

IMS Hands-On Lab

z/OS Connect and IMS OpenAPI 3 – PART 1

Introduction:

This is an opportunity to get your hands dirty and play with the new IMS support for OpenAPI 3 and expose an IMS transaction as an API. This exercise uses the z/OS Connect Designer to create a IMS z/OS Asset and create APIs to access the IMS Phonebook transaction.

PART 1 - Create the API to GET a contact's information from the phonebook

PART 2 – Create the API to POST (add) a contact to the phonebook

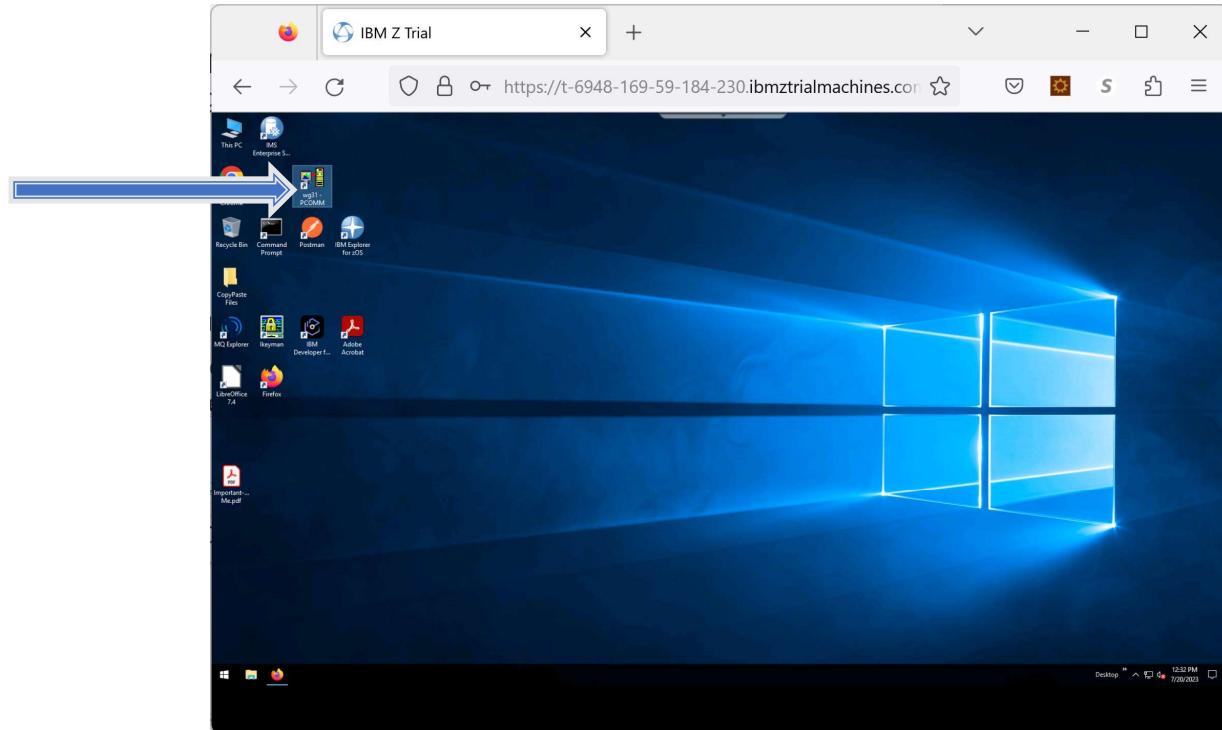
PART 3 – Create the API to UPDATE the contact you added

PART 4 - Create the API to DELETE the contact you added

In this lab, the IMS Phonebook sample application (IVTNO) is used as the target IMS application program to map the example IMS OpenAPI3 definition.

REVIEW the IMS Environment

Check out the IMS environment by double-clicking the **wg31 pcomm** icon



(note that shift- Enter is used as the enter function)

- Logon to TSO using USER1
 - Password is user1

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
[Icons]

Your IP:10.1.1.1          Terminal: TCP00001
07/26/23      z/OS V2R5 LVLI PUT2203/RSU2203      08:38:24
***** Welcome to the Washington Systems Center ****
** >> z/OS Connect Workshop System <<
** *****

Enter TSO userid ... for a TSO session
CICS     ... for a CICS session
IMS      ... for an IMS session

Enter Command ==> tso USER1_

```

24/030

Connected to remote server/host wsg31 using lu/pool TCP00001 and port 23

Desktop 8:38 AM 7/26/2023

- ISPF option D allows access to SDSF

```

Session A - [32 x 80]
File Edit View Communication Actions Window Help
[Icons]

Menu Utilities Compilers Options Status Help
ISPF Primary Option Menu
Option ==> D

0 Settings   Terminal and user parameters
1 View        Display source data or listings
2 Edit         Create or change source data
3 Utilities   Perform utility functions
4 Foreground  Interactive language processing
5 Batch       Submit job for language processing
6 Command    Enter TSO or Workstation commands
7 Dialog Test Perform dialog testing
9 IBM Products IBM program development products
10 SCLM      SW Configuration Library Manager
11 Workplace  ISPF Object/Action Workplace
M More        Additional IBM Products
D SDSF       Spool Search and Display Facility
0 QM          QM MVS/ESR Utility Menu

Enter X to Terminate using log/list defaults

04/015
Connected to remote server/host wsg31 using lu/pool TCP00001 and port 23
Desktop 8:39 AM 7/26/2023

```

- Go to LOG

```

Display Filter View Print Options Search Help
-----+
SDSF MENU V2R5MO ADDCPL S0W1          LINE 1-27 (73)
COMMAND INPUT ==> DD                      SCROLL ==> PAGE
NP NAME          Description      Group   Status
A Active users    Jobs
I Input queue     Jobs
O Output queue    Output
H Held output queue Output
ST Status of jobs Jobs
JG Job groups     JES
SYM System symbols System
LOG System log     Log
SR System requests Log
MRS Members in the MRS JES
JC Job classes    JES
SE Scheduling environments WLM
MWS Managed resources WLM
ENC Entries       WLM
PS Processes      QWVS
SYS System information System
ENO Enqueues      System
ENQC Enqueues contention System
ENDQ Enqueued data sets Sysplex
DYNX Dynamic exits  System
AS Address space memory Jobs
INIT Initiators   JES
PR Printers       JES
PUN Punches       JES
RDR Readers       JES
LINE Lines         Network
NODE Nodes        Network

```

MP A 04/024
Connected to remote server/host wgl1 using luipool TCP00001 and port 23
Windows Taskbar: Desktop, 8:00 AM, 7/26/2023

You will see the outstanding prompts for IMS (IVP1) and IMSConnect IMS15HWS)

- Answer the outstanding prompt number (*nn*) for IMS (IVP1) to check on message region availability and the transaction ENQCT:
 - **/nn/DIS TRAN IVTNO.**
- Make note of the value for ENQCT which will keep track of the number accesses of the IVTNO transaction

```

Display Filter View Print Options Search Help
-----+
SDSF SYSLOG          0.101 S0W1 S0W1 07/26/2023 ZW        4,784  COLUMNS 52- 131
COMMAND INPUT ==> _                                     SCROLL ==> CSR
0290 R 05 /DIS TRAN IVTNO.
0090 IEE6001 REPLY TO 05 IS;/DIS TRAN IVTNO.
0090 DFS0001  TRAN      CLS ENQCT  QCT  LCT  PLCT CP NP LP SEGSZ SEGNO
      PRLM   RC   IVP1
0090 DFS0001  IVTNO   1   0   0 65535 65535 1 1 1   0   0
      NONE   0   IVP1
0090 DFS0001  PSBNAME: DFSIVP1   IVP1
0090 DFS0001  *23207/085309* IVP1
0090 *07 DFS0001 *IMS READY* IVP1
S READY* IVP1
IMS CONNECT READY* IMS15HWS
***** BOTTOM OF DATA *****

MP A 04/021  
Connected to remote server/host wgl1 using luipool TCP00001 and port 23  
Windows Taskbar: Desktop, 8:07 AM, 7/26/2023

```

- You can also issue a display active command to verify that the dependent regions are activating
 - **/nn/DIS A.**

```

Display Filter View Print Options Search Help
SDF SYLOG 0 101 S0W1 S0W1 07/20/2023 2W 4,785 COLUMNS 52- 131
COMMAND INPUT ==> /nn/DIS A.
0090 DFS000I REGID JOBNAME TYPE TRRN/STEP PROGRAM STATUS
0090 DFS000I IVP1 TP WAITING
0090 DFS000I 1 IMSMSG1 TP WAITING
0090 DFS000I IVP1
0090 DFS000I JPRGN JHP NONE IVP1
0090 DFS000I JPRGN JBP NONE IVP1
0090 DFS000I BATCREG BMP NONE IVP1
0090 DFS000I FPRGN FP NONE IVP1
0090 DFS000I DBTRGN DBT NONE IVP1
0090 DFS000I IMS15RC1 DBRC IVP1
0090 DFS000I IMS15RC1 DLS IVP1
0090 DFS000I VTAV ACB OPEN -LOGONS ENABLED IVP1
0090 DFS000I IMSL=N/A,N/A APPC STATUS=DISABLED TIMEOUT=0
MAXC#= 5000 IVP1
0090 DFS000I OTMA GROUP=OTMAGRP STATUS=ACTIVE IVP1
0090 DFS000I APPC/OTMA SHARED QUEUE STATUS - LOCAL=INACTIVE
0090 DFS000I APPC/OTMA SHARED QUEUES LOGGING=N IVP1
0090 DFS000I APPC/OTMA RRS MAX TCBS - 40 ATTACHED TCBS - 2 QUEUED
RRRSKS= 0 IVP1
0090 DFS000I APPLID=IMS15 GRNAME= STATUS=DISABLED
0090 DFS000I TCP/IP.GENIVSID= STATUS=DISABLED IVP1
0090 DFS000I LINE ACTIVE-IN = 1 ACTIV-OUT = 0 IVP1
0090 DFS000I NODE ACTIVE-IN = 0 ACTIV-OUT = 0 IVP1
0090 DFS000I *23207/085407* IVP1
0090 *08 DFS998I *IMS READY* IVP1
S READY* IVP1
04/021
Connected to remote server/host wip31 using lu/pool TCP00001 and port 23
Desktop 8:48 AM 7/26/2023

```

- Answer the outstanding prompt number (**zz**) for IMS Connect (IMS15HWS)
 - Respond with **/zzVIEWHWS**
 - Make sure Datastore=IVP1 status is active

```

Display Filter View Print Options Search Help
SDF SYLOG 0_101 S0W1 S0W1 07/26/2023 2W 4,739 COLUMNS 02- 81
COMMAND INPUT ==> /03viewhws_
PARLM RC IVP1
S DFS000I IVTNO 1
N 4200000 S0W1 23207 08:48:39.44 STC01676 00000090 DFS000I
S NONE 0 IVP1
N 4200000 S0W1 23207 08:48:39.44 STC01676 00000090 DFS000I PSBNAME:
N 4200000 S0W1 23207 08:48:39.44 STC01676 00000090 DFS000I *23207/08483
W 4200000 S0W1 23207 08:48:39.44 STC01676 00000090 *05 DFS998I *IMS READY*
4200000 S0W1 08.48.38 STC01676 *05 DFS998I *IMS READY* IVP1
8000000 S0W1 13.34.00 STC01685 *03 HWS0000I *IMS CONNECT READY* IMS15HWS
***** BOTTOM OF DATA *****

04/031
Connected to remote server/host wip31 using lu/pool TCP00001 and port 23
Desktop 8:48 AM 7/26/2023

```

```

Display Filter View Print Options Search Help
SDF SYLOG 0.101 S0W1 S0W1 07/26/2023 2W 4,759 COLUMNS 52- 131
COMMAND INPUT ==> SCROLL ==> CSR
0090 HWS00001I ODBA IMSPLEX MEMBER=IMS15HWS TARGET MEMBER=PLEX1
0090 HWS00001I DATASTORE=IPV1 STATUS=ACTIVE
0090 HWS00001I GROUP=OTMAGRP MEMBER=HWSMEM
0090 HWS00001I TARGET MEMBER=OTNAMEM STATE=AVAIL
0090 HWS00001I DEFAULT REROUTE NAME=HWSDEF IMS VERSION=15.1
0090 HWS00001I RACF APPL NAME=IMSAAPL MULTIRTP= CASCADE=
0090 HWS00001I OTMA ACCE AGING VALUE=999999
0090 HWS00001I OTMA ACK TIMEOUT VALUE=120
0090 HWS00001I OTMA MAX INPUT MESSAGE=5000
0090 HWS00001I SUPER MEMBER NAME= CMO ACK TOO=
0090 HWS00001I IMSPLEX=PLEX1 STATUS=NOT ACTIVE
0090 HWS00001I MEMBER=IMS15HWS TARGET=PLEX1
0090 HWS00001I NO ACTIVE ODBM
0090 HWS00001I NO ACTIVE MSC
0090 HWS00001I NO ACTIVE ISC
0090 HWS00001I PORT=4000 STATUS=ACTIVE KEEPAV=0 NUMSOC=1
EDIT= TIMEOUT=0
0090 HWS00001I NO ACTIVE CLIENTS
0090 HWS00001I PORT=5555D STATUS=ACTIVE KEEPAV=0 NUMSOC=1
EDIT= TIMEOUT=6000
0090 HWS00001I NO ACTIVE CLIENTS
0090 HWS00001I NO ACTIVE RMTIMSCON
0090 HWS00001I NO ACTIVE RMTCICS
0090 DFS2884I EXTERNAL TRACE DATASET DFSTR001 FULL - SWITCHING TO DFSTR002 .
IPV1
IMS CONNECT READY* IMS15HWS
S READY* IPV1
***** BOTTOM OF DATA *****

```

MR A 04/021
Connected to remote server/host wg31 using lu/pool TCP00001 and port 23

- If PORT=4000 shows an inactive status, issue the command **/zzSTARTPT 4000**

If you want to test the transaction, prior to creating the API (just to make sure everything is working... logon to IMS by opening up another pcomm session by double-clicking on the **wg31 Pcomm** icon on the the desktop

```

Your IP:10.1.1.1 Terminal: TCP00003
07/26/23 z/OS V2R5 LVLI PUT2203/RSU2203 09:03:44
*****
** Welcome to the Washington Systems Center **
** >> z/OS Connect Workshop System << **
****

Enter TSO userid ... for a TSO session
 CICS ... for a CICS session
 IMS ... for an IMS session

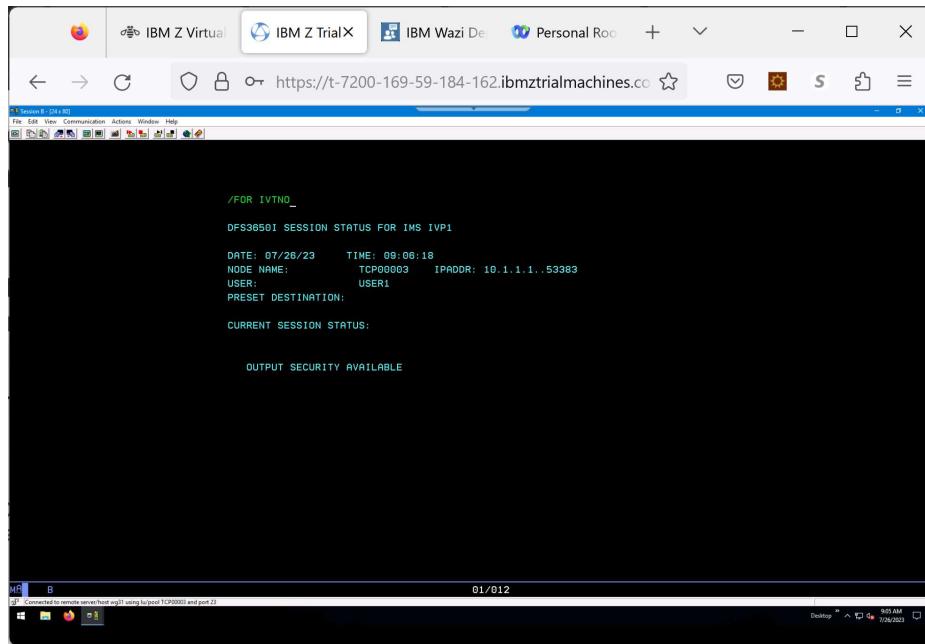
Enter Command ==> IMS_

```

MR B 24/024
Connected to remote server/host wg31 using lu/pool TCP00001 and port 23

Userid and password are USER1/user1

Once you are logged on, use format IVTNO (/FOR IVTNO)

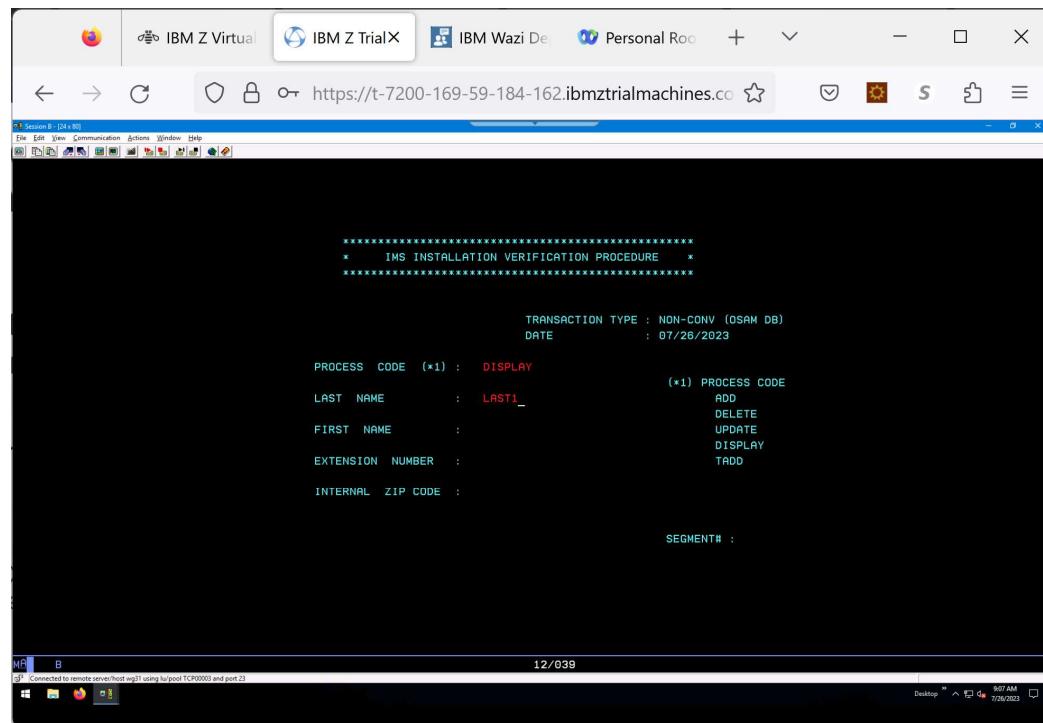


```
/FOR IVTNO_
DFS3050I SESSION STATUS FOR IMS IVP1
DATE: 07/28/23    TIME: 09:08:18
NODE NAME:      TCP00003  IPADDR: 10.1.1.1.5383
USER:          USER1
PRESET DESTINATION:

CURRENT SESSION STATUS:

