

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

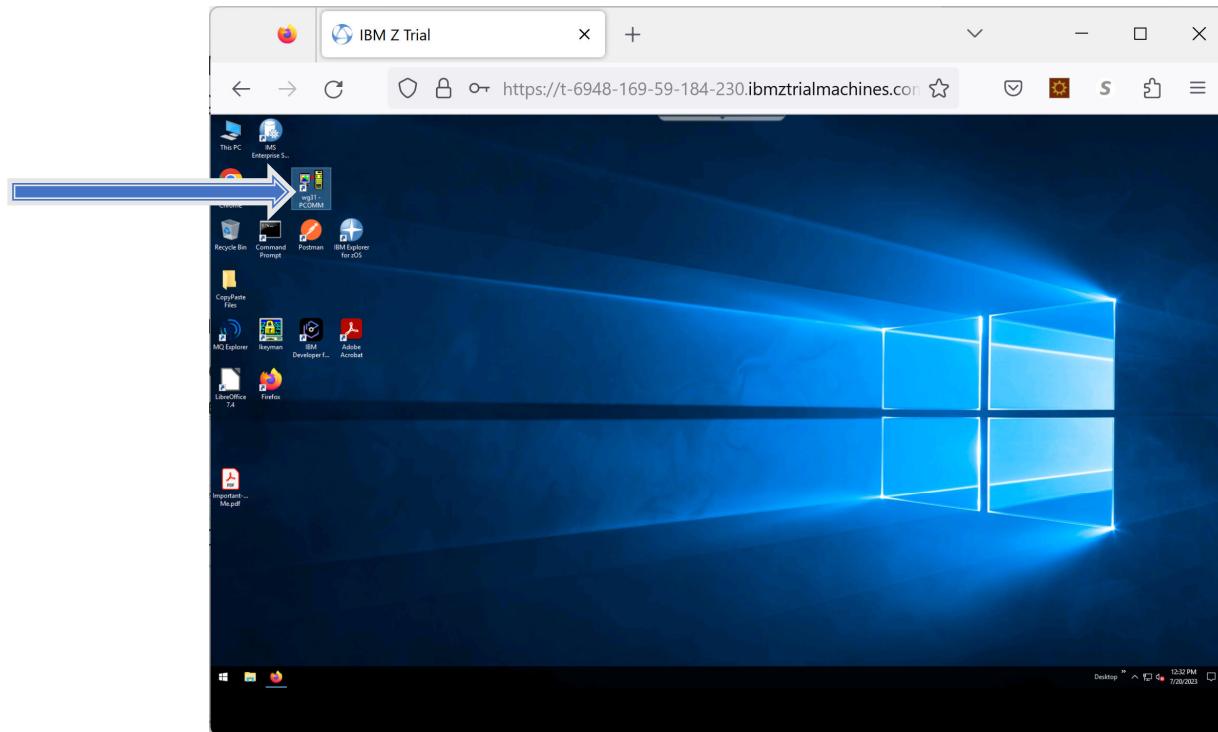
PART 3 – Create the API to UPDATE the contact you added

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

Your IP:10.1.1.1 Terminal: TCP00001
07/26/23 08:38:24

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
** Welcome to the Washington Systems Center **
** >> z/OS Connect Workshop System <<
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

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

Enter Command ==> tso USER1_

- ISPF option D allows access to SDSF

ISPF Primary Option Menu

| | | |
|----------------|------------------------------------|----------------------|
| 0 Settings | Terminal and user parameters | User ID . : USER1 |
| 1 View | Display source data or listings | Time . . : 08:39 |
| 2 Edit | Create or change source data | Terminal . : 3278 |
| 3 Utilities | Perform utility functions | Screen. : 1 |
| 4 Foreground | Interactive language processing | Language. : ENGLISH |
| 5 Batch | Submit job for language processing | Appl ID. : ISR |
| 6 Command | Enter TSO or Workstation commands | TSO logon : DBSPROCC |
| 7 Dialog Test | Perform dialog testing | TSO prefix: USER1 |
| 9 IBM Products | IBM program development products | System ID : S0W1 |
| 10 SCLM | SM Configuration Library Manager | MVS acct. : ACCT# |
| 11 Workplace | ISPF Object/Action Workplace | Release . : ISPF 7.5 |
| M More | Additional IBM Products | |
| D SDSF | Spool Search and Display Facility | |
| Q QM | QM MVS/ESR Utility Menu | |

Enter X to Terminate using log/list defaults

- Go to LOG

Display Filter View Print Options Search Help

SDSF MENU V2R5MO ADCDPL S0W1 LINE 1-27 (73)

COMMAND INPUT ==> LOG SCROLL ==> PAGE

| NP | NAME | DESCRIPTION | Group | Status |
|----|------|-------------------------|---------|--------|
| | DA | 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 | |
| | MAS | Members in the MVS | JES | |
| | JC | Job classes | JES | |
| | SE | Scheduling environments | MLM | |
| | RES | MLM resources | MLM | |
| | ENC | Enclaves | MLM | |
| | PS | Processes | OMVS | |
| | SYS | System information | System | |
| | EVO | Enqueue | System | |
| | ENQC | Enqueue contention | System | |
| | ENQD | Enqueued data sets | Susplex | |
| | 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 | |

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

The screenshot shows a terminal window titled "Session A - (1x2 - 80)" running on "IBM Z Trial". The URL is <https://t-7200-169-59-184-162.ibmtrialmachines.com/dp/>. The terminal displays a series of DFSMS commands related to tape management and file processing.

```
Display Filter View Print Options Search Help
SDSF SYSLOG      0.101 S0W1 S0W1 07/26/2023 2W        4,784  COLUMNS 52- 131
COMMAND INPUT ==> -
0290 R 05./DIS TRAN IVTNO.
0090 IEE8001 REPLY TO 05./DIS TRAN IVTNO.
0090 DFS0001   TRAN    CLS ENQCT  OCT  LCT  PLCT CP NP LP SEGSZ SEGN0
          PARLM   RC   IVP1
0090 DFS0001   IVTNO   1   0   0 65535 65535 1 1   0   0
          NONE     0   IVP1
0090 DFS0001           PSBNM: DFSIVP1   IVP1
0090 DFS0001   *23207/085309*   IVP1
0090 *07 DFS9961 *IMS READY* IVP1
S READY* IVP1
IMS CONNECT READY* IMS15HWS
***** BOTTOM OF DATA *****
```

- 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
-----+
SDSF SYSLOG 0 101 S0W1 S0W1 07/26/2023 2W 4,795 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 IMWMSG1 DLS 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 IMWMSG1 DLS IVP1
0090 DFS000I VTAM ACB OPEN -LOGONS ENABLED IVP1
0090 DFS000I INSLU=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 QUILF STATUS = 1 LOCAL=INACTIVE
0090 DFS000I APPC/OTMA SHARED QUEUES LOGGING=N IVP1
0090 DFS000I APPC/OTMA RRS MAX TCBS = 40 ATTACHED TCBS = 2 QUEUED
RRSOKS= 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 *05 DFS996I *IMS READY* IVP1
S READY* IVP1
04/021
Connected to remote server/host wip31 using lu/pool TCP0001 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
-----+
SDSF SYSLOG 0_101 S0W1 S0W1 07/26/2023 2W 4,739 COLUMNS 02- 81
COMMAND INPUT ==> /03viewhws_
PARLM RC IVP1
S DFS000I IVTNO 1
S DFS000I NONE 0 IVP1
N DFS000I PSBNAME:
N DFS000I *23207/08483
W DFS000I *05 DFS996I *IMS READY* IVP1
4200000 S0W1 08.48.38 STC01676 *05 DFS996I *IMS CONNECT READY* IMS15HWS
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 TCP0001 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  ODBR IMSPLEX MEMBER=IMS15HWS        TARGET MEMBER=PLEX1
0090 HWS00001I  DATASTORE=IPV1   STATUS=ACTIVE
0090 HWS00001I  GROUP=OTMAGRP MEMBER=HWSMEM
0090 HWS00001I  TARGET MEMBER=OTMAMEM   STATE=AVAIL
0090 HWS00001I  DEFAULT REROUTE NAME=HWSDEF  IMS VERSION=15.1
0090 HWS00001I  RACF APPL NAME=IMSAPP 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=    CMD 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 *****

```

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 <<      **
****

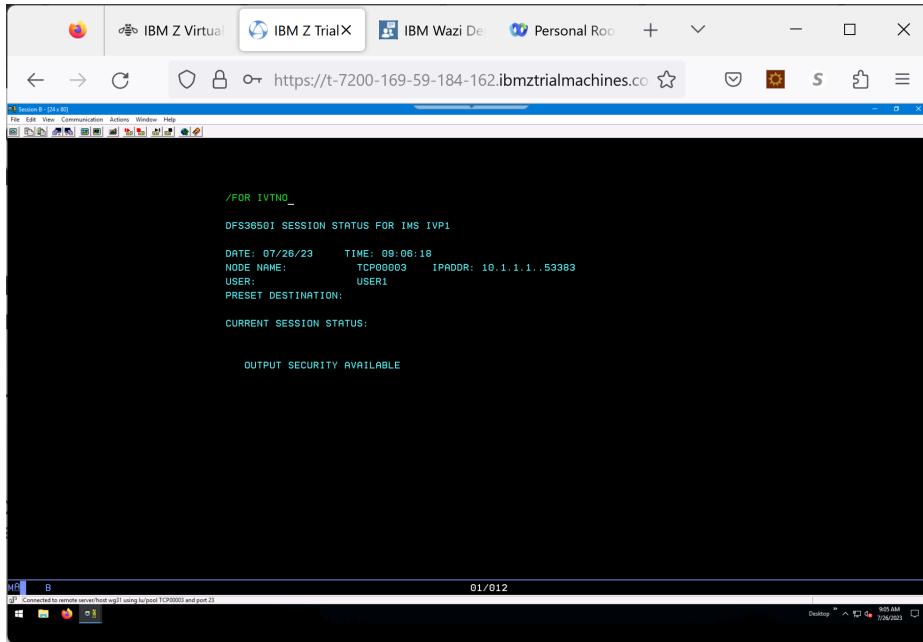
Enter TSO userid ... for a TSO session
CICS      ... for a CICS session
IMS       ... for an IMS session

Enter Command ==> IMS_
```

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)

