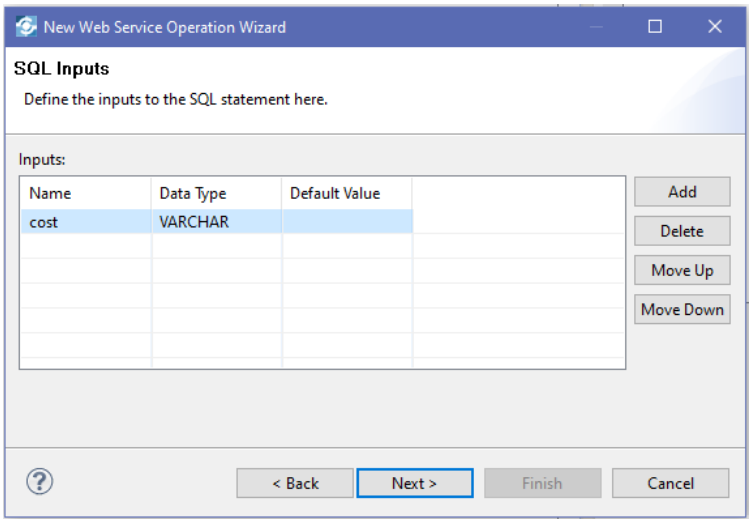
- *ws_item_ref* to *itemIDws_description* to *description*
- *ws_department* to *department*
- *ws-cost* to *cost*
- *ws_in_stock* to *inStock*
- *ws_on_order* to *onOrder*

___35. Click **Finish** to continue.

___36. Use the *Create Operation* wizard under *EXMPCAT* to create a new operation. This operation should be named ***selectByCost*** and the *SQL Statement* should be changed by adding ***WHERE WS_COST = ?*** , see below:



___37. Click **Next** to continue. Since a *WHERE* cause has been added with a variable, providing meaningful names for this variable will be required. The next window to be displayed, *SQL Inputs*, will give us a chance to give a meaningful name to this variable. On this window click on the values of variable name in the *Name* column and change the contents to ***cost***. Click **Next** to continue.



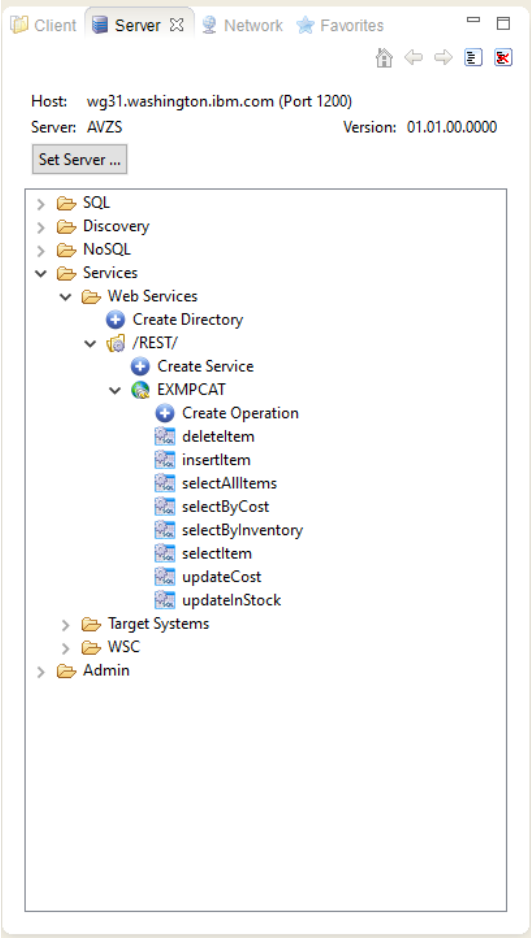
___38. The columns that will be returned are displayed on the *SQL Result Set* window. Change the Variable Names as shown below and then click **Finish** to continue.

Change:

- *ws_item_ref* to *itemID*
- *ws_description* to *description*
- *ws_department* to *department*
- *ws-cost* to *cost*
- *ws_in_stock* to *inStock*
- *ws_on_order* to *onOrder*

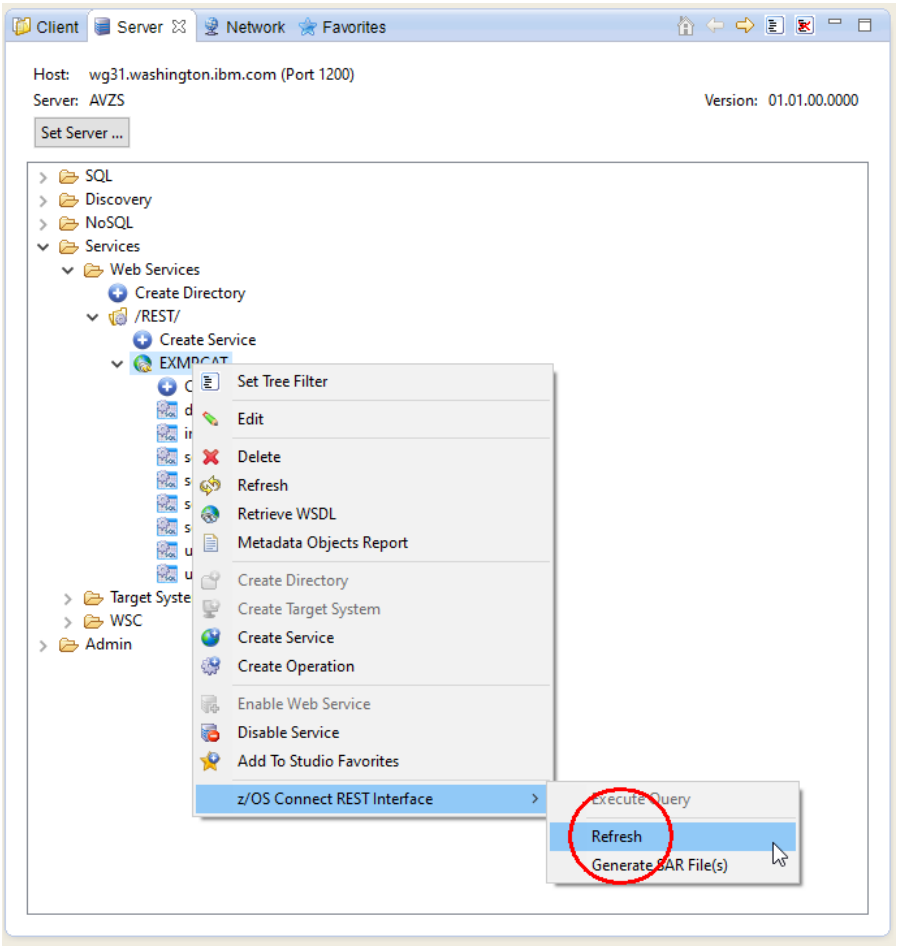
Mitch Johnson (mitchj@us.ibm.com)

All eight operations should now be displayed in as services in the *Server* view.



Deploy the DVM services

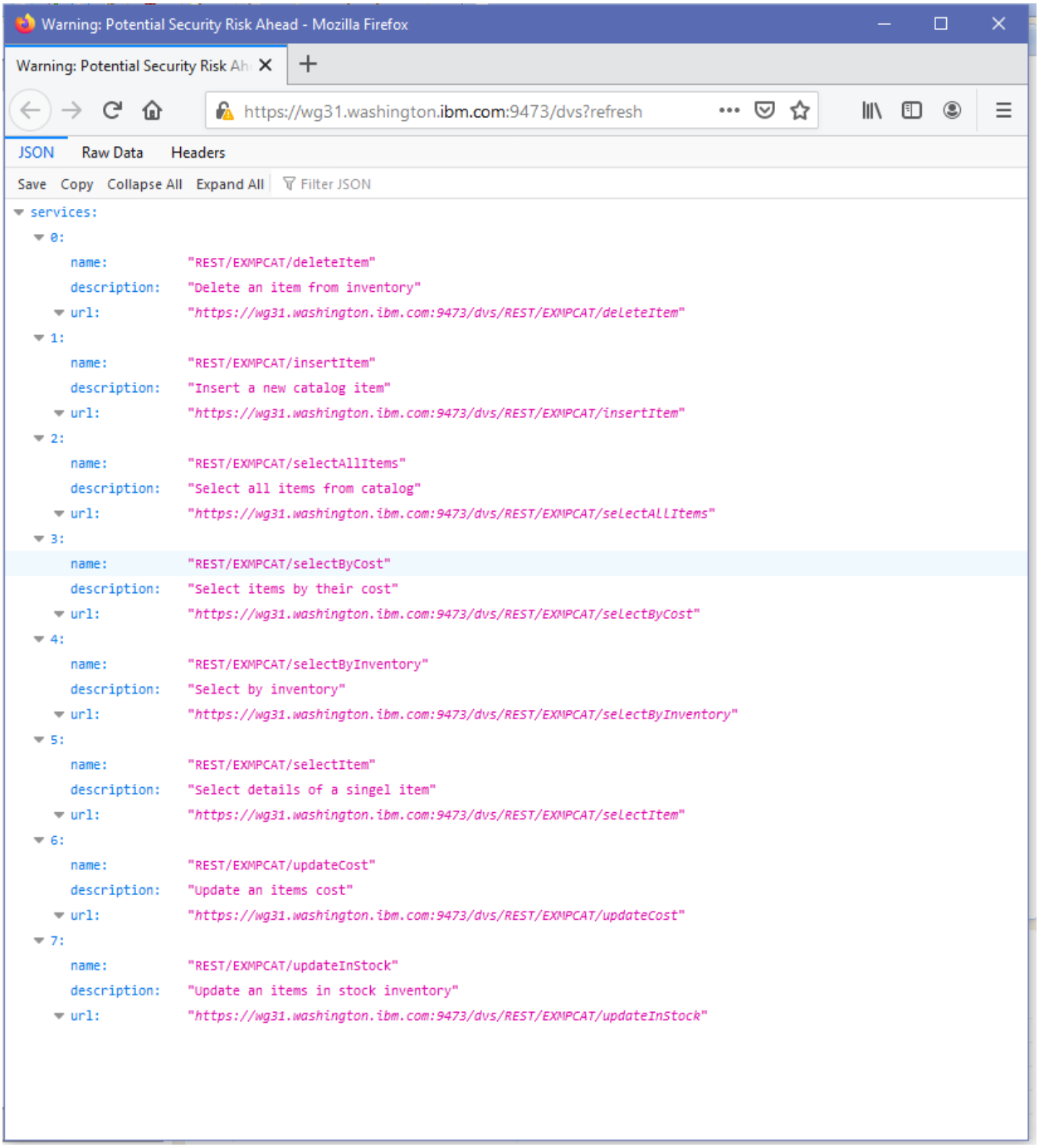
___1. These operations need to be deployed to z/OS Connect server. Select *EXMPCAT* and right mouse button click. Select the *z/OS Connect REST Interface* option then the *Refresh* option. This will install the selected operation into the z/OS Connect EE server as services.



Tech Tip: Operations can be deployed individually by selecting the specific operation and right mouse button clicking and selecting **Refresh**.

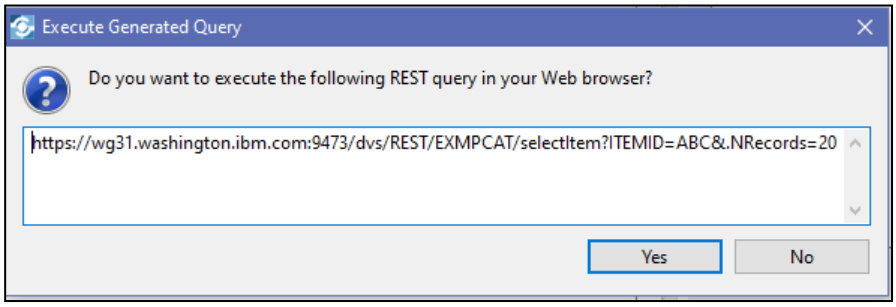
Tech Tip: You may be challenged by Firefox because the digital certificate used by the Liberty z/OS server is self-signed Click the **Advanced** button to continue. Scroll down and then click on the **Accept the Risk and Continue** button. Next you may see a prompt you for a userid and password. If you do see the prompt, enter the username ***USER1*** and password **USER1** and click **OK**.

___2. When finished all the operations should be displayed as services in the z/OS Connect EE server by entering URL ***https://wg31.washington.ibm.com:9473/dvs***.

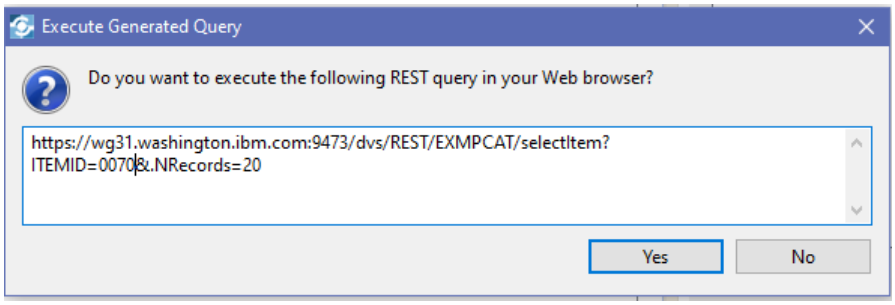


___3. A subset of these z/OS Connect EE services can now be tested using the DVM Data Manager Studio (only the services that do selects) Select the *selectItem* operation and right mouse button click. Select the *z/OS Connect REST Interface option* then the *Execute Query* option.

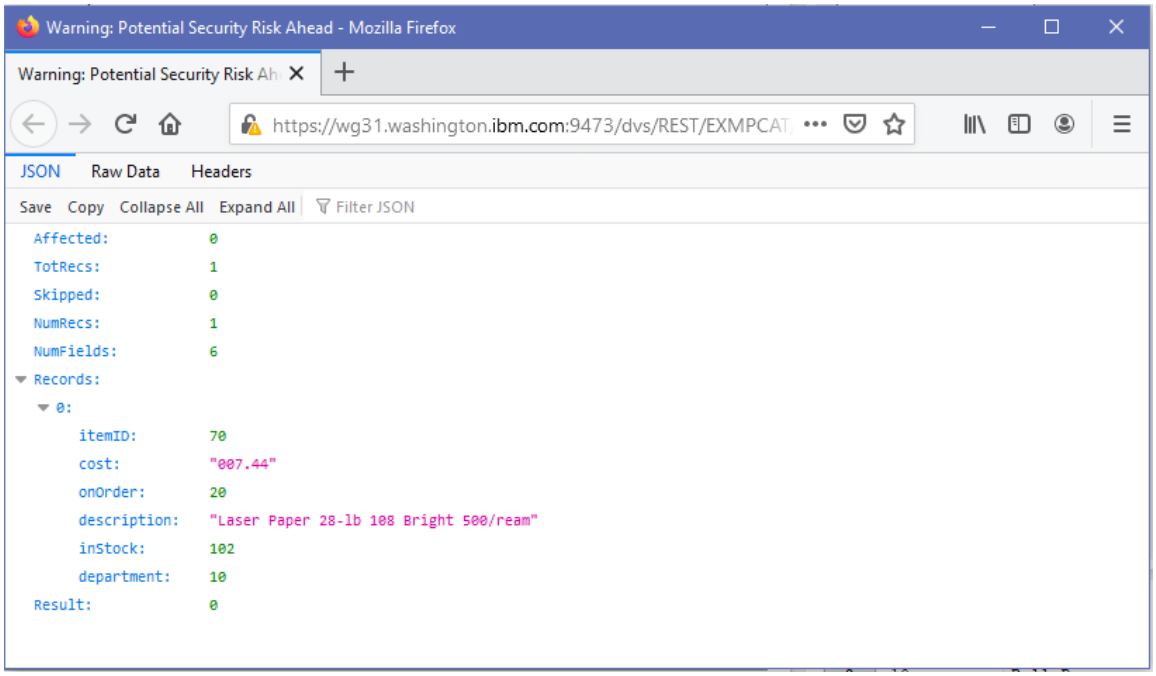
___4. This pop-up window should be displayed.



___5. Change the string *ITEMID=ABC to* ***ITEMID=0070***



___6. Click **Yes** and a web browser tab should be opened with results like these.



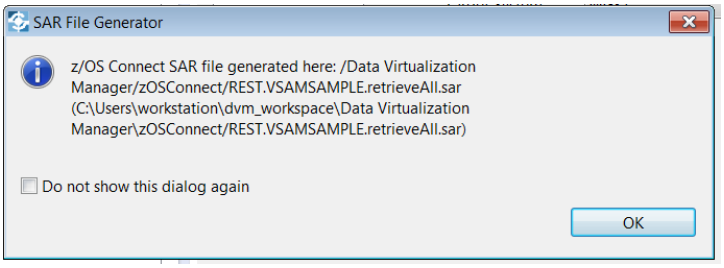
Mitch Johnson (mitchj@us.ibm.com)

Tech-Tip: The above results show some new fields in the response messages. There meanings are provided below:

Affected:	The number of records deleted, updated or inserted by this request.
TotRecs:	The number of records found.
Skipped:	The number of records skipped
NumRecs:	The number of records returned.
NumFields:	The number of fields returned for each record.

Export the DVM services as z/OS Connect service archive files

Finally, the Service Archive (SAR) files need to be exported from the DVM Studio for use in the z/OS Connect EE API Editor. Select *EXMPCAT* and right mouse button click. Select the *z/OS Connect REST Interface* option then the *Generate SAR File(s)* option. Click **OK** to continue.



Tech-Tip: The directory where the SAR file is exported may be different on your system. Make a note of this directory name so you will know from where to import the SAR file later. On some images, this directory will be *C:\Users\administrator\dvm_workspace\Data Virtualization Manager\zOSConnect.*

This exports the SAR files to a subdirectory in the DVM Toolkit's workspace directory, e.g *C:\Users\workstation\dvm_workspace\Data Virtualization Manager\zOSConnect*. This pop-up will be repeated for each operation. This directory will be referenced in a latter section of this exercise.

Using DVM services in a z/OS Connect EE API

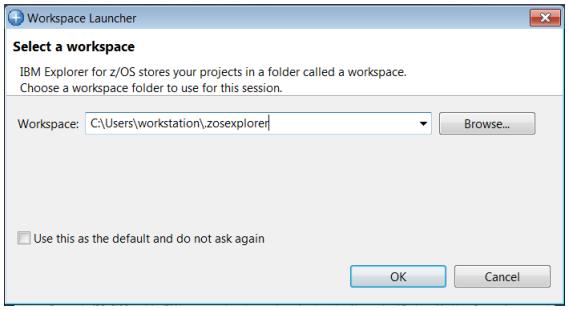
Connect to a z/OS Connect EE Server

Begin by establishing a connection to DVM z/OS Connect server from IBM z/OS Explorer.

___1. On the workstation desktop, locate the *z/OS Explorer* icon and double click on it to open the Explorer.

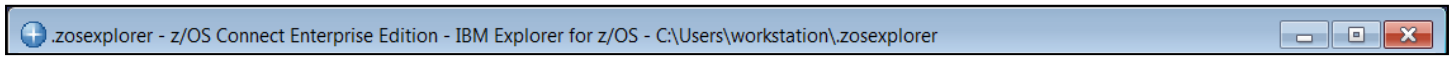
> **Tech-Tip:** Windows desktop tools can be opened either by double clicking the icon or by selecting the icon and right mouse button clicking and then selecting the *Open* option.

___2. You will be prompted for a workspace:



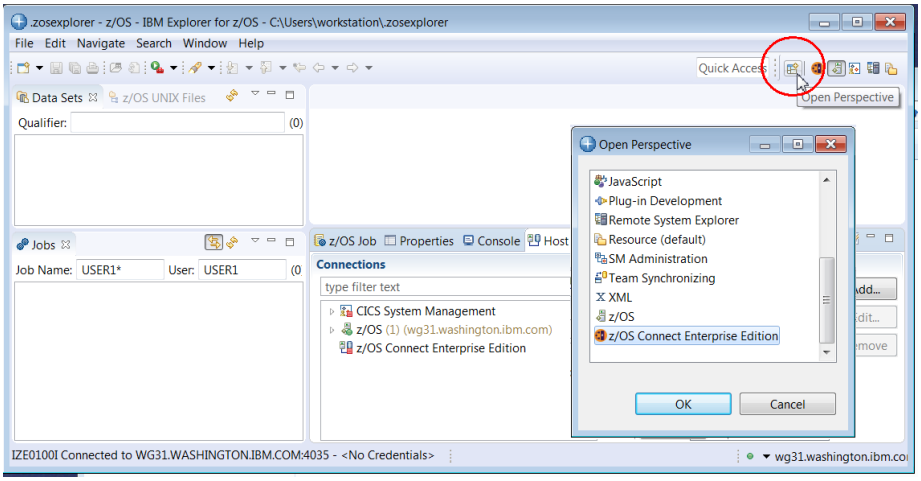
Take the default value by clicking **OK**.

___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:



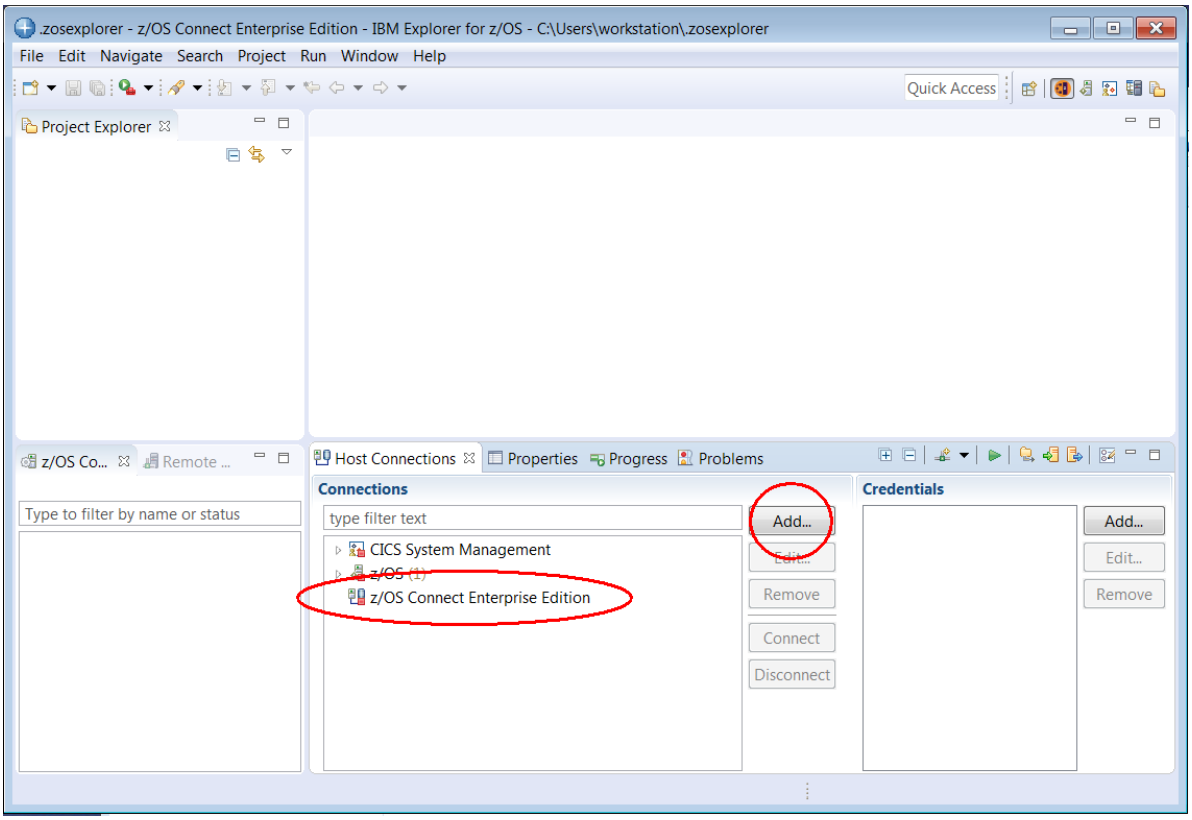
N.B. If a *Welcome* screen is displayed then click the white X beside *Welcome* to close this view.