OUTPUT SECURITY AVAILABLE
```

Test the transaction using a process code of **DISPLAY** and a LAST NAME of **LAST1** (by the way, this is what we will be exposing using an API)



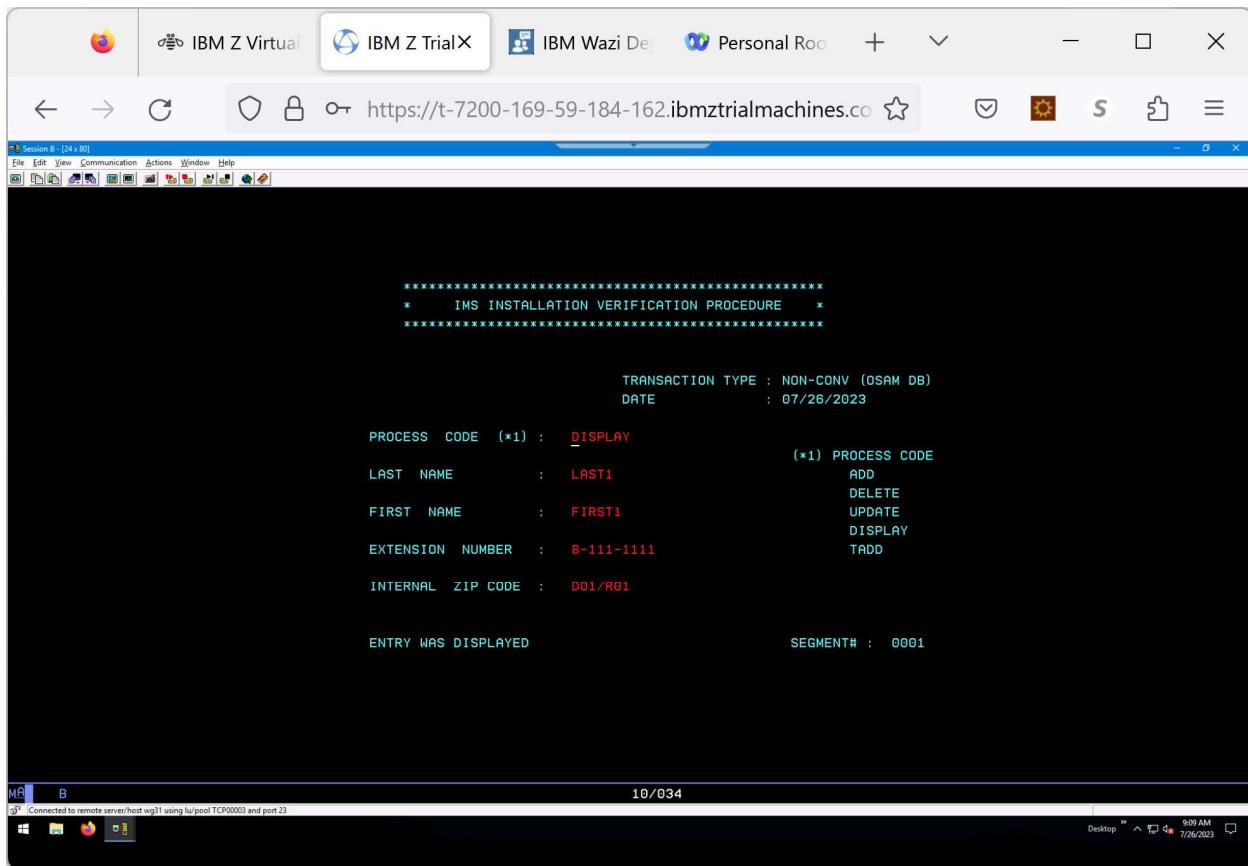
```
*****
*   IMS INSTALLATION VERIFICATION PROCEDURE  *
*****
```

TRANSACTION TYPE :	NON-CONV (OSAM DB)
DATE :	07/26/2023
PROCESS CODE (*1) :	DISPLAY
LAST NAME :	LAST1_
FIRST NAME :	
EXTENSION NUMBER :	
INTERNAL ZIP CODE :	

(*1) PROCESS CODE
ADD
DELETE
UPDATE
DISPLAY
TADD

SEGMENT# :

If everything on the IMS system is working properly, you will receive the results as follows



The screenshot shows a web browser window with the URL <https://t-7200-169-59-184-162.ibmtrialmachines.co>. The page displays an IMS installation verification procedure output. The output includes:

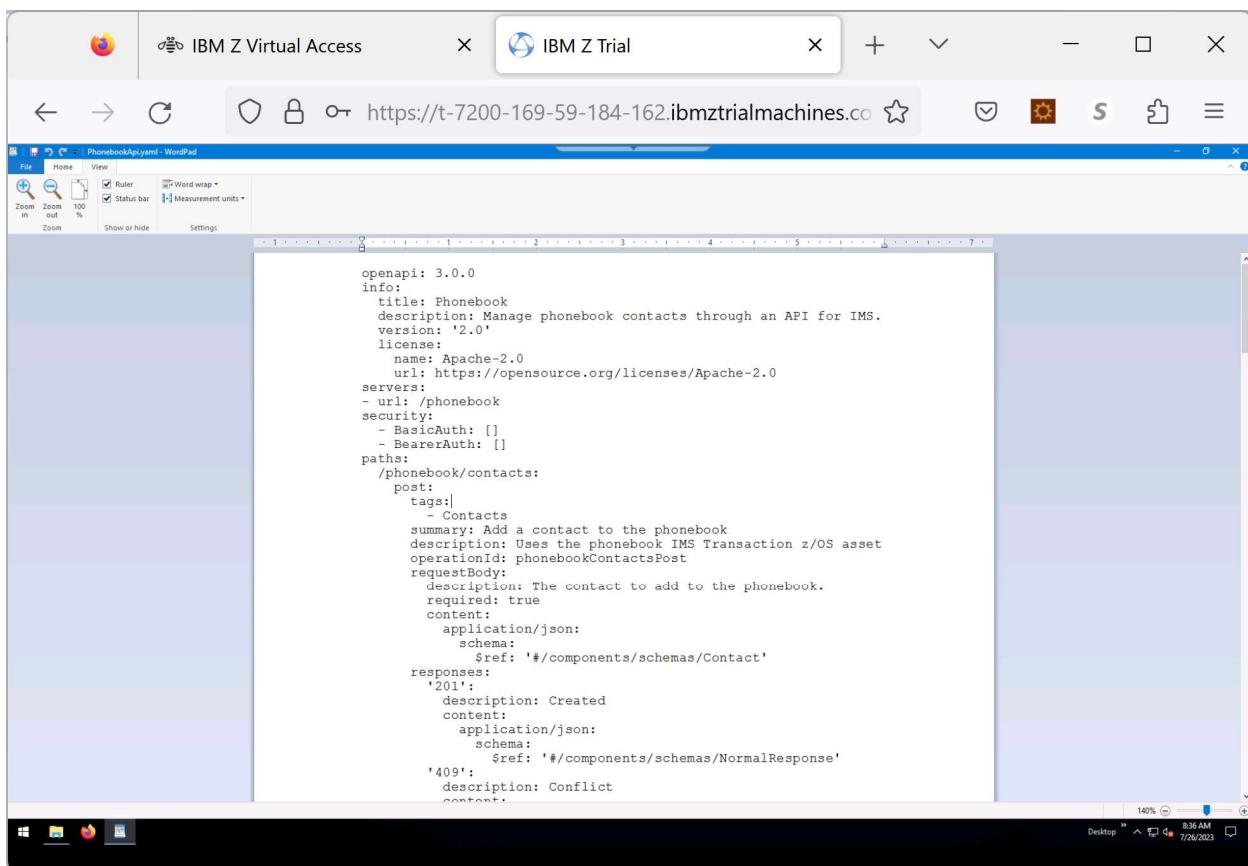
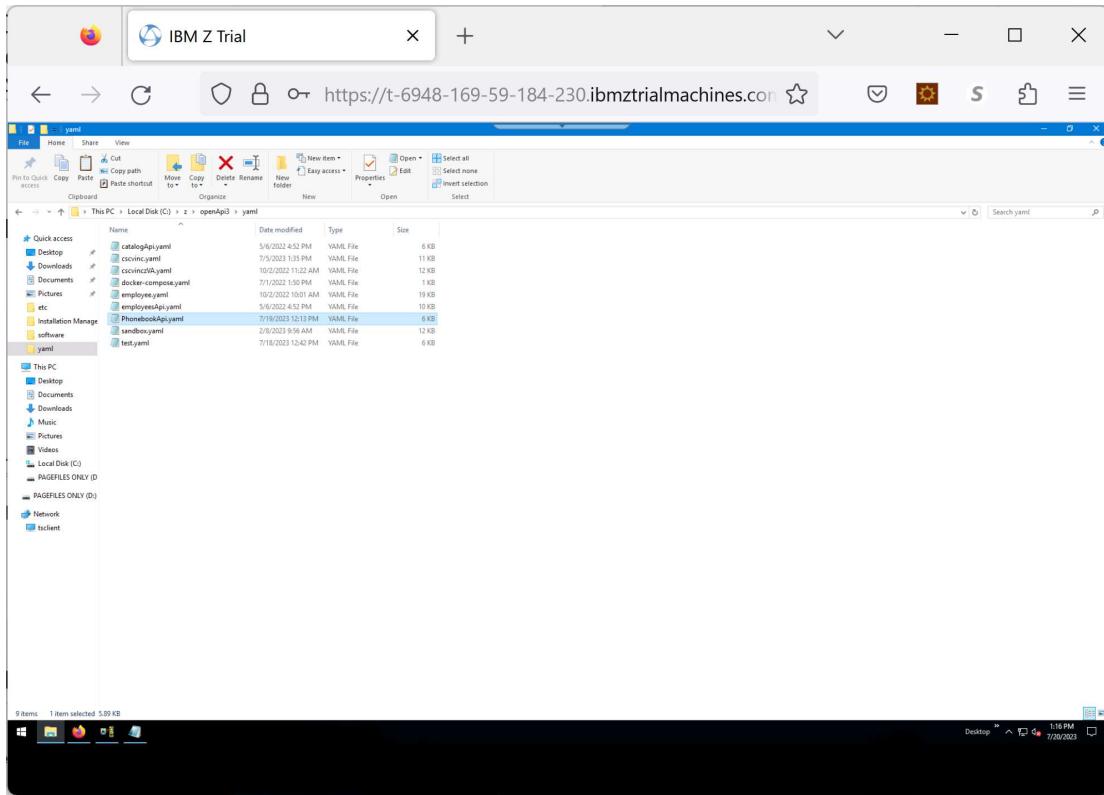
```
*****
*      IMS INSTALLATION VERIFICATION PROCEDURE      *
*****  
TRANSACTION TYPE : NON-CONV (OSAM DB)  
DATE          : 07/26/2023  
  
PROCESS CODE (*1) : DISPLAY          (*1) PROCESS CODE  
LAST NAME       : LAST1            ADD  
FIRST NAME      : FIRST1           DELETE  
EXTENSION NUMBER: 8-111-1111      UPDATE  
INTERNAL ZIP CODE: D01/R01        DISPLAY  
TADD  
  