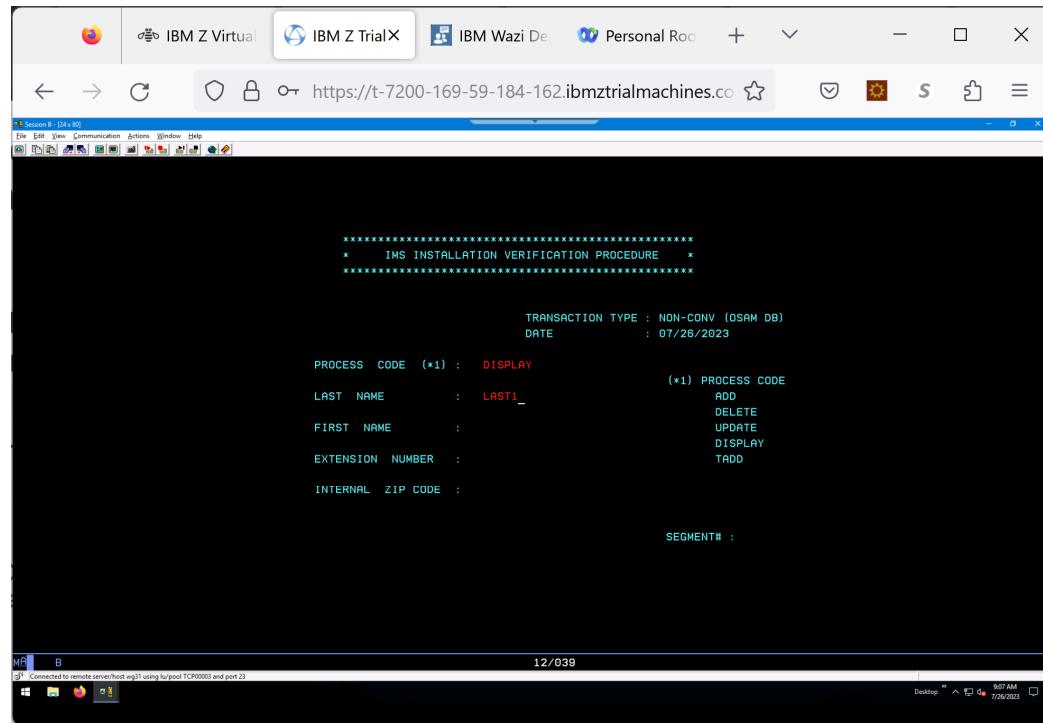


```
/FOR IVTNO_
DFS3050I SESSION STATUS FOR IMS IVP1
DATE: 07/26/23    TIME: 09:08:18
NODE NAME:      TCP00003  IPADDR: 10.1.1.1..5383
USER:          USER1
PRESET DESTINATION:

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)



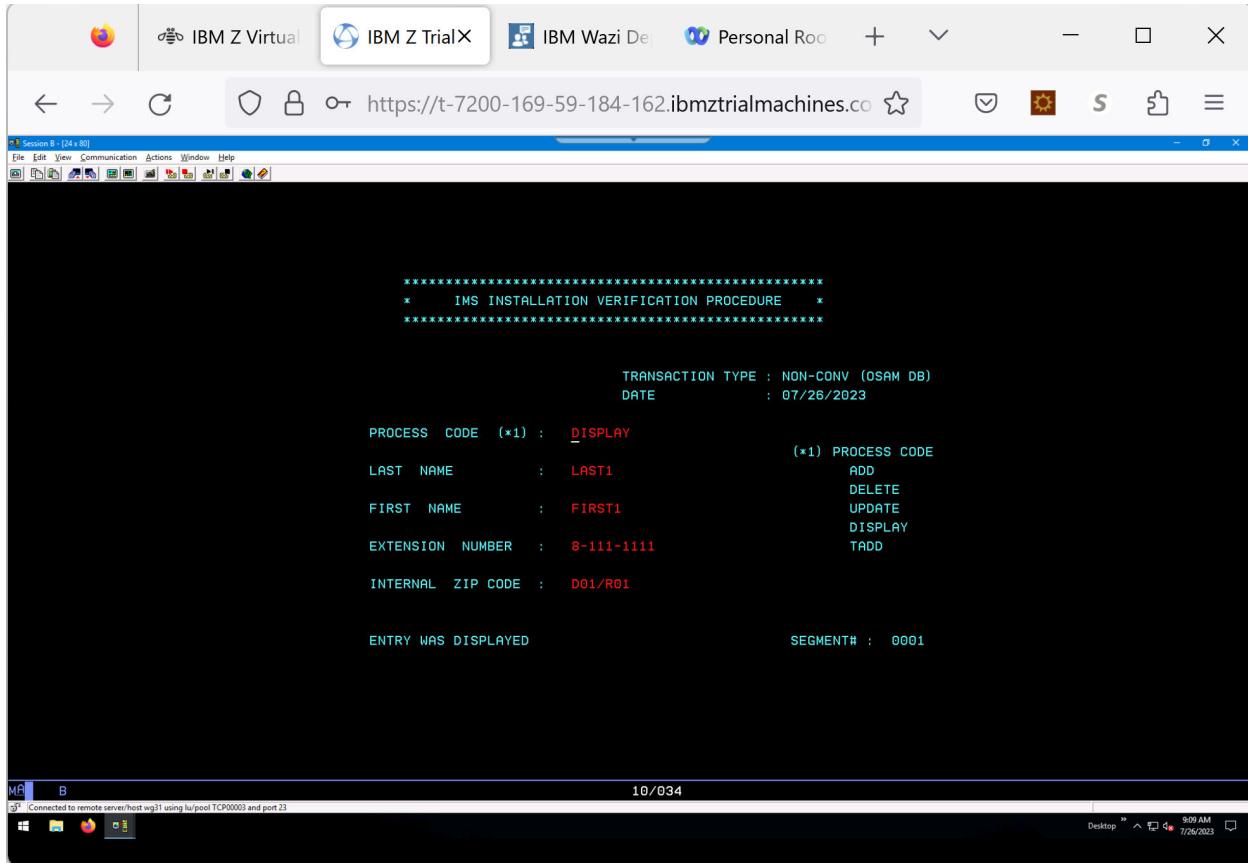
```
*****
*   IMS INSTALLATION VERIFICATION PROCEDURE  *
*****
```

| | |
|---------------------|--------------------|
| 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:

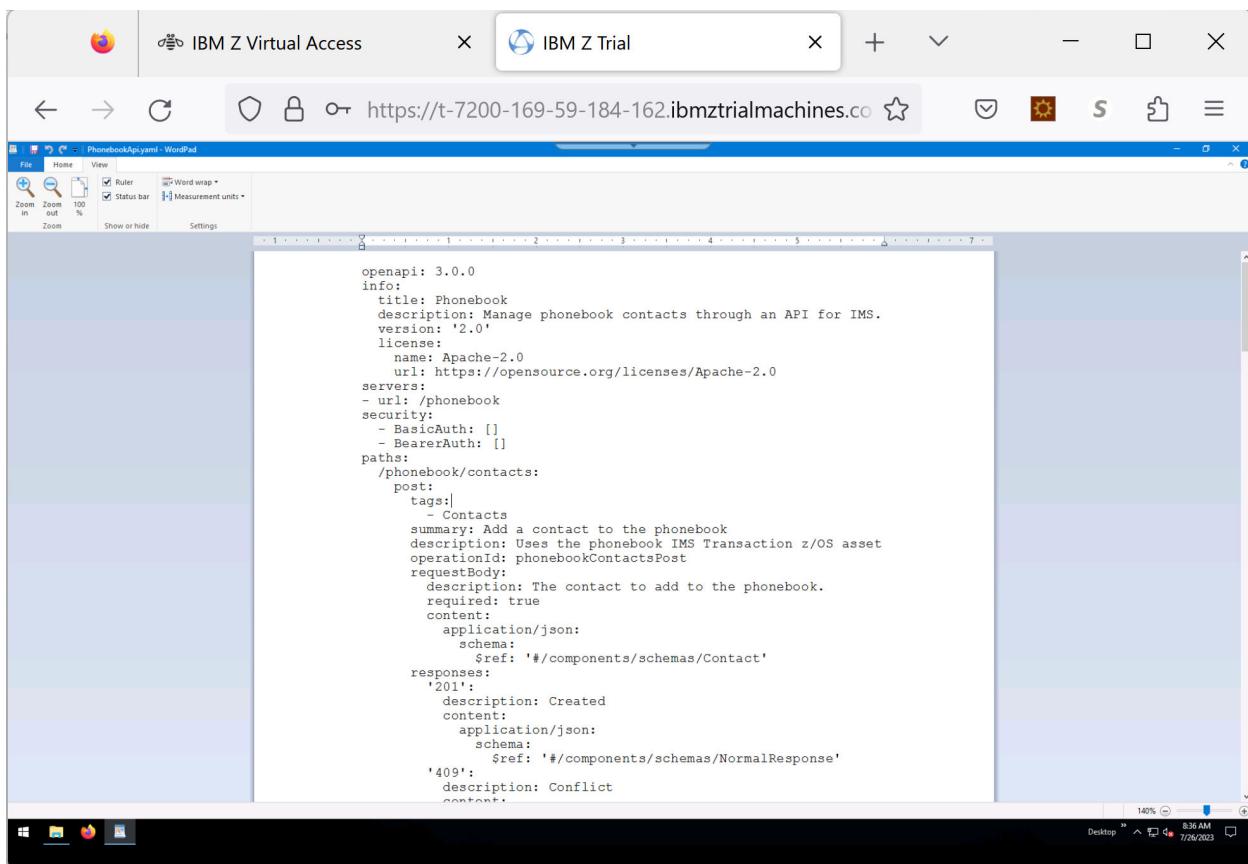
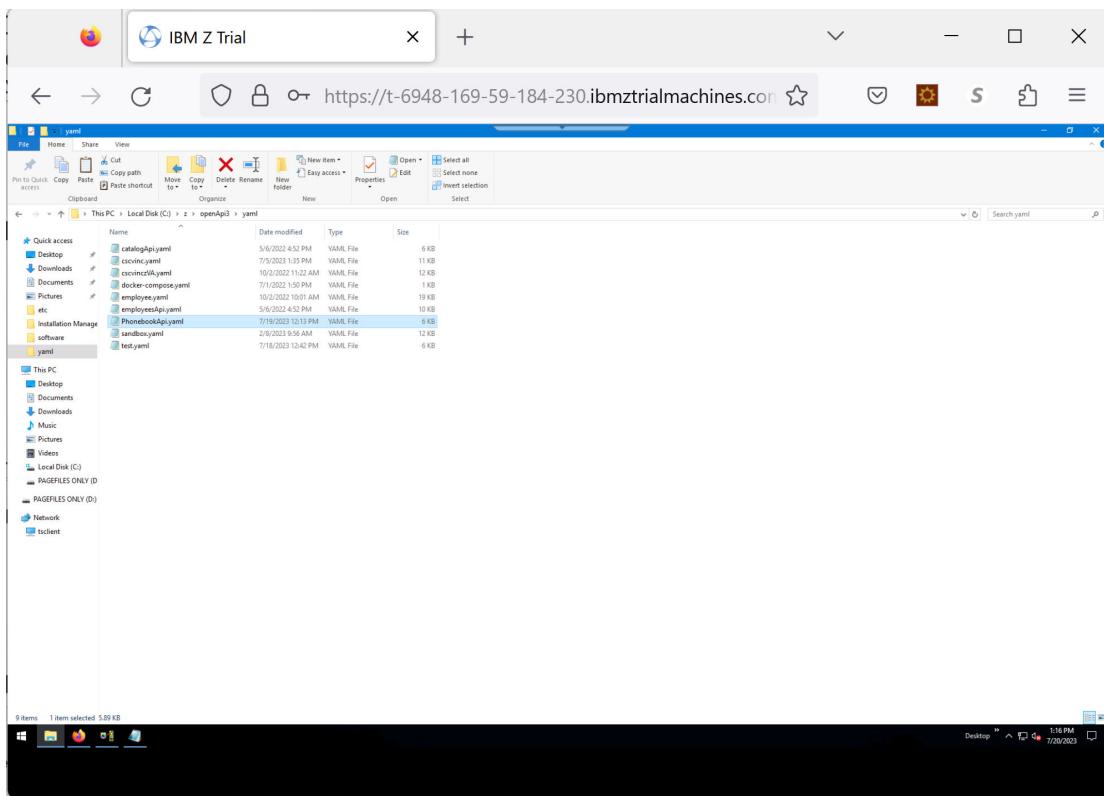
```
*****
*     IMS INSTALLATION VERIFICATION PROCEDURE      *
*****  
TRANSACTION TYPE : NON-CONV (OSAM DB)  
DATE          : 07/26/2023  
  
PROCESS CODE (*1) : DISPLAY          (*1) PROCESS CODE  
LAST NAME      : LAST1            ADD  
FIRST NAME     : FIRST1           DELETE  
EXTENSION NUMBER : 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 wq31 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