___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.



Mitch Johnson (mitchj@us.ibm.com)

___5. To add a connection to the z/OS Connect Server, select *z/OS Connect Enterprise Edition* connection in the *Host connection*s tab in the lower view and then click the **Add** button.



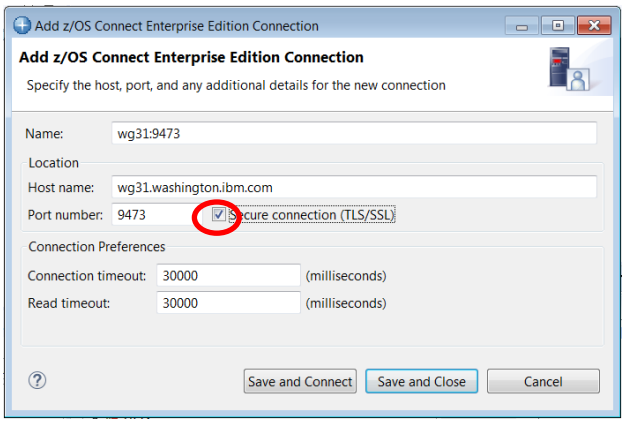
Tech-Tip: Eclipse based development tools like z/OS Explorer; provide a graphical interface consisting of multiple views within a single window.

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.

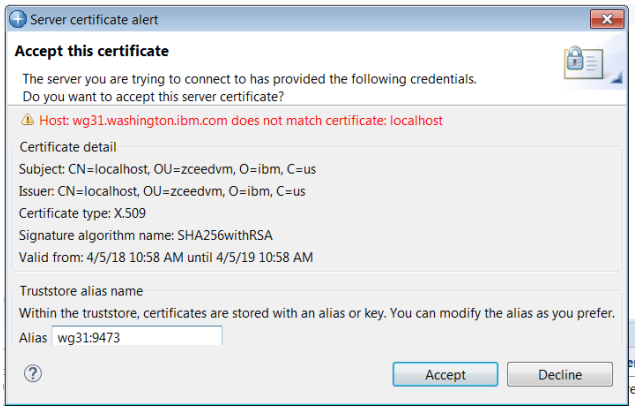
Eclipse based tools also can display multiple views based on the current role of the user. In this context, a window is known as a perspective. The contents (or views) of a perspective are based on the role the user, i.e., developer or administrator.

___6. In the pop-up list displayed, select *z/OS Connect Enterprise Edition* and on the *Add z/OS Connect Enterprise Edition Connection* window enter ***wg31.washington.ibm.com*** for the *Host name*, 9473 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 ***USER1*** and for *Password or Passphrase* enter USER's password. Click **OK** to continue.

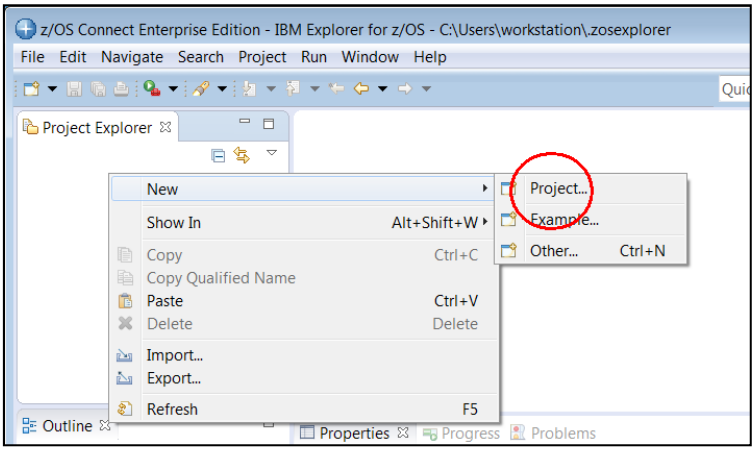
___8. Click the **Accept** button on the *Server certificate alert – Accept this certificate* screen. You may be presented with another prompt for a userid and password, enter ***USER1*** and USER1's password again.



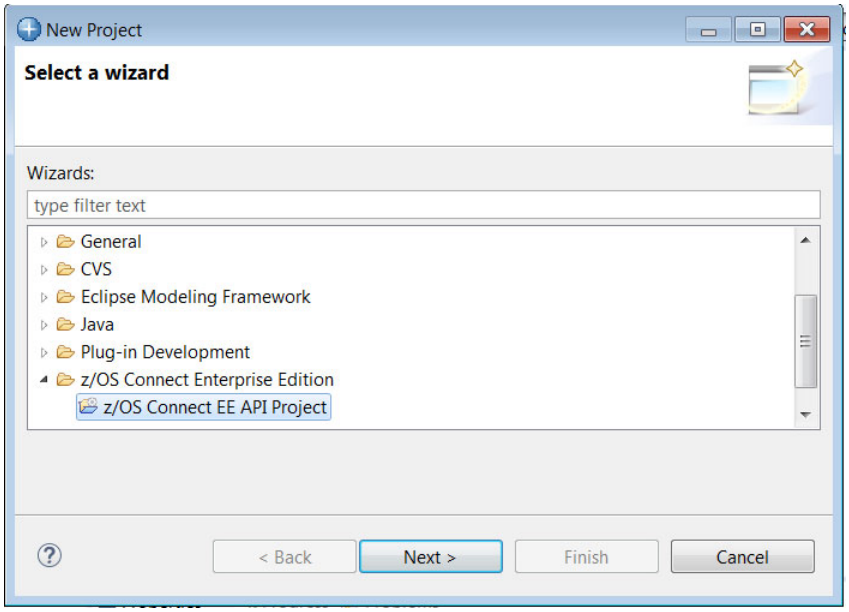
___9. The status icon beside *wg31:9473* should now be a green circle with a lock. This shows that a secure connection has been established between the z/OS Explorer and the z/OS Connect server. A red box indicates that no connection exists.

Create the VSAMCICS DVM 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*:

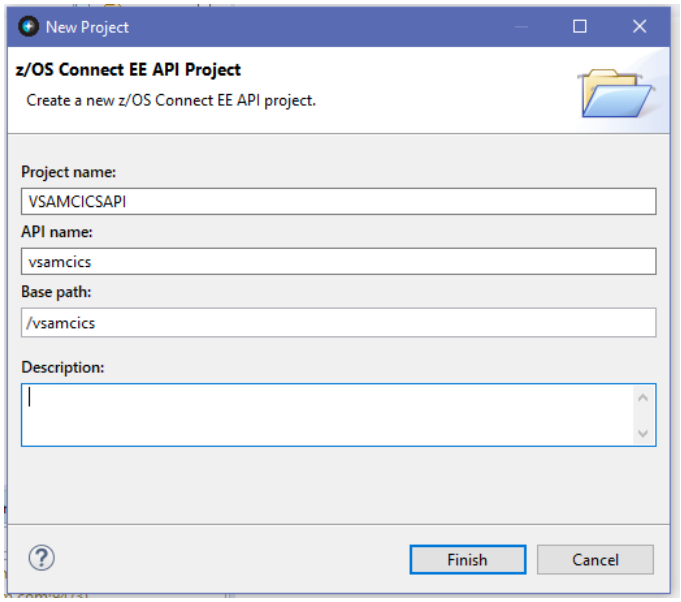


___2. In the *New Project* window, scroll down and open the *z/OS Connect Enterprise Edition* folder and select *z/OS Connect EE API Project* and then click the **Next** button.



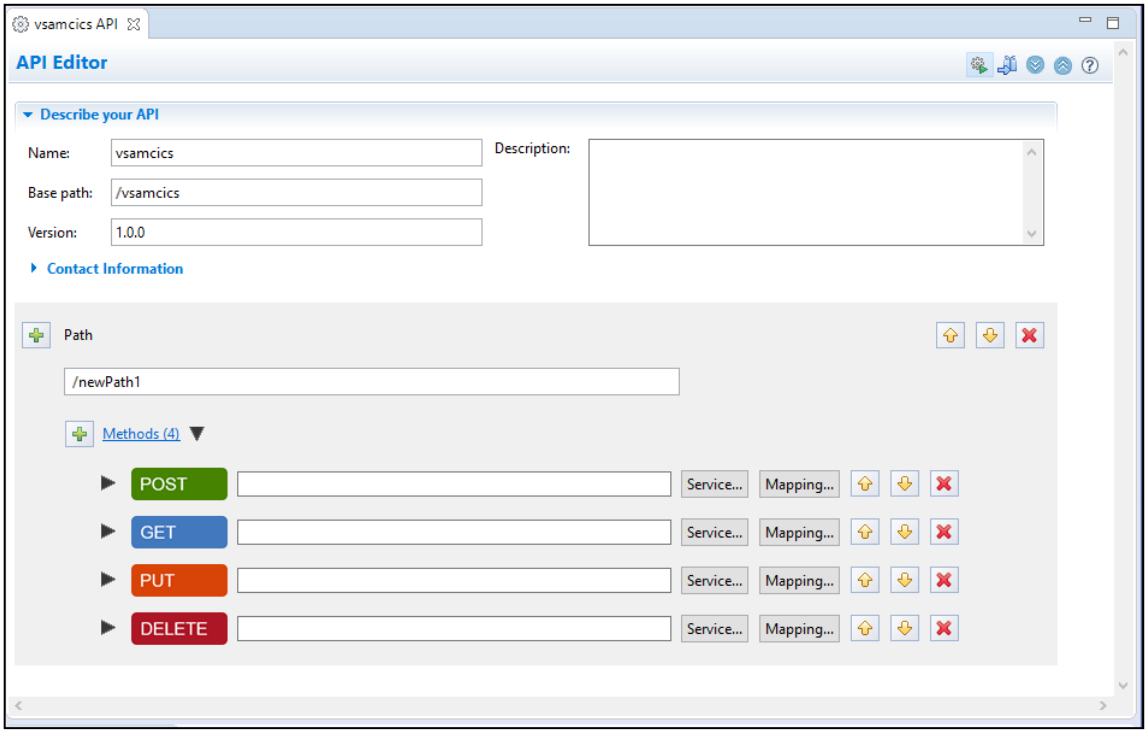
Mitch Johnson (mitchj@us.ibm.com)

___3. Enter ***VSAMCICSAPI*** for the *Project name.* Be sure the *API name* is set to ***vsamcics*** and the *Base path is* set to ***/vsamcics.*** Click **Finish** to continue.



Note: The Base path name of */vsamcics* is used to distinguish a request for this API from other APIs in the same server. It can be any value as long as the value is unique within the server. The same is true of any sub path names added to the base path. Sub path names are used to distinguish one service from another within an API.

___4. You should now see something like the view below. The view may need to be adjusted by dragging the view boundary lines.

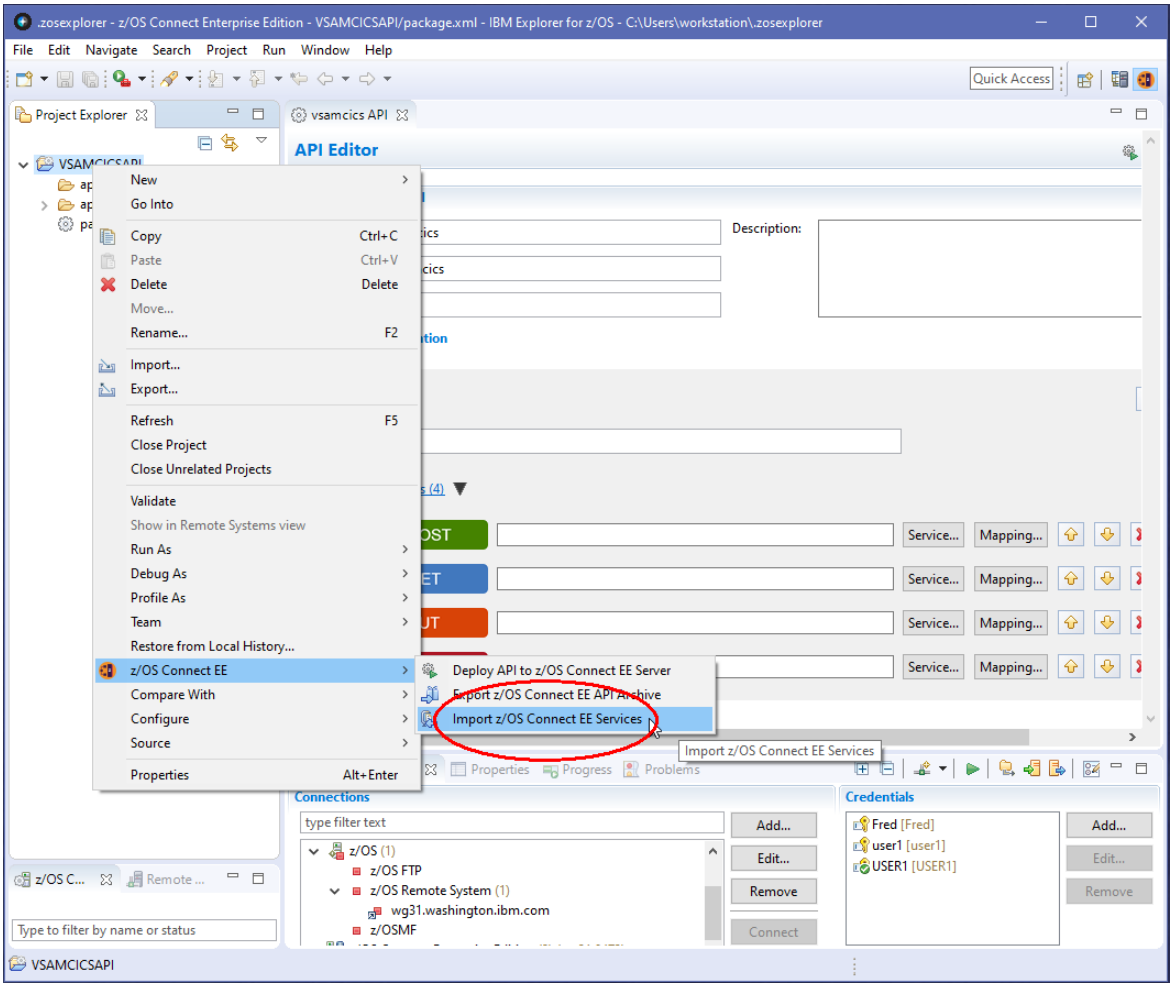


Mitch Johnson (mitchj@us.ibm.com)

Tech-Tip: If the API Editor view is closed, it can be reopened by double clicking the *package.xml* file in the API project.

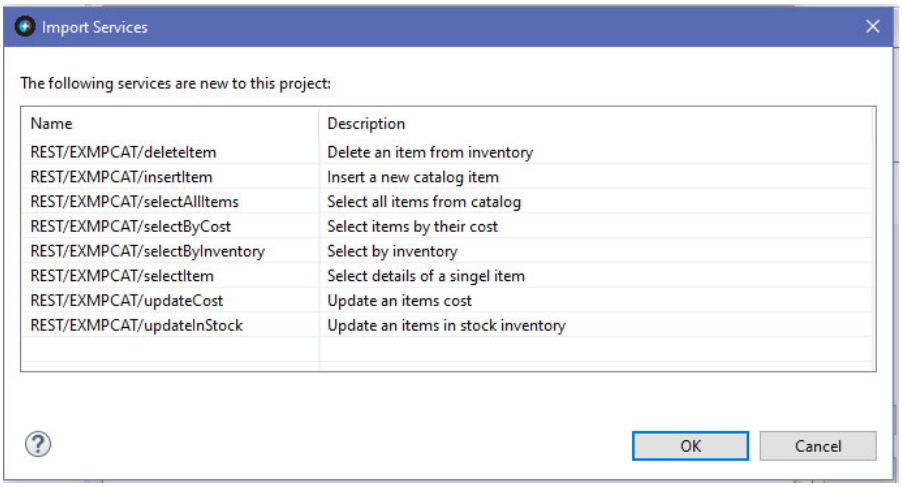
Import the SAR files generated by the DVM Studio

___1. In the z/OS Explorer in the *z/OS Connect Enterprise Edition* perspective in the the *Project Explorer* view (upper left), right-click on the *VSAMCICSAPI* project, then select *z/OS Connect EE* and then *Import z/OS Connect EE Services* (see below):

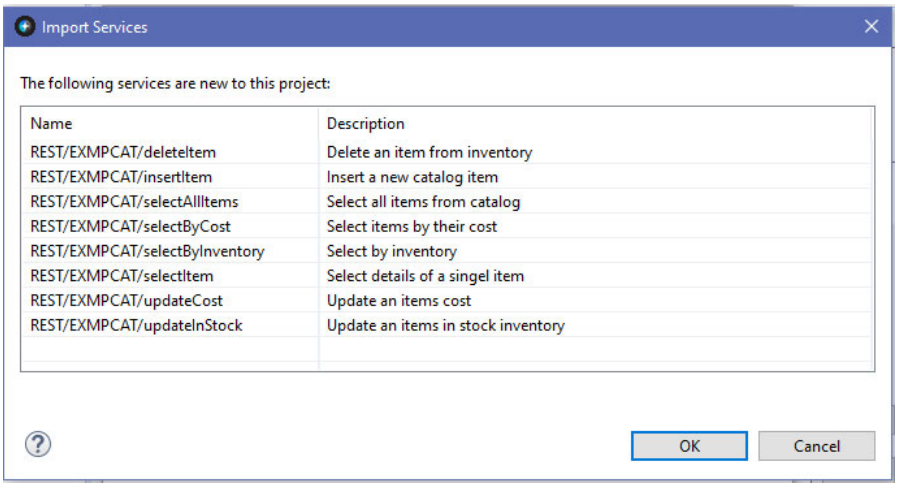


___2. In the *Import z/OS Connect EE Services* window, click on the **File System** button and navigate to directory *C:\Users\workstation\dvm_workspace\Data Virtualization Manager\zOSConnect*. Select all the SAR files and click on the **Open** button:

> **Tech-Tip:** Remember from step 7 on page 34, the directory where the SAR file is to imported from may be different on your system. On some images, this directory will be *C:\Users\administrator\dvm_workspace\Data Virtualization Manager\zOSConnect.*

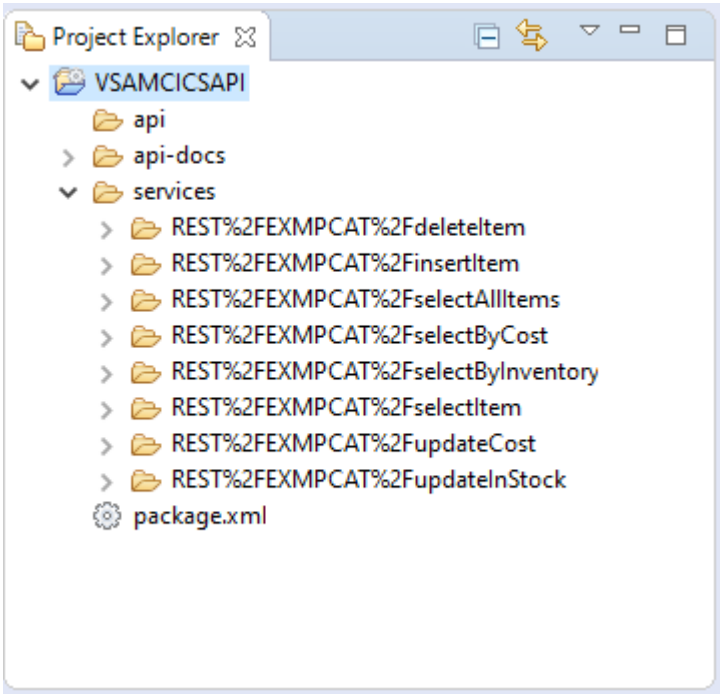


___3. The service archive files should appear in the *Import Services* window. Click the **OK** button twice to import them into the workspace.