ENTRY WAS DISPLAYED          SEGMENT# : 0001  
  
10/034
```

At the bottom of the browser window, it says "Connected to remote server/host wazi using lu/pool TCP00003 and port 23". The taskbar at the bottom right shows the date as 7/26/2023 and the time as 9:09 AM.

CREATE an IMS z/OS Connect API

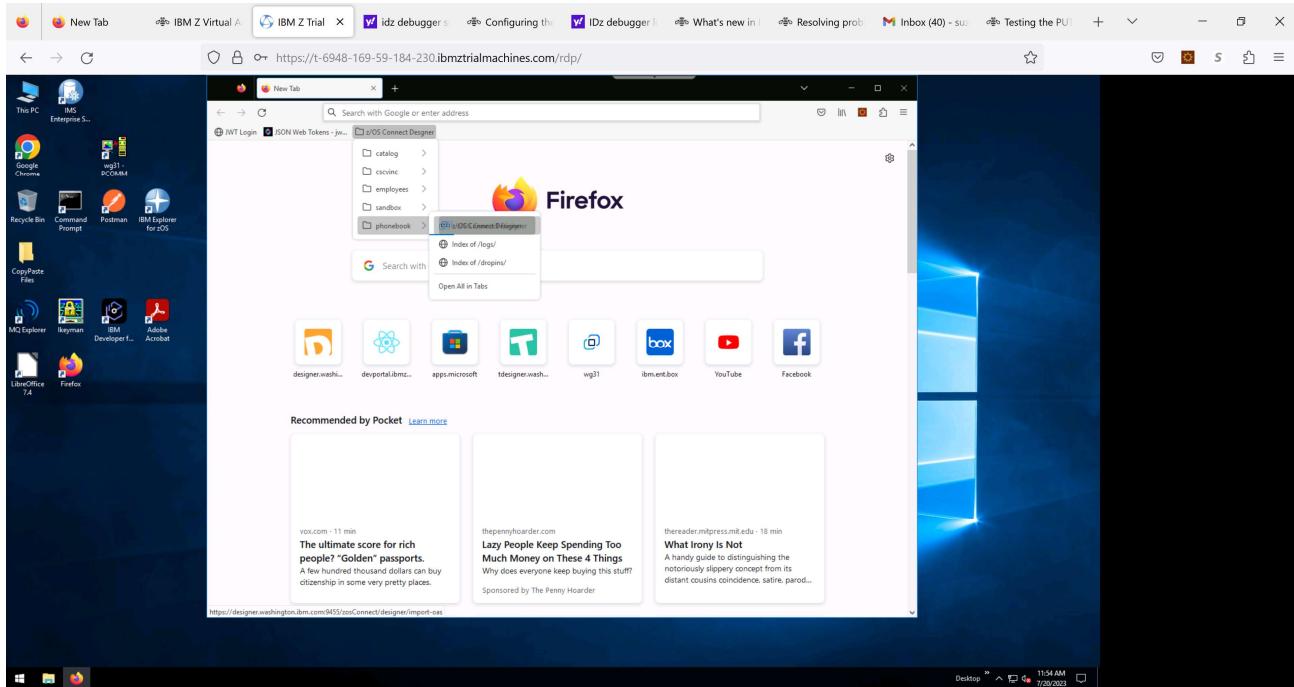
For this lab, a **PhonebookAPI.yaml** file has been provided in the directory

C: z/openapi3/yaml

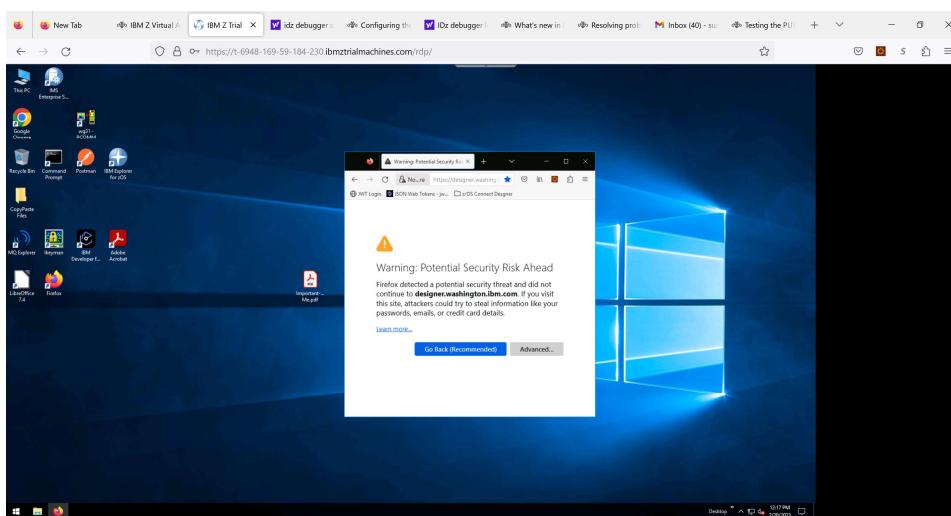


To create the API, open up a **Firefox browser**

- Click on **z/OS Connect Designer > phonebook > zOS Connect designer**
 - Make sure you choose phonebook

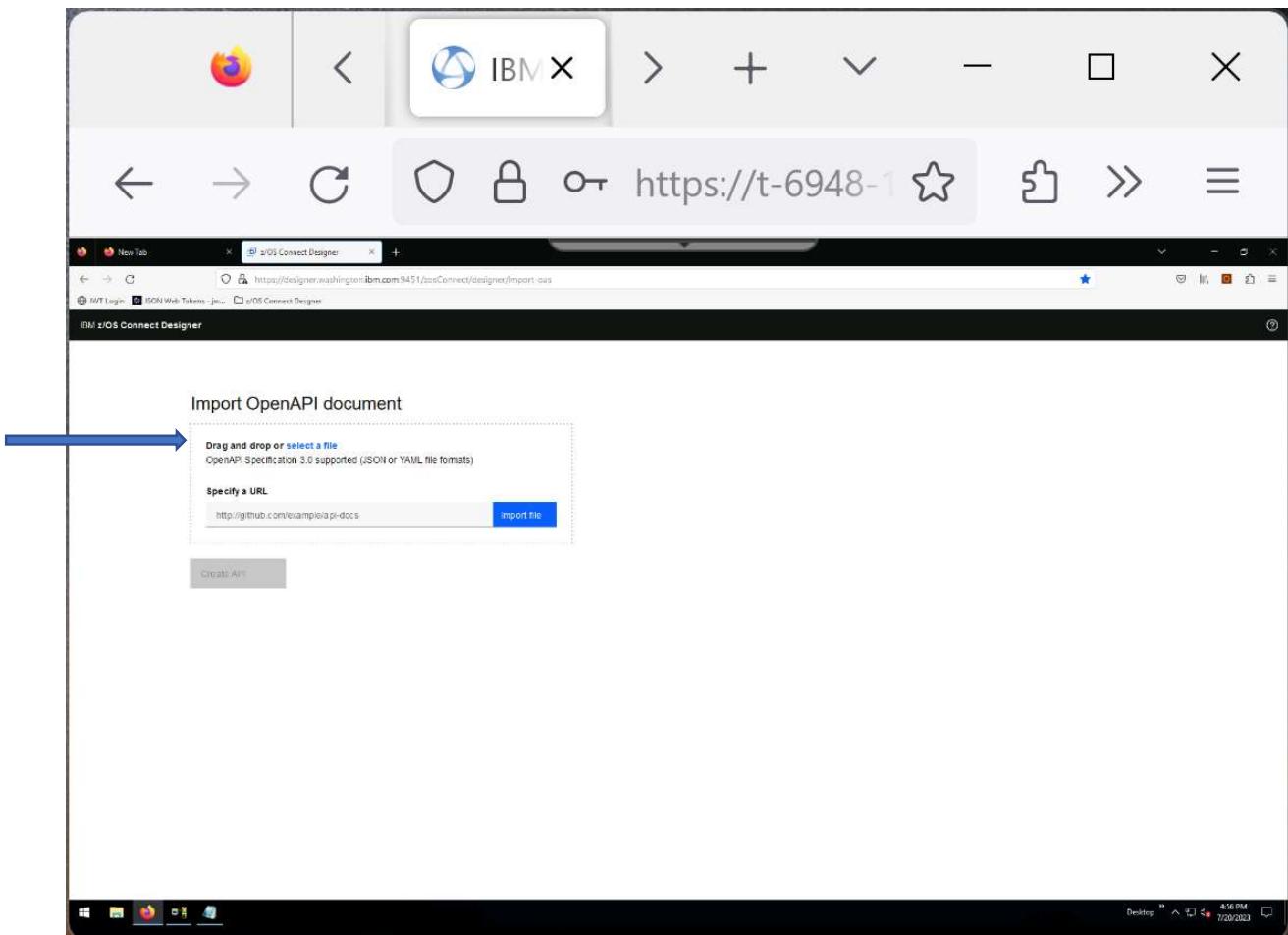


- Accept the risk (Advanced) and wait (it takes a while to set it up)

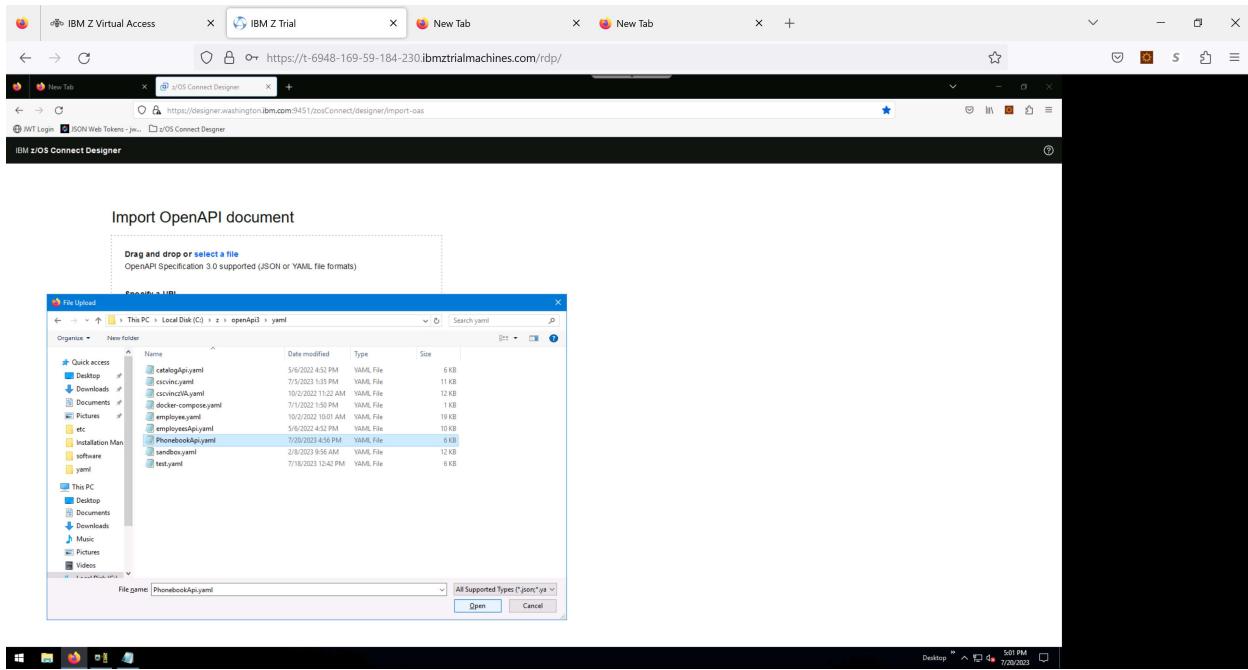


The tool will ask you to choose an OpenAPI document (yaml)

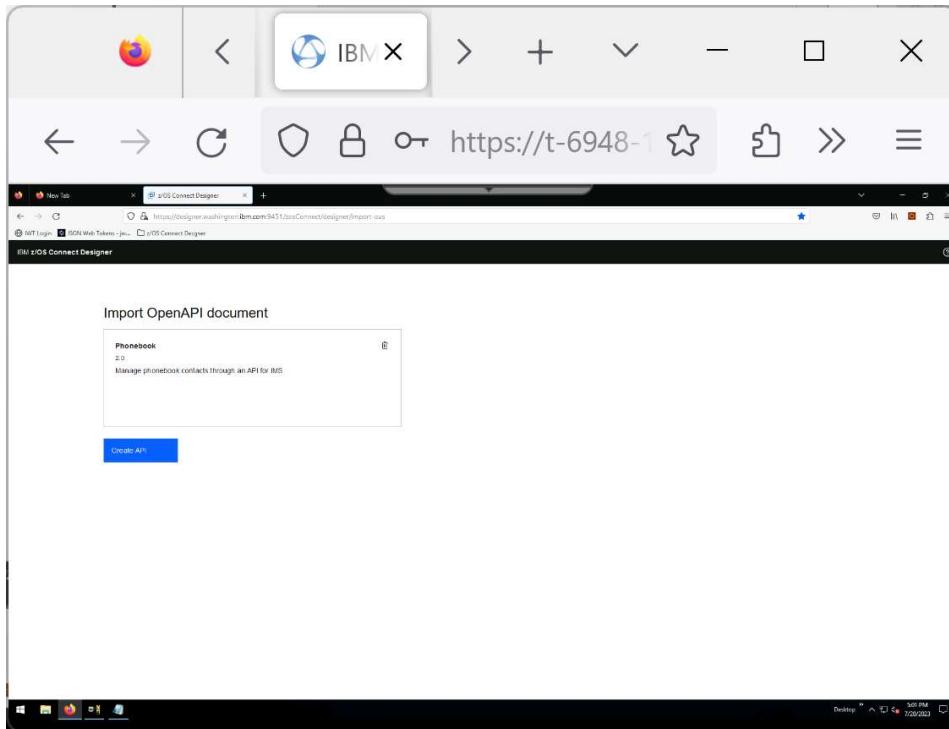
- Click on **select a file**



- Follow the path to **C:/z/openAPI3/yaml/phonebook.yaml**



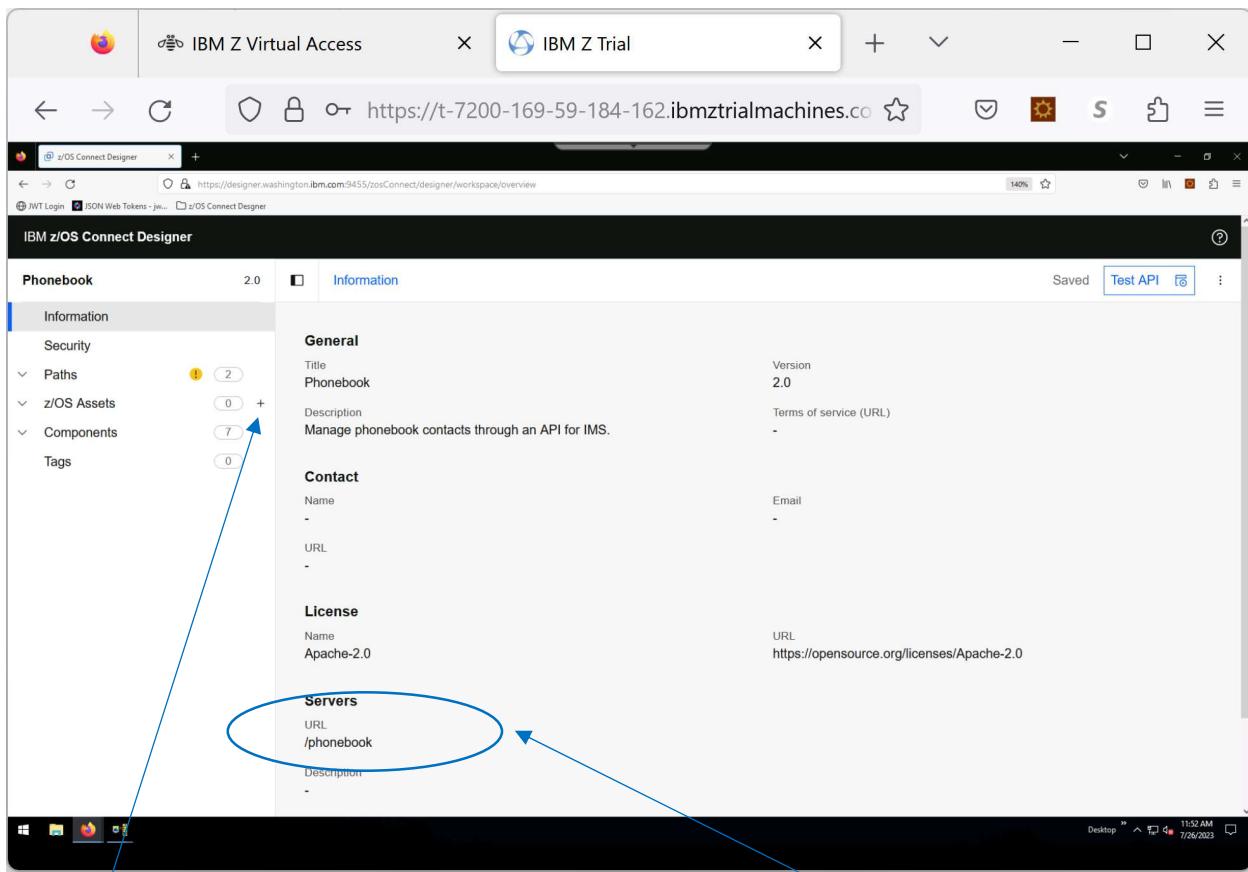
- Click on Create API



The information from the Phonebook.yaml file is now available – this is what the API would expect to request and retrieve when accessing the z/OS asset (IMS transaction)

On the other hand, access to and from the z/OS environment (e.g., the IMS transaction) has to be created as a z/OS asset. Ultimately, the information from the yaml file will need to be mapped to the z/OS asset to complete the API. This is what is known as ‘meet-in-the-middle’ since both sides have to be mapped to each other.

CREATE the z/OS asset

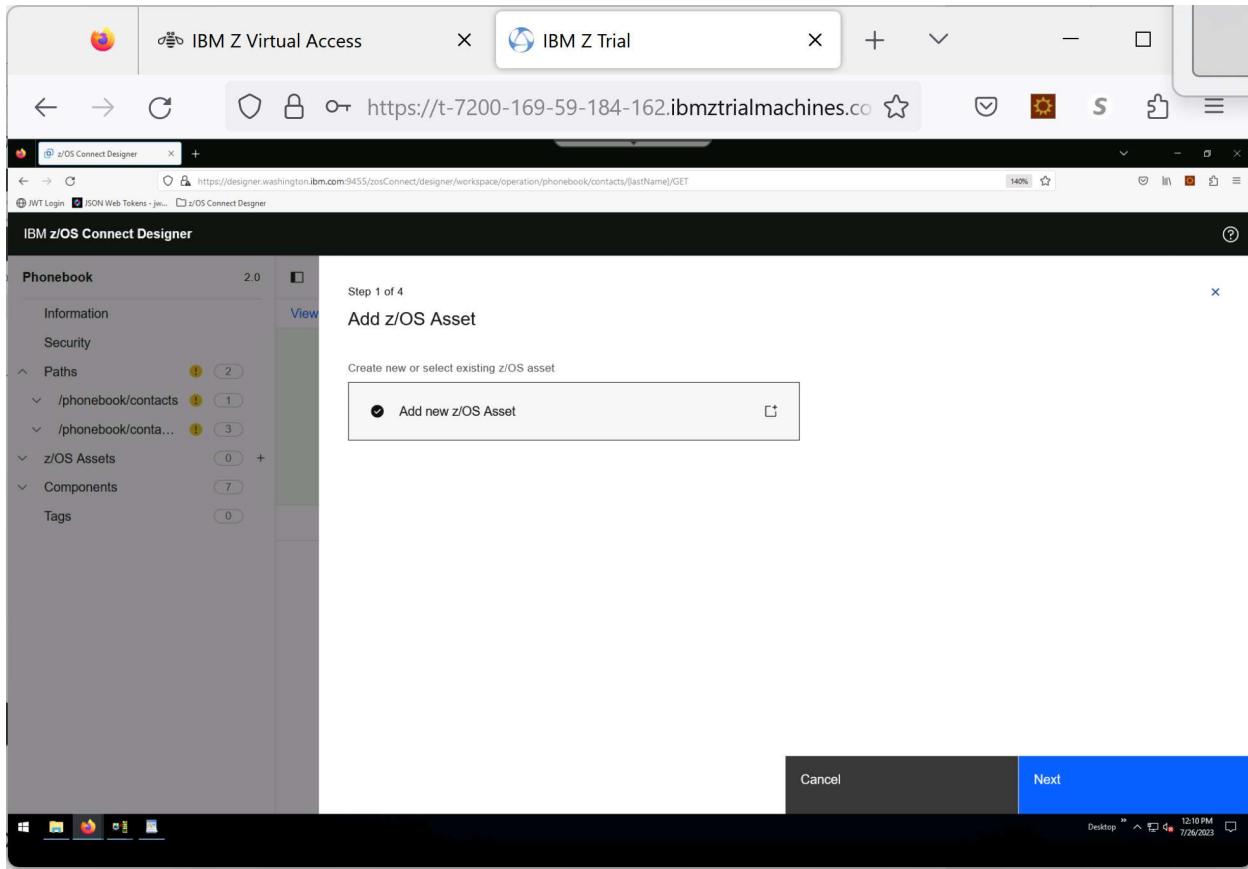


Click on the plus (+) sign by z/OS Assets

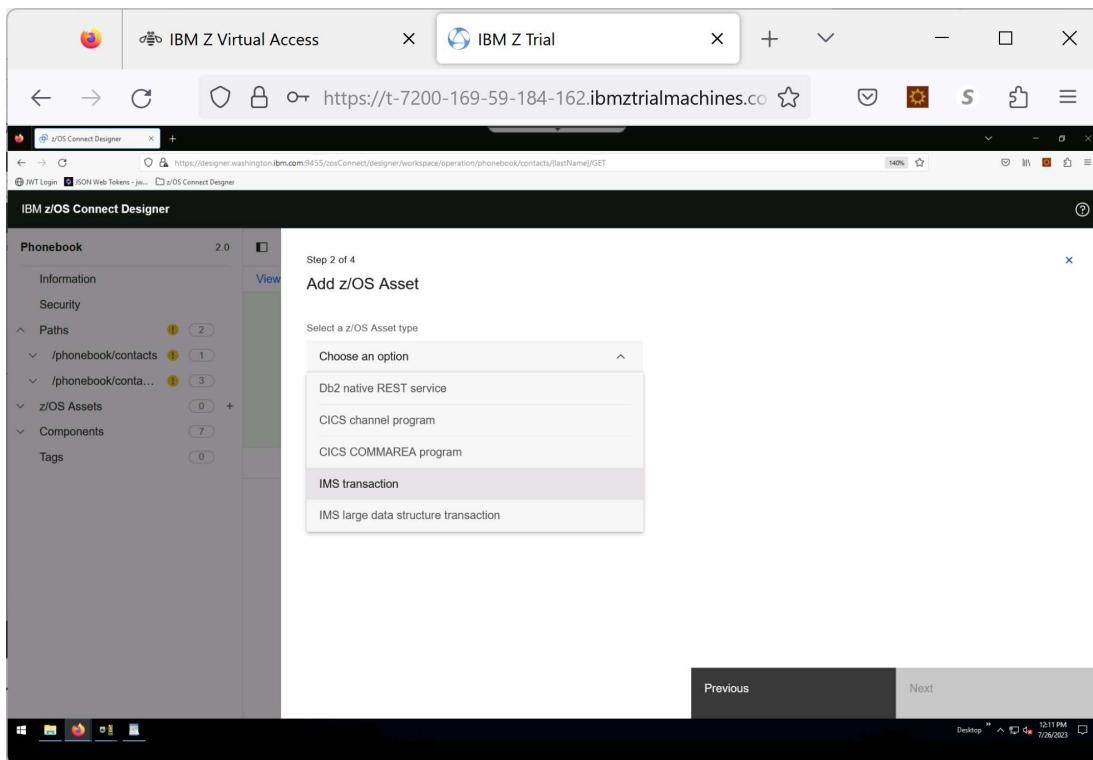
(Note: the URL of the server you will use to test later)

- Select **Add New z/OS Asset** and click Next

The IMS z/OS asset describes the formats of the request and response data structure from IMS and specific characteristics such as the code page, transid, etc.



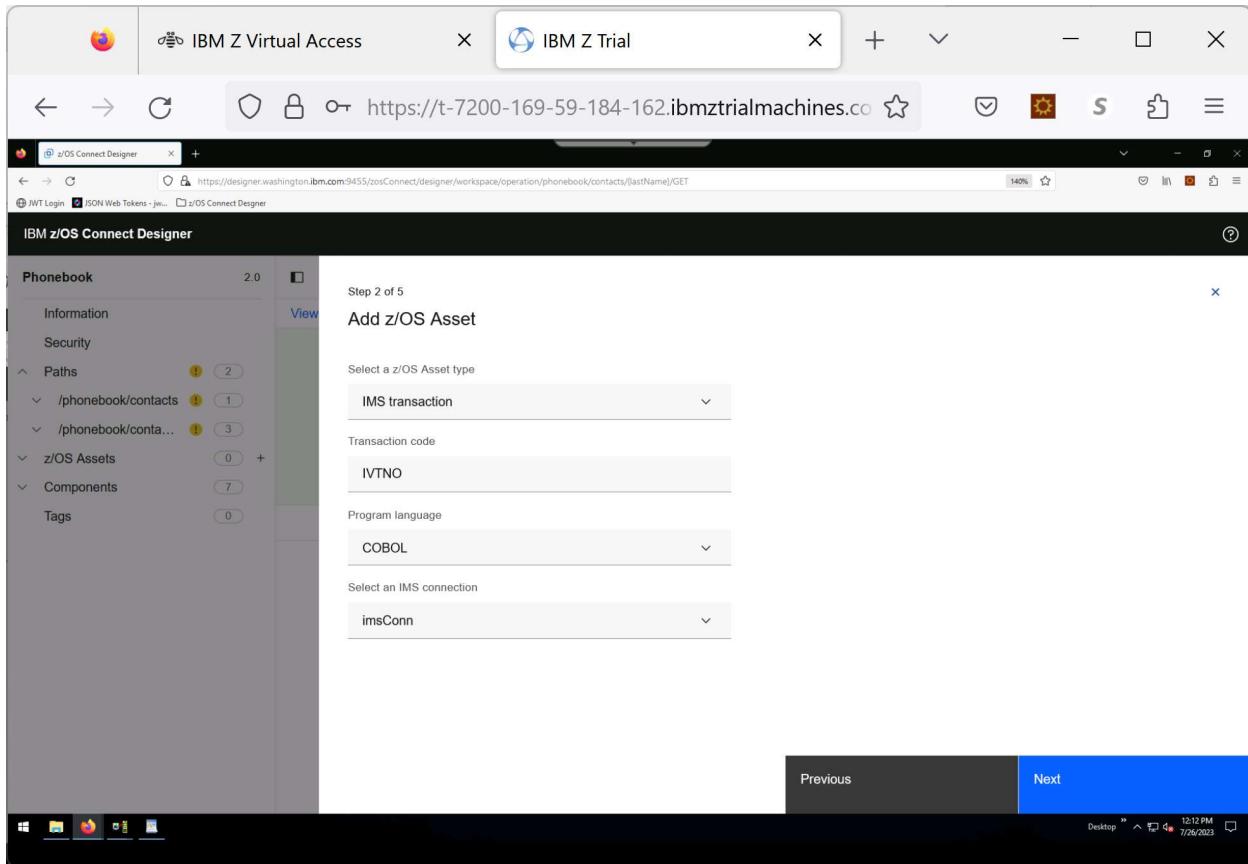
- Highlight “Add new z/OS Asset” and click Next.



- Select **IMS transaction** as the Asset type.

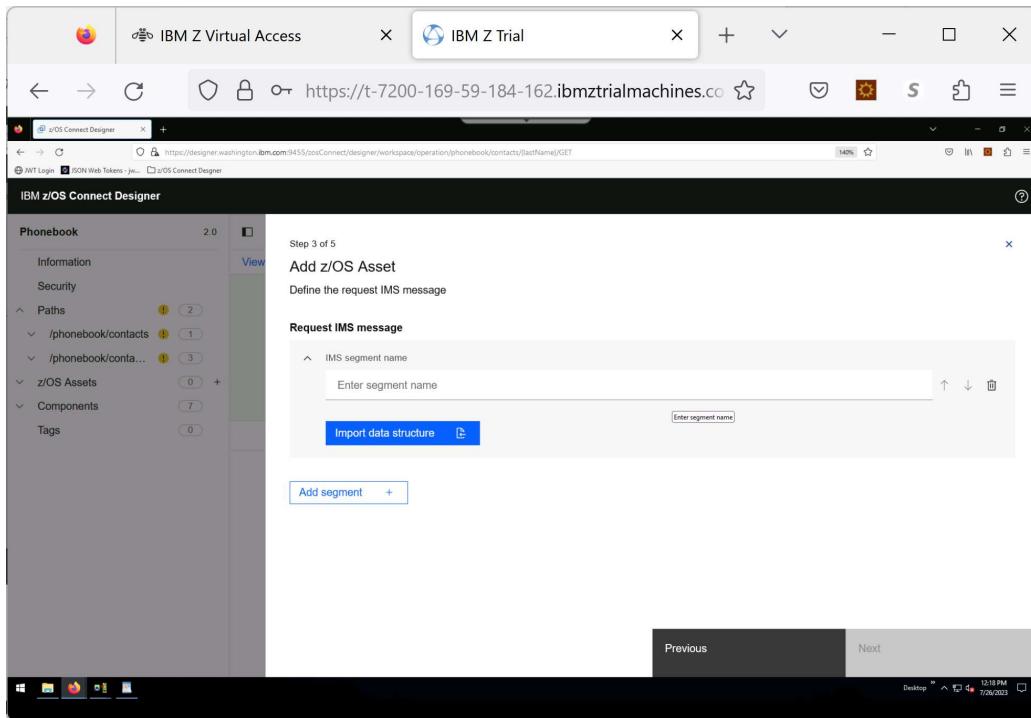
When the window opens,

- Key in **IVTNO** for the Transaction code
- Use the pulldown to select **COBOL** for the Program language
- Use the pulldown to select **ImsConn** for the IMS connection

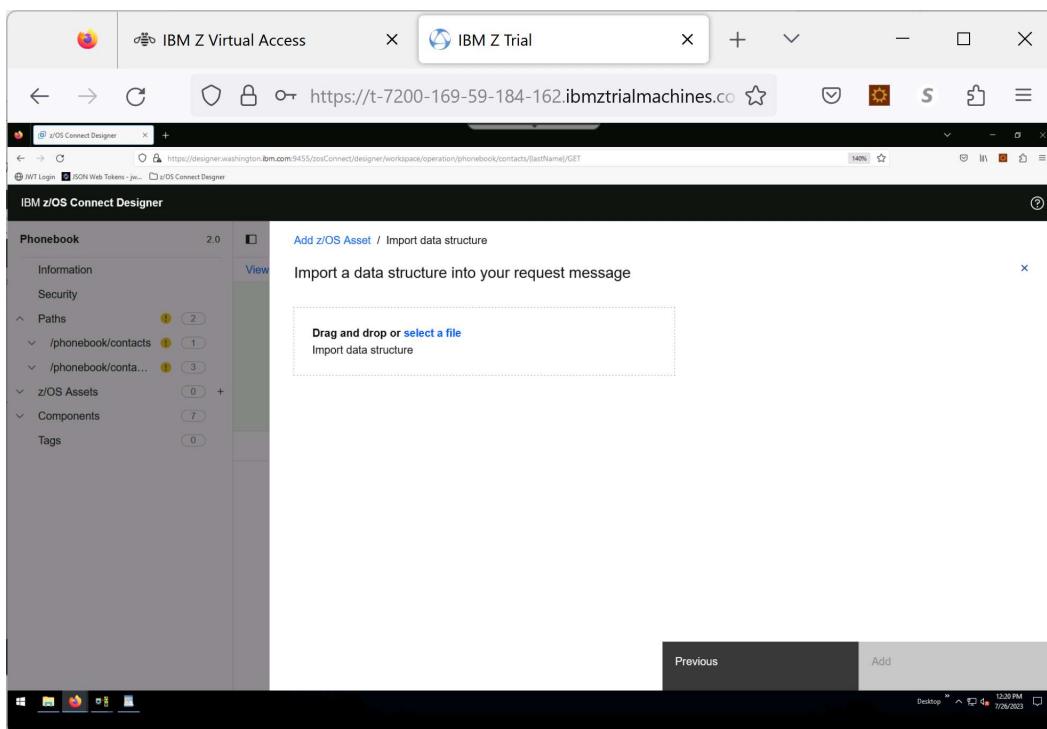


Click **Next**.

Define the request IMS message

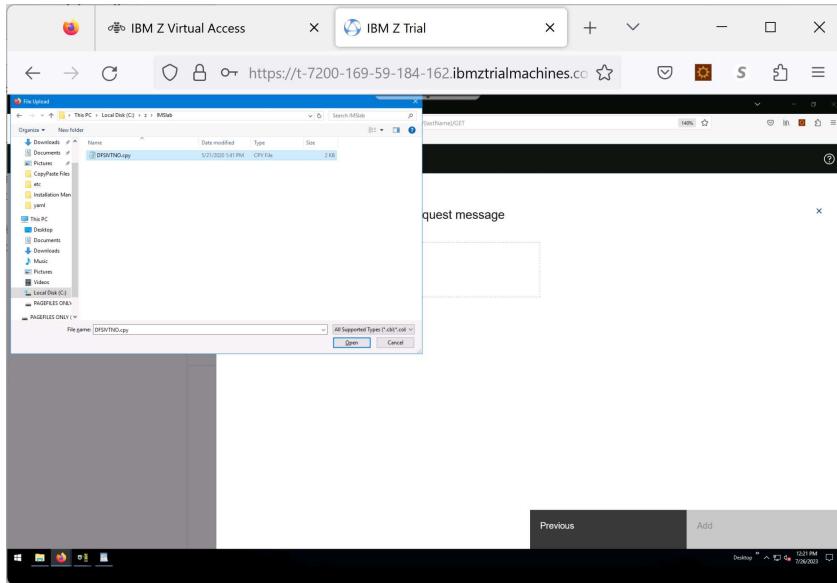


- Key in **INPUT-MESSAGE** as the segment name and click on **Import data structure**



Click on **select a file**

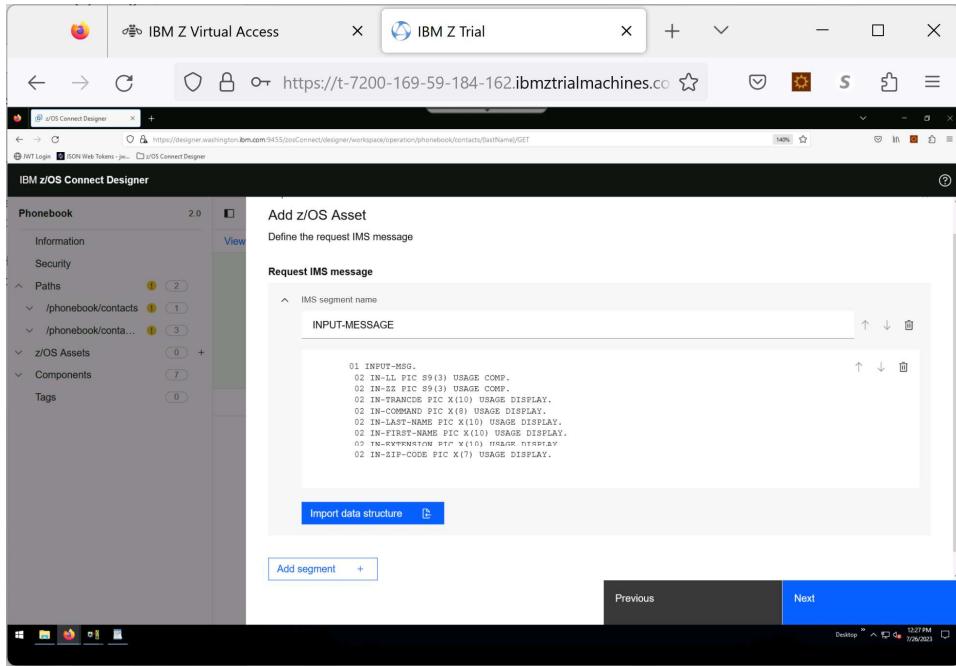
- Select **DFSIVTNO.CPY** which is in **C:/z/imslab**
 - This is the COBOL copybook for the input/output message structures associated with the IVTNO transaction



A screenshot of the "IBM z/OS Connect Designer" interface. The left sidebar shows a navigation tree with sections like "Phonebook", "Paths", "z/OS Assets", and "Components". The main area is titled "Add z/OS Asset / Import data structure" and contains a sub-section "Import a data structure into your request message". It features a "Drag and drop or select a file" input field and a table listing imported data structures. The table has columns: "Copybook name", "Data structure", and "Status". Two rows are listed: "DFSIVTNO.cpy" under "INPUT-MSG" with "Imported" status, and "DFSIVTNO.cpy" under "OUTPUT-AREA" with "Imported" status. A blue arrow points to the "Imported" status of the first row.