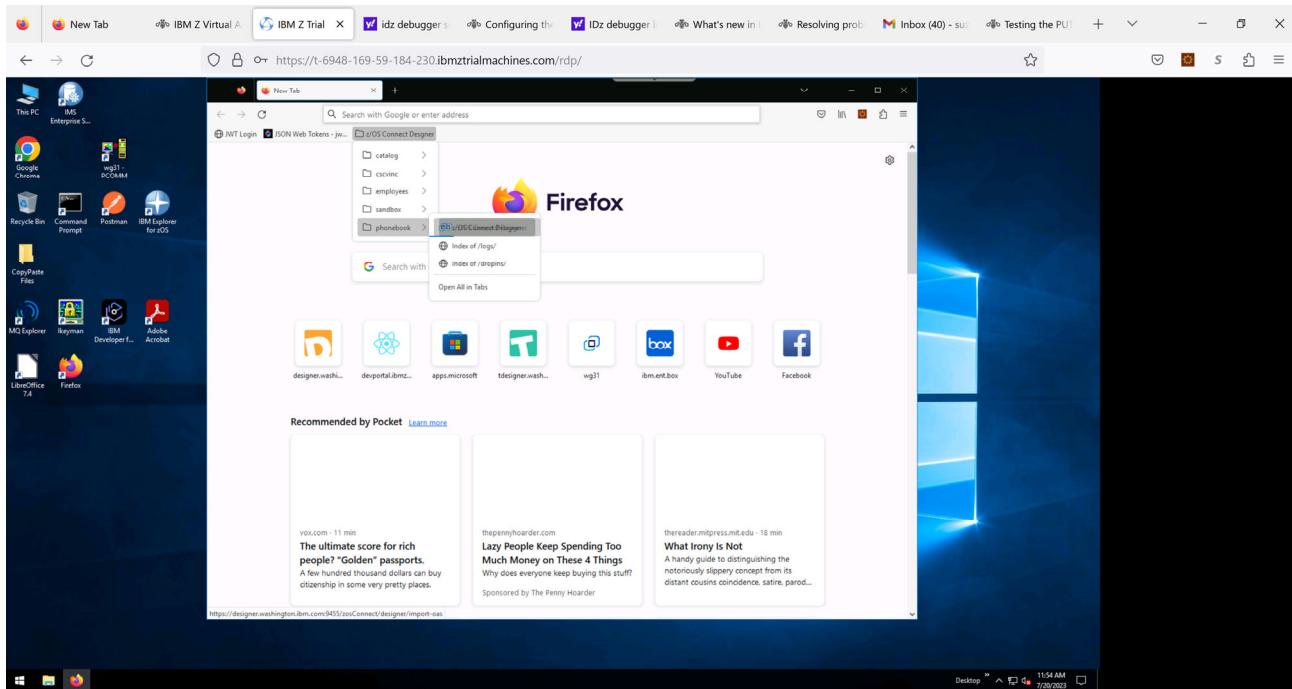
For this lab, a **PhonebookAPI.yaml** file has been provided in the directory

C: z/openapi3/yaml

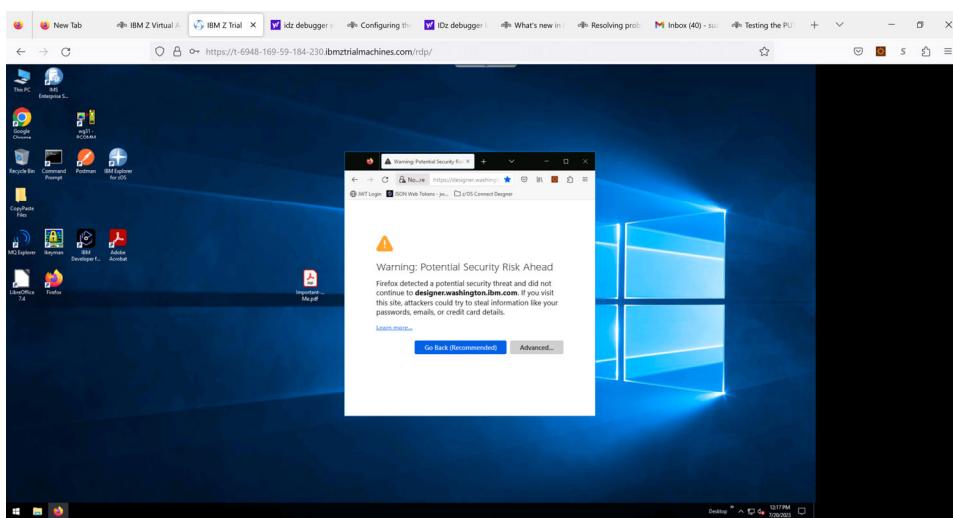


To create the API, open up a **Firefox browser**

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

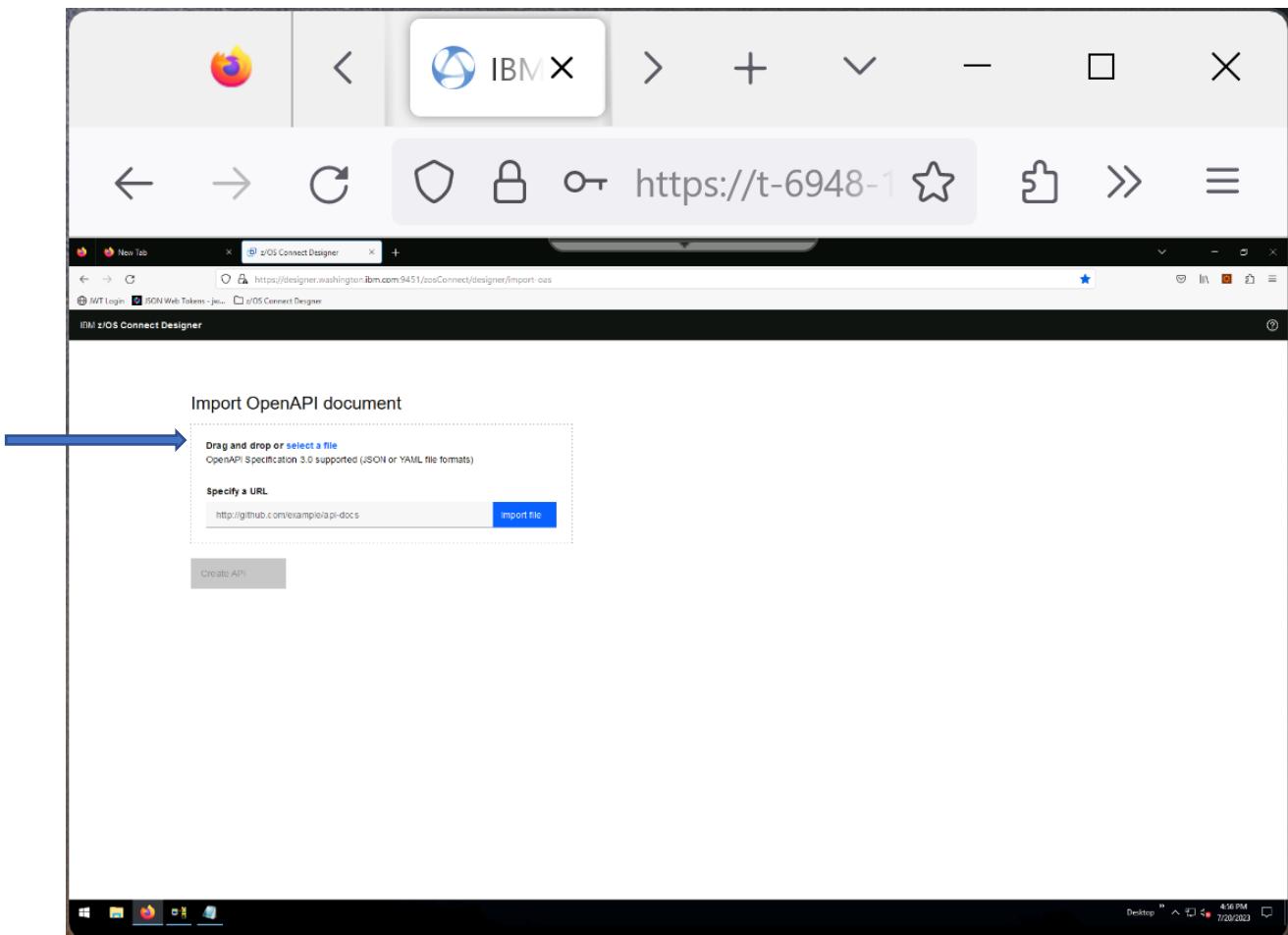


- Accept the risk and wait (it takes a while to set it up)

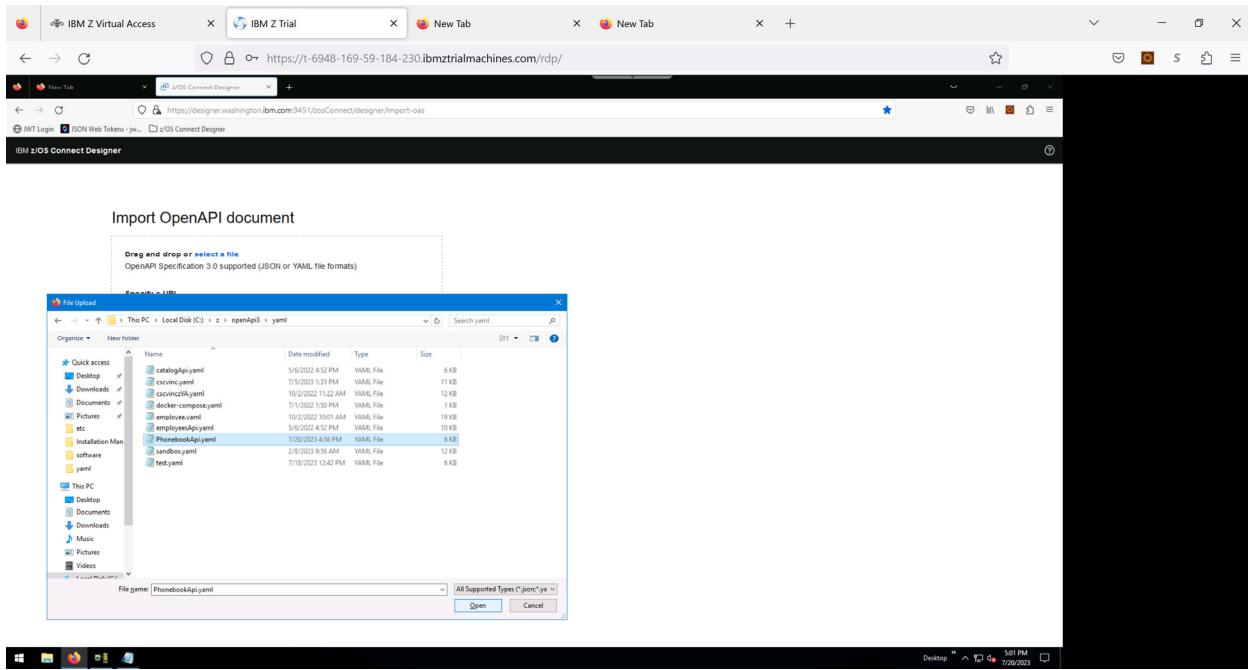


The tool will ask you to choose an OpenAPI document (yaml)