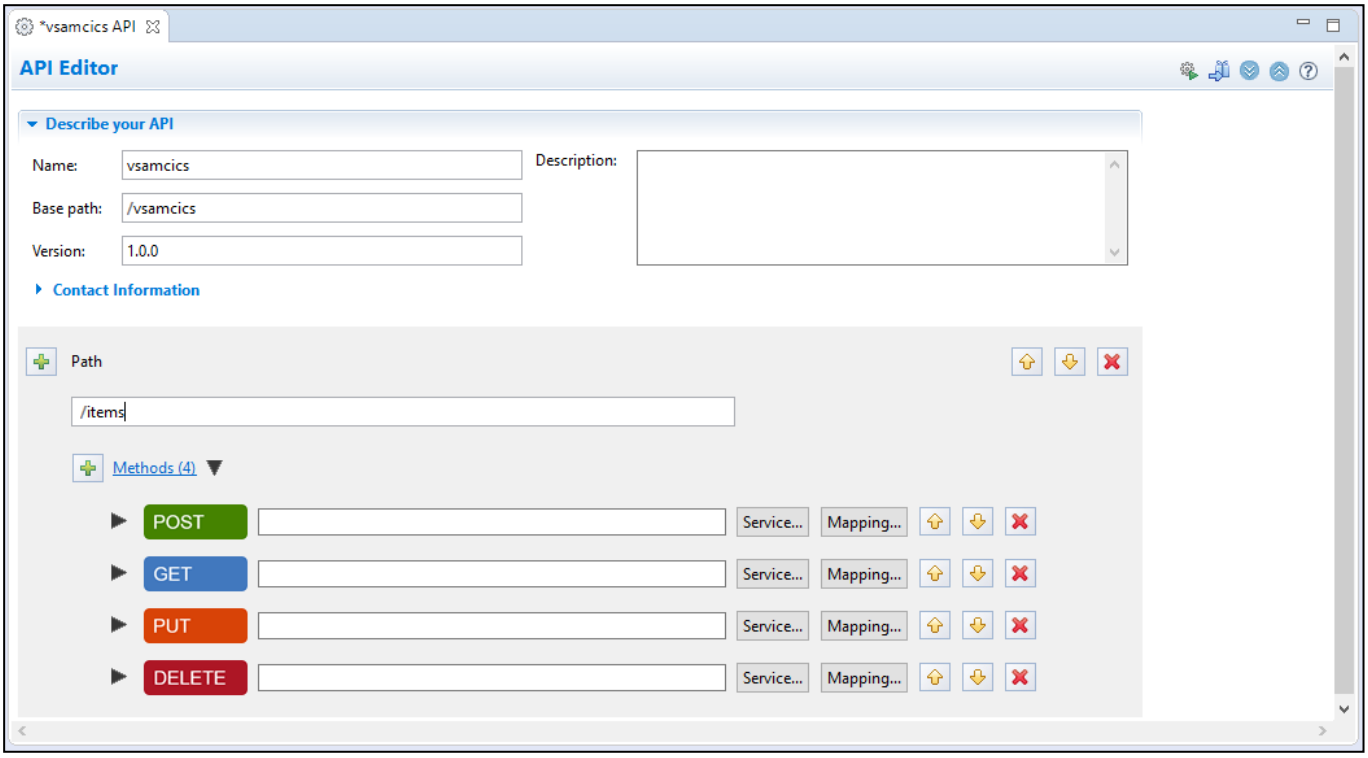
Mitch Johnson (mitchj@us.ibm.com)

___4. In the *Project Explorer* view (upper left), expand the *services* folder to see the the imported service:



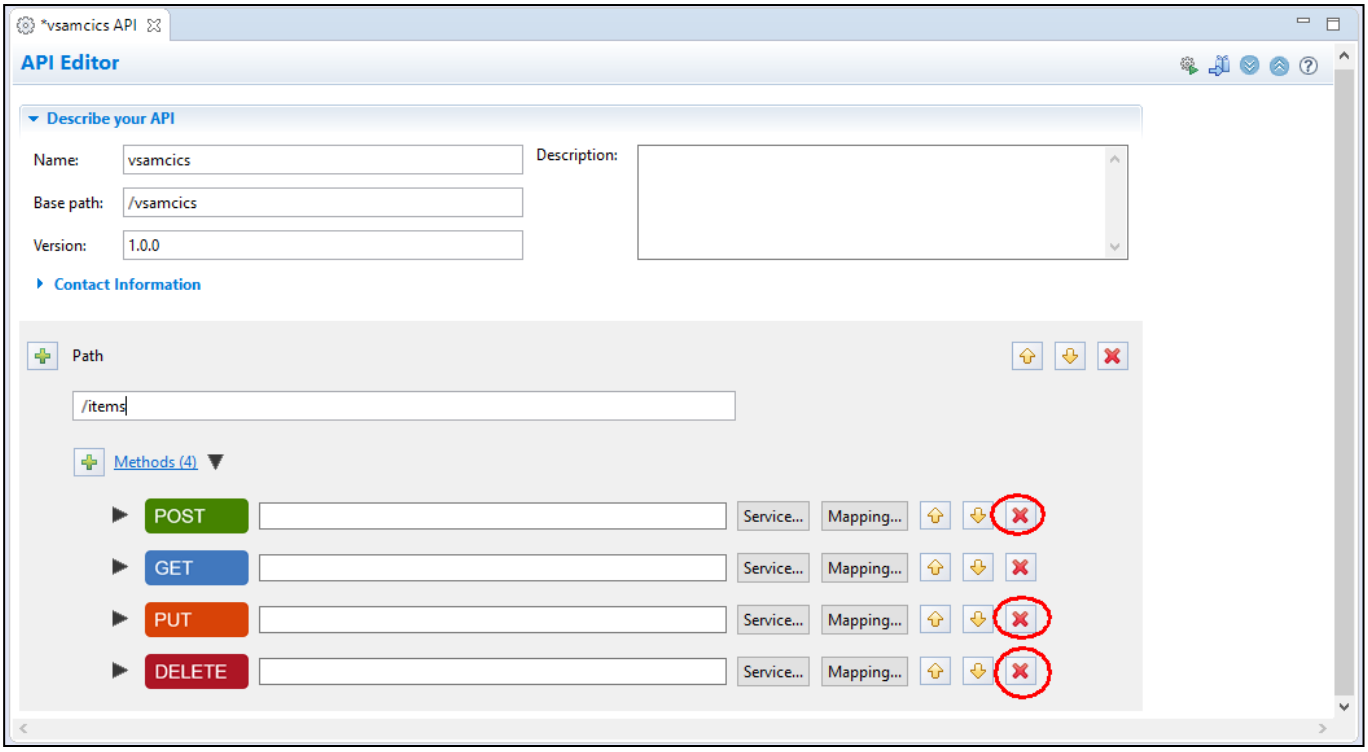
Compose an API for the DVM Rest Services

___1. Start by entering a *Path* of ***/items*** in the *z/OS Connect EE API Editor* view as shown below:



Mitch Johnson (mitchj@us.ibm.com)

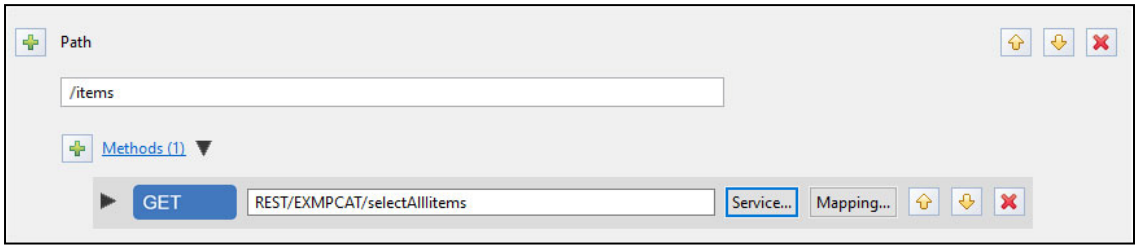
___2. The initial API to be added will be when no path or query parameter will be required, the supported HTTP methods will only be the **GET** method. Remove the **POST**, **PUT** and **DELETE** methods by clicking the red *X* icon to the right of each method.



Mitch Johnson (mitchj@us.ibm.com)

___3. That should leave you with just the **GET** method.

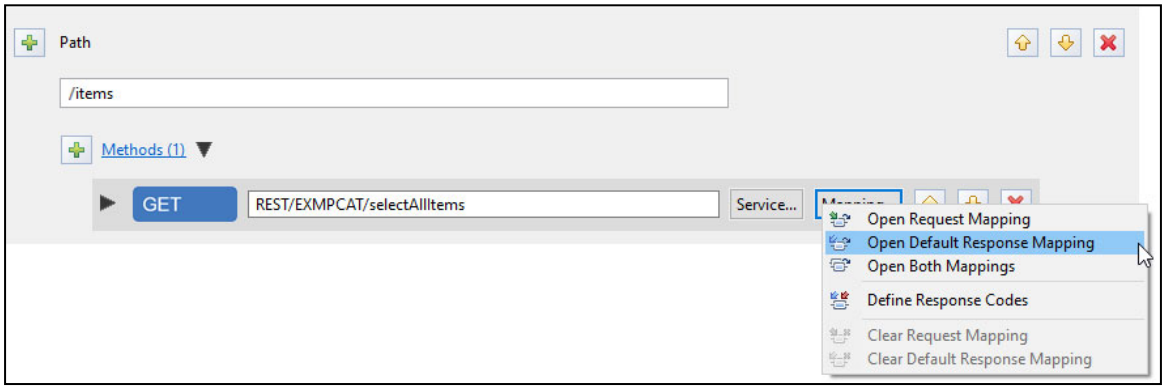
___4. Click on the **Service** button to the right of the **GET** method. Then select the *REST/EXMPCAT/selectAllItems s*ervice from the list of services and click **OK**. This will populate the field to the right of the method.



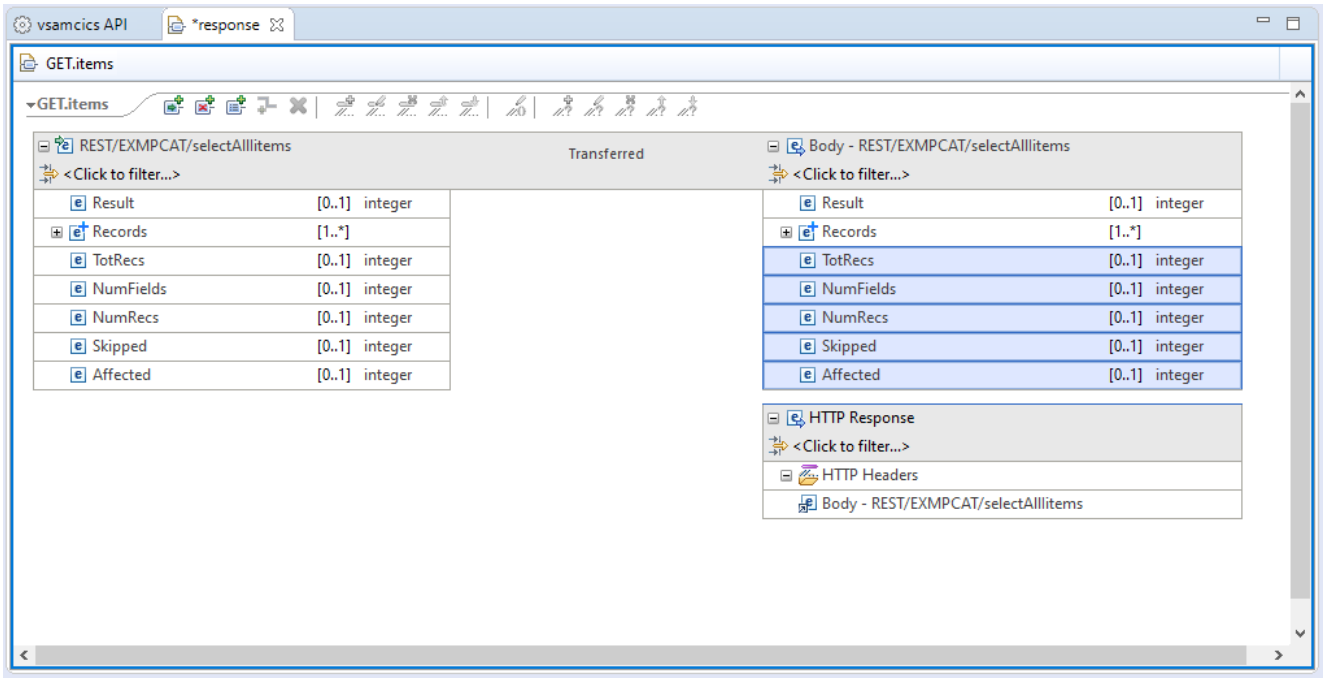
___5. Save the changes so far by using the key sequence **Ctrl-S.**

> **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.

___6. Next, click on the **Mapping** button beside the **GET** method and then select *Open Default Response Mapping*:



___7. Use the left mouse button and draw a dotted line box that **fully** includes the *TotRecs, NumFields, NumRecs, Skipped* and *Affected* fields. When you release the button, these fields should be selected (the background should be blue).

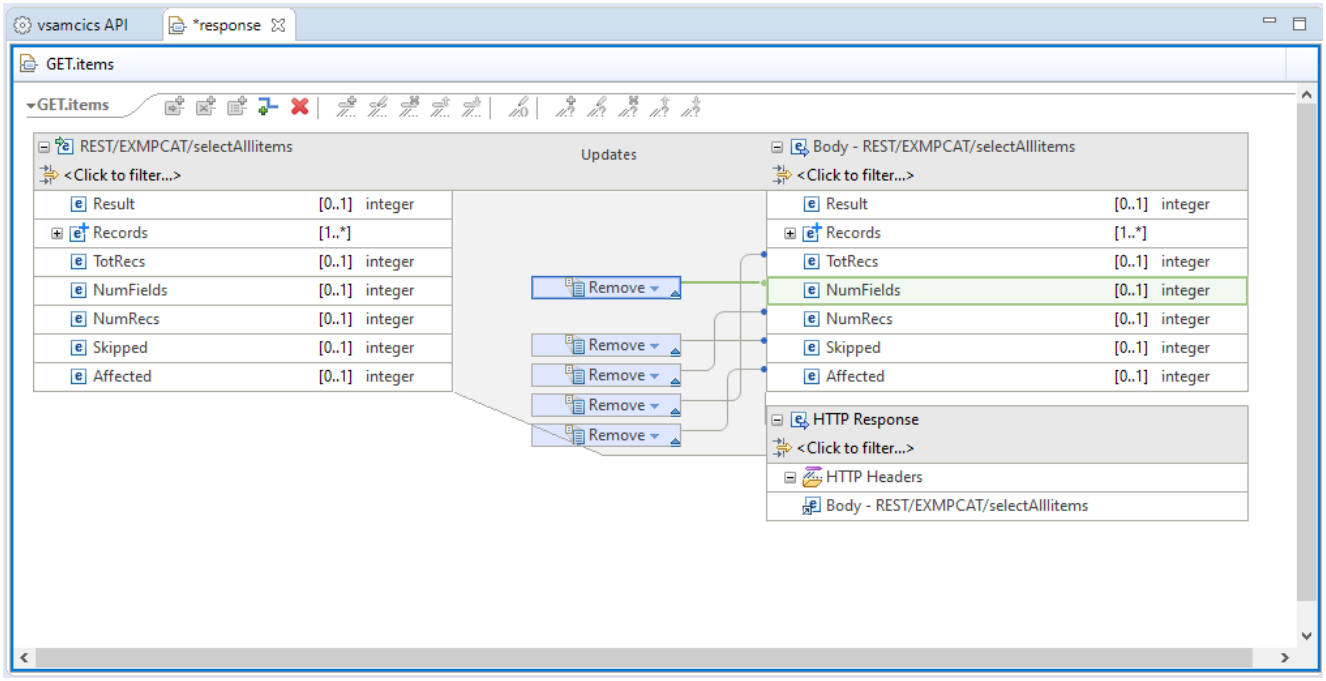


Tech-Tip: This step is being done just to demonstrate that fields can be removed from the service interface response message. We will be using these fields later to analyze the results of the DVM service.

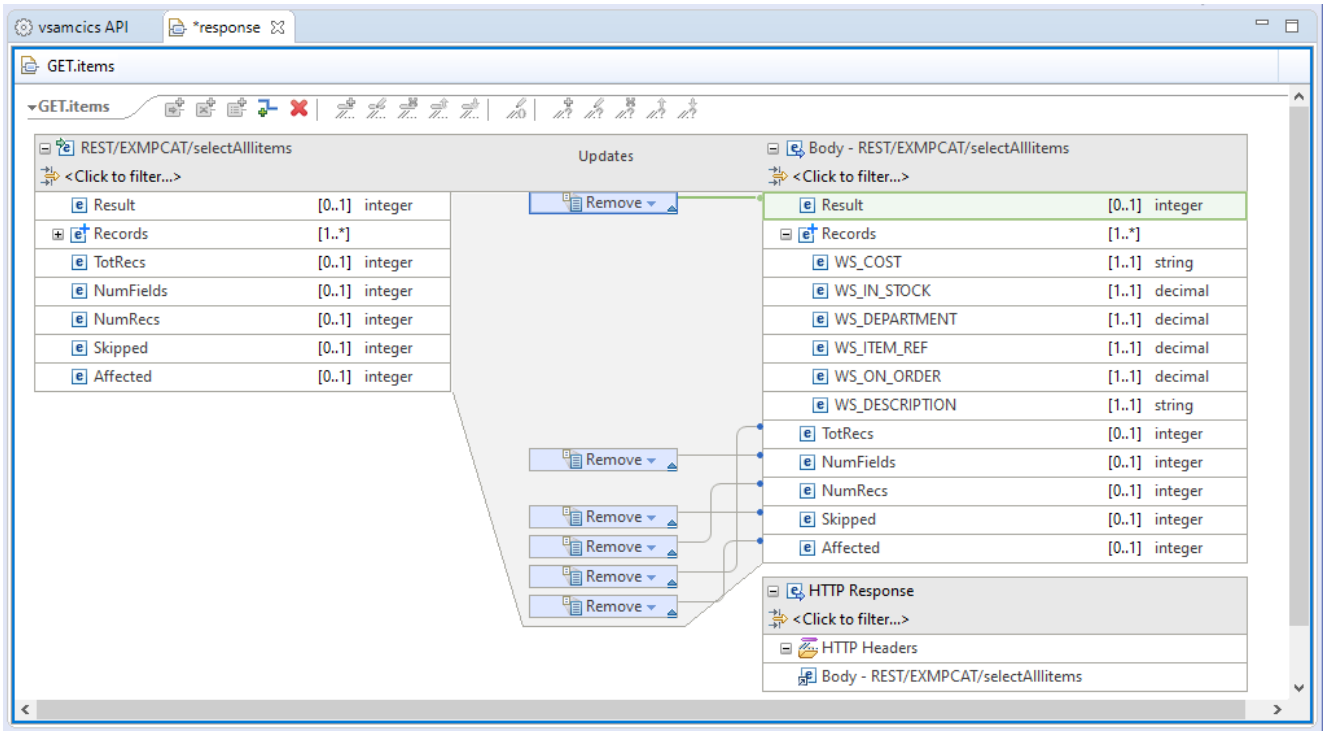
Tech-Tip: The other response files like *TotRecs, Affected, NumRecs*, etc. can be checked like this to set appropriate HTTP response codes. For example, if you were doing a GET (a SELECT) and the value of *NumRecs* was zero, the HTTP response code could be set to *404 – Not Found.*

___8. Right mouse button click on any of the selected fields and select the *Add Remove transform* from the list of options.

___9. This action generates multiple “Remove” requests (see below) for the selected fields. These fields are not required so they will be removed from the response.



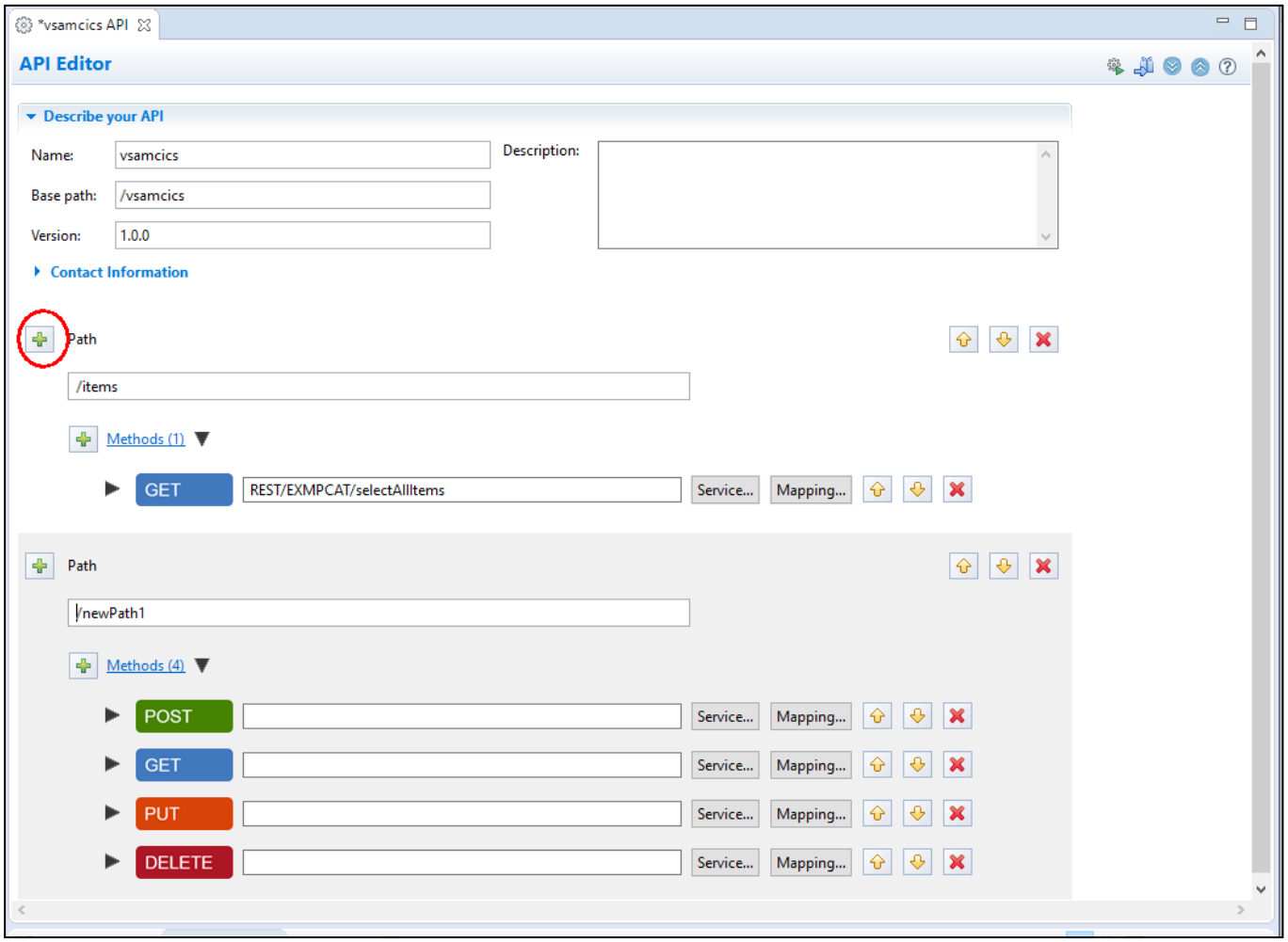
___10. Select the *Result* field and remove it from the response. If not expanded already, expand the *Records* structure and you should see the ‘columns’ that will be displayed in the response.



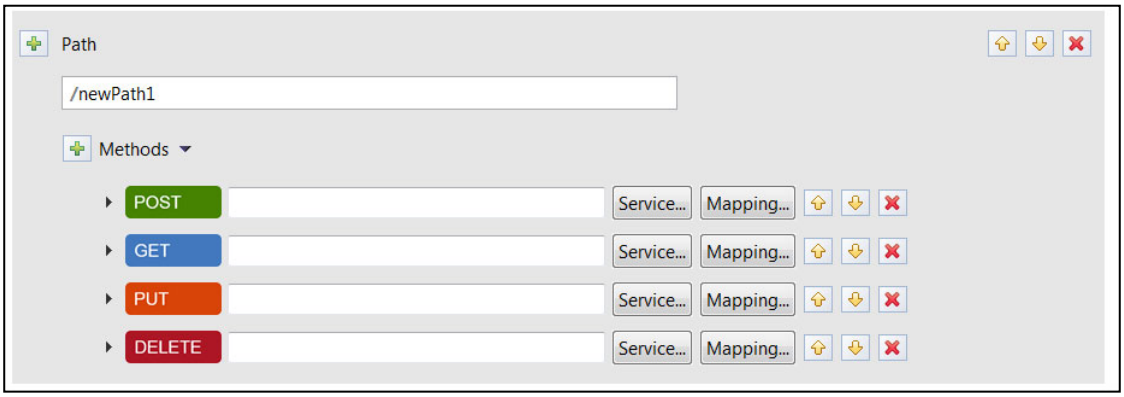
___11. Use the ***Ctrl-S*** key sequence to save all changes and close the *GET.item* response view.