Copybook name	Data structure	Status
DFSIVTNO.cpy	INPUT-MSG	Imported
DFSIVTNO.cpy	OUTPUT-AREA	Imported

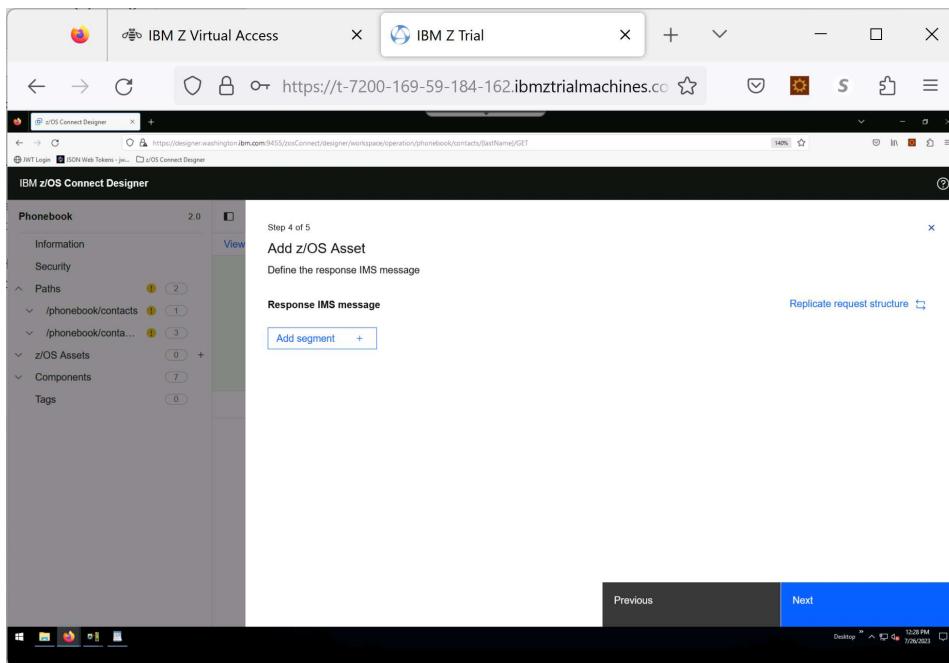
To include the request message, ONLY select the INPUT-MSG data structure. Click **Add**.



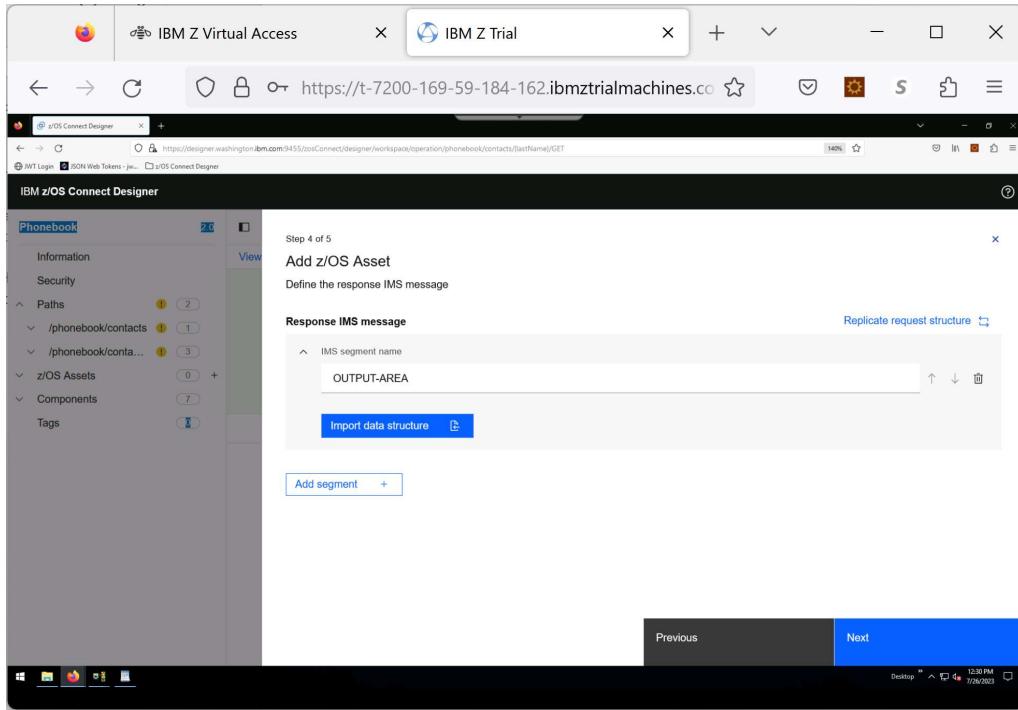
This is the COBOL copybook associated with the input message for IVTNO.

Click **Next**.

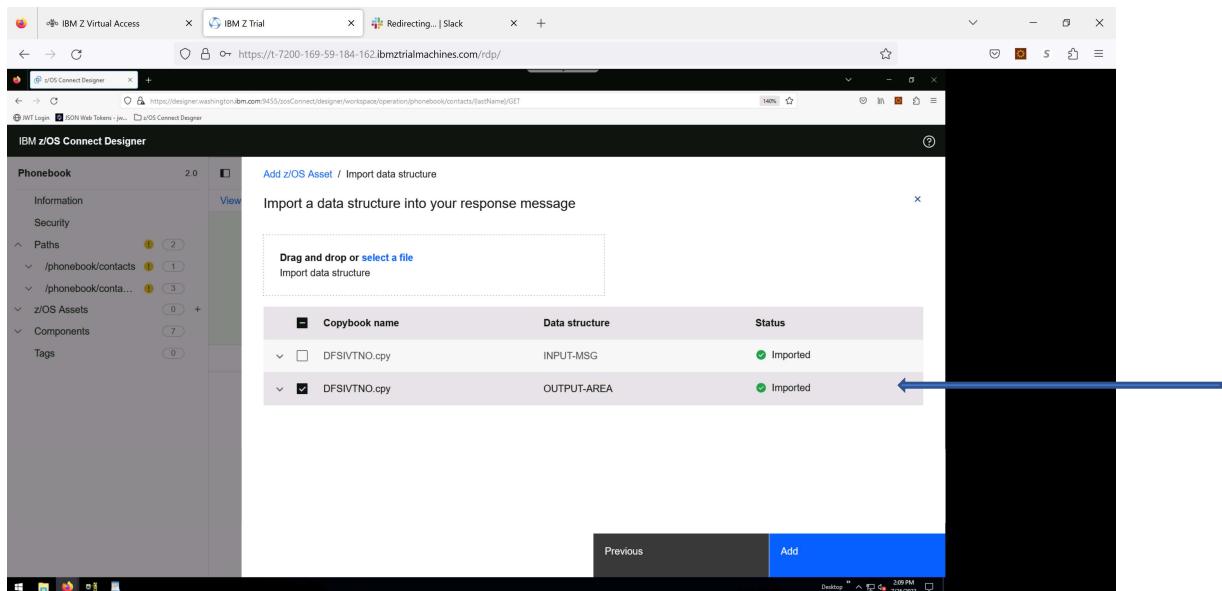
Add the segment for the **response** message.



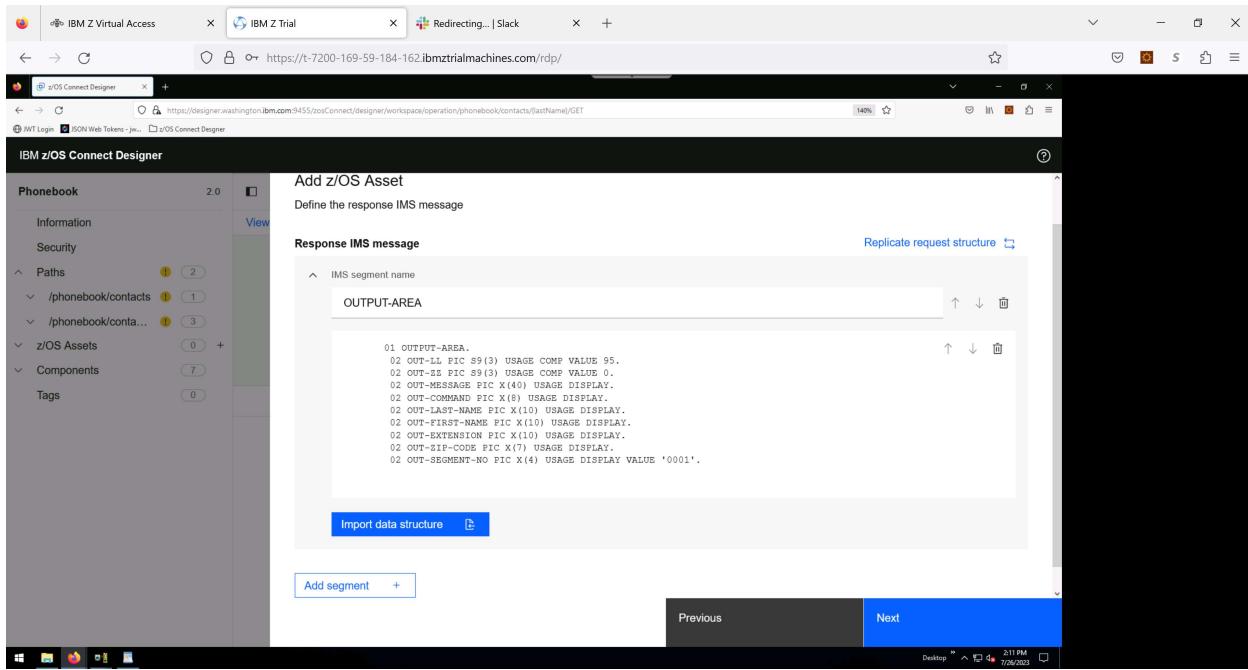
For segment name, specify **OUTPUT-AREA**



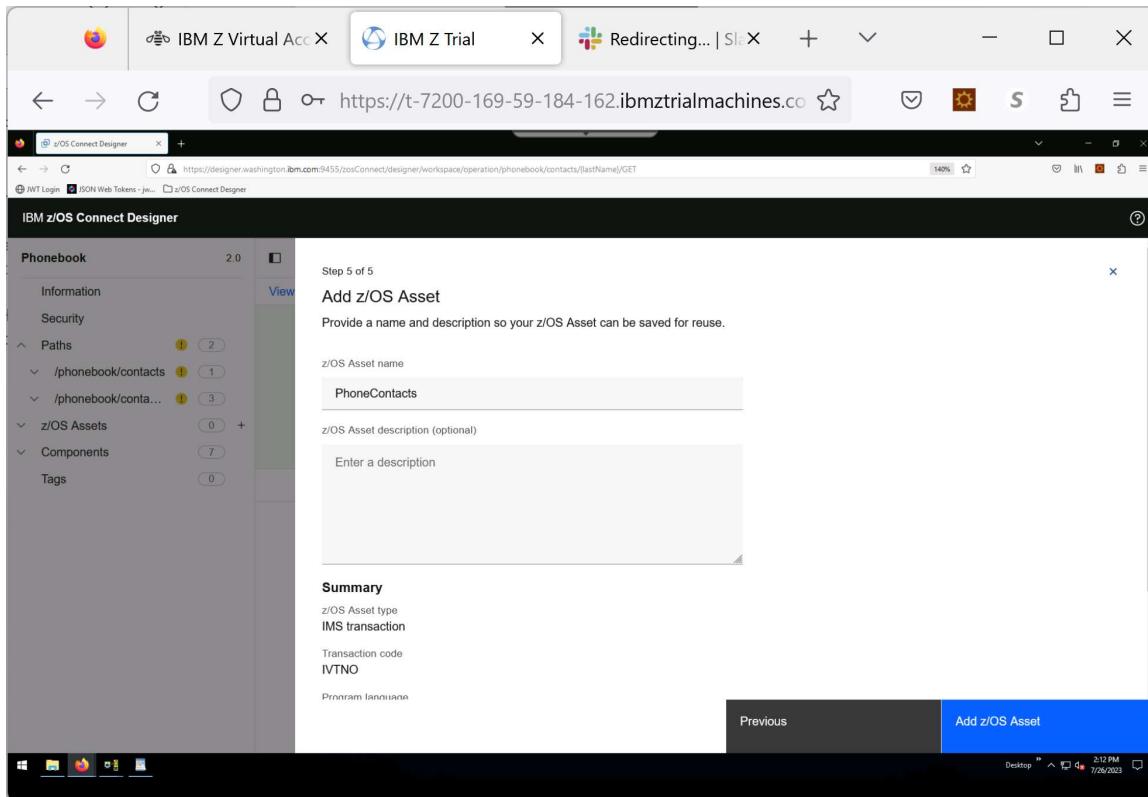
- Click **Import data structure** and follow the steps, as above, to retrieve the DFSIVTNO.CPY file



- Check the second box to include the copybook response structure.
Click **Add**.



Click Next.



- Name the z/OS Asset – e.g., **PhoneContacts**
- Click on “Add z/OS Asset”

You have built the **PhoneContacts** z/OS Asset for the IMS transaction.

The screenshot shows the IBM z/OS Connect Designer application window. The left sidebar displays the asset structure under 'Phonebook' version 2.0:

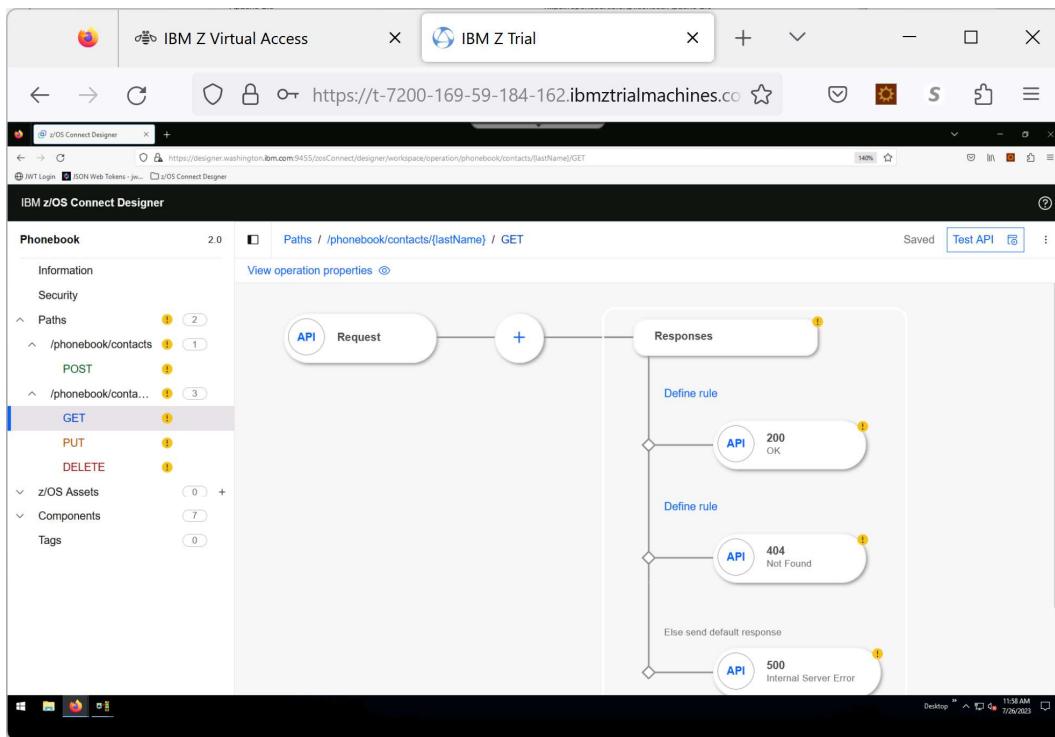
- Information
- Security
- Paths
 - /phonebook/contacts (1)
 - /phonebook/conta... (3)
 - GET (1)
 - PUT (1)
 - DELETE (1)
- z/OS Assets
 - PhoneContacts (selected)
- Components (7)
- Tags (0)

The main panel shows the configuration for the selected 'PhoneContacts' asset:

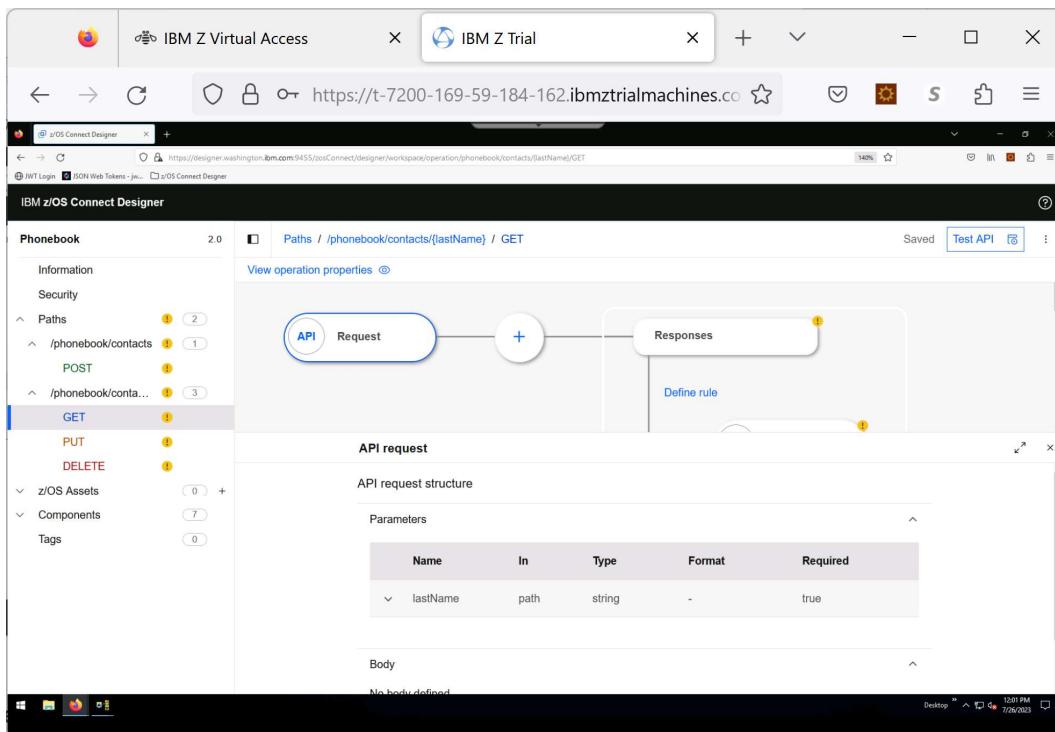
- General**: Name is 'PhoneContacts', Type is 'IMS transaction'.
Description is empty.
- IMS transaction**: Transaction code is 'IVTNO', Program language is 'COBOL'. Connection profile is 'imsConn'.
- Request structure**: INPUT-MESSAGE
- Response structure**: OUTPUT-AREA

Create the API – GET method

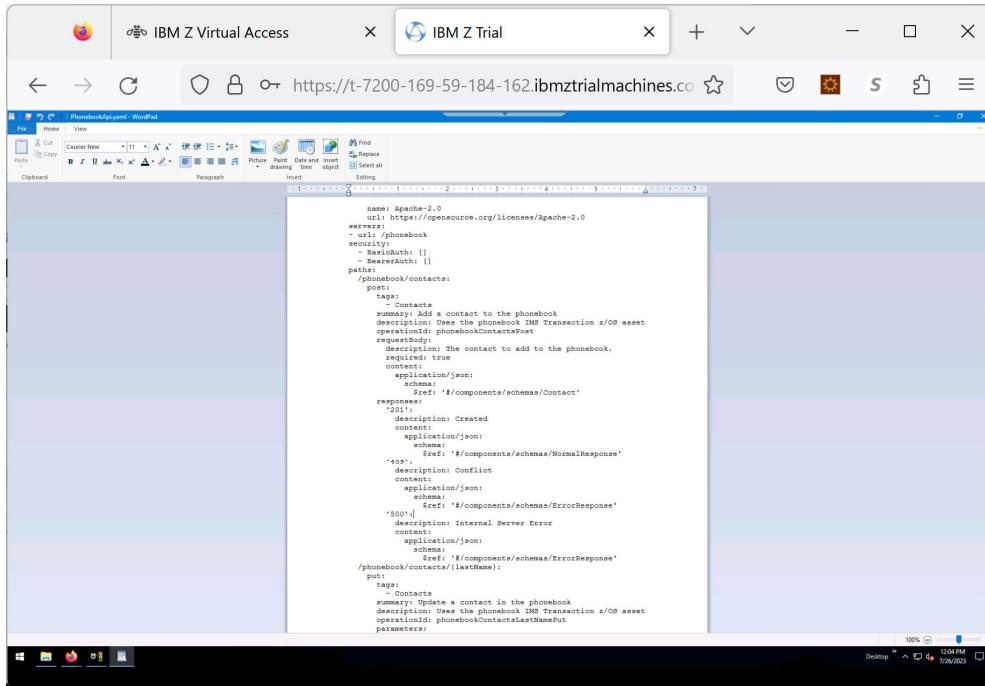
Click on the GET method to view the operation properties. This opens the GET /phonebook/contacts/{lastName} method



Click on request

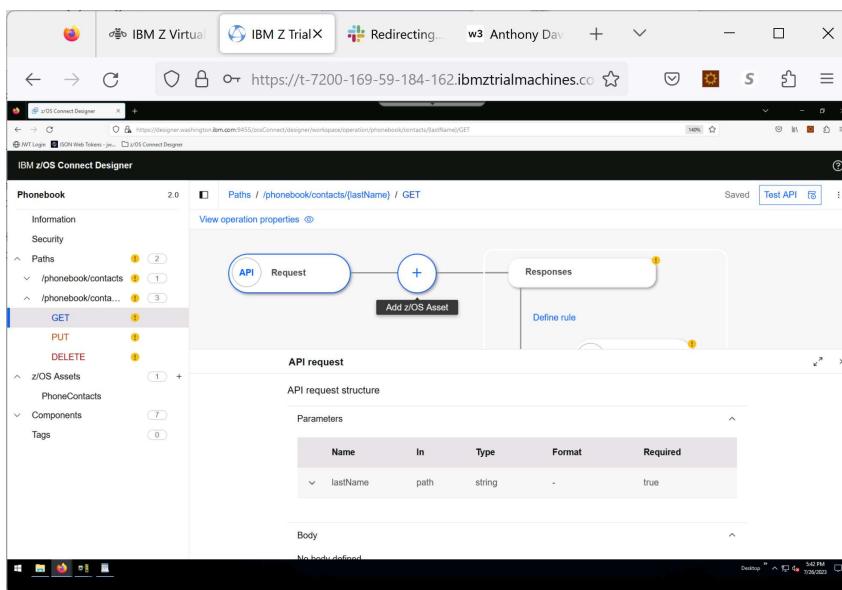


Notice that the lastName parameter is part of the request – this parameter was defined in the phonebook.yaml

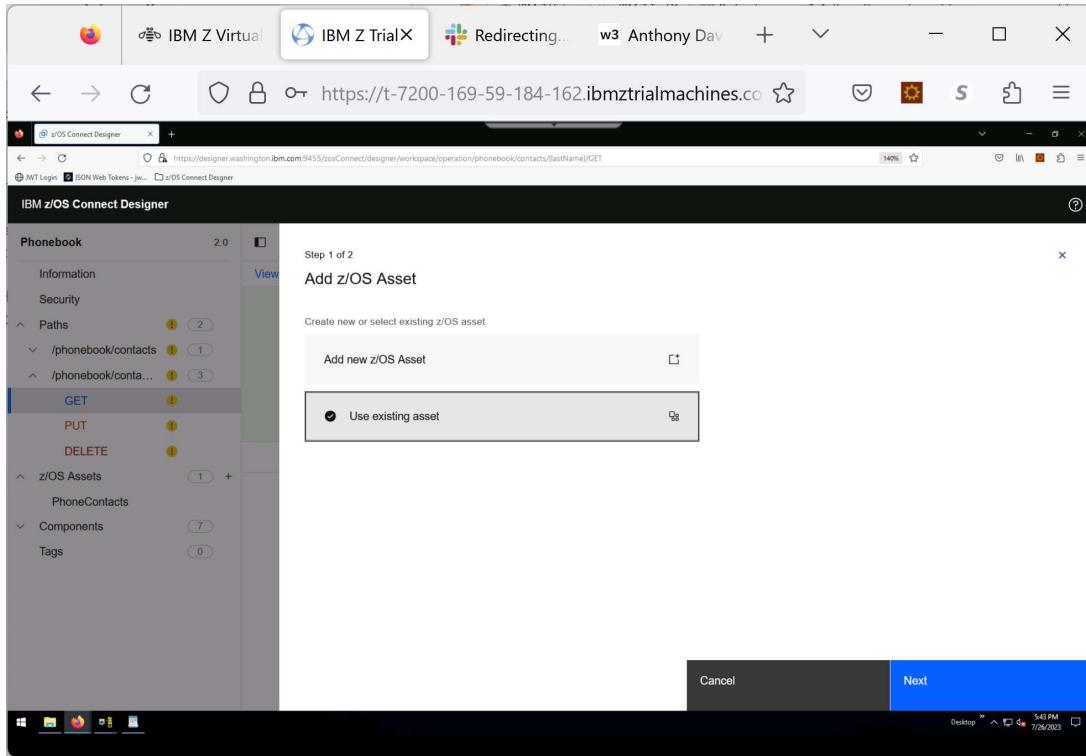


Map the API Request to the z/OS asset

Click on the + (plus sign) in the center to add a z/OS asset

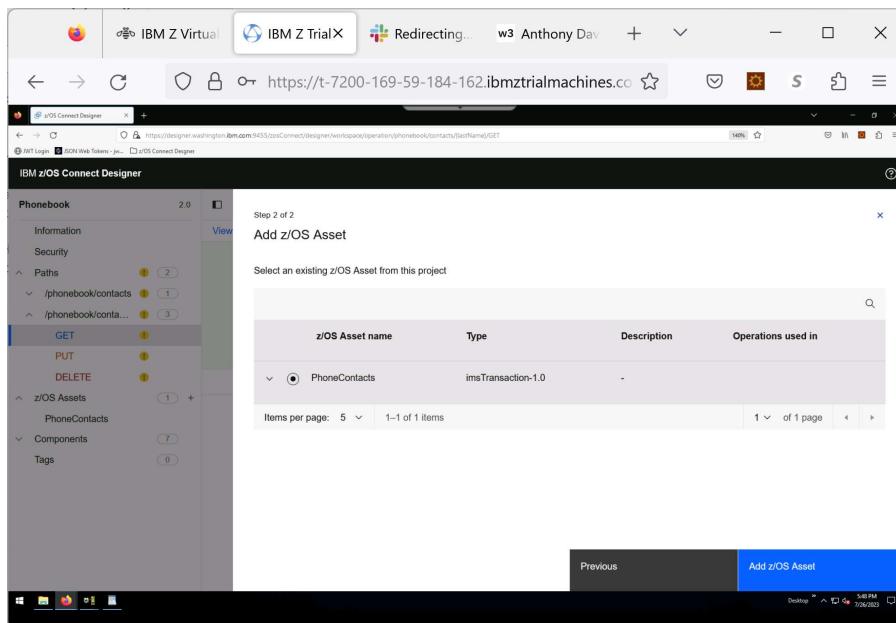


- Click on **Use existing asset** since we will use the one you just created.



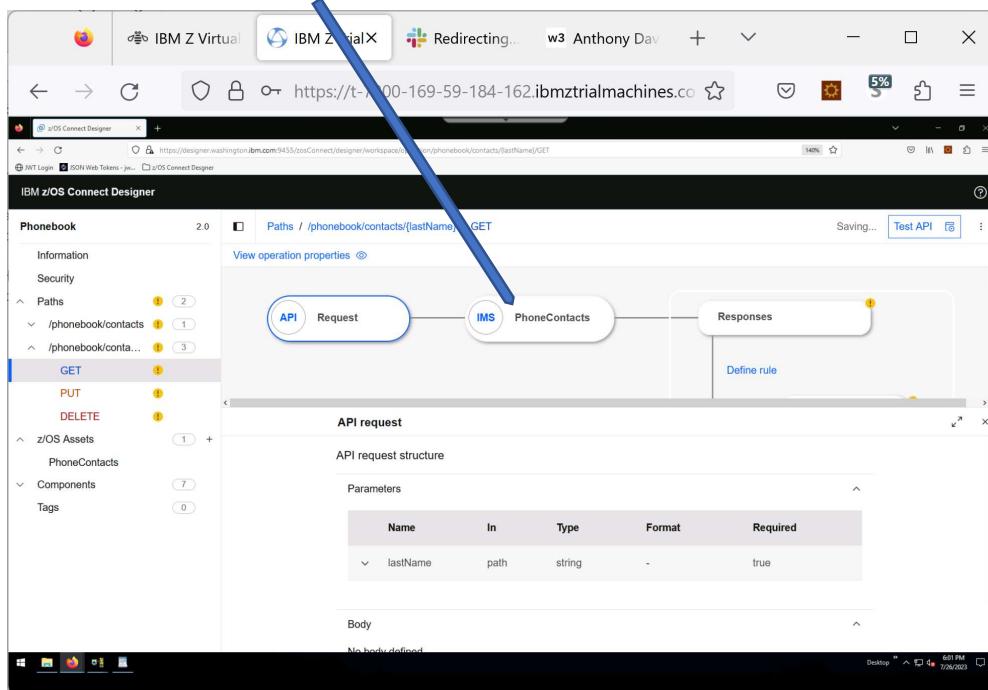
- Click **Next**.

The z/os Asset previously created is displayed

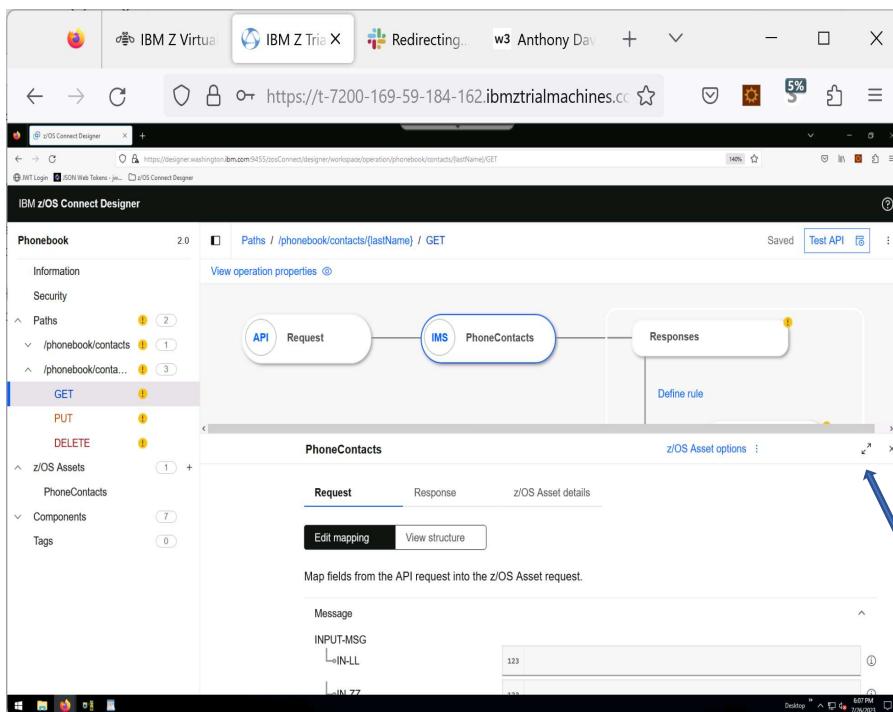


- Click **Add z/OS Asset**

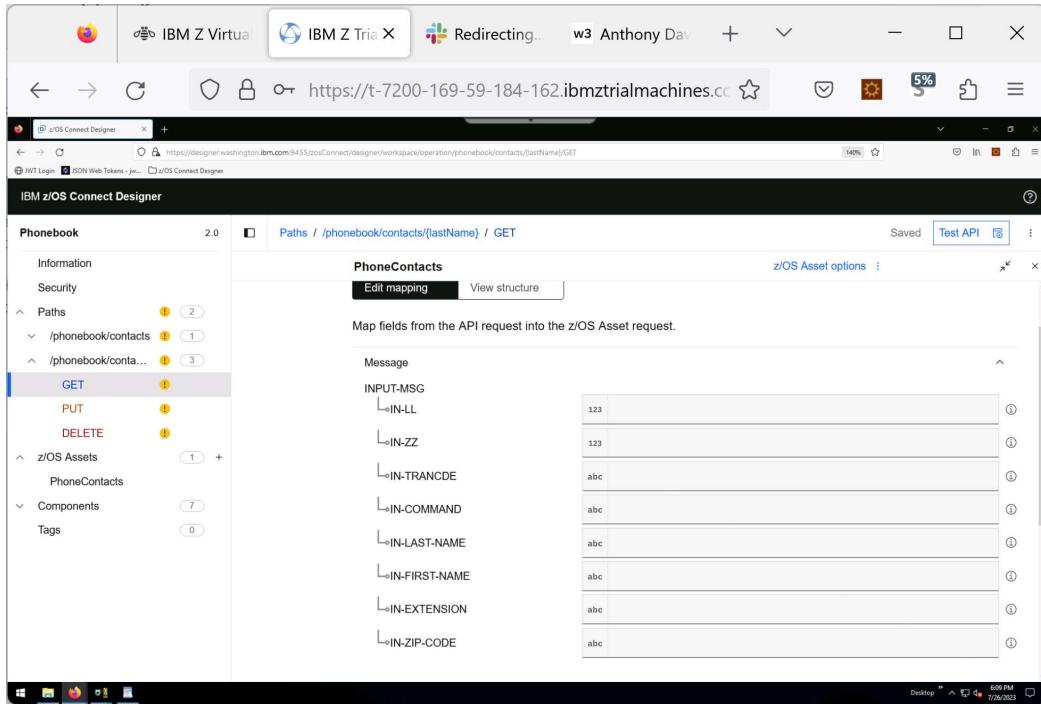
Note that the asset is now listed in the Operations flow diagram.



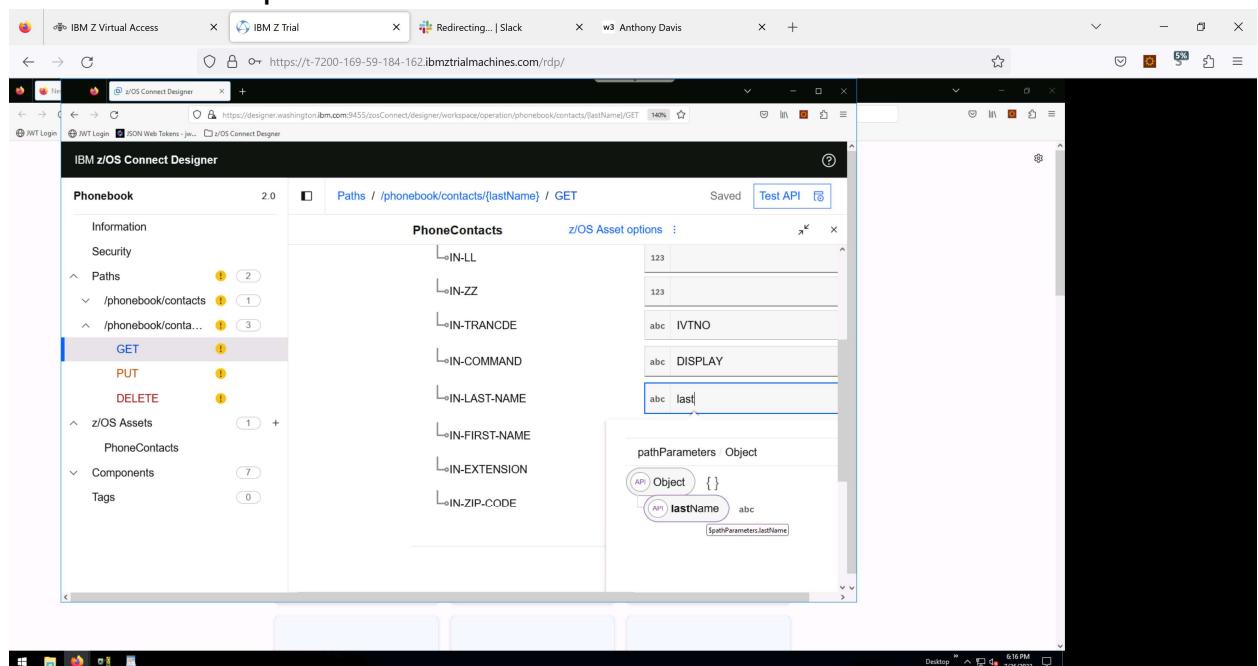
- Click the asset you created e.g., “PhoneContacts”

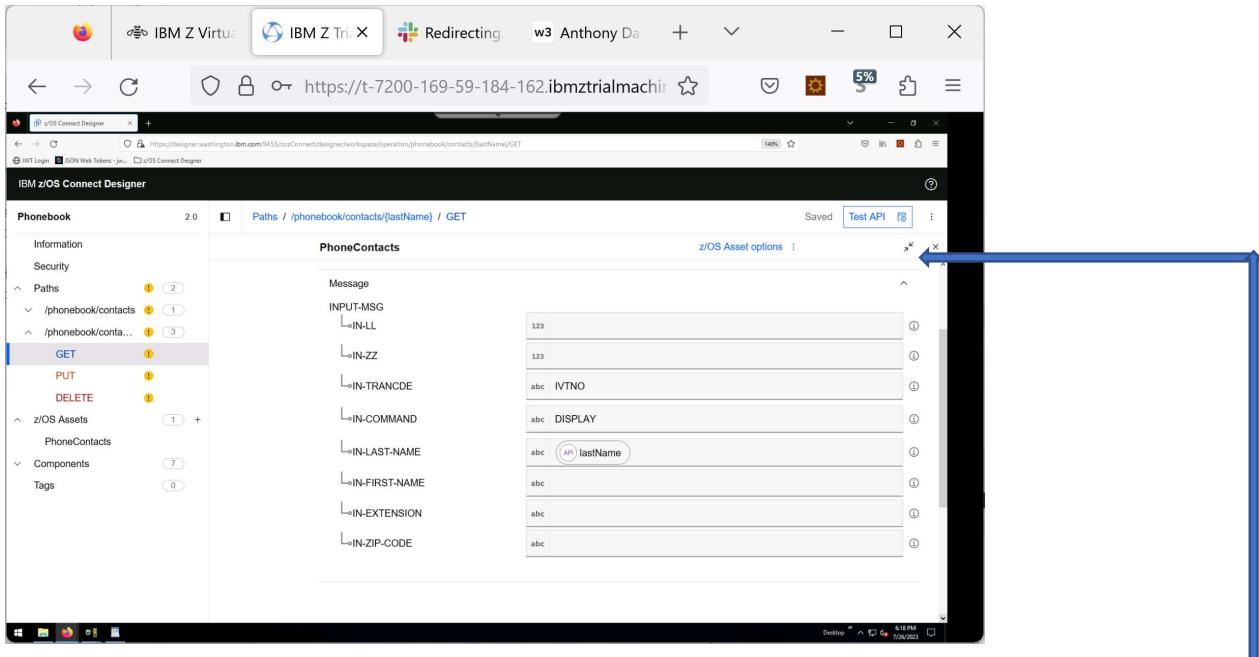


- Click on **Edit Mapping** under **Request**
- Expand the bottom pane by clicking on the two arrows on the top right



- Key in **IVTNO** (uppercase) in the **IN_TRANCDE** field
- Key in **DISPLAY** (uppercase) in the **IN-COMMAND** field
- Map the API Request parameter `lastName` into the **IN-LAST-NAME** z/OS Asset Request field.
 - Type **lastName** in the **IN-LAST-NAME** field. Note that when you start typing, the **Available Mappings** menu opens with the available parameters. Select **lastName** from the list.





- lastName is the only request parameter that is needed.

Just as a note, to the right of each text box are two icons.



Is used to select a Path parameter from the list. When the list opens, be sure that you look in the correct group.