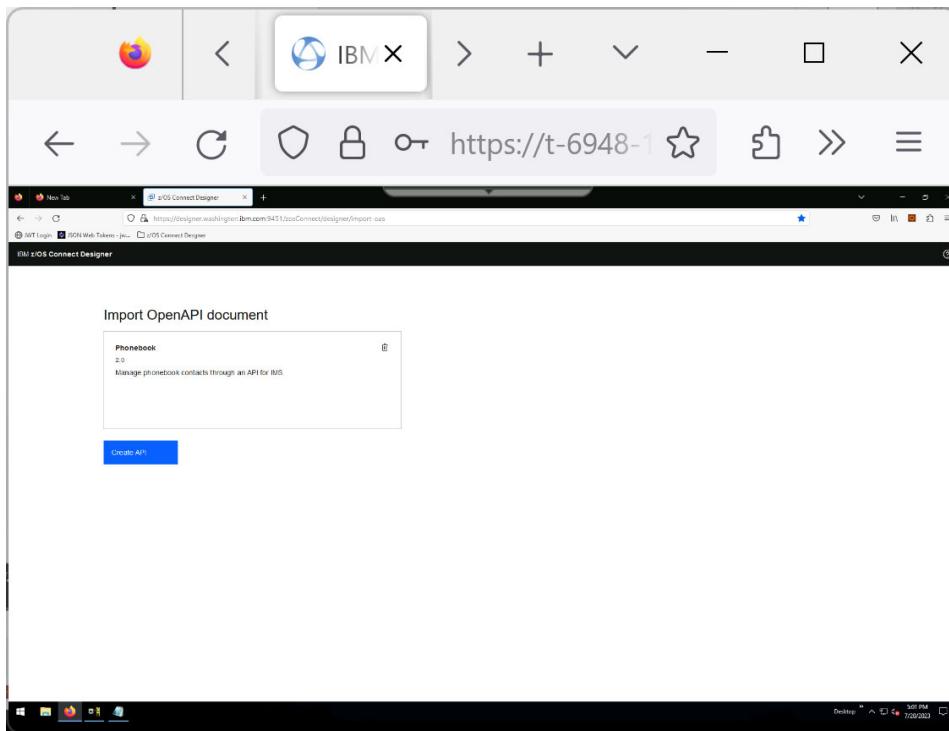
- Click on **select a file**



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



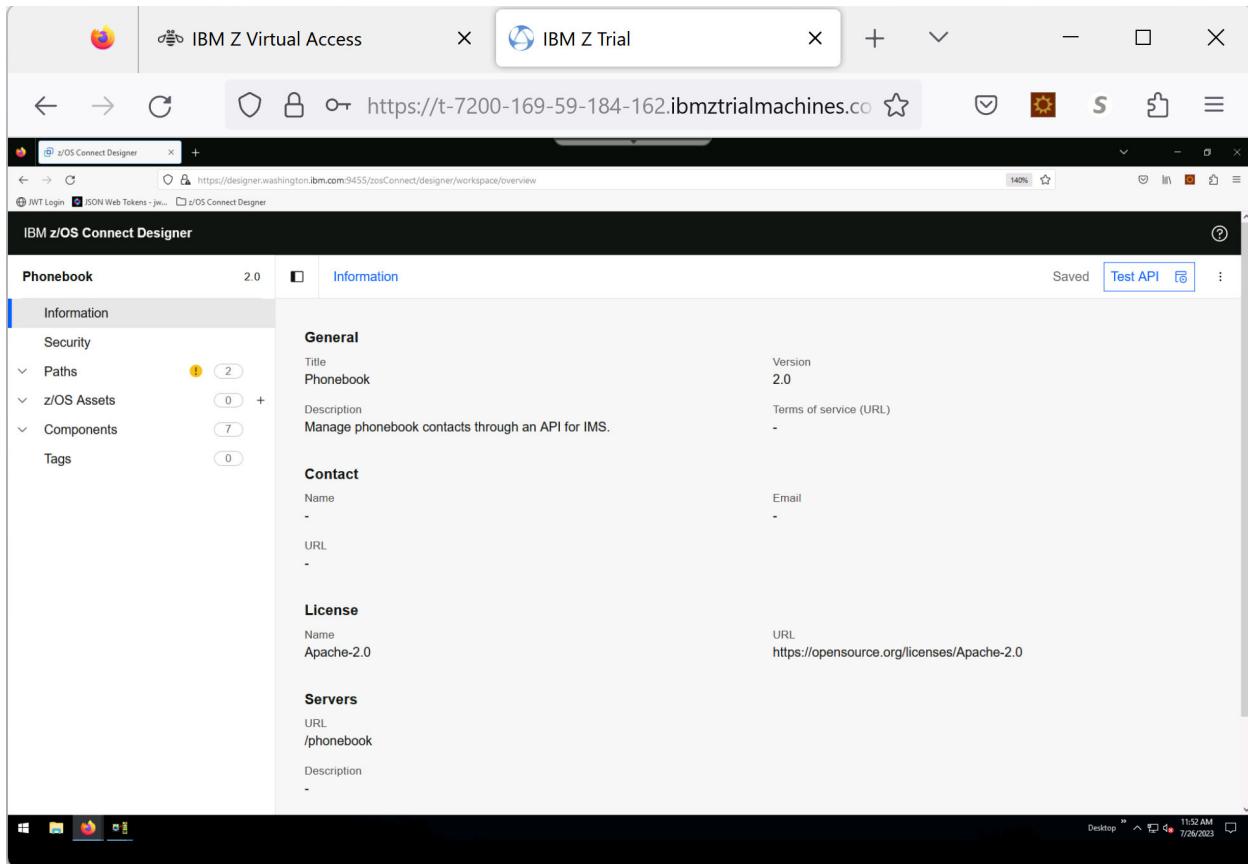
- Click on Create API



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.

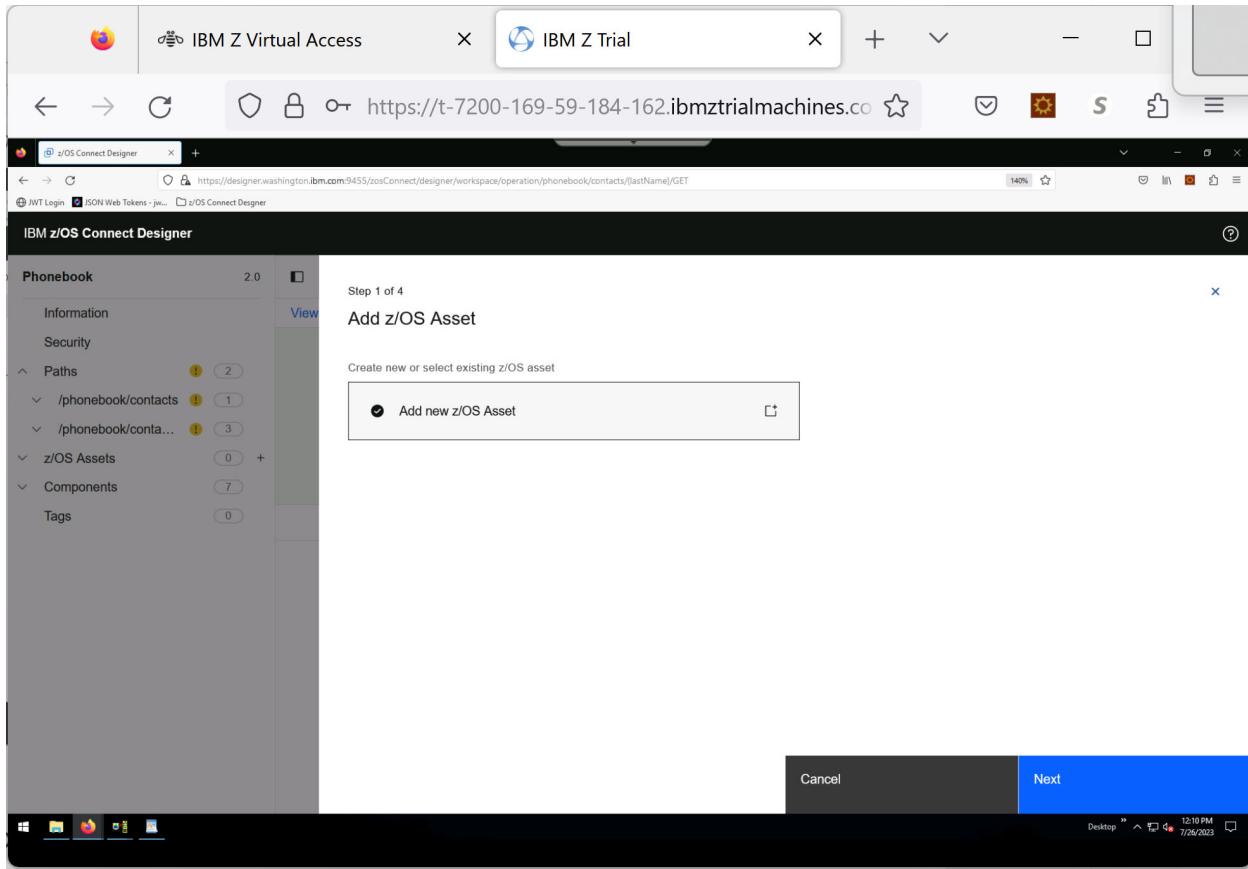
CREATE the z/OS asset



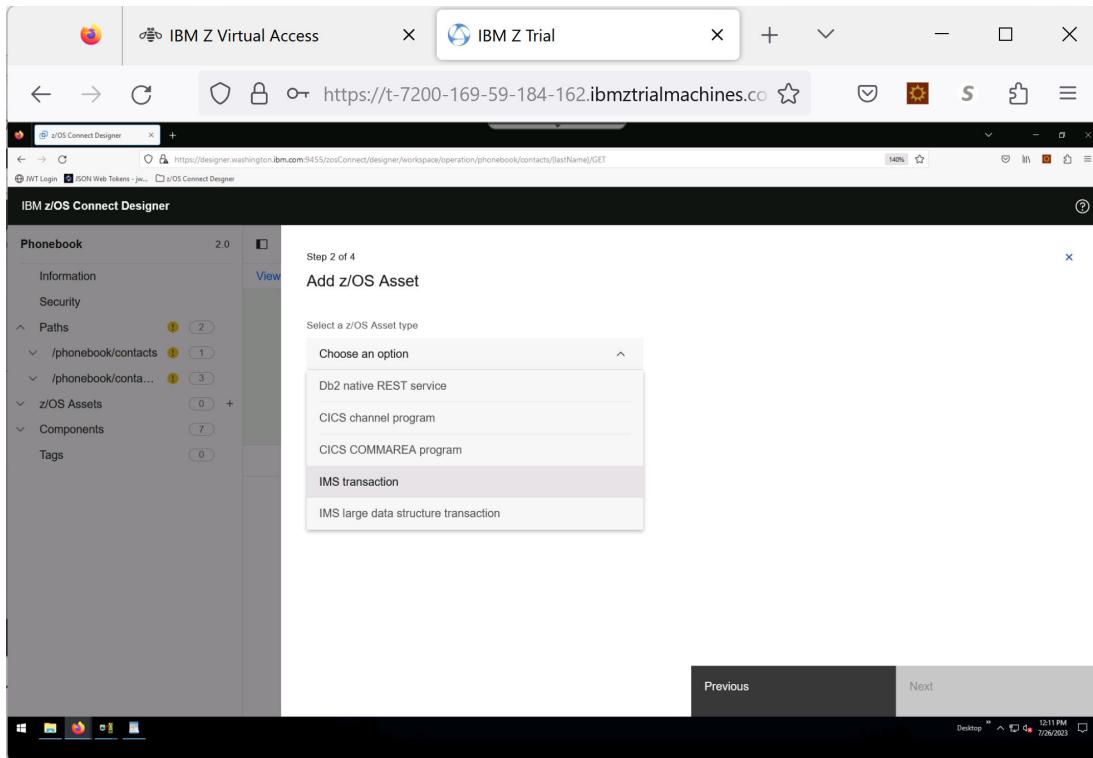
Click on the plus (+) sign by z/OS Assets