___12. Next, click on the **Mapping** button beside the **GET** method and then select *Open Request Mapping* for this method. Note there are no fields in a request message. Close the request view.

___13. Next, we want to add a *Path* for a **GET** method for the *selectByCost* service. Click the plus icon beside *Path* on the z/OS Connect EE API Editor view to add another path to the API.



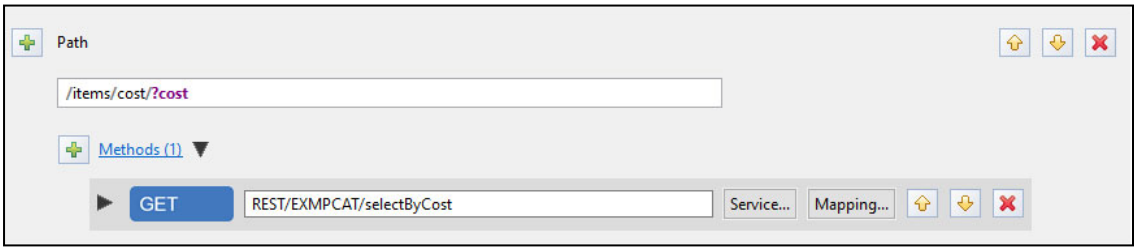
The result is another full set of methods for the new *PATH*.



Mitch Johnson (mitchj@us.ibm.com)

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

___14. Enter a path value of ***/items/cost/?cost*** and remove the **PUT**, **POST** and **DELETE** methods.



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

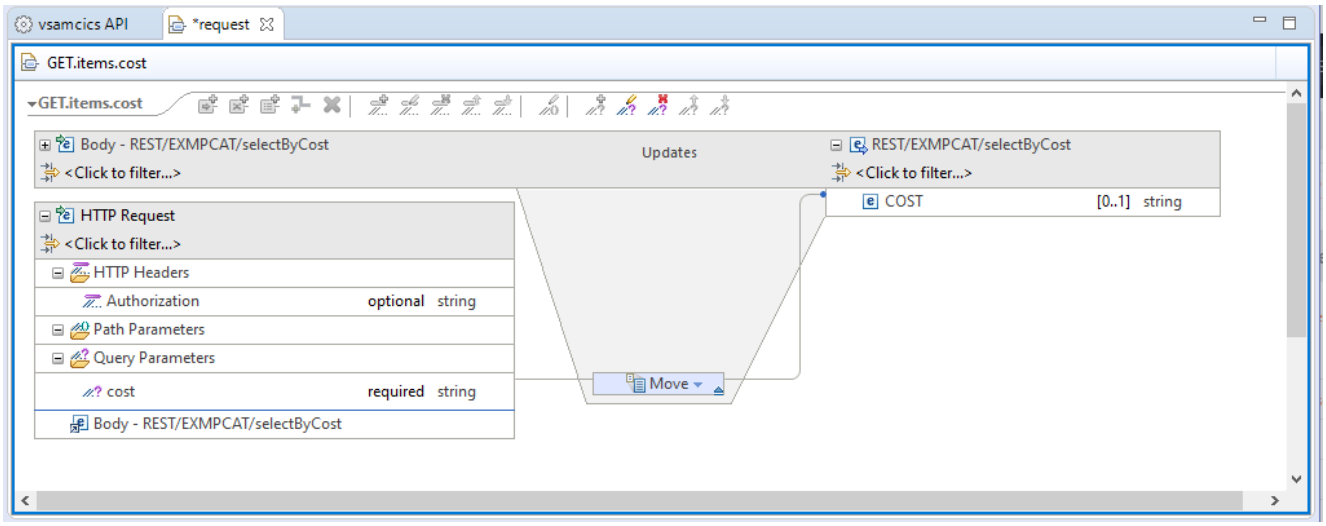
The *?cost* element is a query parameter in the URL that will be used to provide the key of the record for get requests.

The full URL to invoke the methods for this path of the API will be
https://hostname:port/vsamcics/items/cost/?cost=###.##
where ###.## is the cost to be used to select records in the VSAM data set

___15. Save the changes by using the key sequence **Ctrl-S**.

___16. Click on *Mapping* button beside the **GET** method and select *Open request mapping*.

___17. Use the left mouse button to drag the *cost* query parameter from the left-hand side to the *COST* field on the right side.



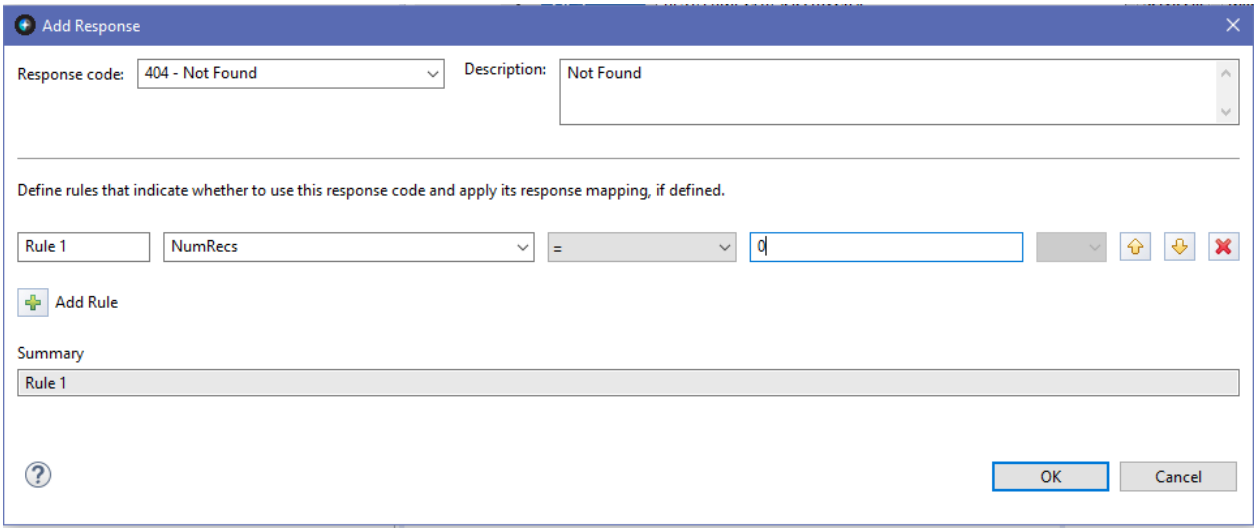
Mitch Johnson (mitchj@us.ibm.com)

___18. Save the the changes using the key sequence **Ctrl-S** and close the *GET.items.cost* request view.

___19. Define a *Response Code* for this request that if the number of records returned equal zero set the HTTP response code to 404. Click on the **Mapping** button and select the *Define Response Codes* option.

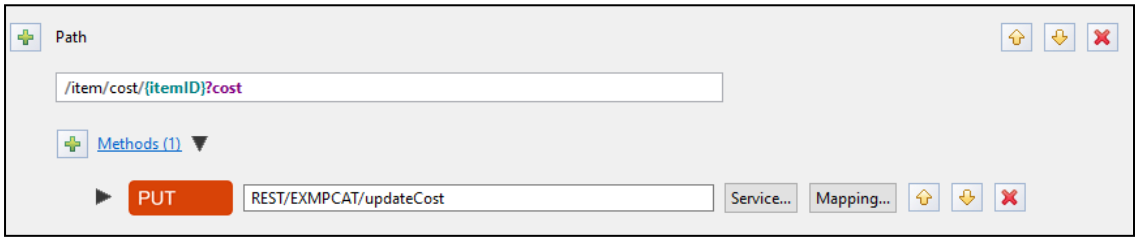
___20. Click the plus sign beside **Add Response** to open the *Add Response* window.

___21. Use the pull-down arrow to select *404 – Not Found* for the *Response Code.* Use the pull-down arrows to select field *NumRecs* and the equal sign for *Rule 1.* Enter **0** in the open area for *Rule 1*. When finished your windows should look like the one below. Click **OK** to continue.



___22. Next, we want to add a *Path* for a **PUT** method for the *updateCost*. Click the plus icon beside Path on the z/OS Enter a path value of ***/item/cost/{itemID}?cost*** and remove the **GET, POST** and **DELETE** methods.

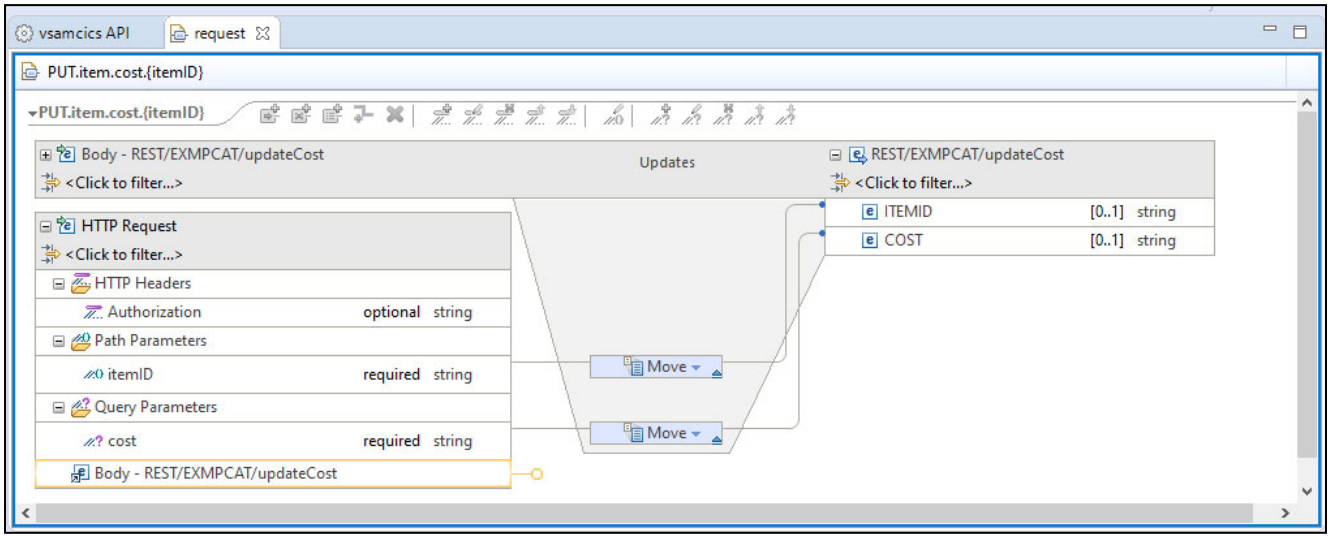
___23. Click the **Service** button beside **PUT** and select the *REST/EXMPCAT/updateCost* service.



___24. Save the changes by using the key sequence **Ctrl-S**.

___25. Click on *Mapping* → *Open request mapping*.

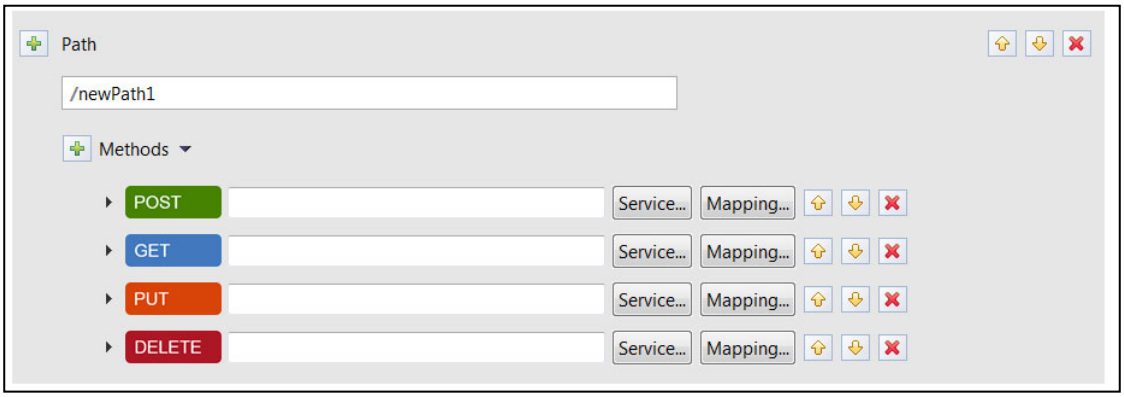
___26. Use the left mouse button to drag the *cost* query parameter from the left-hand side to the *COST* field on the right side. Repeat this to drag the *itemID* path parameter from the left-hand side to the *ITEMID* field on the right side.



___27. Save the the changes using the key sequence **Ctrl-S** and close the *PUT.item.cost request* view.

___28. Next, we want to add another Path for a **GET** method for the *selectByInventory* service. The **GET** API will return the list of items in the inventory where the number of in stock items is below a certain level. This number of items on order will be set to the maximum value of this field. Click the plus icon beside Path on the z/OS Connect EE API Editor view to add another path to the API.

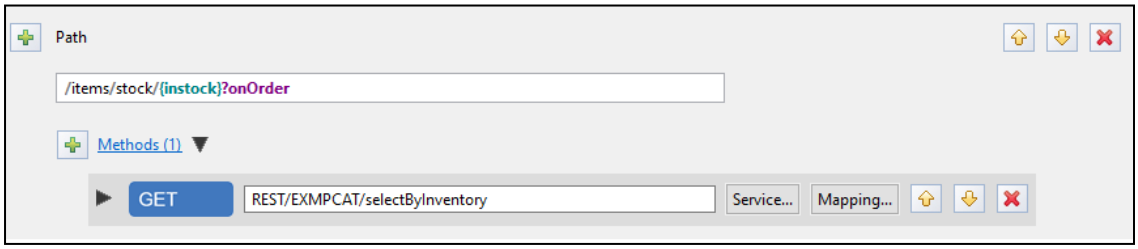
The result is another full set of methods for the new *PATH*.



___29. Enter a path value of ***/items/stock/{inStock}?onOrder*** and remove the **PUT, POST** and **DELETE** methods.

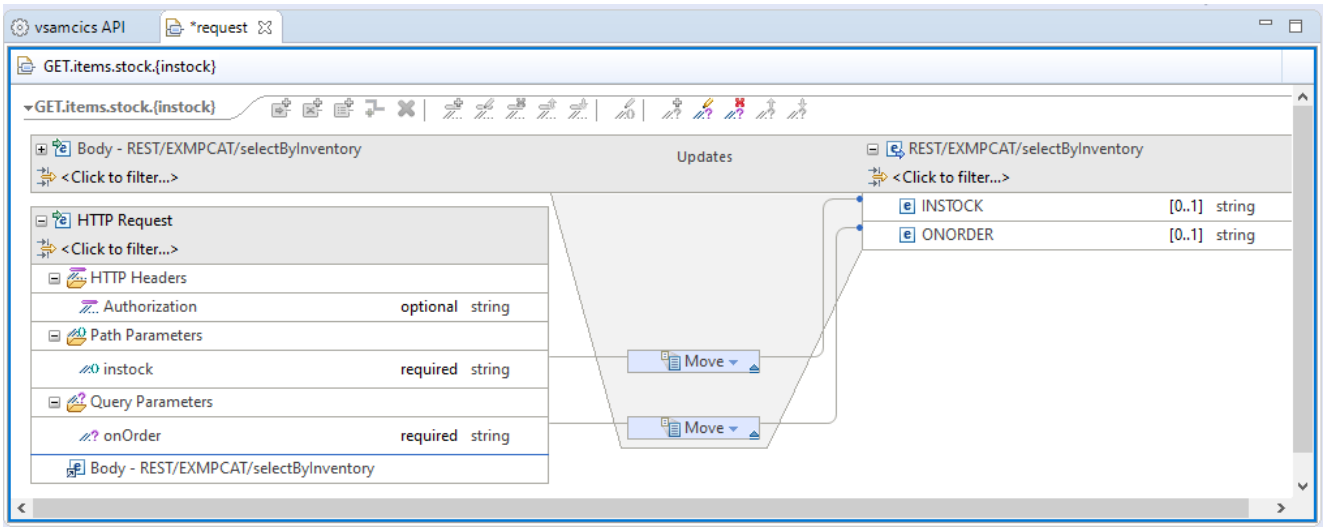
___30. Click the **Service** button beside **GET** and select the *REST/EXMPCAT/selectByInventory* service.

___31. Save the changes by using the key sequence **Ctrl-S**.



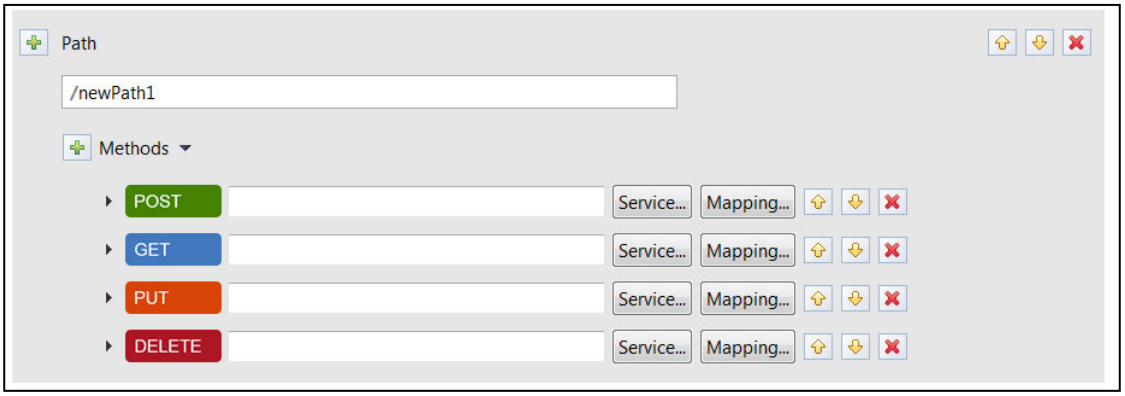
___32. Click on *Mapping* → *Open request mapping*.

___33. Use the left mouse button to drag the *inStock* path parameter from the left-hand side to the *INSTOCK* field on the right side. Select the *onOrder* query parameter from the left-hand side and drag it to the ONORDER field on the right-hand side.



___34. Save the the changes using the key sequence **Ctrl-S** and close this view.

___35. Next, we want to add another Path for a **PUT** method for the *updateInStock* service. The **PUT** API will update the number of items in stock for a specific item. Click the plus icon beside on the z/OS Connect EE API Editor view to add another path to the API.



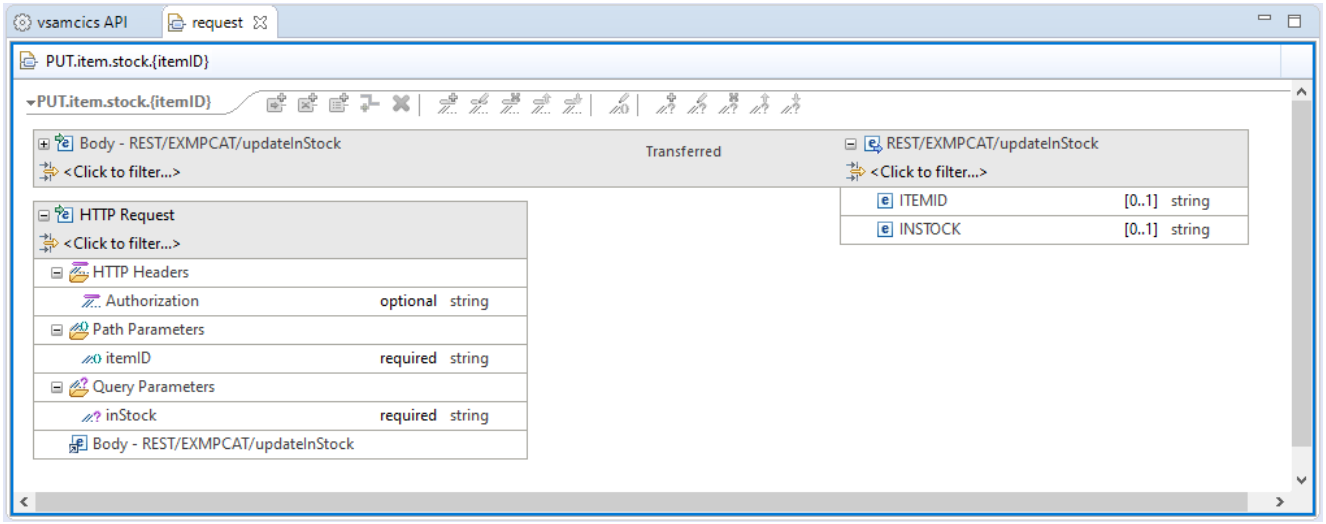
___36. Enter a path value of ***/item/stock/{itemID}?inStock*** and remove the **GET, POST** and **DELETE** methods.

___37. Click the **Service** button beside **PUT** and select the *REST/EXMPCAT/updateInStock* service.

___38. Save the changes by using the key sequence **Ctrl-S**.

___39. Click on *Mapping* → *Open request mapping*.

___40. Use the left mouse button to drag the *itemID* path parameter from the left-hand side to the *ITEMID* field on the right side. Select the *inStock* query parameter from the left-hand side and drag it to the INSTOCK field on the right-hand side.



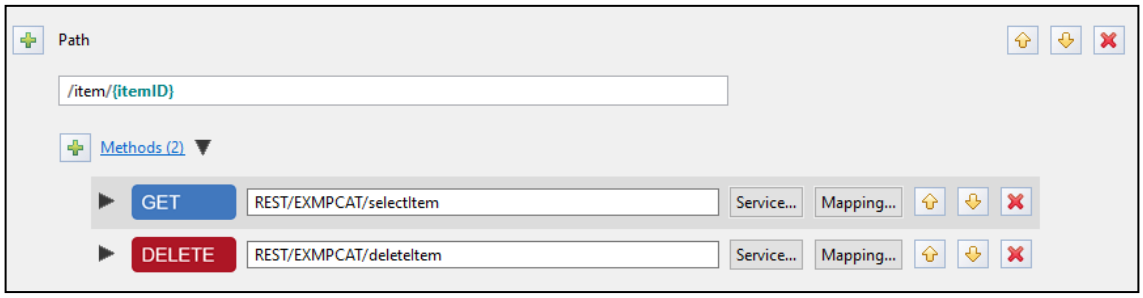
___41. Save the the changes using the key sequence **Ctrl-S** and close this view.