Is used to insert a function. *Functions are not used in this lab.*

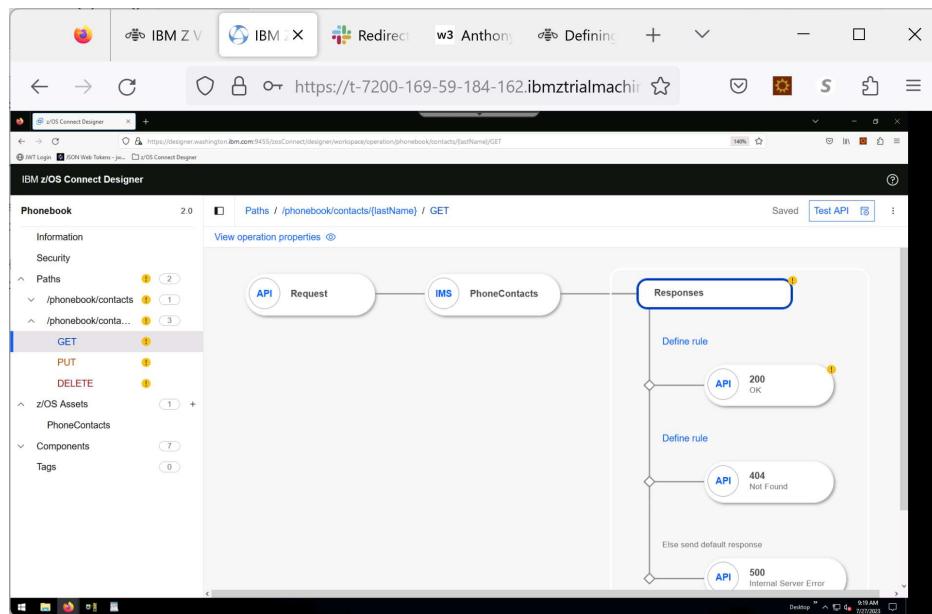
- You can now minimize this panel by clicking on the arrows on the top right. This will bring you back to the main operations flow diagram.

Map the API Response fields

- Click **Responses** on the Operation flow diagram. The Responses configuration pane opens. Responses are evaluated from top to bottom where the final response is the default response.

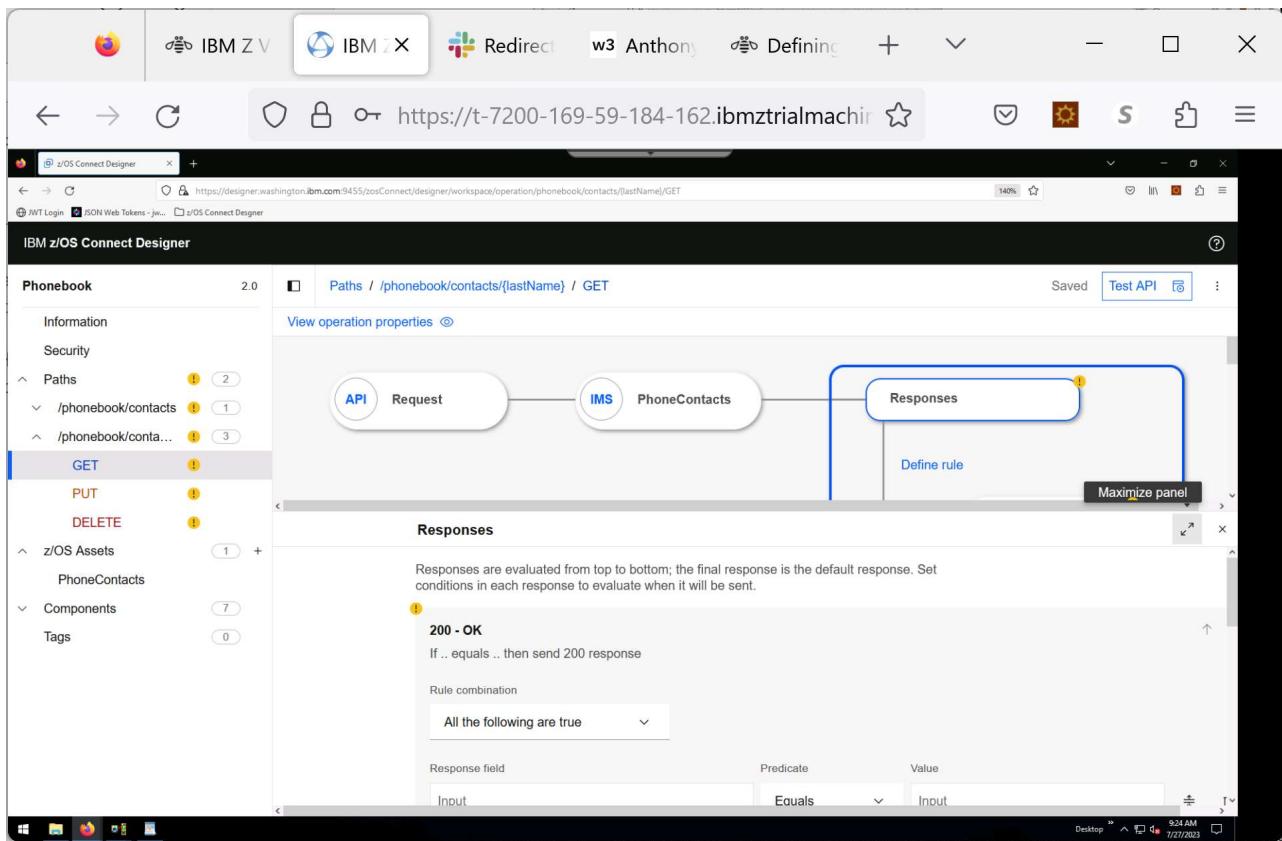
Each response has the following properties:

- A condition with three fields, *response*, *predicate*, and *value*.
- One or more *conditions*.
- You can change the order of the responses by using the ↑ and ↓ buttons next to each response case.
- The sequence of the conditions within a response can be changed. Click ↕ to change the position in the sequence.
- Conditions can be deleted.



The default order of the responses is such that 200 - OK is the first to be evaluated and 500 - Internal server error is the last and therefore the default response. (Best practice is to configure 500 - Internal server error as the default response to capture any errors in the conditional logic of the response.)

- Set the **200** response code condition
 - This code indicates that the requested contacts were found and the information is returned in an array. The contact record properties will need to be mapped to the fields of the API Response.
- Maximize the lower panel.



Set the conditions for the 200 OK response condition.

- Key in the following into the **Response field** – *note* the case and quotes
\$zosAssetResponse.message."OUTPUT-AREA"."OUT-MESSAGE"

OUTPUT-AREA is the name of the segment that was given when creating the z/OS asset – if a different name was used, then that name should be used instead of OUTPUT-AREA.

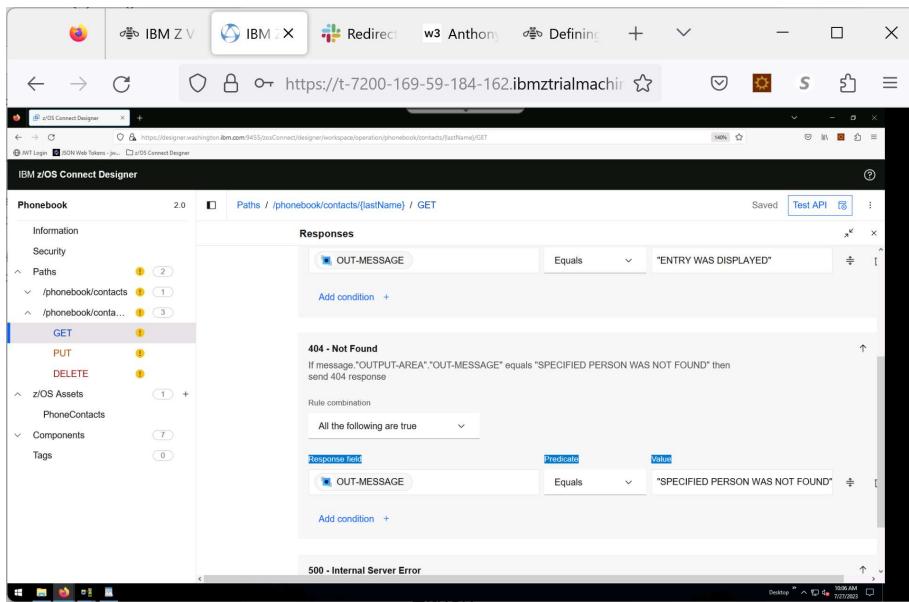
- Also key in “**ENTRY WAS DISPLAYED**” (note the double quotes) in the corresponding Value field.

The screenshot shows the IBM z/OS Connect Designer interface. On the left, there's a sidebar with sections like Information, Security, Paths, z/OS Assets, Components, and Tags. Under Paths, there are entries for /phonebook/contacts (1), /phonebook/contact... (3), and a selected GET entry (1). The main panel shows the configuration for the GET /phonebook/contacts/{lastName} endpoint. It has a 'Responses' section with a note: "Responses are evaluated from top to bottom; the final response is the default response. Set conditions in each response to evaluate when it will be sent." Below this is a "200 - OK" section with a rule combination: "All the following are true". A condition is defined: Response field \$zosAssetResponse.message."OUTPUT-AREA"."OUT-MESSAGE" Equals "ENTRY WAS DISPLAYED". There's also a "404 - Not Found" section with a note: "If .. equals .. then send 404 response". The status bar at the bottom indicates "Saved" and "Test API".

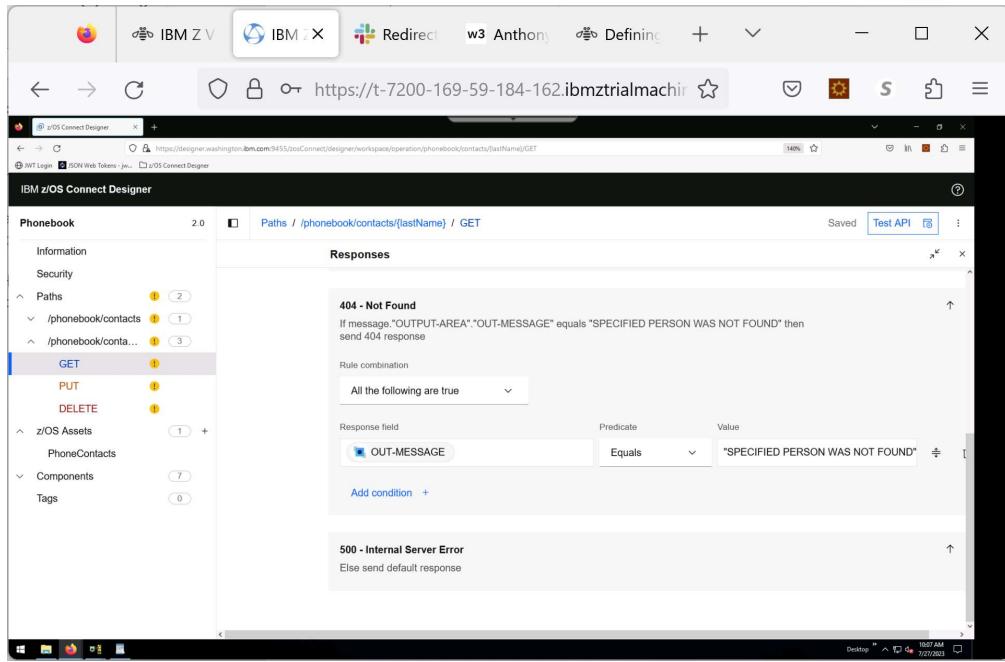
If done correctly, the ! mark associate with the 200- OK should disappear.

This screenshot is identical to the one above, showing the same configuration for the GET /phonebook/contacts/{lastName} endpoint. The key difference is that the exclamation mark (!) mark associated with the "200 - OK" section in the previous screenshot has disappeared, indicating that the configuration is now correct.

- Set the **404** response code condition – Not Found response.
- Either:
 - Copy and paste the condition from the 200 code