- 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, and tramsod usage/



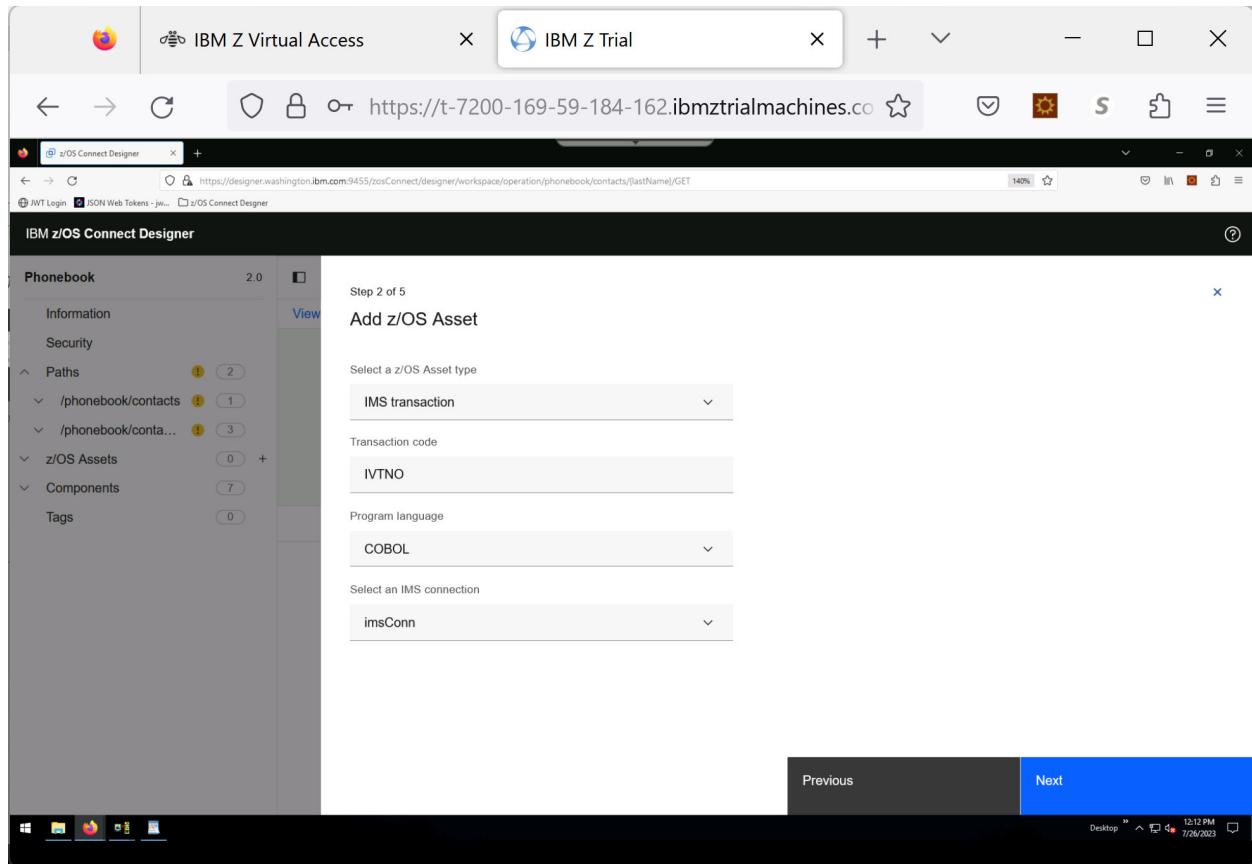
- Highlight “Add new z/OS Asset” and click Next.



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

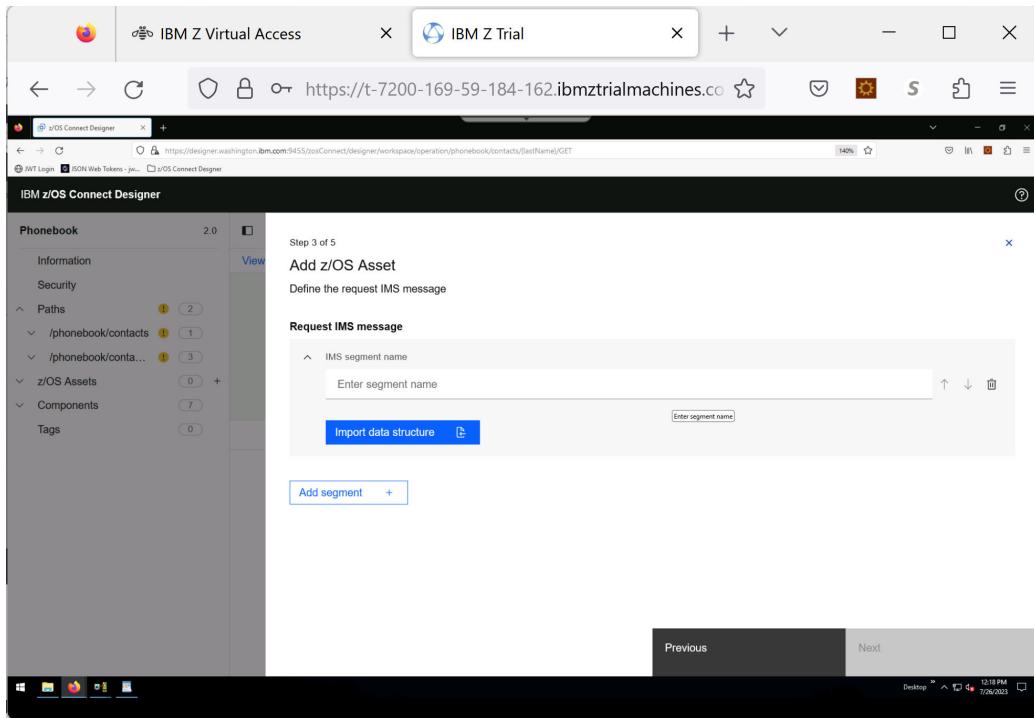
When the window opens,

- 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

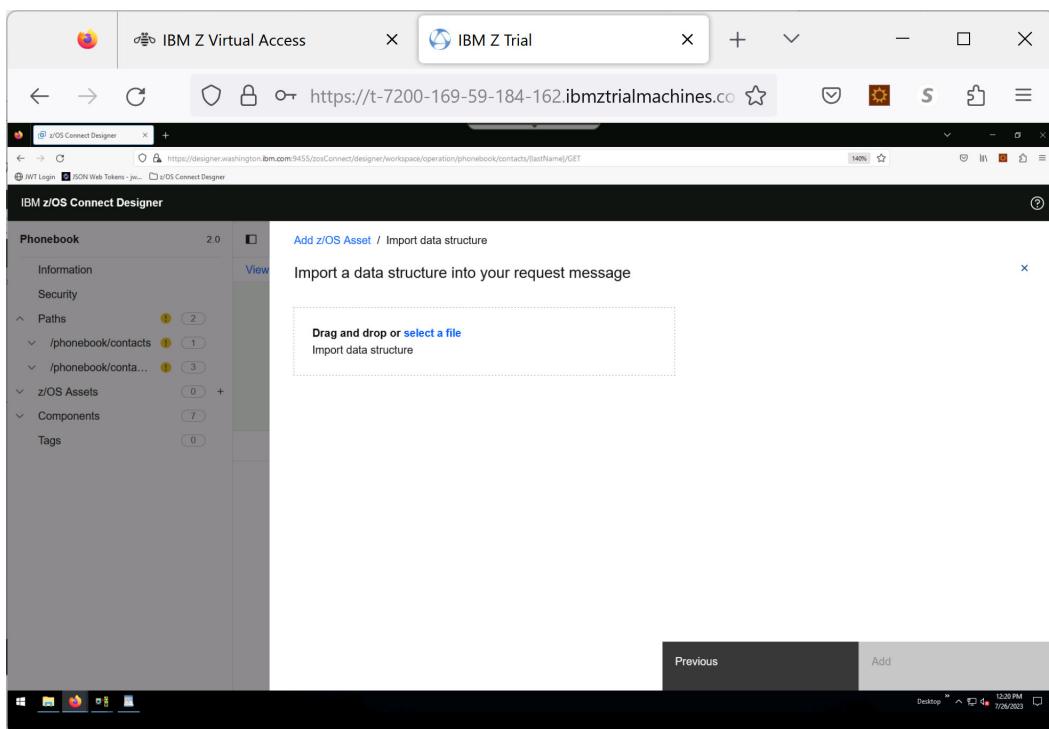


Click **Next**.

Define the request IMS message

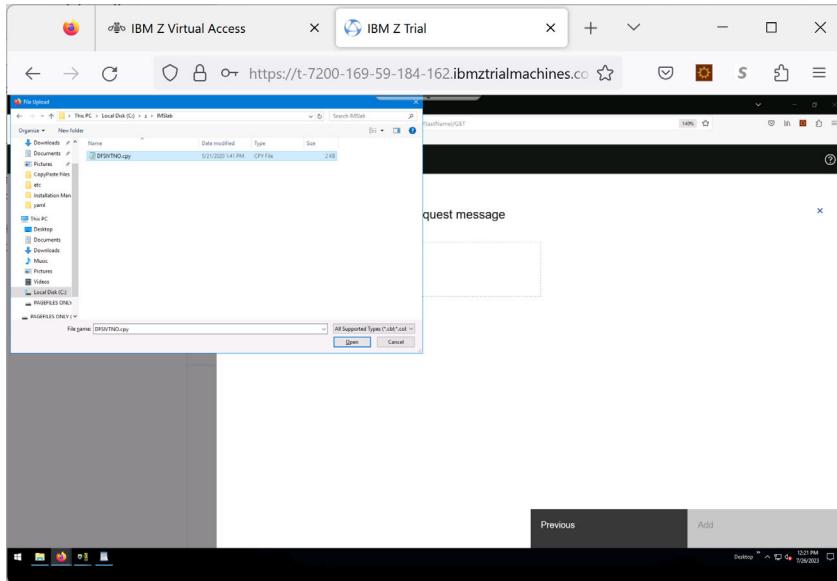


- Key in **INPUT-MESSAGE** as the segment name and click on **Import data structure**



Click on **select a file**

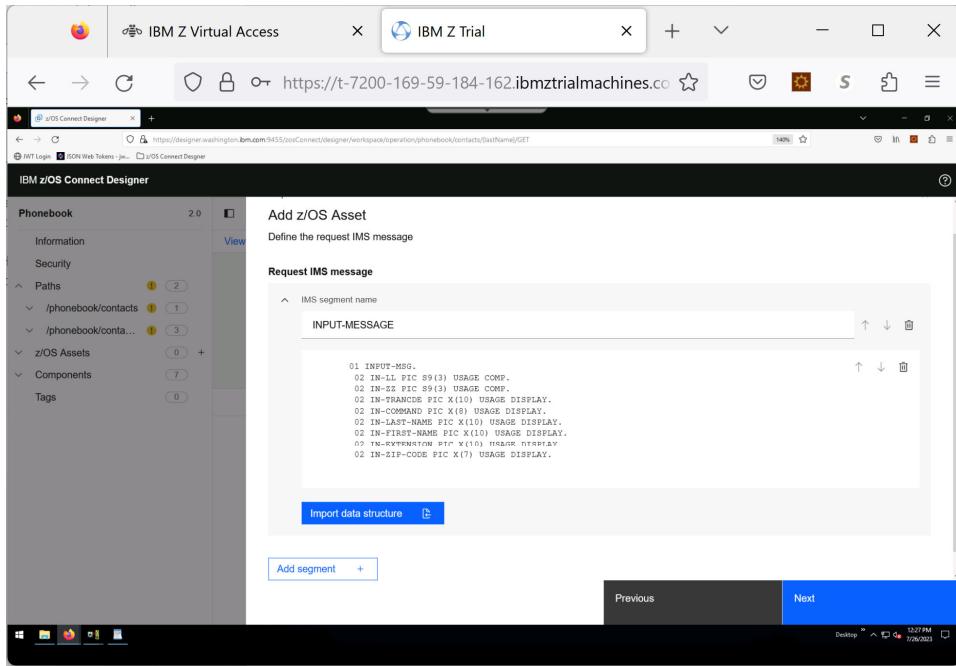
- Select **DFSIVTNO.CPY** which is in **C:/z/imslab**
 - This is the COBOL copybook for the input/output message structures associated with the IVTNO transaction



A screenshot of the 'z/OS Connect Designer' interface. The main title is 'IBM z/OS Connect Designer'. On the left, there's a sidebar with 'Phonebook' and other navigation options. The main area has a heading 'Add z/OS Asset / Import data structure' and a sub-instruction 'Import a data structure into your request message'. Below this is a section titled 'Drag and drop or select a file' with the sub-instruction 'Import data structure'. To the right is a table:

| Copybook name | Data structure | Status |
|--|----------------|----------|
| <input checked="" type="checkbox"/> DFSIVTNO.cpy | INPUT-MSG | Imported |
| <input type="checkbox"/> DFSIVTNO.cpy | OUTPUT-AREA | Imported |