___42. Next, we want to add another path for a **GET** and **DELETE** methods for the *selectItem* and *deleteItem* services. Note that these services have one parameter, an item number. So, they can share a path.

___43. Add a new *Path* and enter a path value of ***/item/{itemID}*** and remove the **POST** *and* **PUT** methods.

___44. Click the **Service** button beside **GET** and select the *REST/EXMPCAT/selectItem* service.

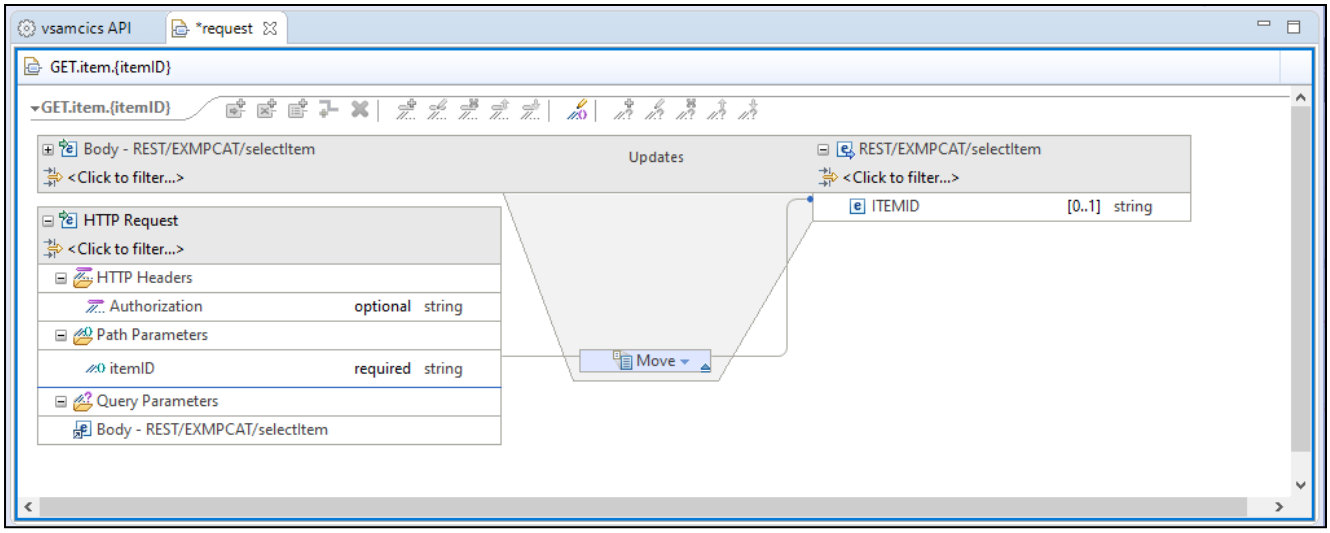
___45. Click the **Service** button beside **DELETE** and select the *REST/EXMPCAT/deleteItem* service.



___46. Save the changes by using the key sequence **Ctrl-S**.

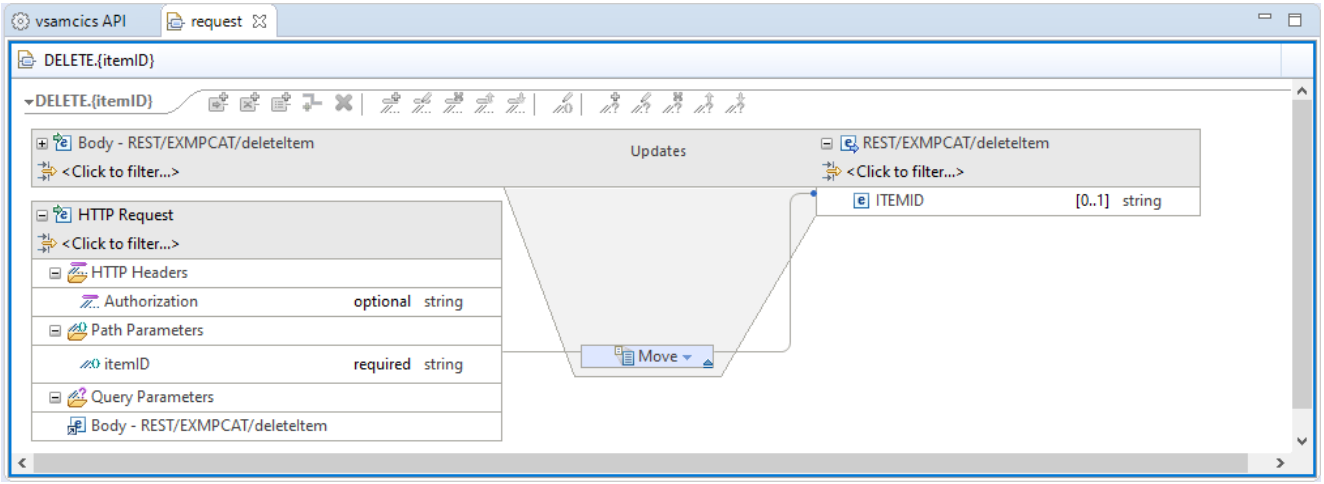
___47. Click on *Mapping* → *Open request mapping* for the **GET** method.

___48. Use the left mouse button to drag the *itemID* path parameter from the left-hand side to the *ITEMID* field on the right side. Save and close this view.



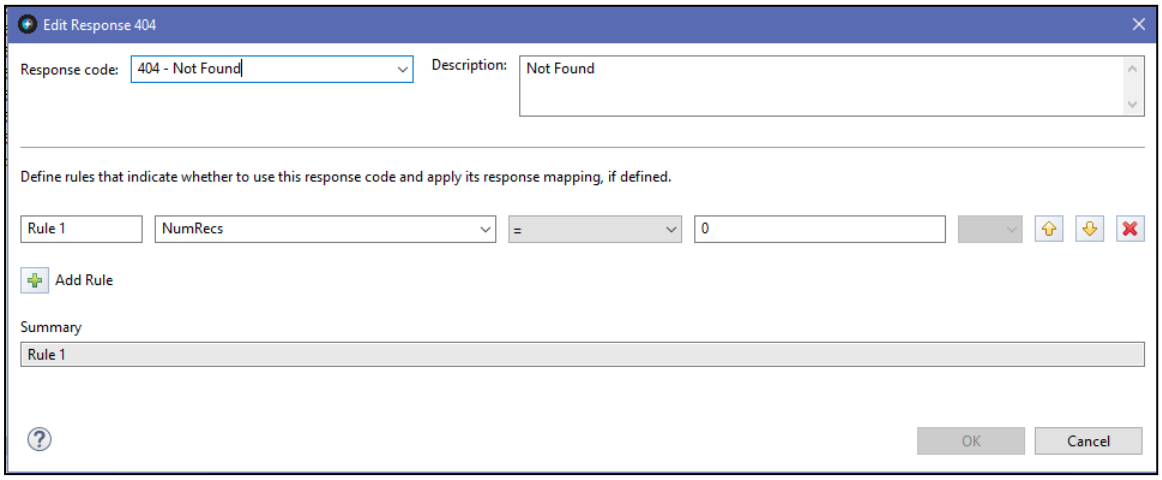
Mitch Johnson (mitchj@us.ibm.com)

___49. Repeat these two steps and do the same for the **DELETE** method.



___50. Save the changes using the key sequence **Ctrl-S**.

___51. Define an additional *Response Code* for both the **GET** and **DELETE** methods so if the number of records returned is zero the HTTP status code is set to *404 – Not Found*, see below.

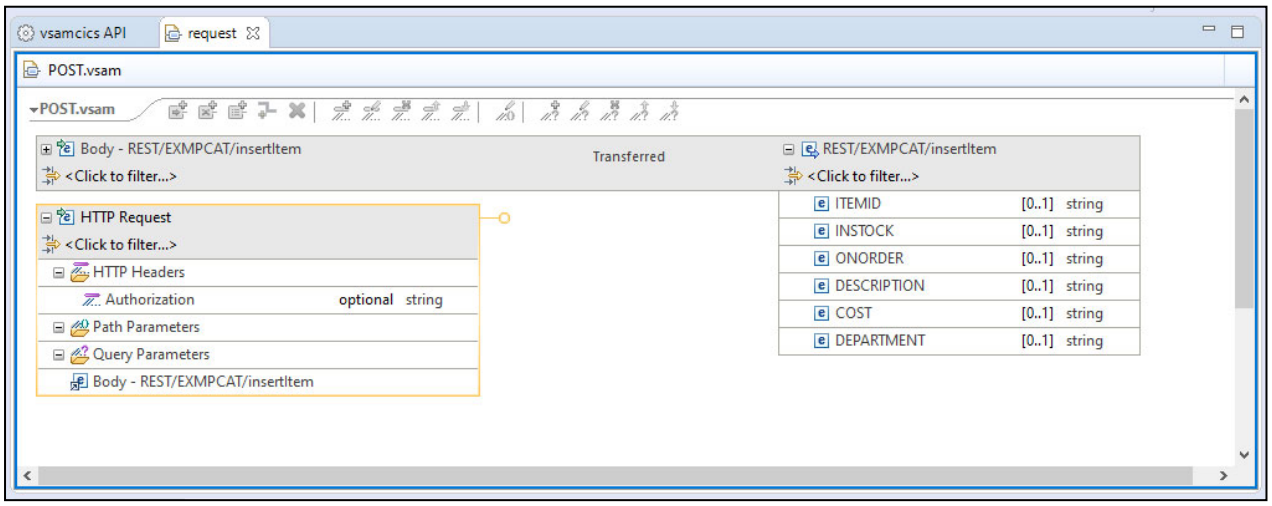


___52. Finally, we want to add another Path for a **POST** method for the *insertItem* service.

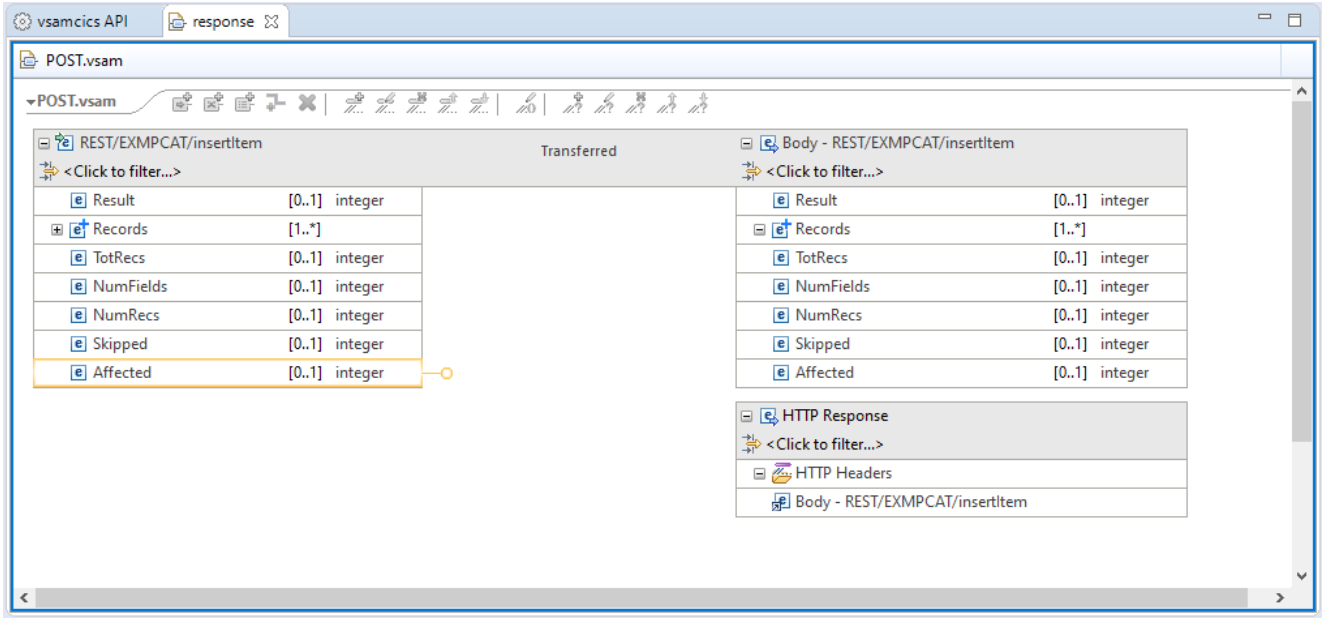
___53. Enter a path value of ***/item*** and remove the **GET**, **PUT** and **DELETE** methods.

Mitch Johnson (mitchj@us.ibm.com)

___54. Next, click on the **Mapping** button beside the **POST** method and then select *Open Request Mapping*. Note the request message has the fields derived from the record layout. Close this view.



___55. Next, click on the **Mapping** button beside the **POST** method and then select *Open Default Response Mapping*. Note the response message returns no 'column' fields in the response message. Close this view.



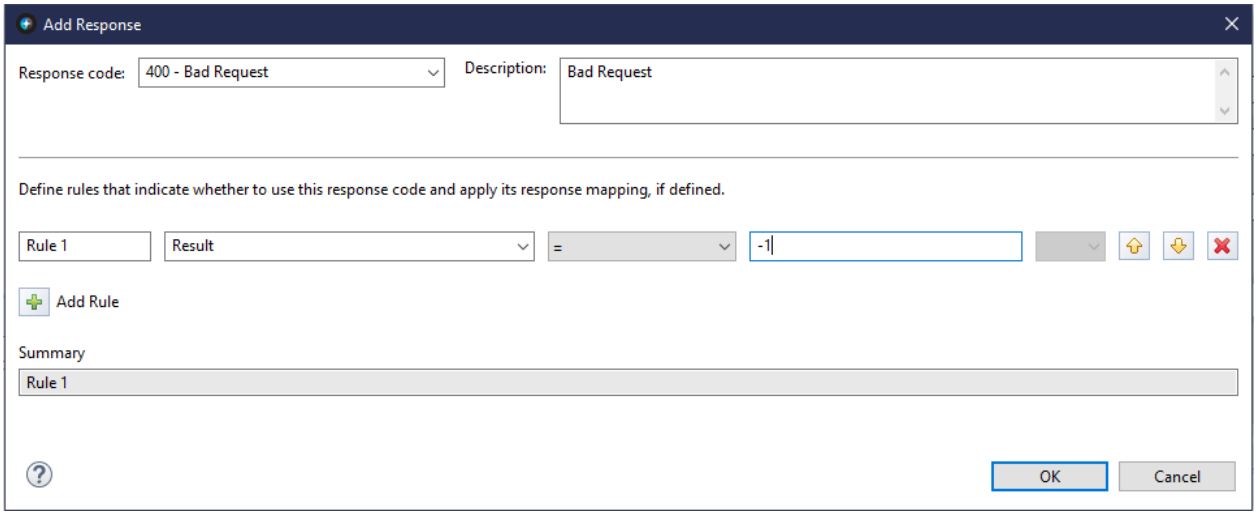
Tech-Tip: Records are return only when the underlying service is a select request.

Mitch Johnson (mitchj@us.ibm.com)

___56. Next, click on the **Mapping** button beside the **POST** method and then select *Define Response Code*.

___57. Click the plus sign beside **Add Response** to open the *Add Response* window.

___58. Use the pull-down arrow to select *400 – Bad Request* for the *Response Code.* Use the pull-down arrows to select field *Result* and the equal sign for *Rule 1.* Enter ***-1*** in the open area for *Rule 1*. When finished your windows should look like the one below:



___59. Click *OK* to continue.

___60. Close any open *request* or *response* mapping tabs.

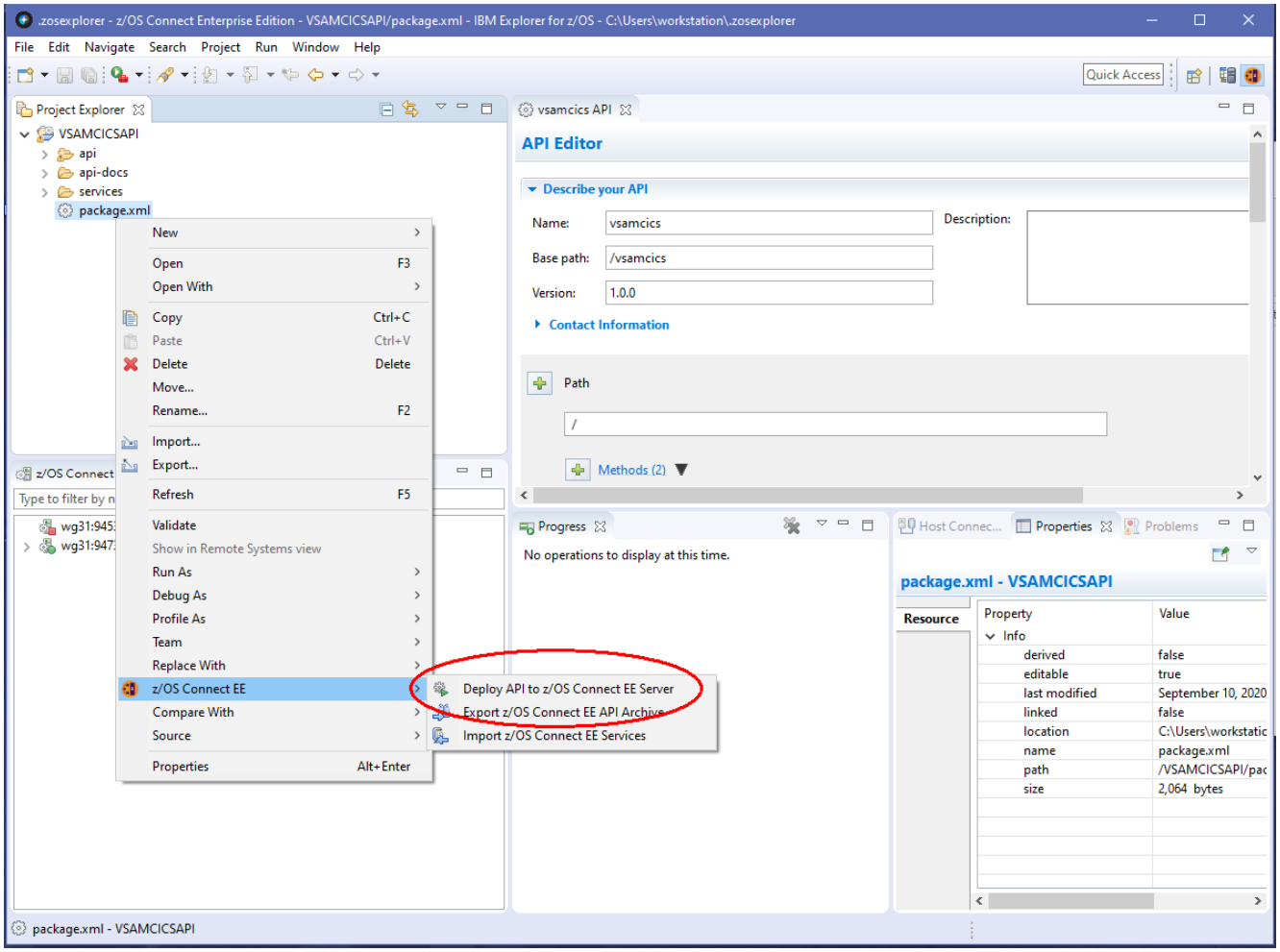
Summary

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

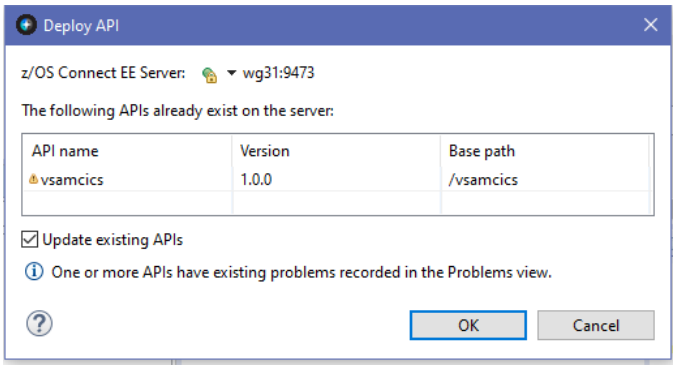
Mitch Johnson (mitchj@us.ibm.com)

Deploy the API to a z/OS Connect EE Server

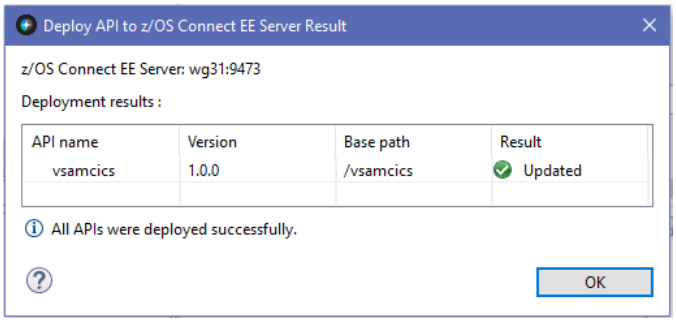
___1. In the *Project Explorer* view (upper left), right-mouse click on the *VSAMCICSAPI* folder, then select *z/OS Connect EE→ Deploy API to z/OS Connect EE Server*.



___2. If the z/OS Explorer is connected to only one z/OS Connect server there is only one choice (*wg31:9473*). 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 *wg31:9473* from the list. Click **OK** on this window to continue.