 - Or, once again type in:
\$zosAssetResponse.message."OUTPUT-AREA"."OUT-MESSAGE"
- Also key in “**SPECIFIED PERSON WAS NOT FOUND**” in the **Value** field (note the double quotes).

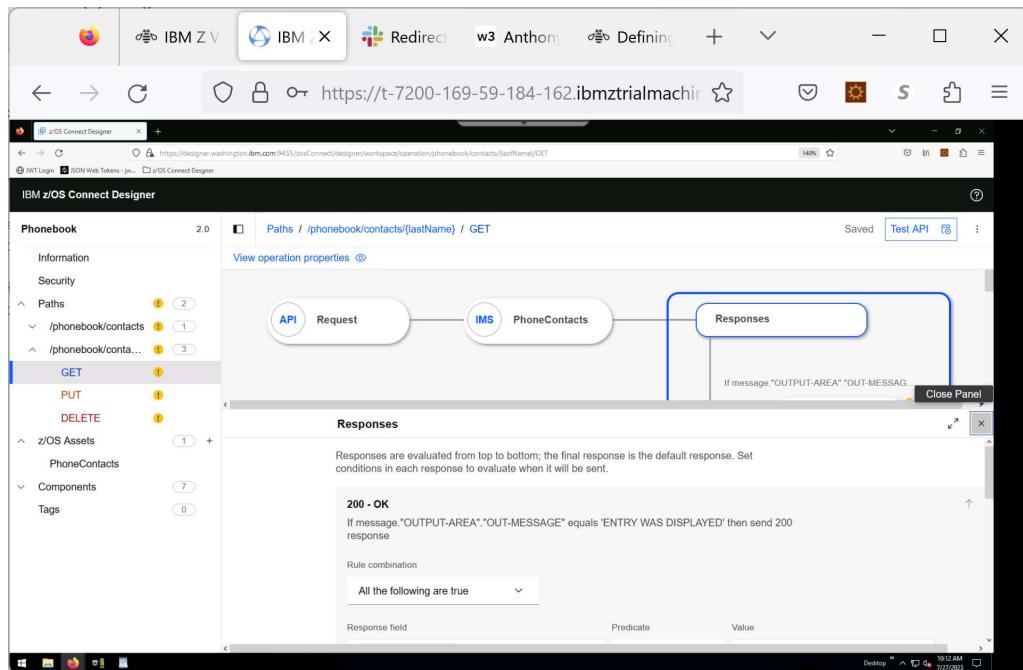


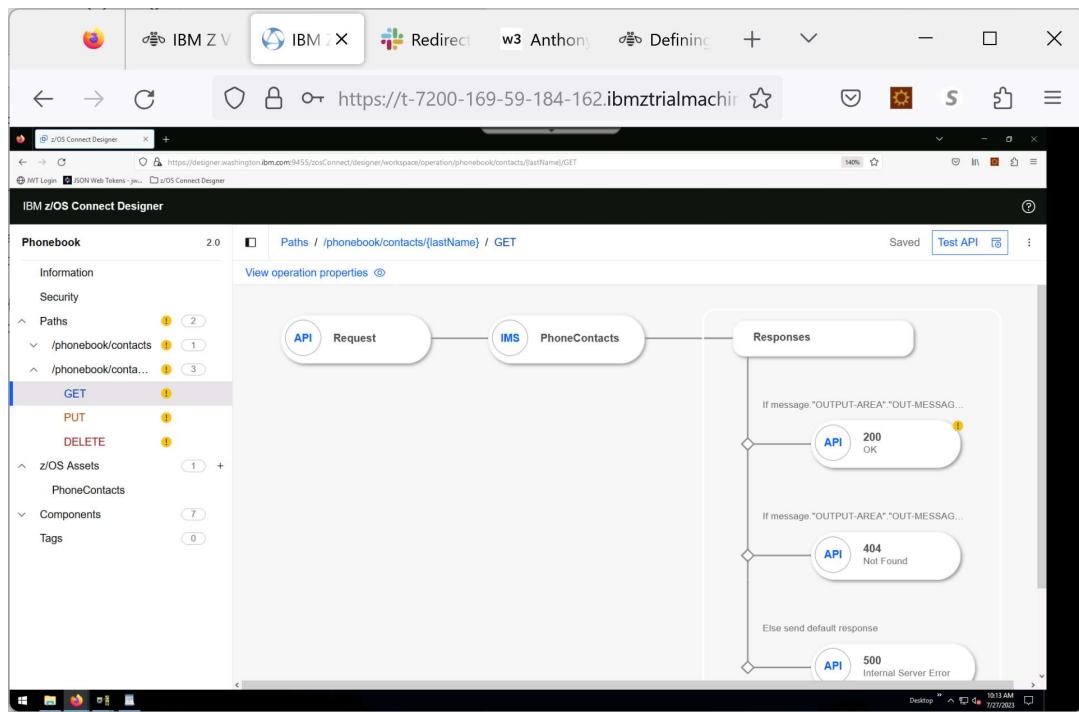
The 500 -Internal server error response is the default, so it has no conditions and must be the last entry in the table.



The final set of tasks before running a test, is to map the responses from the z/OS asset (IMS response) to the API Response fields.

- Minimize the panel you have been working on by clicking the double arrows at the top right to get back to the primary window. You can also close the panel.

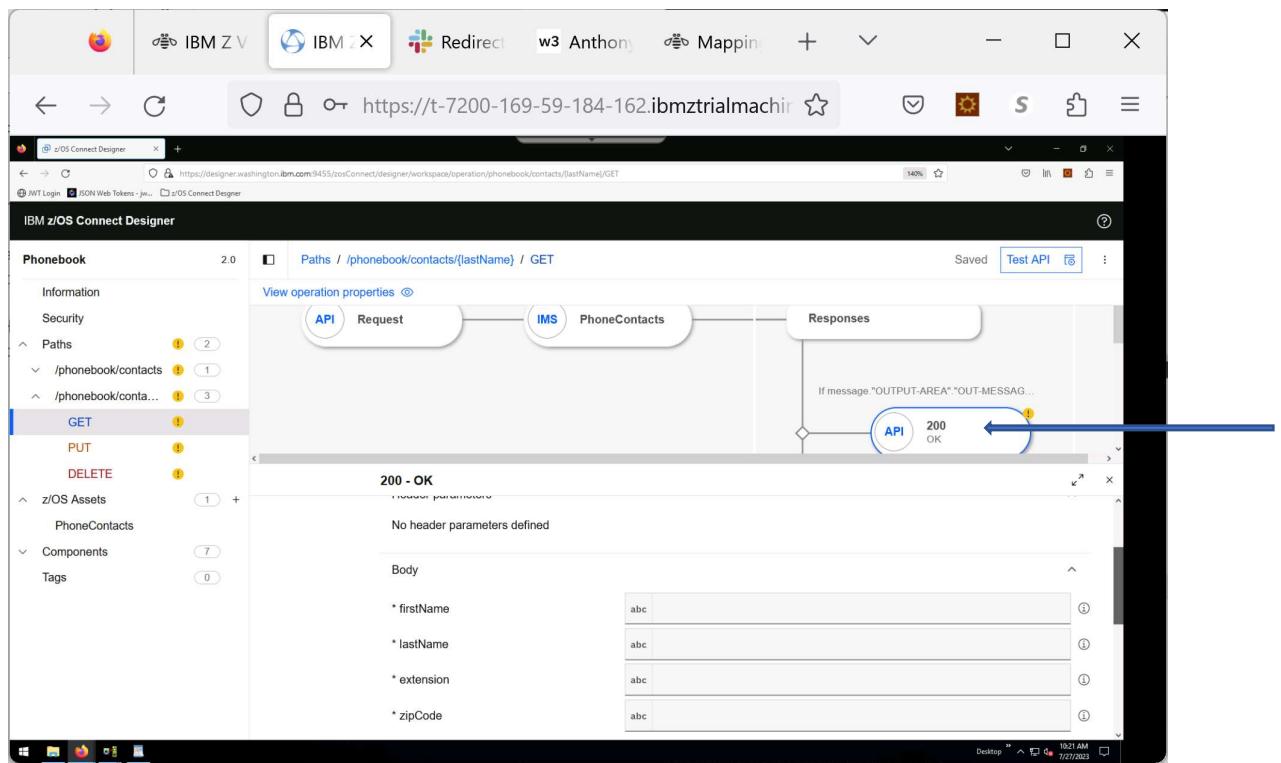




Map the 200 response.

- In the Operation flow diagram, click the **200** response node.

A 200 response code indicates that the requested catalog contacts were found and the information is returned in an array. The contact record properties need to be mapped to the fields of the API response.



- Map the z/OS Asset response input **OUT-FIRST-NAME** to the firstName API response field.
 - ***Start keying in OUT-FIRST-NAME, the tool will allow you to select the appropriate field.***
- Map the z/OS Asset response input **OUT-LAST-NAME** to the lastName API response field.
- Map the z/OS Asset response input **OUT-EXTENSION** to the extension API response field.
- Map the z/OS Asset response input **OUT-ZIP-CODE** to the zipCode API response field.

The screenshot shows the IBM z/OS Connect Designer interface. On the left, there's a navigation sidebar with sections like Phonebook, Information, Security, Paths, z/OS Assets, Components, and Tags. The main area displays a GET operation for the path `/phonebook/contacts/{lastName}`. The mapping section on the right shows how fields from the z/OS Asset response are mapped to the API response. Fields like `firstName`, `lastName`, `extension`, and `zipCode` are mapped to `OUT-FIRST-NAME`, `OUT-LAST-NAME`, `OUT-EXTENSION`, and `OUT-ZIP-CODE` respectively.

- Minimize the panel to go back to the Operation flow diagram

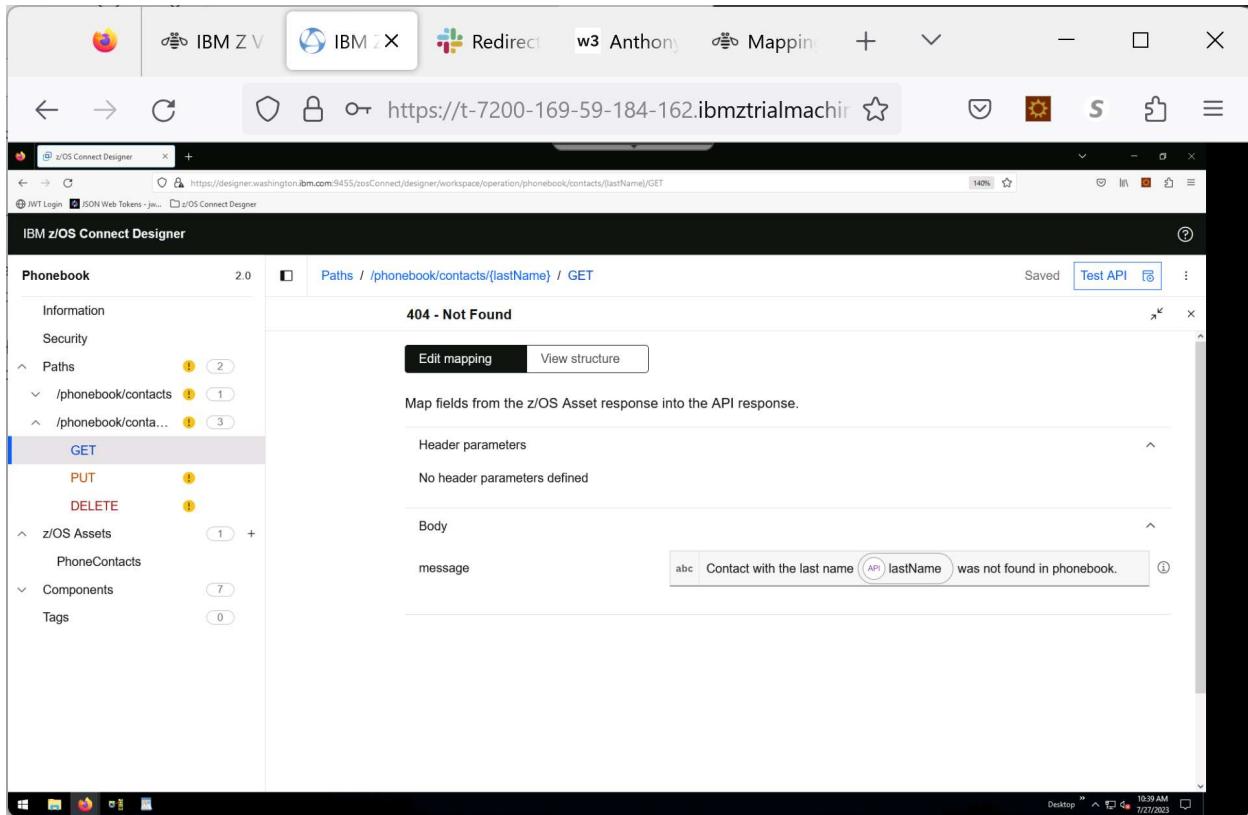
Map the 404 response.

The screenshot shows the operation flow diagram for the `/phonebook/contacts/{lastName}` GET operation. The flow starts with an API node labeled "200 OK". There are two parallel paths from this node. One path leads to an API node labeled "404 Not Found" if the condition "If message.OUTPUT-AREA.OUTPUT-MESSAGE" is true. The other path leads to an API node labeled "500 Internal Server Error" if the condition is false. A blue arrow points to the "404 Not Found" node.

Configure the 404 response to return a message to explain that the contact was not found.

- Key the following into the **message** field (be aware of case, and brackets):

Contact with the last name {{\\$apiRequest.pathParameters.lastName}}
was not found in phonebook.

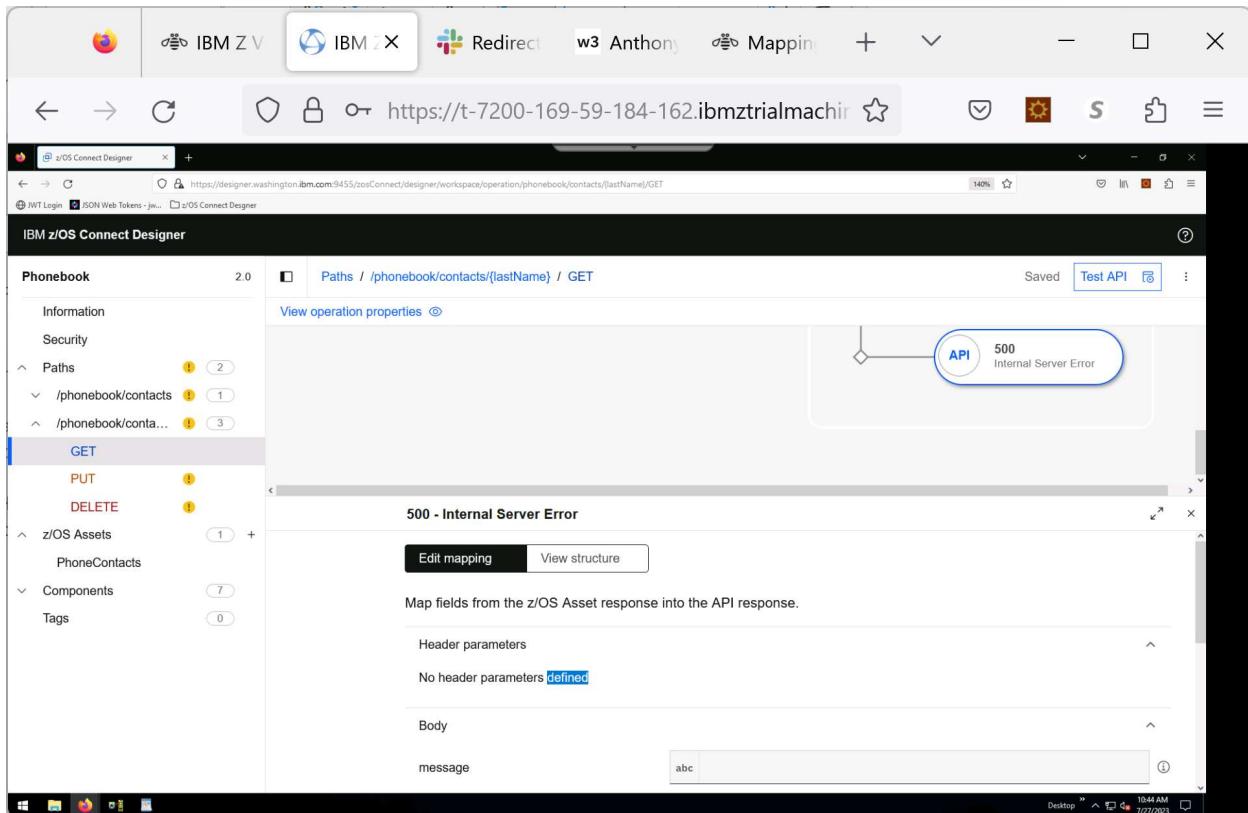


Note how the tooling pulls the lastName from the structure.

- Click **X** at the top right of this panel to close it and return to the Operations flow diagram.

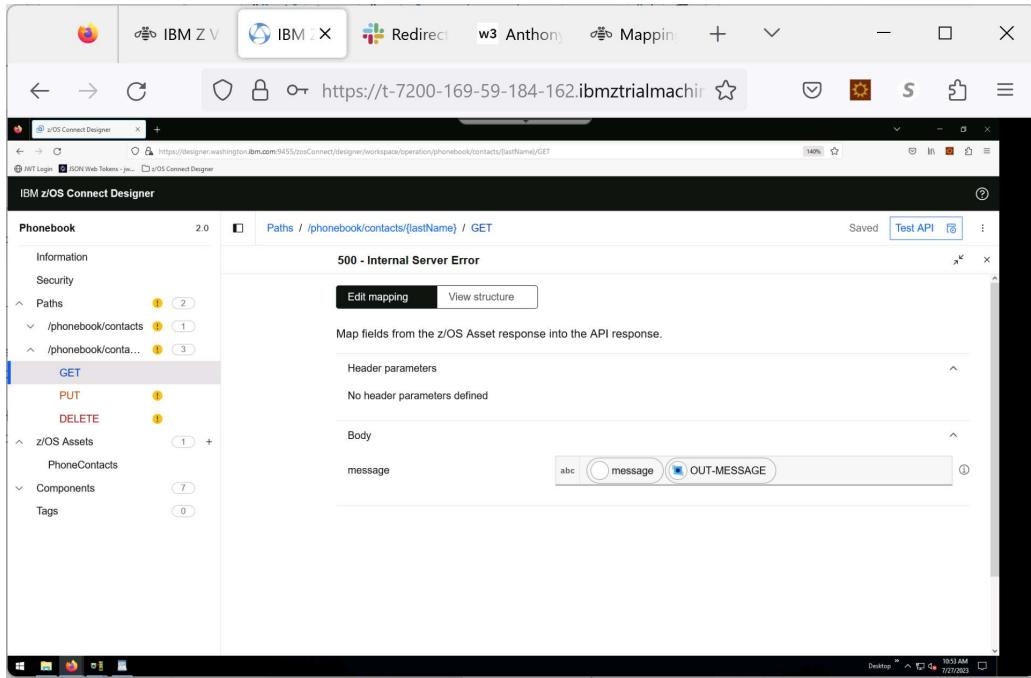
Map the 500 response.

- Click on the **500** node to open up the mapping panel.



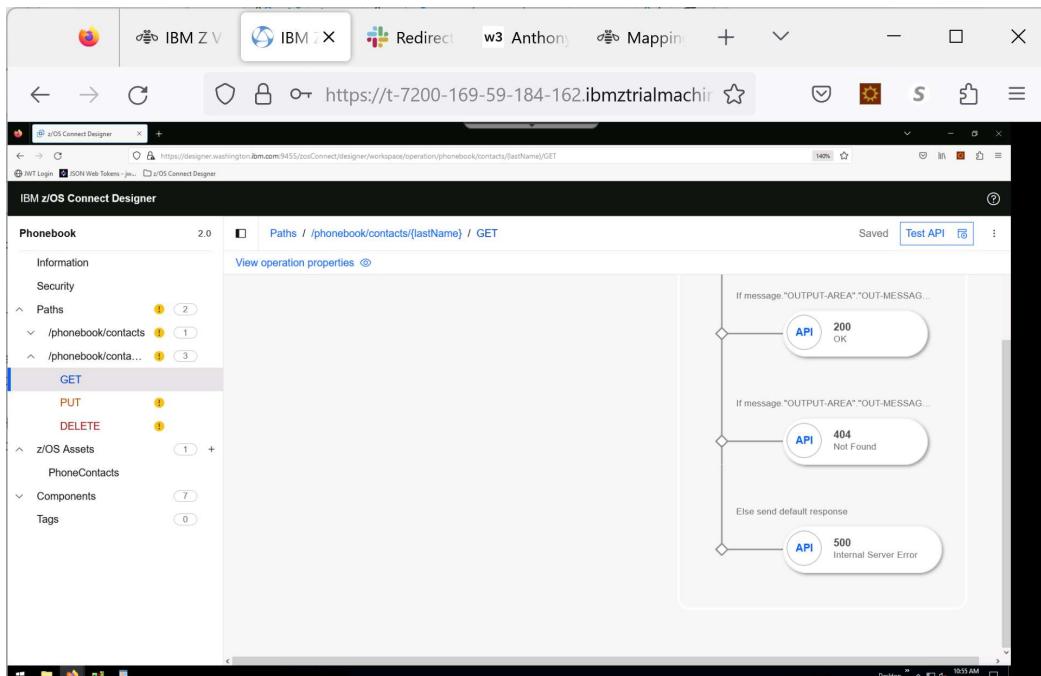
A 500 response code indicates an internal server error. Configure the 500 response to return the z/OS Connect error message by typing the following into the message field (Note the case, brackets, and quotes):