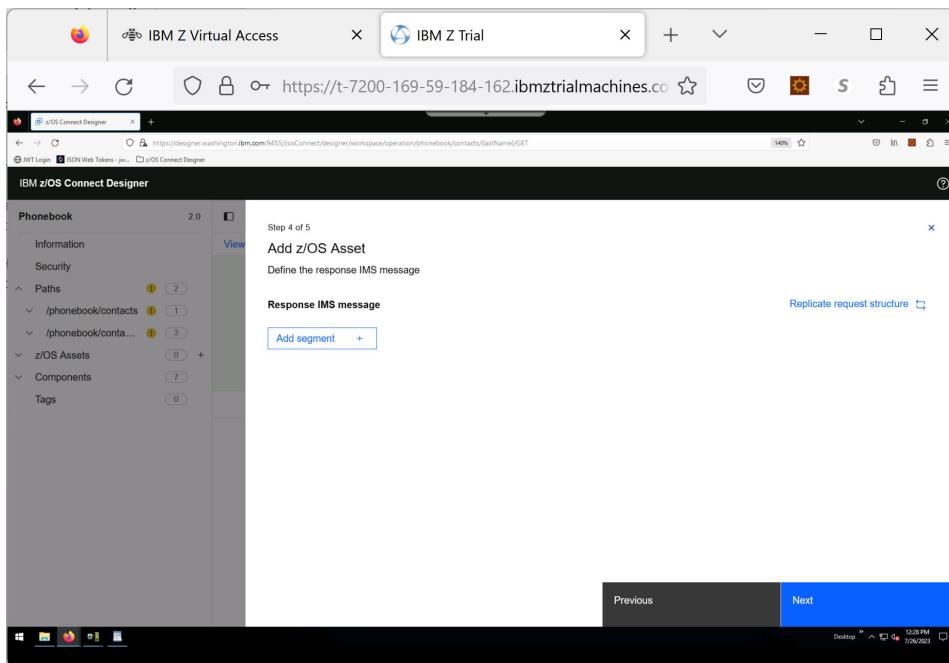
To include the request message, ONLY select the INPUT-MSG data structure. Click **Add**.



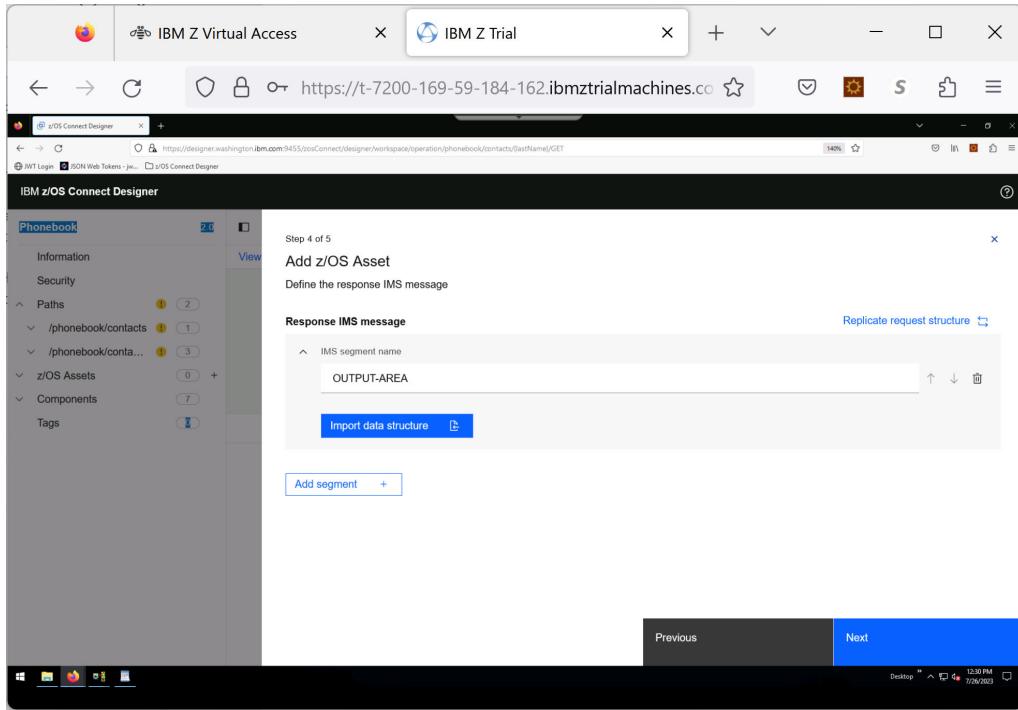
This is the COBOL copybook associated with the input message for IVTNO.

Click **Next**.

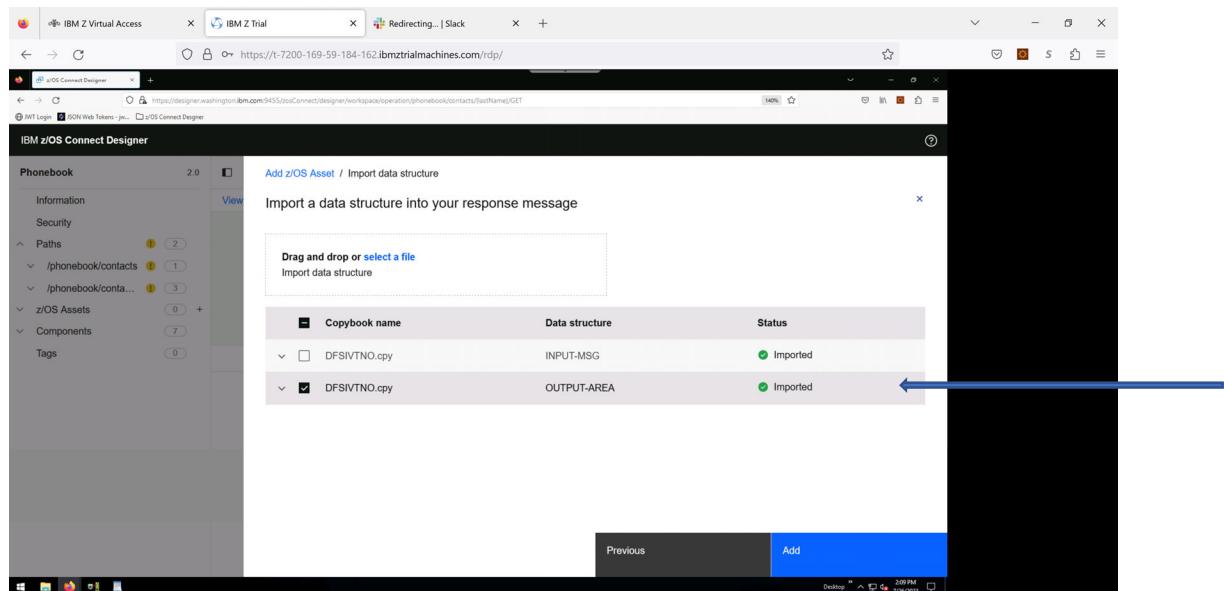
Add the segment for the **response** message.



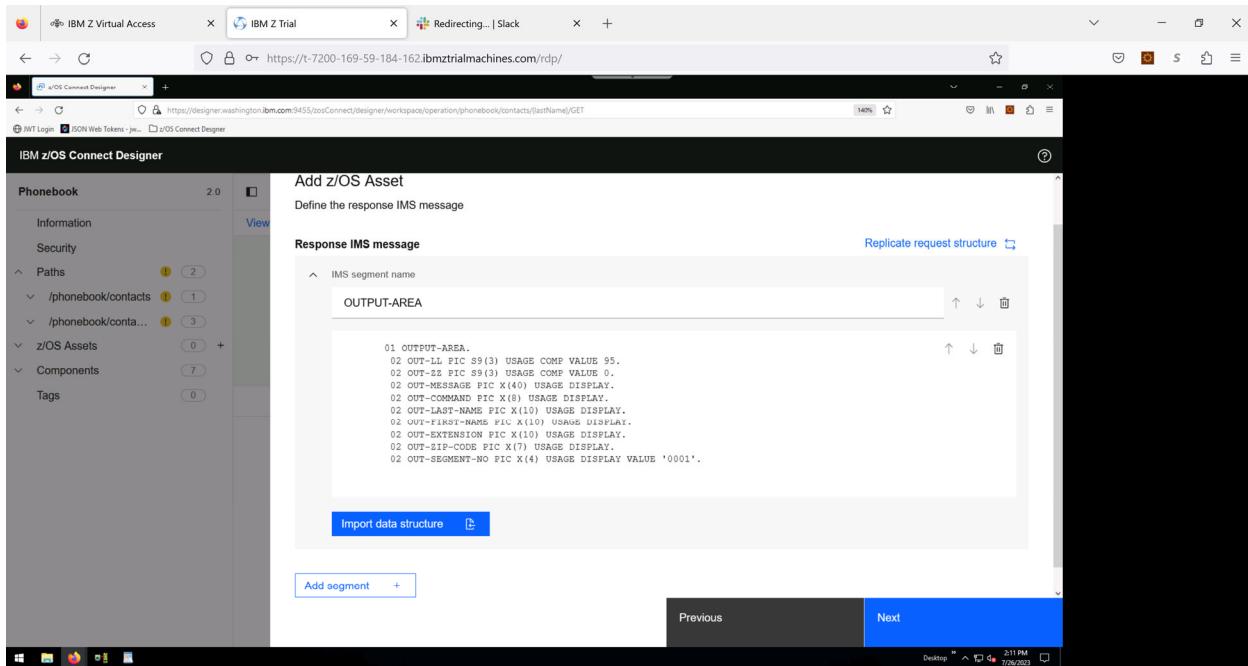
For segment name, specify **OUTPUT-AREA**



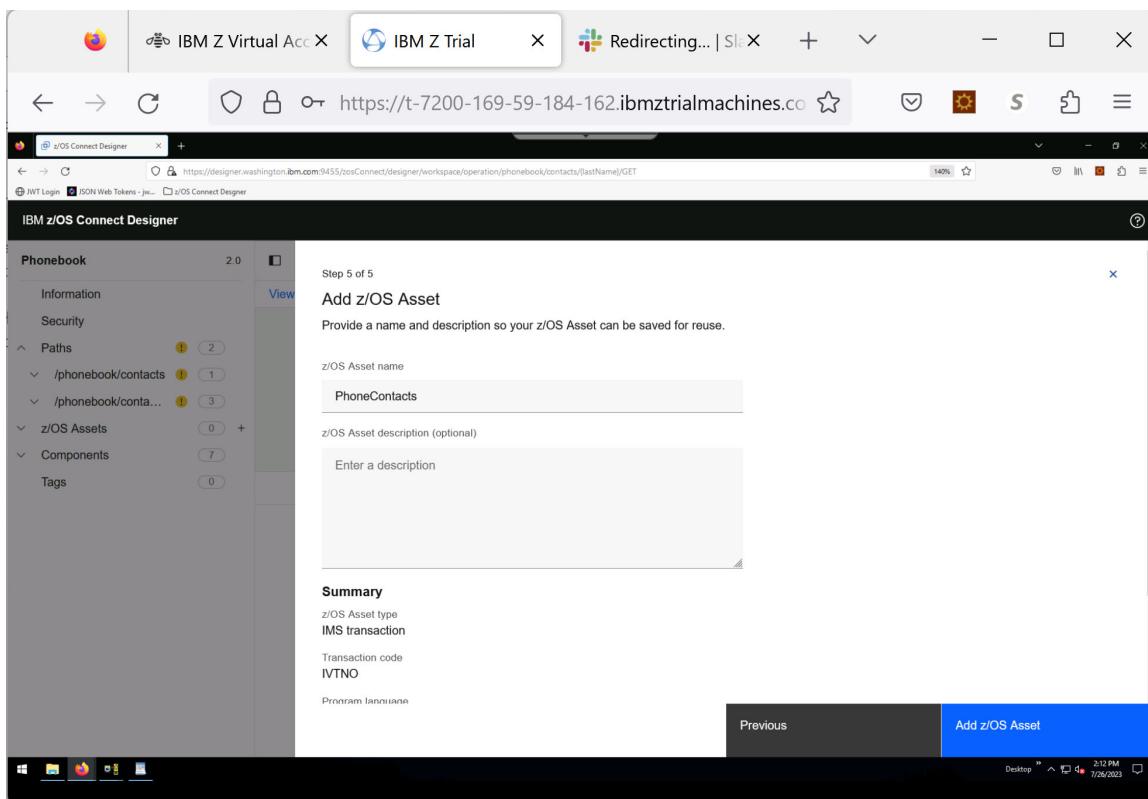
- Click **Import data structure** and follow the steps, as above, to retrieve the DFSIVTNO.CPY file