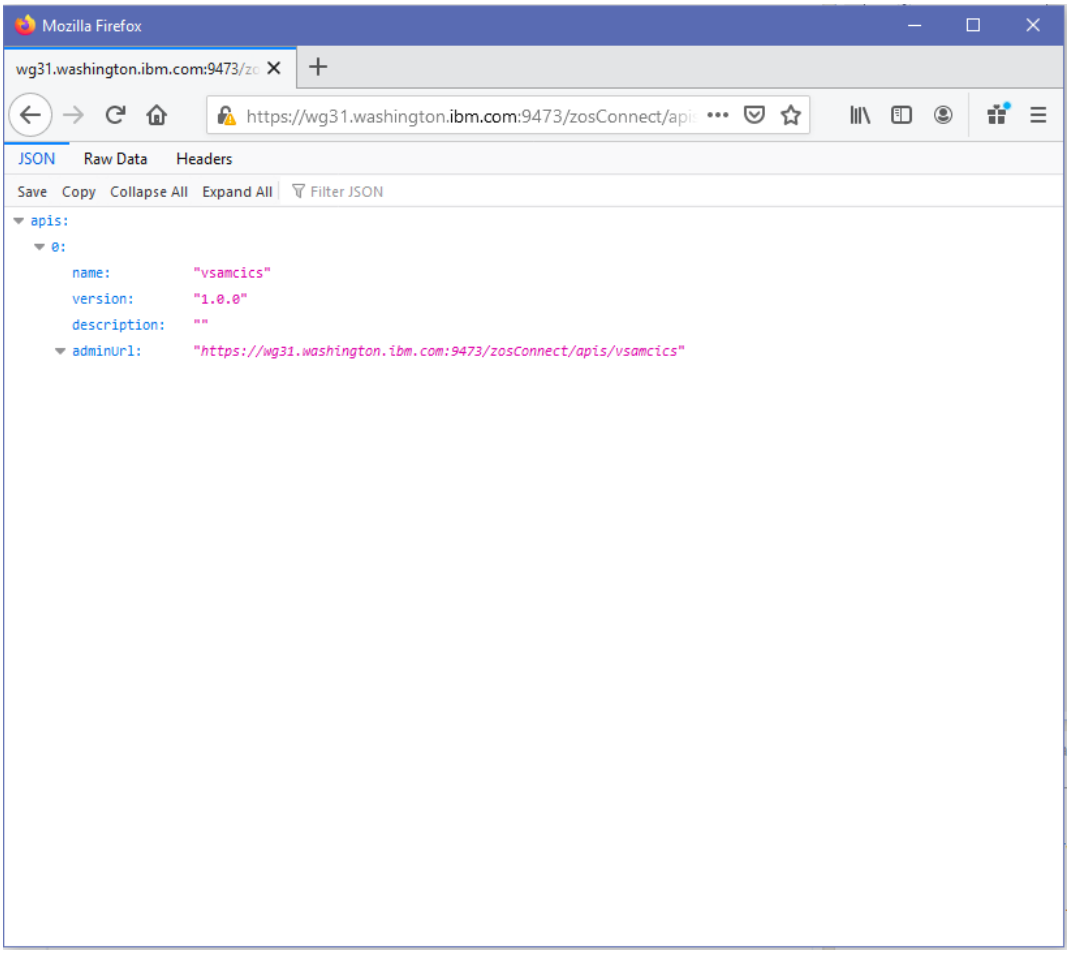
___3. The API artifacts will be transferred to z/OS in an API archive (AAR) file and copied into the */var/ats//zosconnect/servers/zceedvm/resources/zosconnect/apis* directory.



Test the VSAM CICS APIs using Swagger UI

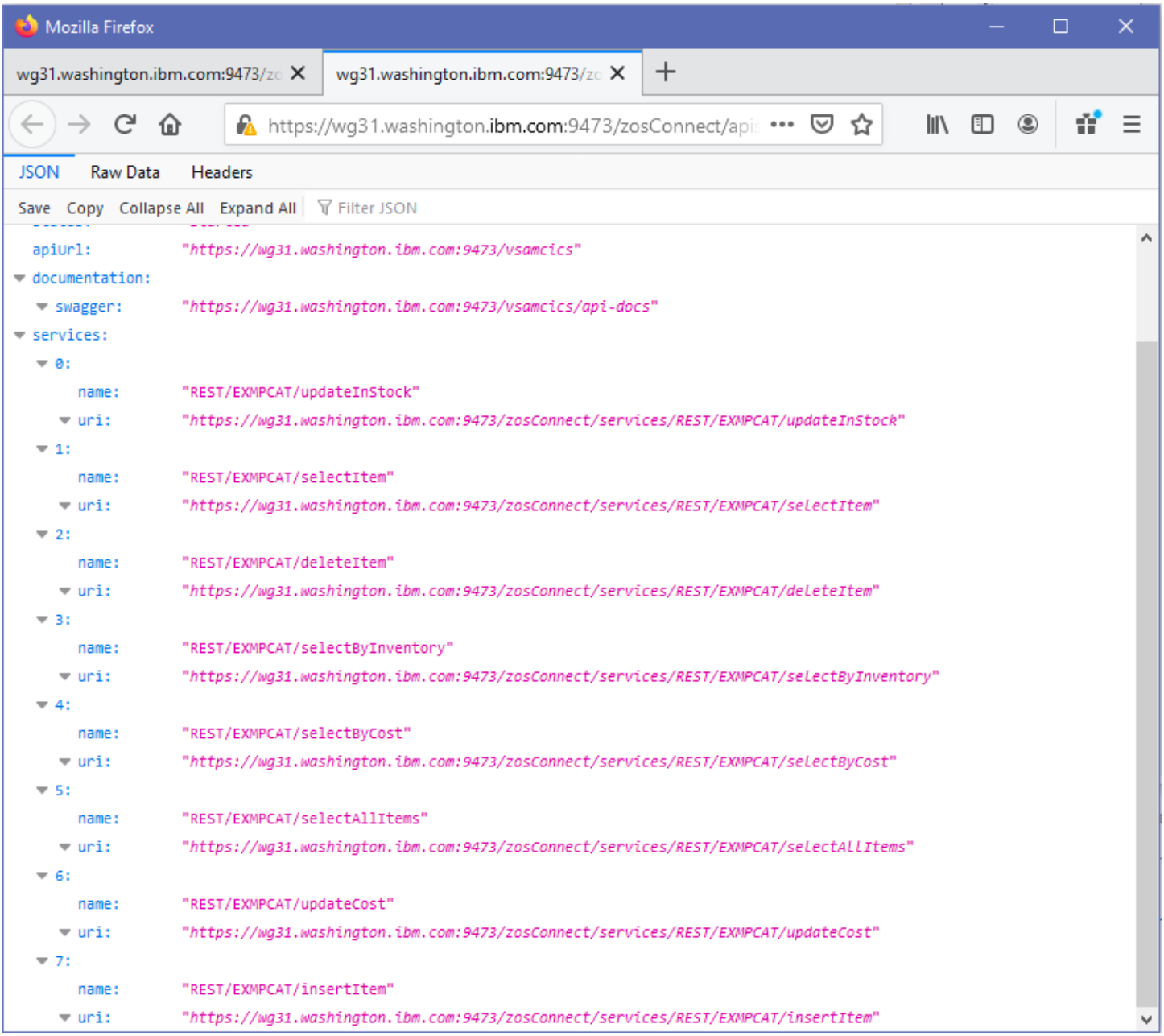
___1. Next enter URL ***https://wg31.washington.ibm.com:9473/zosConnect/apis*** in the Firefox browser and you should see the window below.

> **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 ***USER1*** and password **user1** and click **OK**.

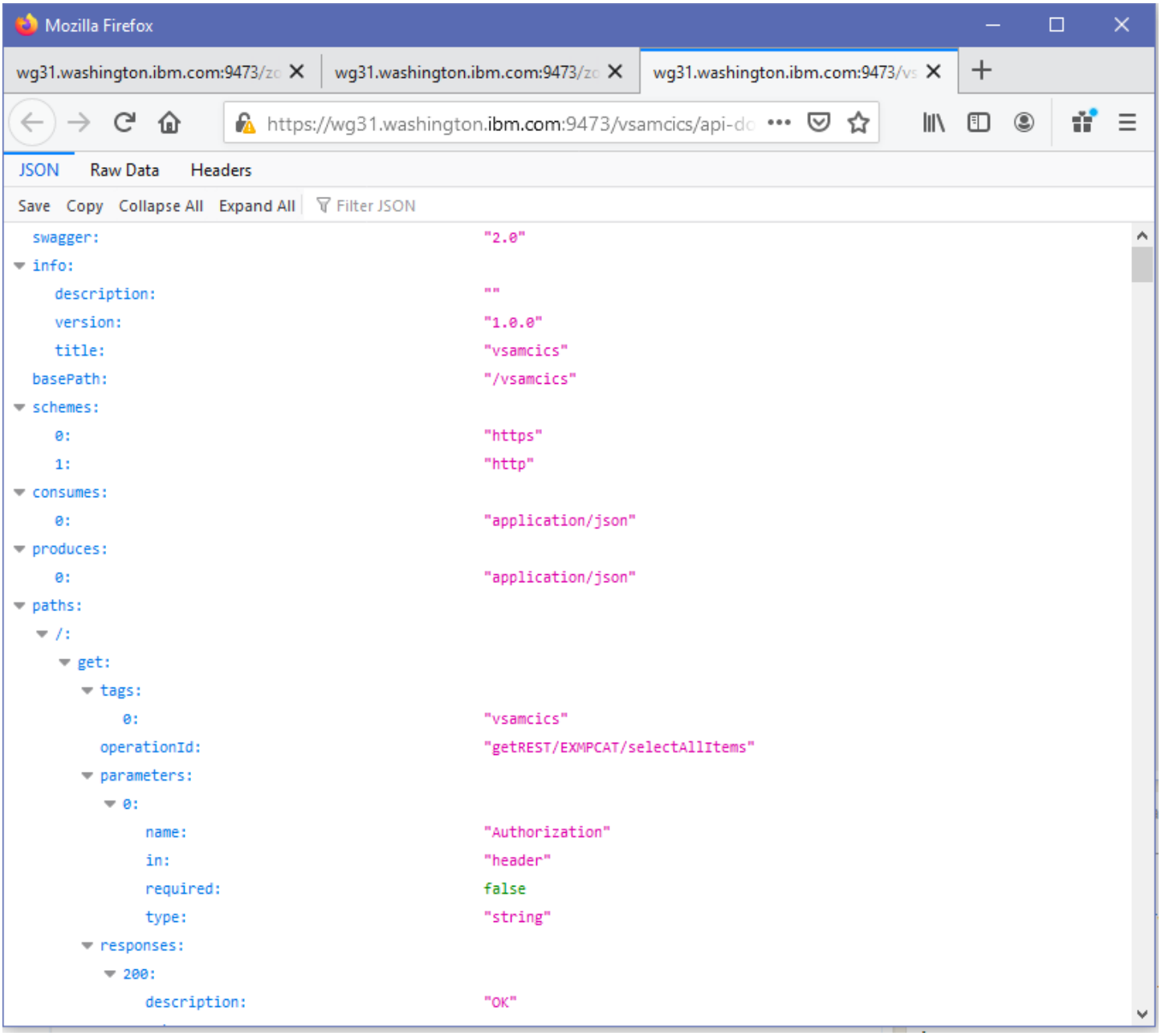


> **Tech Tip:** It is very important to access the z/OS Connect server from a browser prior to any testing using the Swagger UI. Accessing a z/OS Connect URL from a browser starts an SSL handshake between the browser and the server. If this handshake has not performed prior to performing any test the 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.

___2. If you click on *adminUrl* URL the window below should be displayed:

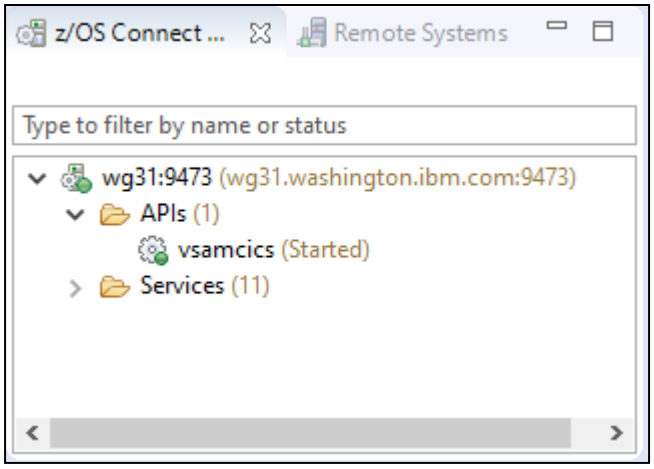


___3. Finally. click on the *swagger* URL for and you should see the Swagger document associated with this API.

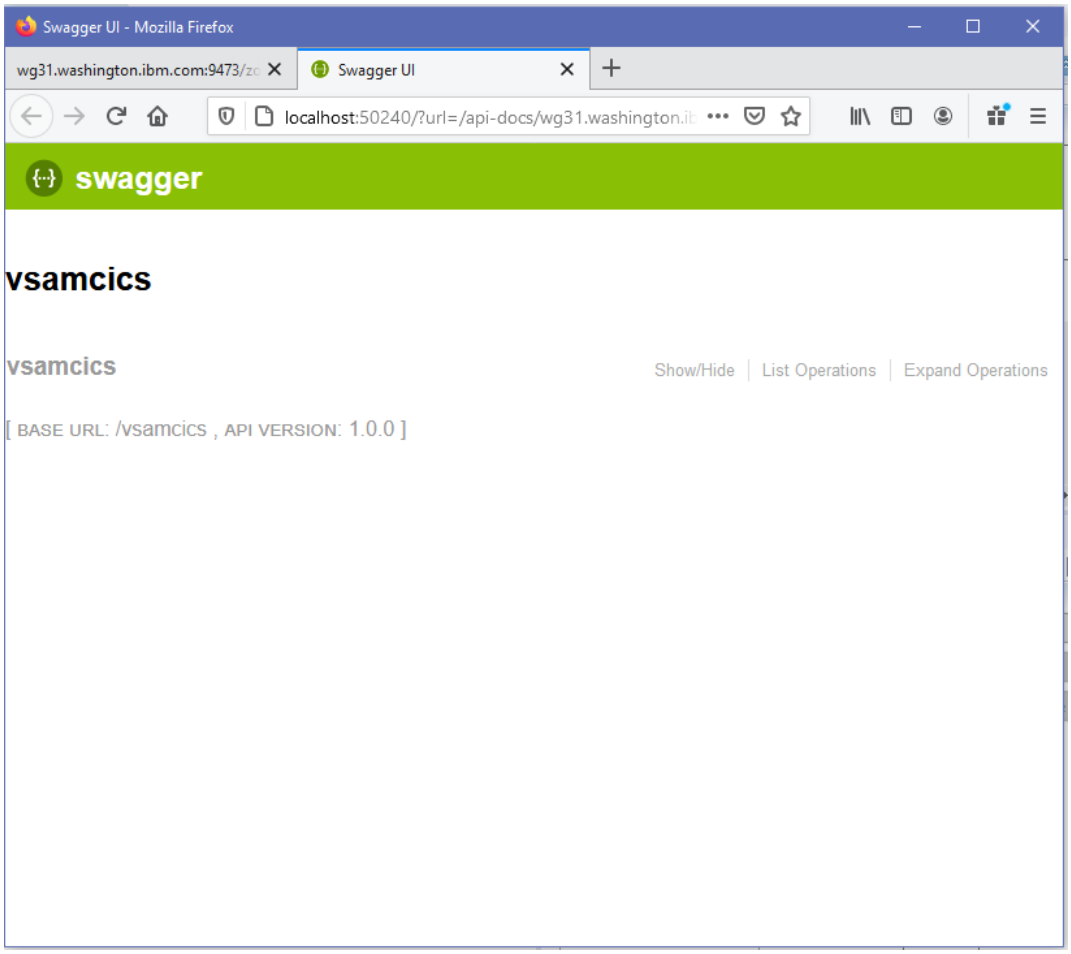


Explore this Swagger document and you will see the results of the request and response mapping performed earlier. This Swagger document can be used by a developer or other tooling to develop REST clients for this specific API.

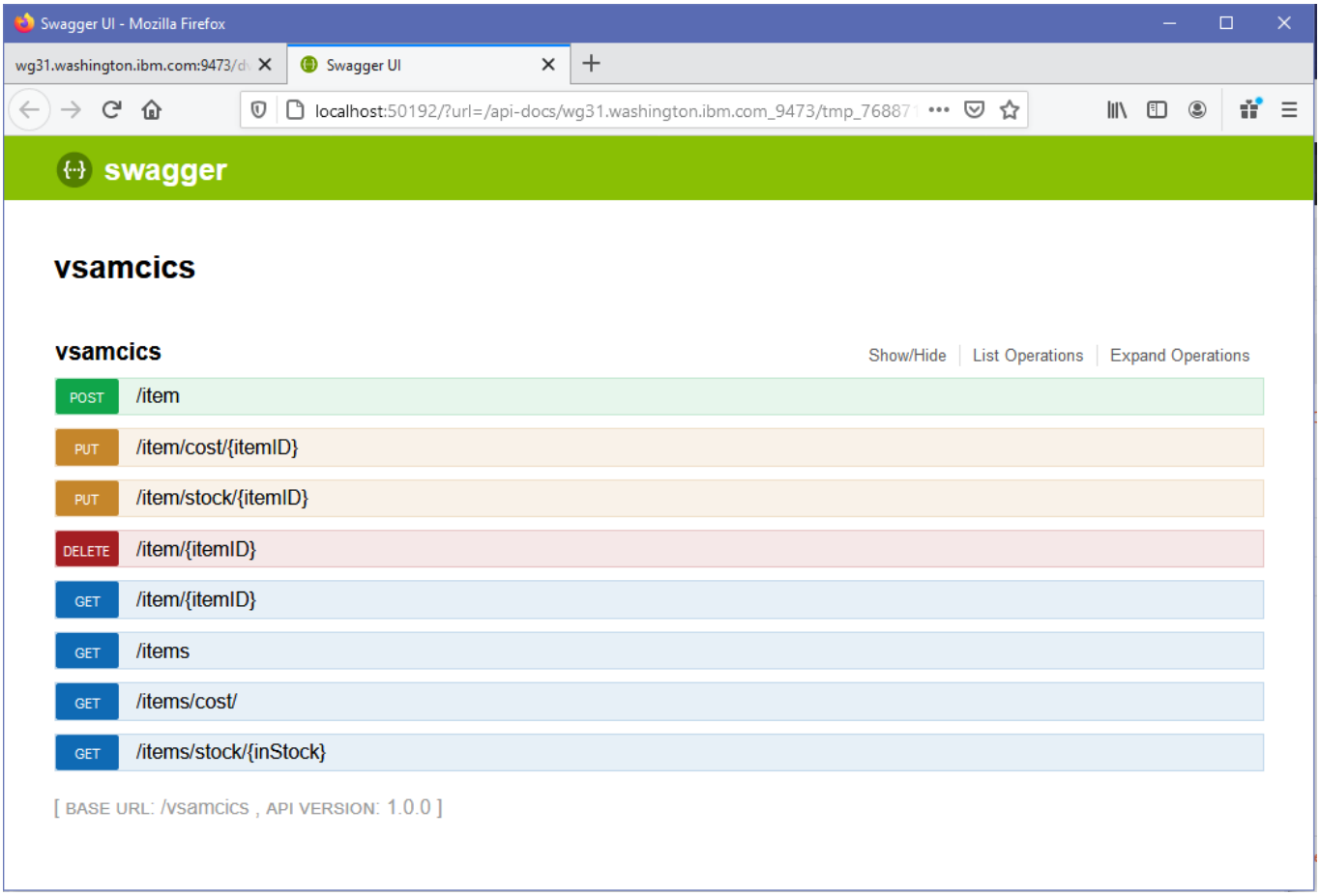
___4. In the lower left-hand side of the *z/OS Connect Explorer* perspective there is view entitled *z/OS Connect EE Servers*. Expand *wg31:9473* and the expand the *APIs* folder. You should see a list of the APIs installed in the server.



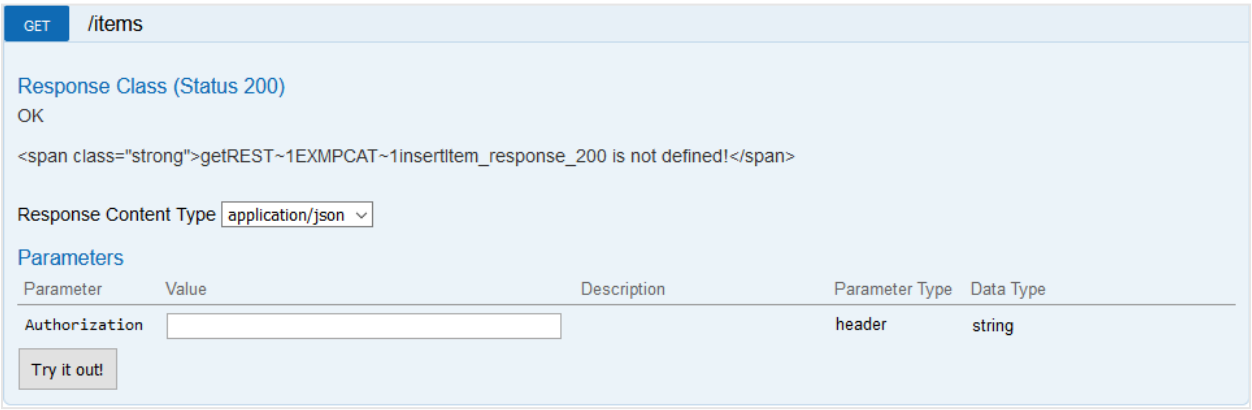
___5. Right mouse button click on *vsamcics* and select *Open in Swagger UI.* Click **OK** if an informational prompt appears. This will open a new view showing a *Swagger* test client (see below).



___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 all records from the VSAM data set by clicking on the */items* URI string. Remember this was the *Path* specified for the *GET* method for the *selectAllItems* 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 ***Basic VVNFUjE6VVNFUjE=*** in the box beside *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.

> **Tech Tip:** The string *VVNFUjE6VVNFUjE=* is the string USER1:USER1 encoded in base 64. See URL https://www.base64encode.org/ for information on how this string was generated.

Mitch Johnson (mitchj@us.ibm.com)

___9. Scroll down the view and you should see the *Response Body* which contains the results of the GET method (see below). Note that the columns removed from the interface in an earlier step are not present.

Response Body


```
{
  "Records": [
    {
      "ws_department": 10,
      "ws_item_ref": 10,
      "ws_cost": "003.90",
      "ws_in_stock": 14,
      "ws_on_order": 0,
      "ws_description": "Ball Pens Black 24pk"
    },
    {
      "ws_department": 10,
      "ws_item_ref": 20,
      "ws_cost": "002.90",
      "ws_in_stock": 6,
      "ws_on_order": 50,
      "ws_description": "Ball Pens Blue 24pk"
    },
    {
```


___10. Select the *GET* method for selecting a single record from the VSAM data set by clicking on the */item/{itemID}* URI string. Remember this was the *Path* specified for the *GET* method for the *selectItem* 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).

GET /item/{itemID}

Response Class (Status 200)

OK

getREST~1EXMPCAT~1insertItem_response_200 is not defined!