```
{{error.message}}{{$zosAssetResponse.message."OUPUT-AREA"."OUT-MESSAGE"}}
```



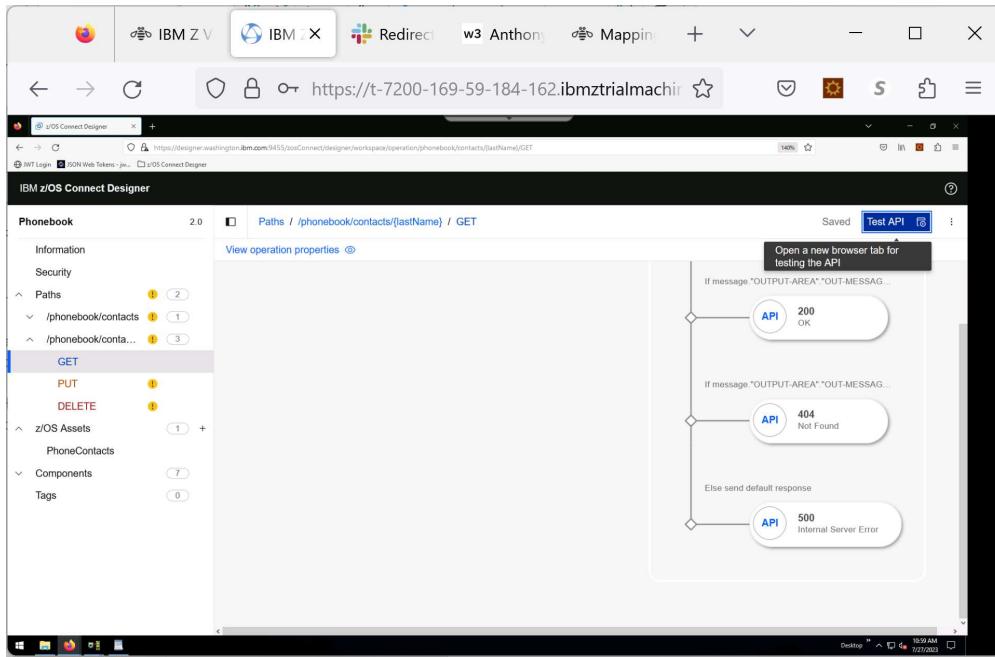
- Click X at the top right of this panel to close it and return to the Operations flow diagram.

Note that on the top right of the panel that your work have been **Saved** and that on the left, the ! has disappeared by the GET method.

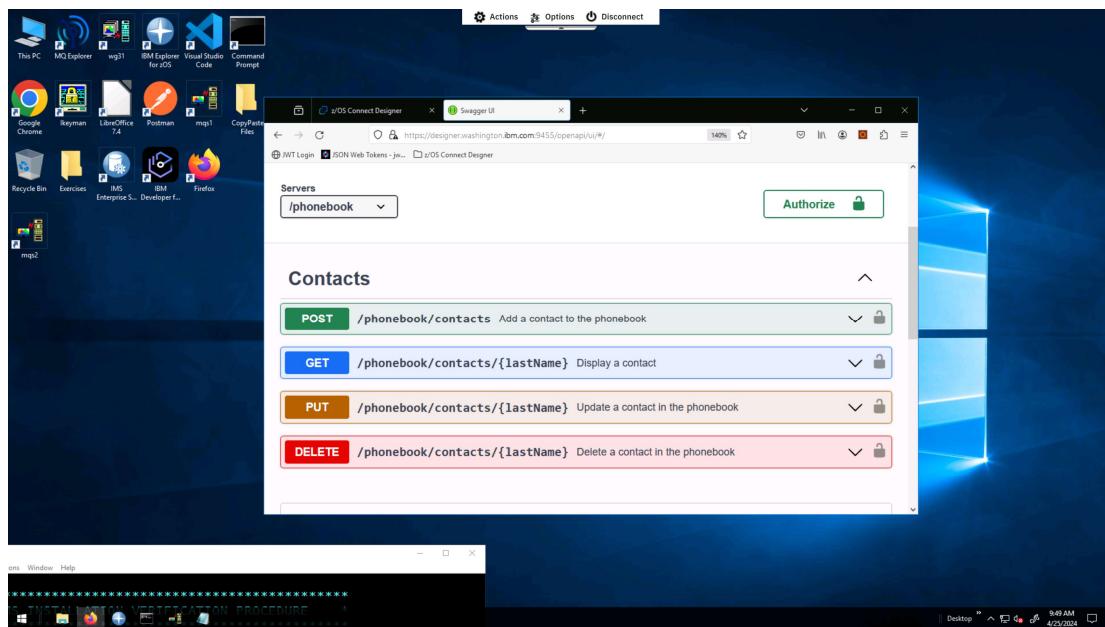


Test the API GET method.

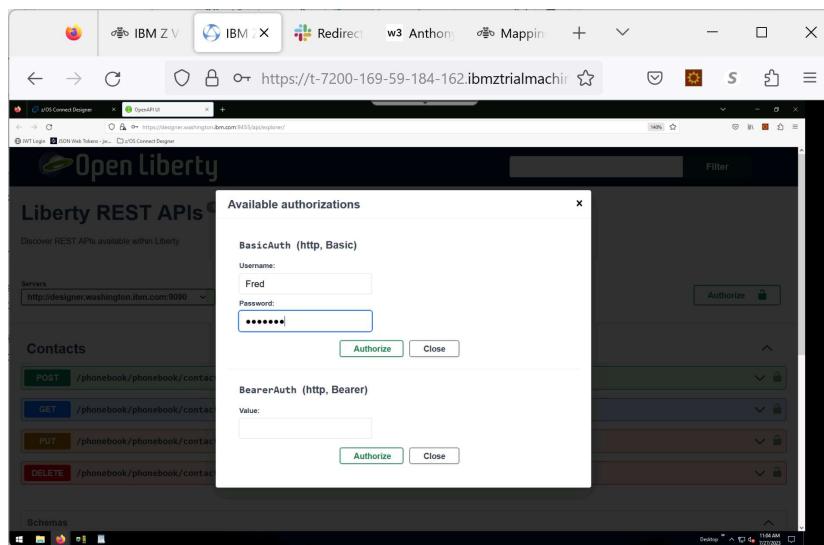
- Click on the **TEST API** button on the top right of the operations diagram.



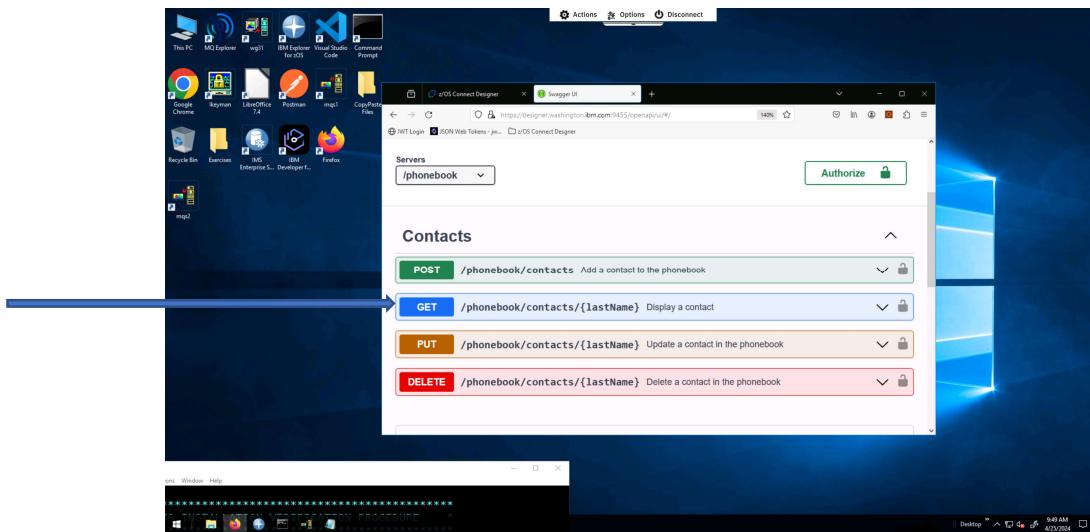
- On the **Servers** drop down, select **/phonebook**
- Click on **Authorize**



- Click the **Authorize** button and choose BasicAuth.
 - Two usernames (user1 and Fred) are available. You can use either one
 - Otherwise, if the username is blank
 - Key in **Fred** (note the capital **F**) for the Username
 - Key in **fredpwd** (Note all lowercase) for the password
- Click **Authorize** and close the panel



Back on the main panel, you will see all the possible methods that can be used for this API. The only one configured at this point is the **GET** method so it is the only one that can be tested.



- Click **GET**

- Click **Try it out**
- Key in **LAST1** in the **lastName** field, and then click **Execute**.

GET /phonebook/phonebook/contacts/{lastName} Display a contact

Uses the phonebook IMS Transaction z/OS asset

Parameters

Name	Description
lastName * required	LAST1

Responses

Code	Description	Links
200	OK	No links

Media type: application/json

You should see the following:

Curl