- 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 interface. The left sidebar displays the asset structure under 'Phonebook' version 2.0:

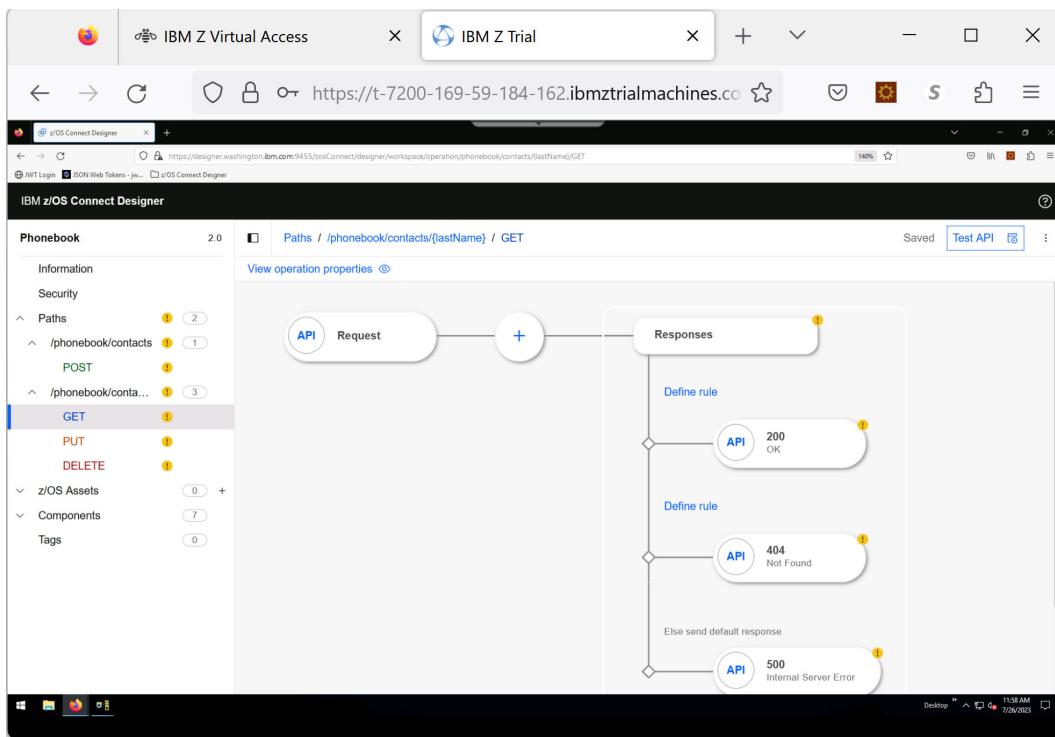
- Information
- Security
- Paths
 - /phonebook/contacts (1)
 - /phonebook/conta... (3)
 - GET (1)
 - PUT (1)
 - DELETE (1)
- z/OS Assets
 - PhoneContacts (selected)
- Components (7)
- Tags (0)

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

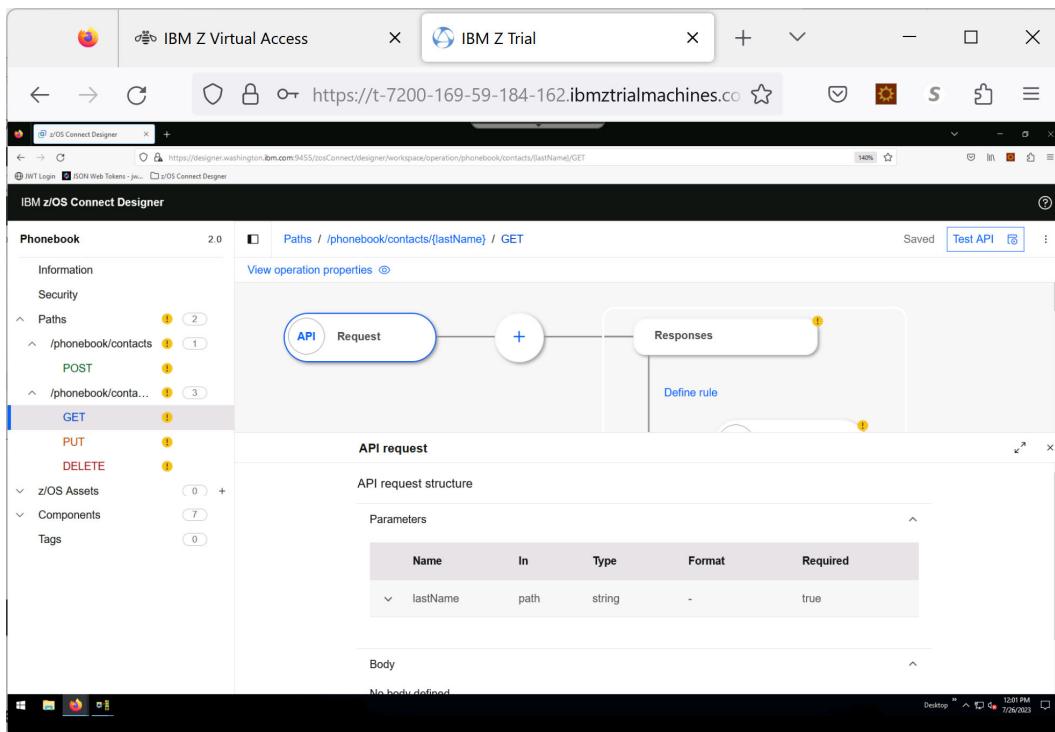
- General**: Name is 'PhoneContacts', Type is 'IMS transaction'.
Description is empty.
z/OS Asset options are available.
- IMS transaction**: Transaction code is 'IVTNO', Program language is 'COBOL'.
Connection profile is 'imsConn'.
- Request structure**: INPUT-MESSAGE
- Response structure**: OUTPUT-AREA

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



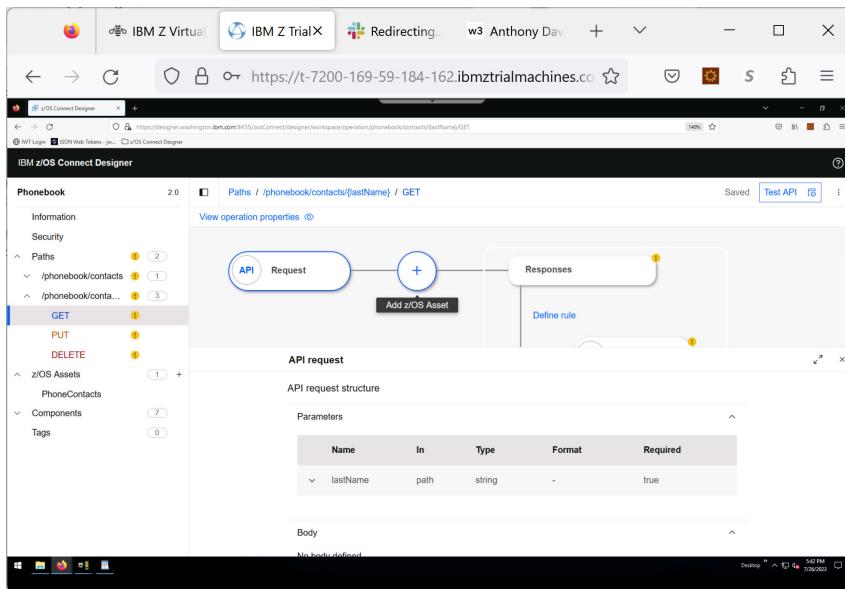
Notice that the lastName parameter is part of the request – this parameter was defined in the phonebook.yaml

```

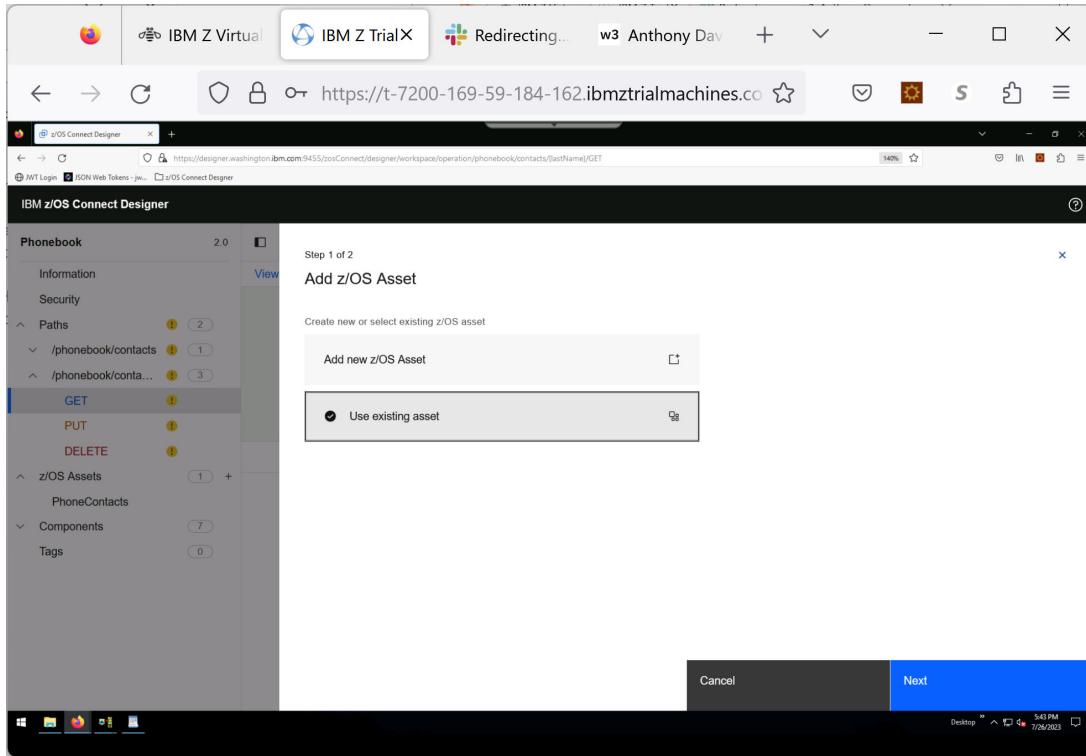
name: Apache-2.0
url: https://opensource.org/licenses/Apache-2.0
server:
  url: /phonebook
  security:
    - BasicAuth: []
    - OAuth2Auth: []
  paths:
    /phonebook/contacts:
      get:
        tags:
          - Contacts
        summary: Add a contact to the phonebook
        description: Use the phonebook IM9 Transaction z/OS asset
        requestBody:
          description: The contact to add to the phonebook.
          required: true
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Contact'
        responses:
          '201':
            description: Created
            content:
              application/json:
                schema:
                  $ref: '#/components/schemas/NormalResponse'
          '409':
            description: Conflict
            content:
              application/json:
                schema:
                  $ref: '#/components/schemas/Conflict'
          '400':
            description: Bad Request
            content:
              application/json:
                schema:
                  $ref: '#/components/schemas/NormalResponse'
          '500':
            description: Internal Server Error
            content:
              application/json:
                schema:
                  $ref: '#/components/schemas/ErrorResponse'
        parameters:
          lastName:
            description: The last name of the contact to add
            type: string
            format: null
            required: true
            example: Smith
      put:
        tags:
          - Contacts
        summary: Update a contact in the phonebook
        description: Use the phonebook IM9 Transaction z/OS asset
        operationId: phonebookContactsLastNamePut
        parameters:
          lastName:
            description: The last name of the contact to update
            type: string
            format: null
            required: true
            example: Smith
  