Response Content Type application/json

Parameters

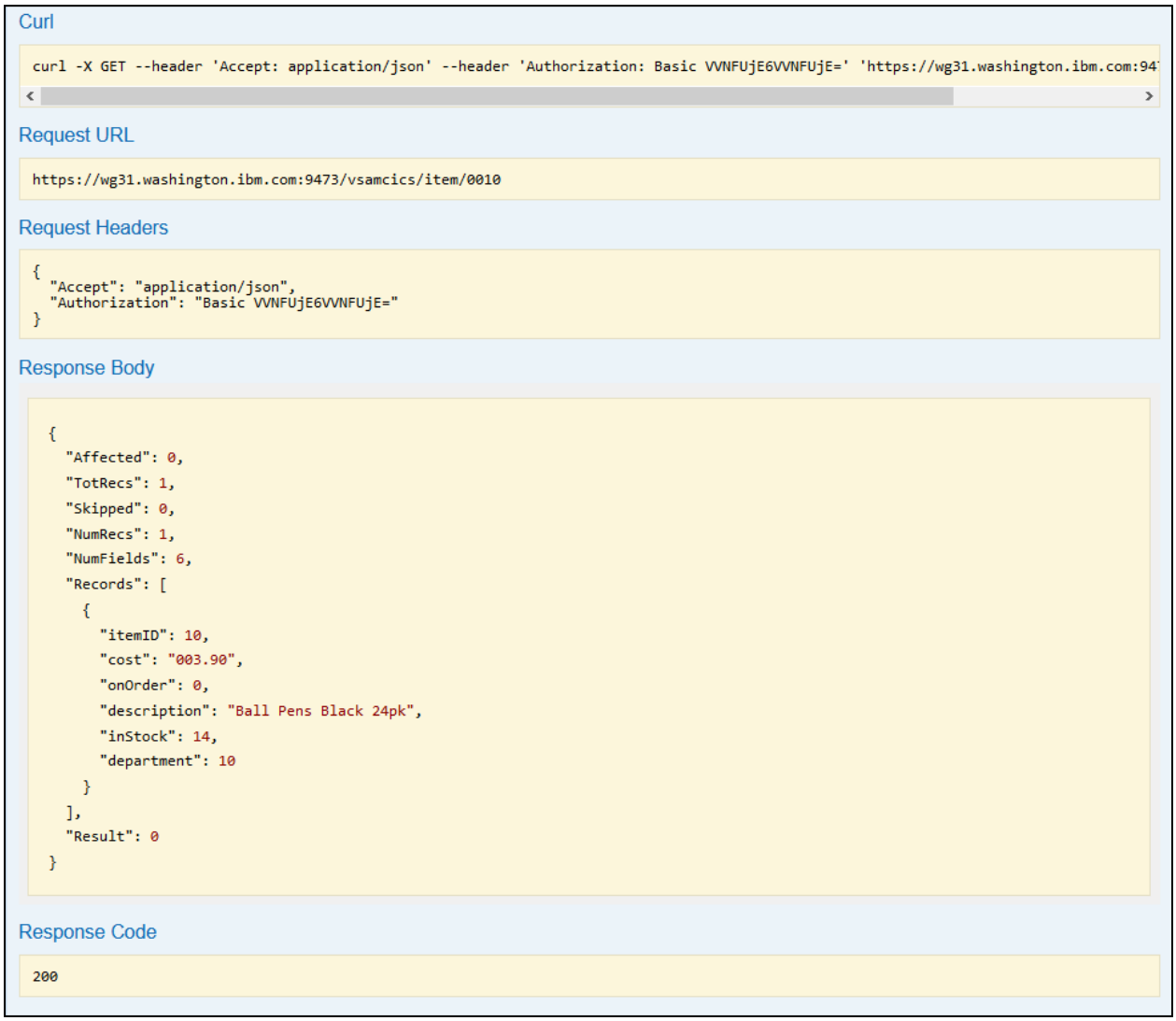
Parameter	Value	Description	Parameter Type	Data Type
Authorization	Basic VVNFUjE6VVNFUjE=		header	string
itemID	0010		path	string

Try it out! Hide Response

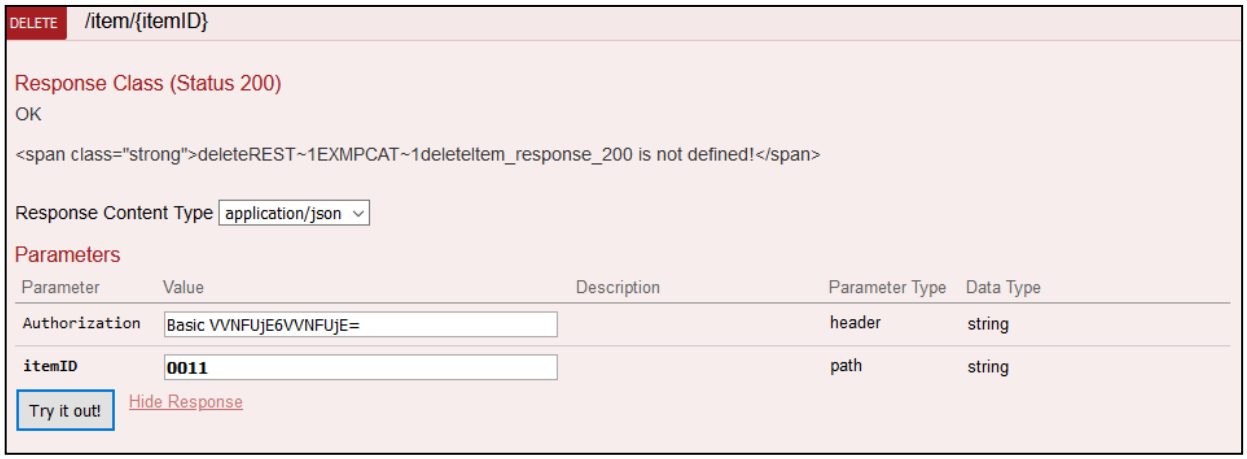
Mitch Johnson (mitchj@us.ibm.com)

___11. Enter ***Basic VVNFUjE6VVNFUjE=*** in the box beside *Authorization* and ***0010*** in the area beside *itemID* and press the **Try it out!** button.

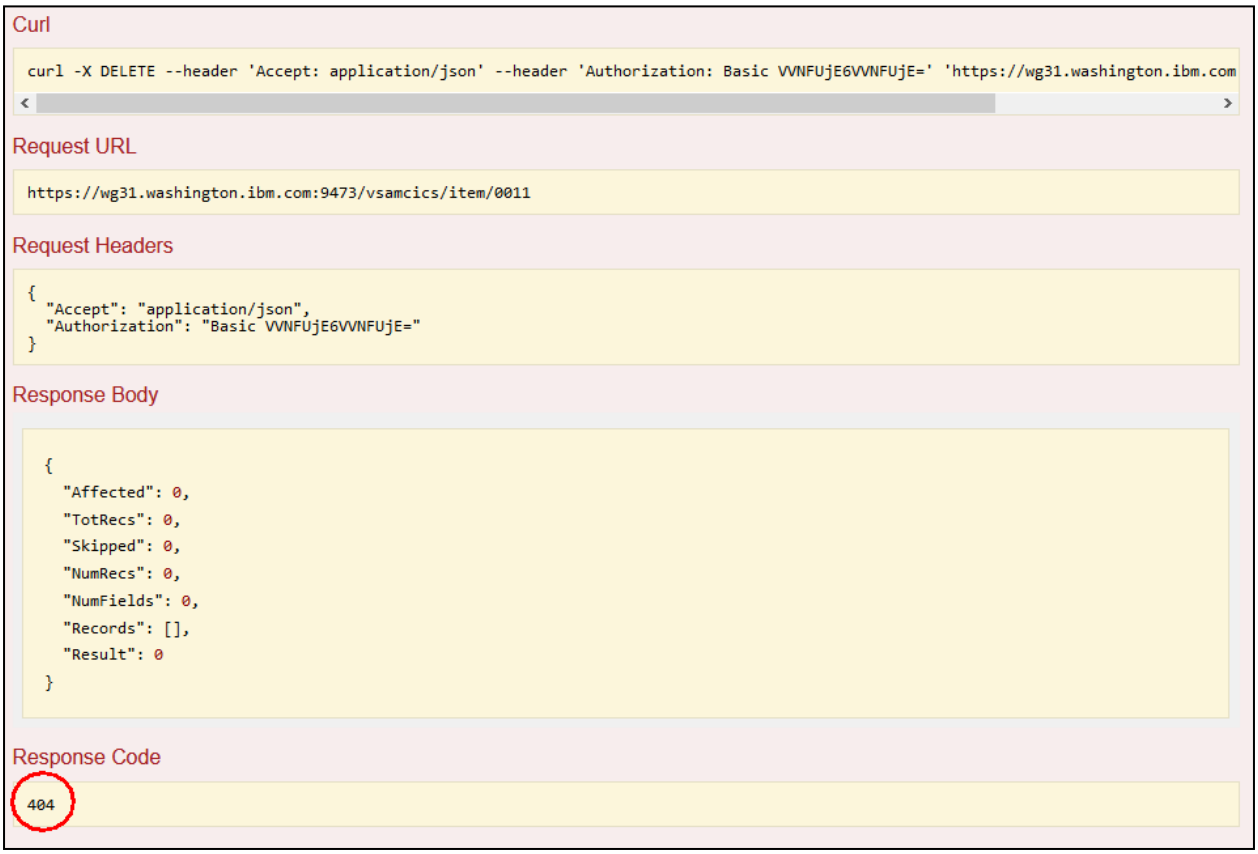
___12. Scroll down the view and you should see the *Response Body* which contains the results of the GET method (see below). Note that the columns removed from the interface in an earlier step are not present.



___13. Select the *DELETE* method for deleting a single record from the VSAM data set by clicking on the *DELETE /item/{itemID}* URI string. Remember this was the *Path* specified for the *DELETE* method for the *deleteItem* service when the API was defined. Enter ***Basic VVNFUjE6VVNFUjE=*** in the box beside *Authorization* and ***0011*** in the area beside *itemID* and press the **Try it out!** button.



___14. Scroll down the view and you should see the *Response Body* which contains the results of the DELETE method (see below). Note the 404-response code. The record was not found so it could not be deleted.



Mitch Johnson (mitchj@us.ibm.com)

Below are the contents of the VSAM data set.

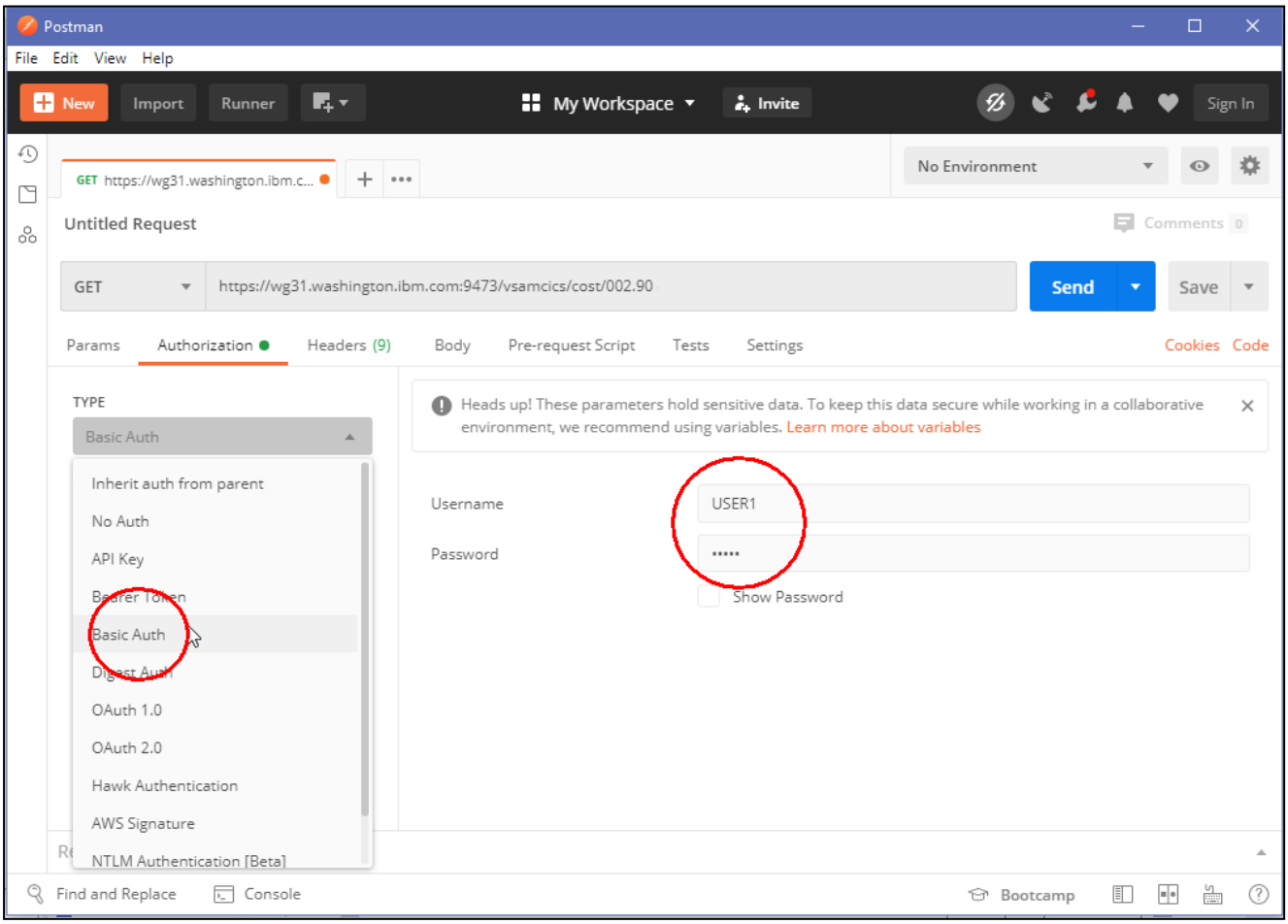
Item#	Description	Dept	Cost	In Stock	On Order
0010	Ball Pens Black 24pk	010	002.90	0135	000
0020	Ball Pens Blue 24pk	010	002.90	0006	050
0030	Ball Pens Red 24pk	010	002.90	0106	000
0040	Ball Pens Green 24pk	010	002.90	0080	000
0050	Pencil with eraser 12pk	010	001.78	0083	000
0060	Highlighters Assorted 5pk	010	003.89	0013	040
0070	Laser Paper 28-lb 108 Bright 500/ream	010	007.44	0102	020
0080	Laser Paper 28-lb 108 Bright 2500/case	010	033.54	0025	000
0090	Blue Laser Paper 20lb 500/ream	010	005.35	0022	000
0100	Green Laser Paper 20lb 500/ream	010	005.35	0003	020
0110	IBM Network Printer 24 - Toner cart	010	169.56	0012	000
0120	Standard Diary: Week to view 8 1/4x5 3/4	010	025.99	0007	000
0130	Wall Planner: Erasable 36x24	010	018.85	0003	000
0140	70 Sheet Hard Back wire bound notepad	010	005.89	0084	000
0150	Sticky Notes 3x3 Assorted Colors 5pk	010	005.35	0036	045
0160	Sticky Notes 3x3 Assorted Colors 10pk	010	009.75	0067	030
0170	Sticky Notes 3x6 Assorted Colors 5pk	010	007.55	0064	030
0180	Highlighters Yellow 5pk	010	003.49	0088	010
0190	Highlighters Blue 5pk	010	003.49	0076	020
0200	12 inch clear rule 5pk	010	002.12	0014	010
0210	Clear sticky tape 5pk	010	004.27	0073	000

Test the VSAM CICS APIs using Postman

The other API services will be tested using Postman.

___1. Open the *Postman* tool icon on the desktop. If necessary reply to any prompts and close any welcome messages.

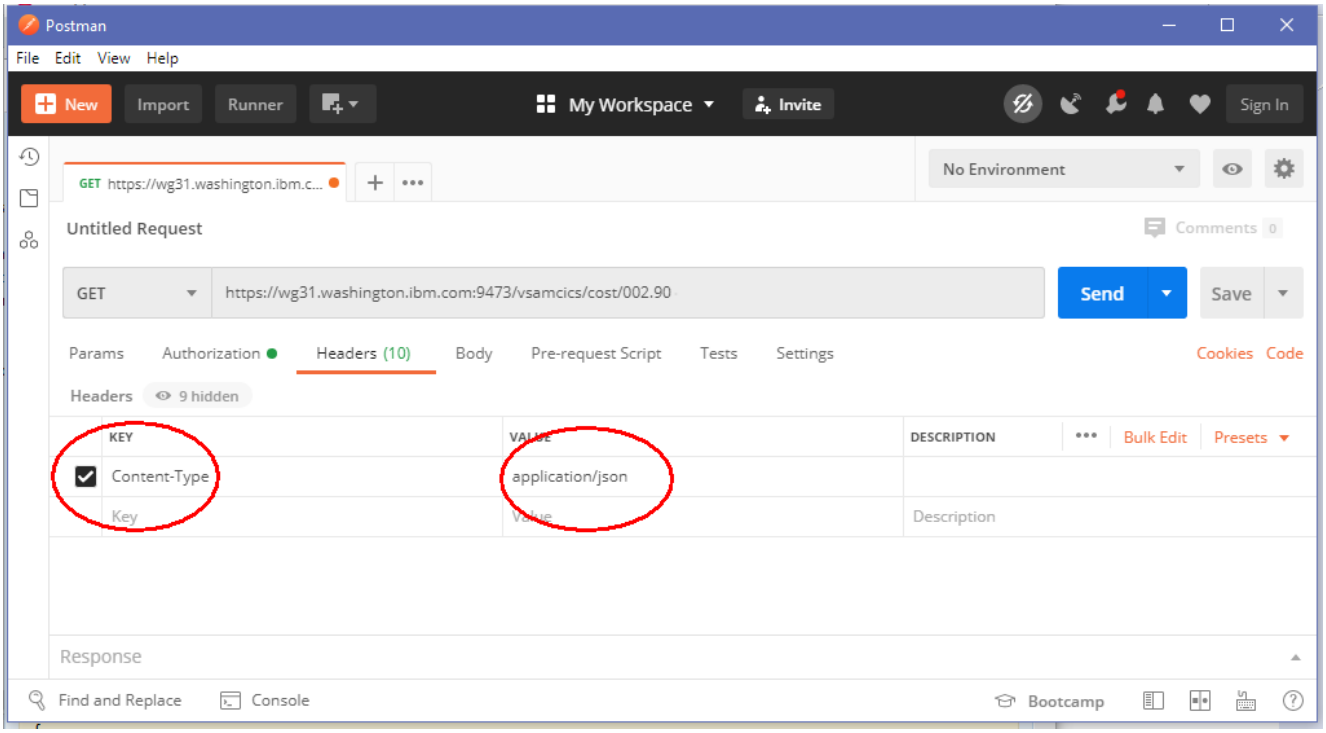
___2. Next, select the *Authorization* tab to enter an authorization identity and password. Use the pull down arrow to select *Basic Auth* and enter *USER1* as the *Username* and *USER1* as the Password.



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.

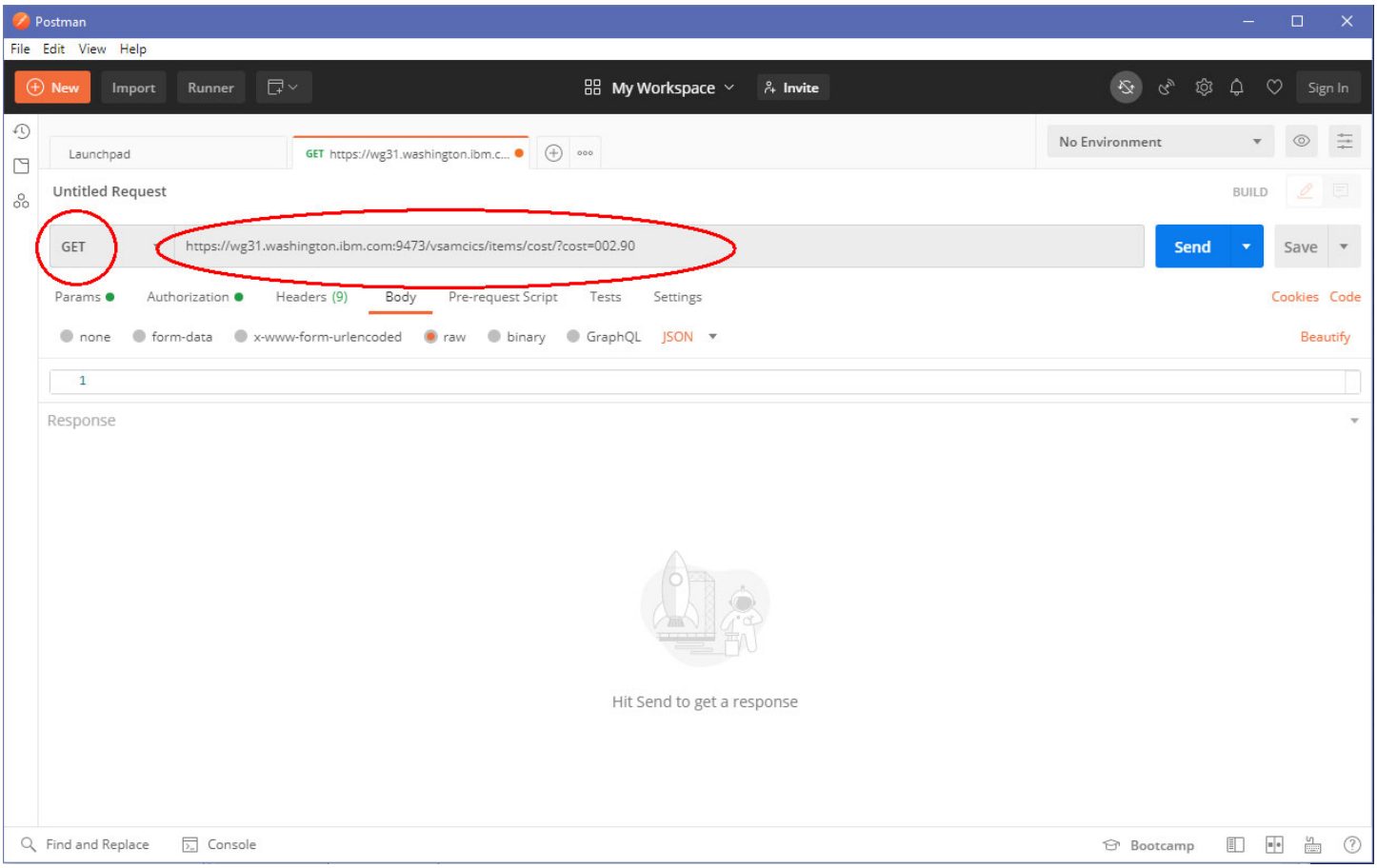
Mitch Johnson (mitchj@us.ibm.com)

___3. Next, select the *Headers* tab. Under *KEY* use the code assist feature to enter ***Content-Type***, and under *VALUE*, use the code assist feature to enter ***application/json***.