```
curl -X 'GET' \
  'https://designer.washington.ibm.com:9455/phonebook/phonebook/contacts/LAST1' \
  -H 'Accept: application/json' \
  -H 'Authorization: Basic RnZlZDpmcnVkcHdk'
```

Request URL

https://designer.washington.ibm.com:9455/phonebook/phonebook/contacts/LAST1

Server response

Code	Details
200	Response body

```
{ "firstName": "FIRST1", "lastName": "LAST1", "extension": "8-111-1111", "zipCode": "D01/R01" }
```

Note that the Response body of the 200 code shows the entry in the phone book.

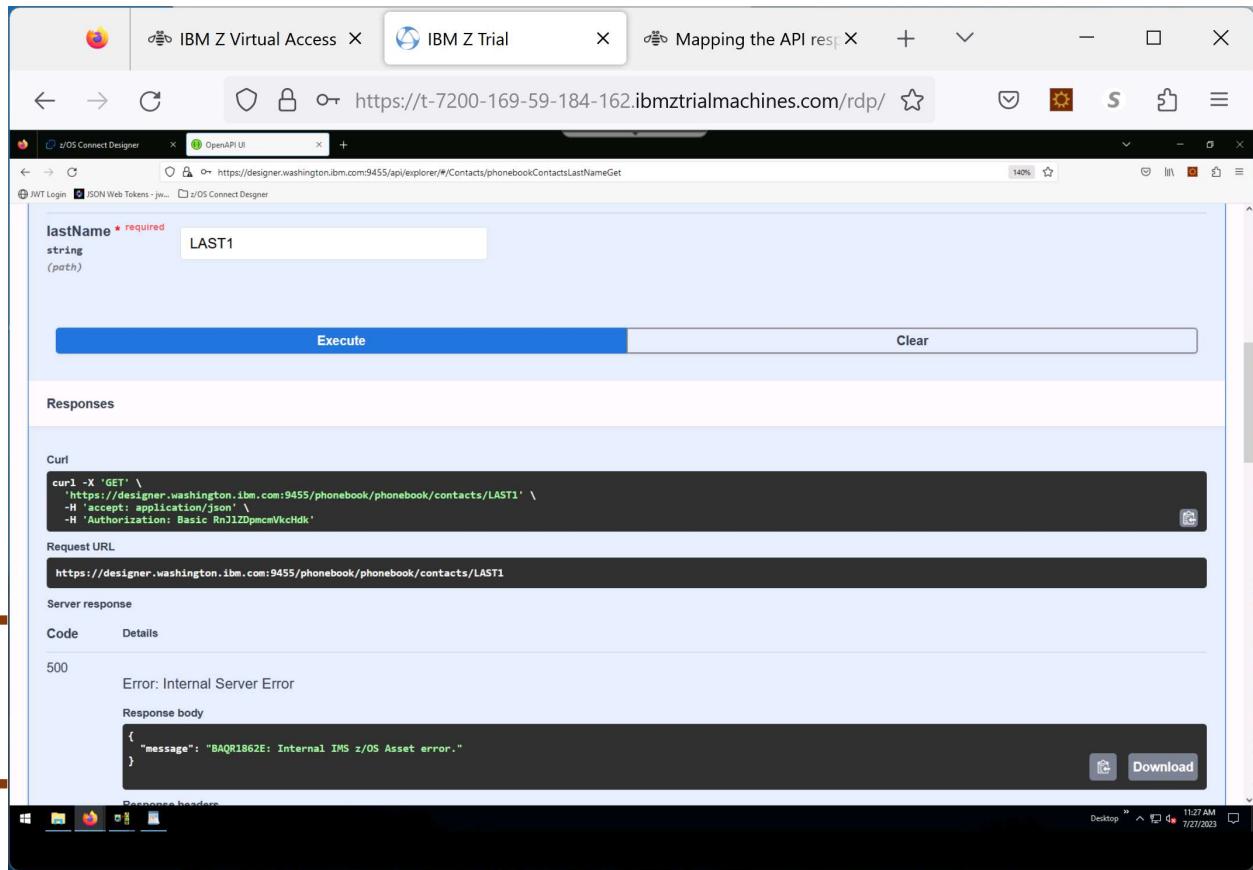
Note: after you complete Part 2 (POST) where you add your name into the phone book, you can return to this GET method to see what you successfully added.

- Try a last name (e.g., ZZZZZ) that doesn't exist in the phonebook to see what message comes back.

The screenshot shows a web browser window with three tabs: 'IBM Z Virtual Access', 'IBM Z Trial', and 'Mapping the API resp...'. The main content area displays an OpenAPI UI for a 'Contacts/phonebookContactsLastNameGet' endpoint. A form field labeled 'lastName * required' contains the value 'ZZZZZ'. Below the form is a blue 'Execute' button and a 'Clear' button. Under the 'Responses' section, there's a 'Curl' block with a command line, a 'Request URL' block with 'https://designer.washington.ibm.com:9455/phonebook/phonebook/contacts/ZZZZZ', and a 'Server response' table. The table has two rows: one for 'Code' (404) and one for 'Details' (Error: Not Found). The 'Details' row includes a 'Response body' block containing the JSON object: { "message": "Contact with the last name ZZZZZ was not found in phonebook." }. A brown bracket on the left side of the screenshot points to the 'Server response' table.

You should see the **404** error message

- What would you see if the IVTNO transaction was stopped in IMS?



The screenshot shows a Firefox browser window with three tabs open:

- IBM Z Virtual Access
- IBM Z Trial
- Mapping the API resp

The main content area displays the z/OS Connect Designer interface. A search bar at the top shows the URL: <https://t-7200-169-59-184-162.ibmztrialmachines.com/rdp/>. Below the search bar, there's a form field labeled "lastName * required" with the value "LAST1".

Below the form are two buttons: "Execute" and "Clear".

The "Responses" section contains the following details:

- Curl:**

```
curl -X 'GET' \
  https://designer.washington.ibm.com:9455/phonebook/phonebook/contacts/LAST1' \
  -H 'Accept: application/json' \
  -H 'Authorization: Basic RnJlZDpmcmVkcldk'
```
- Request URL:**
`https://designer.washington.ibm.com:9455/phonebook/phonebook/contacts/LAST1`
- Server response:**

Code	Details
500	Error: Internal Server Error

Response body:

```
{
  "message": "BAQR186ZE: Internal IMS z/OS Asset error."
}
```

Response headers:

Congratulations! You have completed the exercise for GET method.