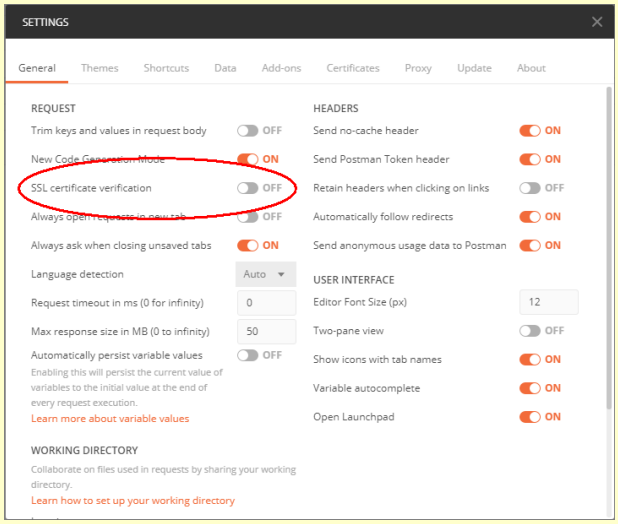


Tech-Tip: Code assist simply means that when text is entered in field, all the valid values for that field that match the typed text will be displayed. You can select the desired value for the field from the list displayed and that value will populate that field.

___4. To test the *selectByCost* service, use the down arrow to select **GET** and enter ***https://wg31.washington.ibm.com:9473/vsamcics/items/cost/?cost=002.90*** in the URL area (see below) to select all items that cost 002.90.

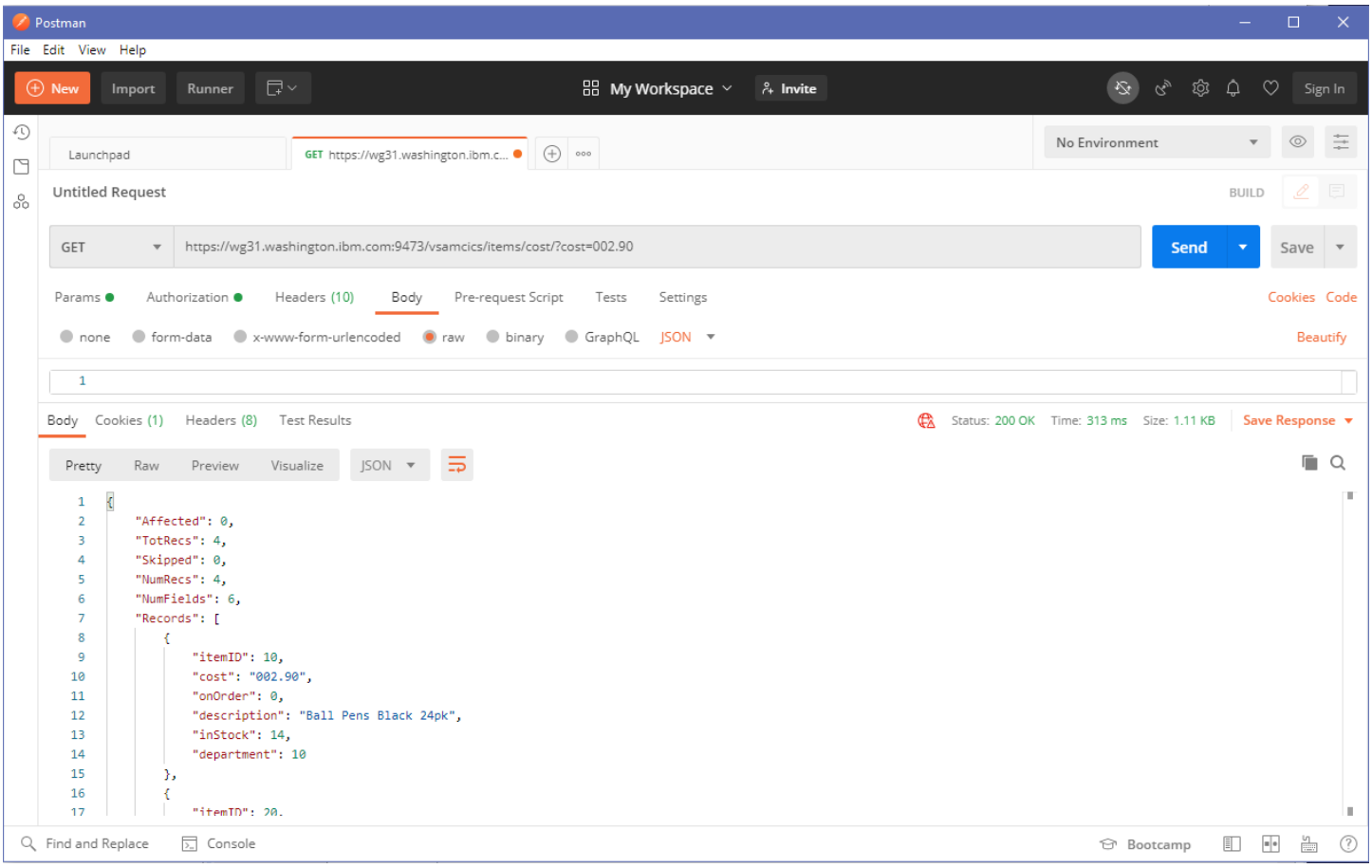


Tech-Tip: The Postman settings have been changed to disable *SSL certificate verification*, a settings option.



Mitch Johnson (mitchj@us.ibm.com)

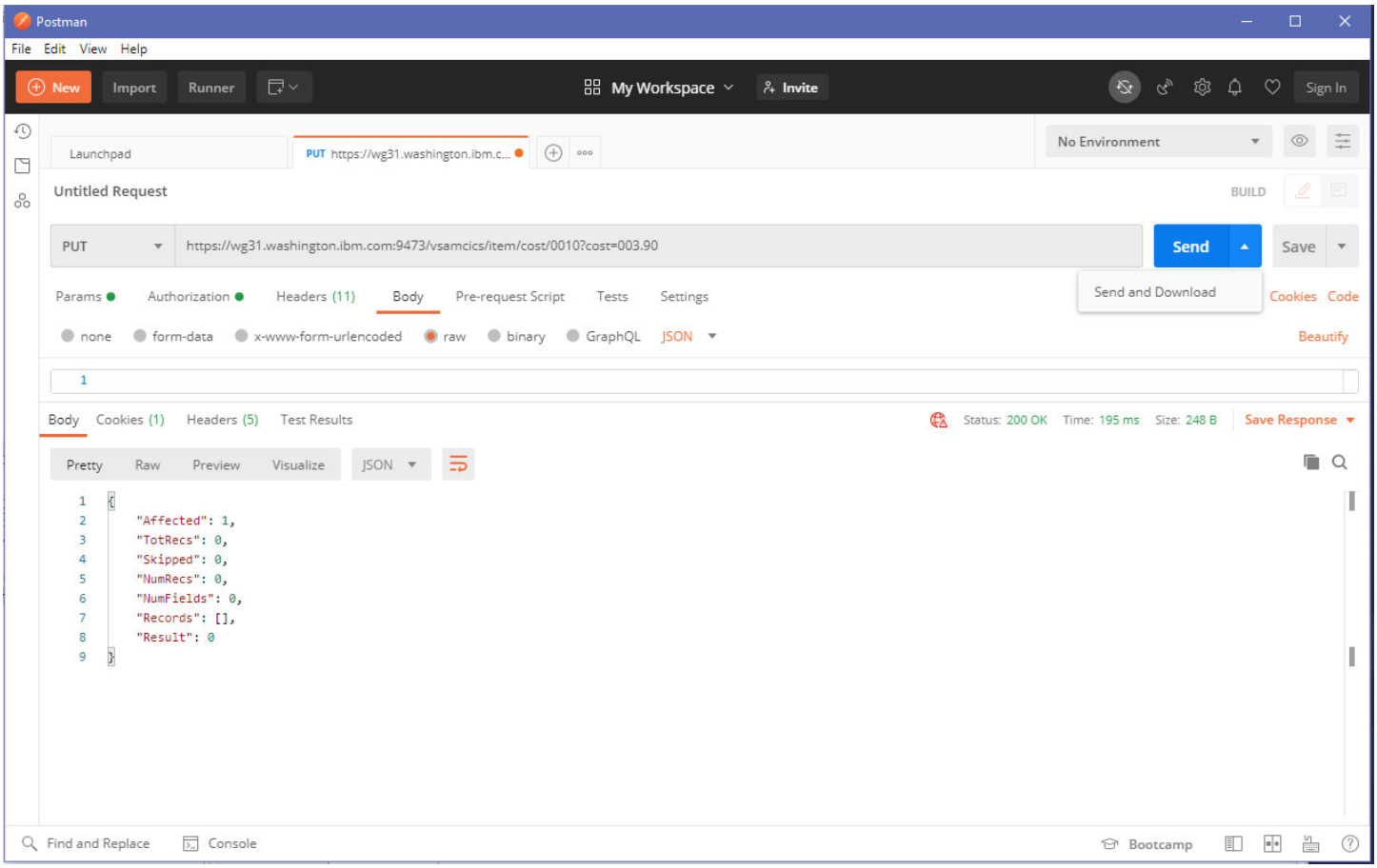
___5. Next select the *Body* tab and select the *raw* radio button. Then press the **Send** button. A response message should come back indicating the service has been started and other details about the service. You may have to 'drag' the response body area up to display the response message. You should see that 4 records that match the cost were returned.



___6. Use the table provided earlier and try other amounts in the URL and observe the results. If you use an amount without a corresponding item,e.g *https://wg31.washington.ibm.com:9473/vsamcics/cost/003.90*, your should see zero records returend (*NumRecs=0).*

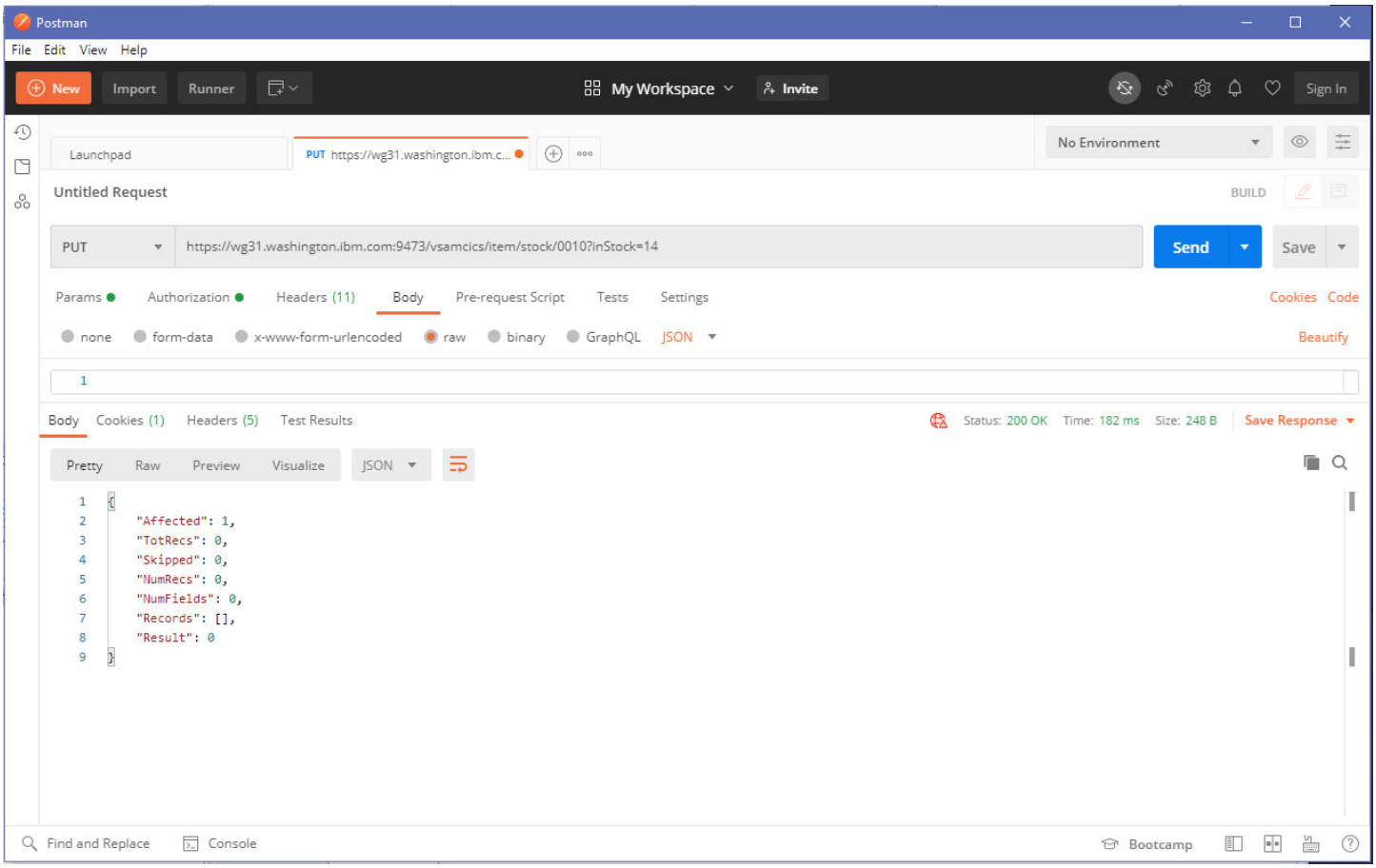
Mitch Johnson (mitchj@us.ibm.com)

___7. To test the *updateCost* service, use the down arrow to select **PUT** and enter ***https://wg31.washington.ibm.com:9473/vsamcics/item/cost/0010?cost=003.90*** in the URL area (see below) to change the cost for item 0010 to 003.90. Press the **Send** button. Note that one record was changed (affected). Observe the behavior when an invalid item ID is provided. A response code was not defined for this service so the runtime still returned a 200 status code.



Tech-Tip: Use a GET to URL https://wg31.washington.ibm.com:9473/vsamcics/item/0010 to display the item and confirm the update has taken place.

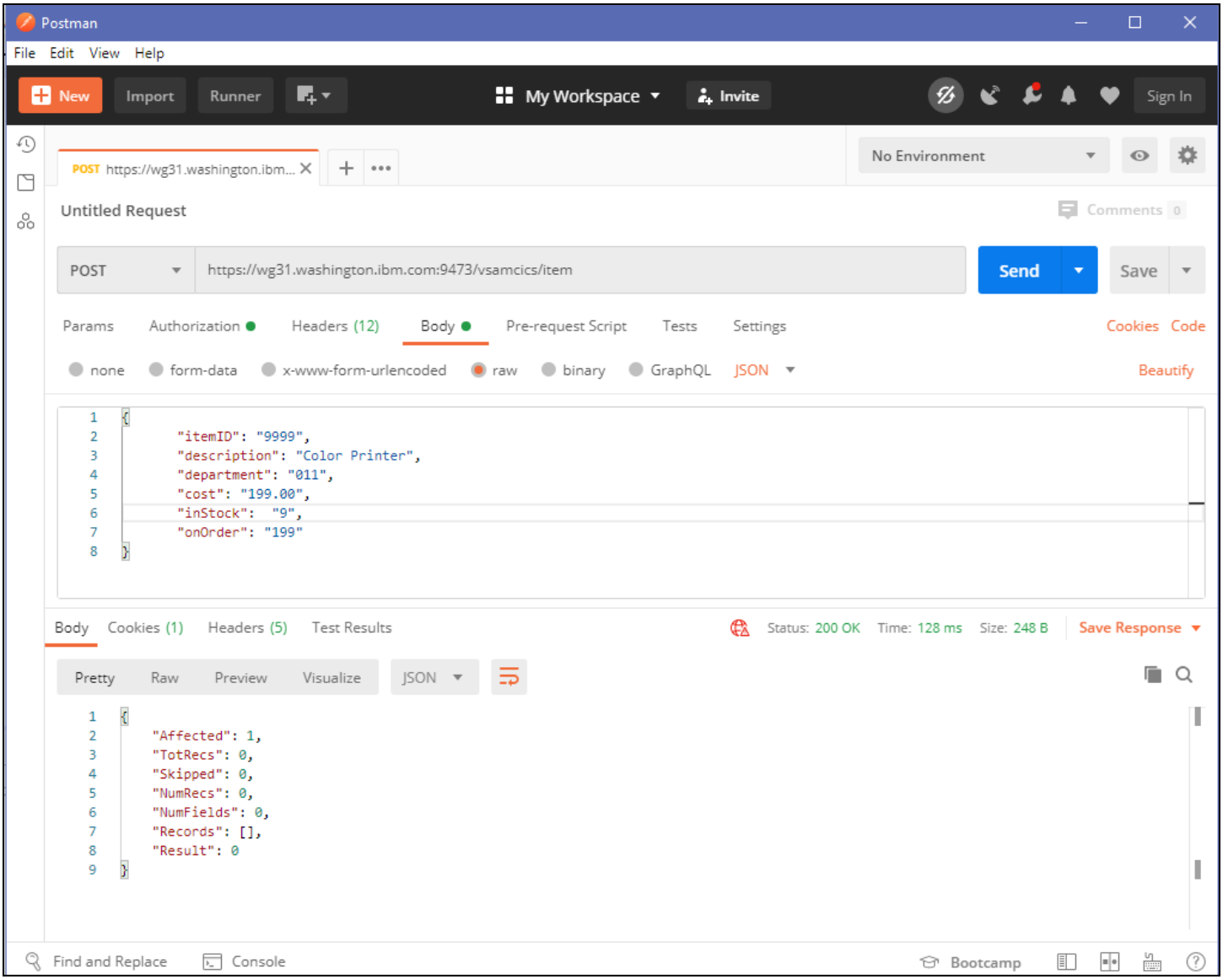
___8. To test the *updateInStock* service. use the down arrow to select **PUT** and enter ***https://wg31.washington.ibm.com:9473/vsamcics/item/stock/0010?inStock=14*** in the URL area (see below) and press the **Send** button. Note that one record was changed (affected). Observe the behavior when an invalid item ID is provided. A response code was not defined for this service so the runtime still returned a 200 status code.



Tech-Tip: Use a GET to URL https://wg31.washington.ibm.com:9473/vsamcics/item/0010 to display the item and confirm the update has taken place.

___9. To test the *insertItem* service use the down arrow to select **POST** and enter ***https://wg31.washington.ibm.com:9473/vsamcics/item*** in the URL area (see below). Enter the JSON below in the *Body* area. and press the **Send** button. Note that one record was changed (affected). Observe the behavior when an invalid item ID is provided. A response code was not defined for this service so the runtime still returned a 200 status code.

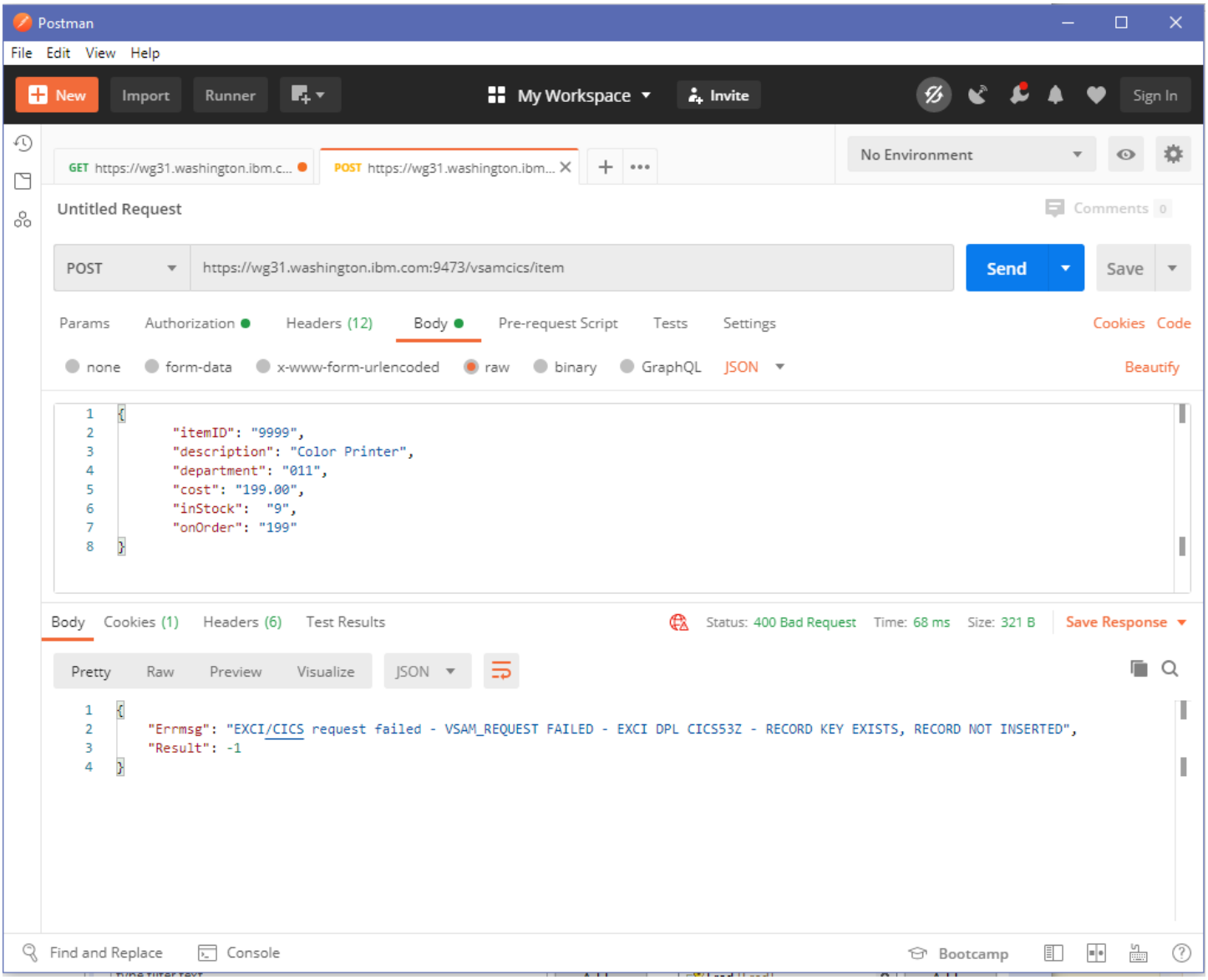
```
{
    "itemID": "9999",
    "description": "Color Printer",
    "department": "011",
    "cost": "199.00",
    "inStock":  "9",
    "onOrder": "199"
}
```



Mitch Johnson (mitchj@us.ibm.com)

Tech-Tip: Clear the Body area and use a GET to URL https://wg31.washington.ibm.com:9473/vsamcics/item/9999 to display the item and confirm the item has been inserted.

If the POST request in Step 9 is repeated the results will be quite different, see below.



Summary

You use DVM to develop 8 services. The SAR files for the 8 services were imported in the API Editor of z/OS Connect EE. The API Editor was used to develop a RESTful API. You have verified the API. The API layer provided a further level of abstraction and allows a more flexible use of HTTP verbs, and better mapping of data via the API editor function.

Mitch Johnson (mitchj@us.ibm.com)