```

Map the API Request to the z/OS asset

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

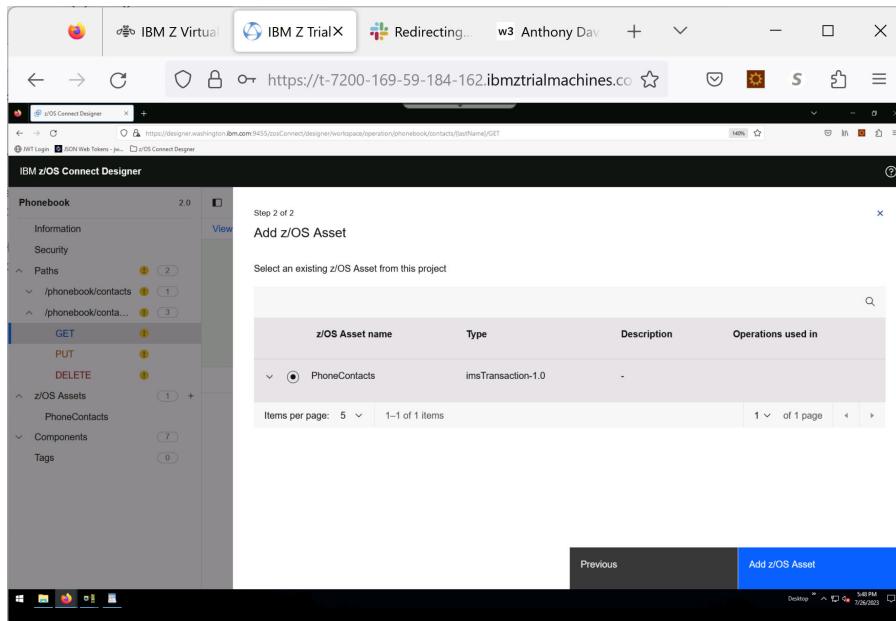


- Click on **Use existing asset** since we will use the one you just created.



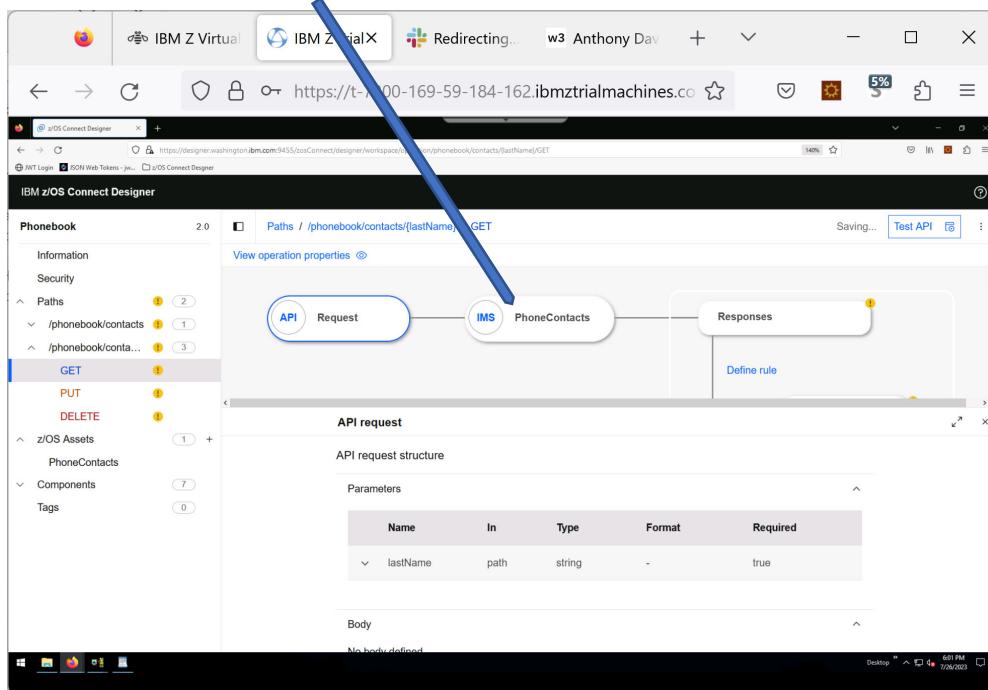
- Click **Next**.

The z/os Asset previously created is displayed

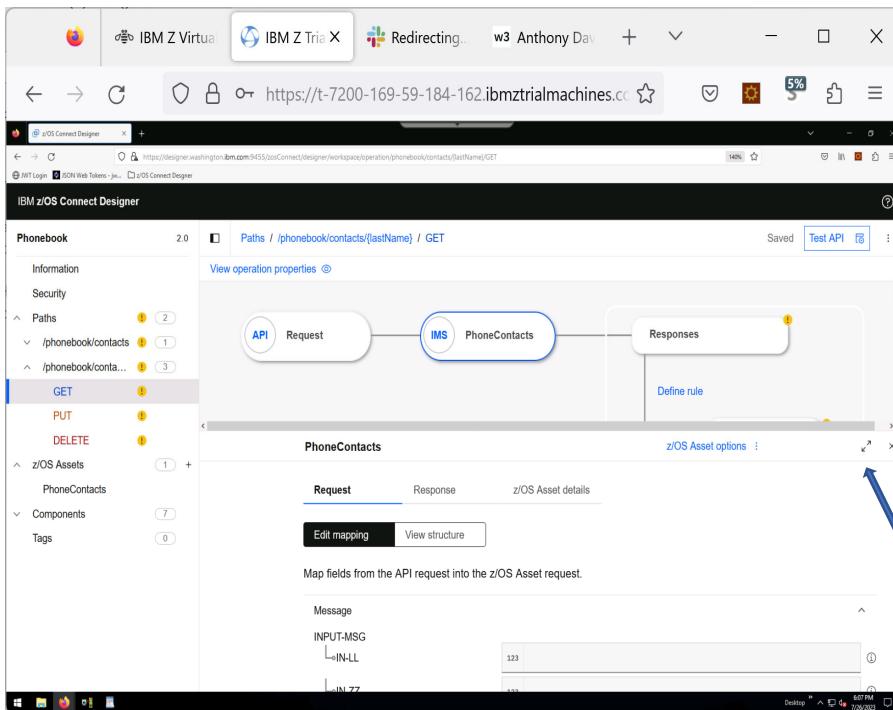


- Click **Add z/OS Asset**

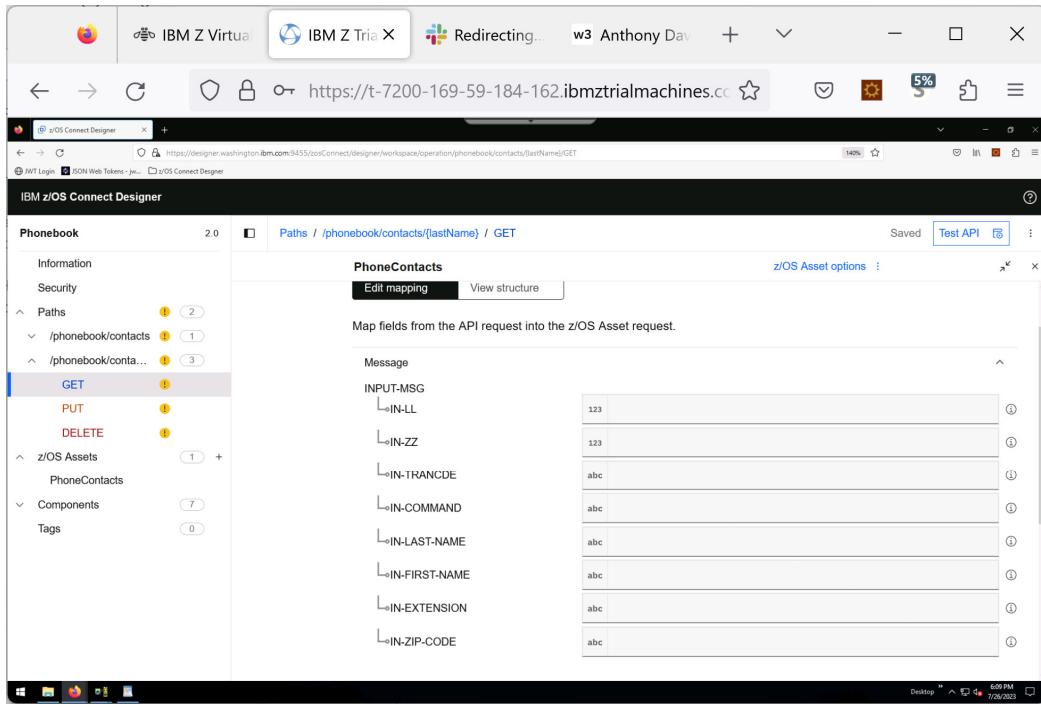
Note that the asset is now listed in the Operations flow diagram.



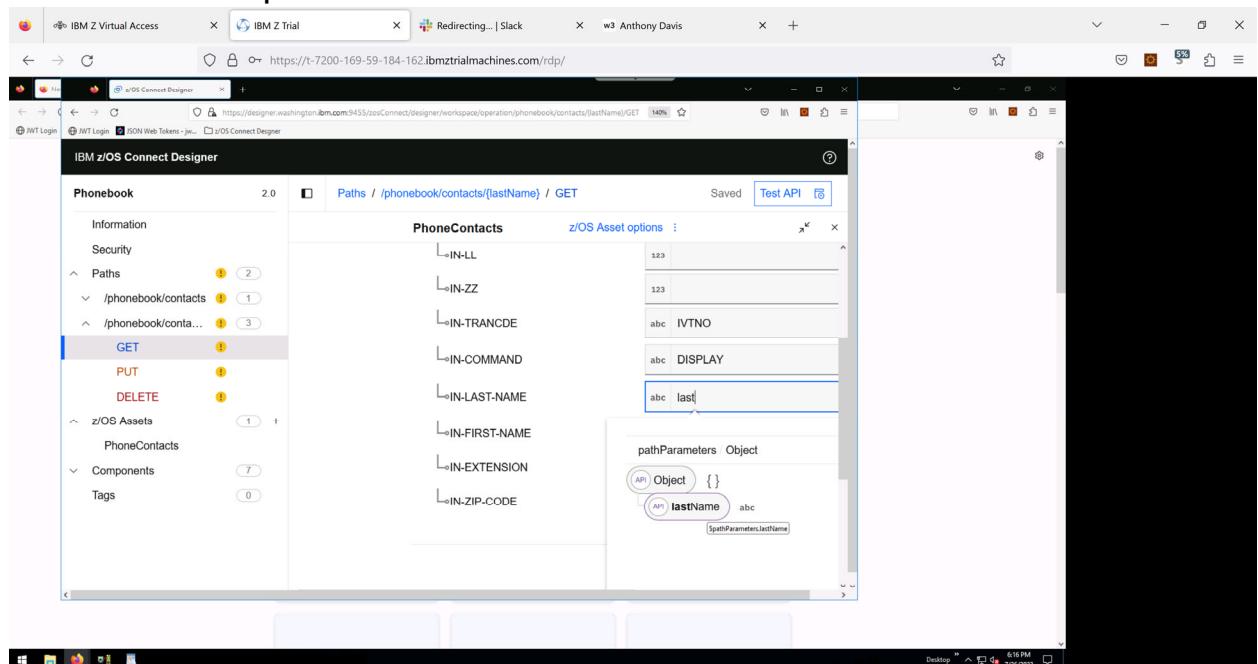
- Click the asset you created e.g., “PhoneContacts”

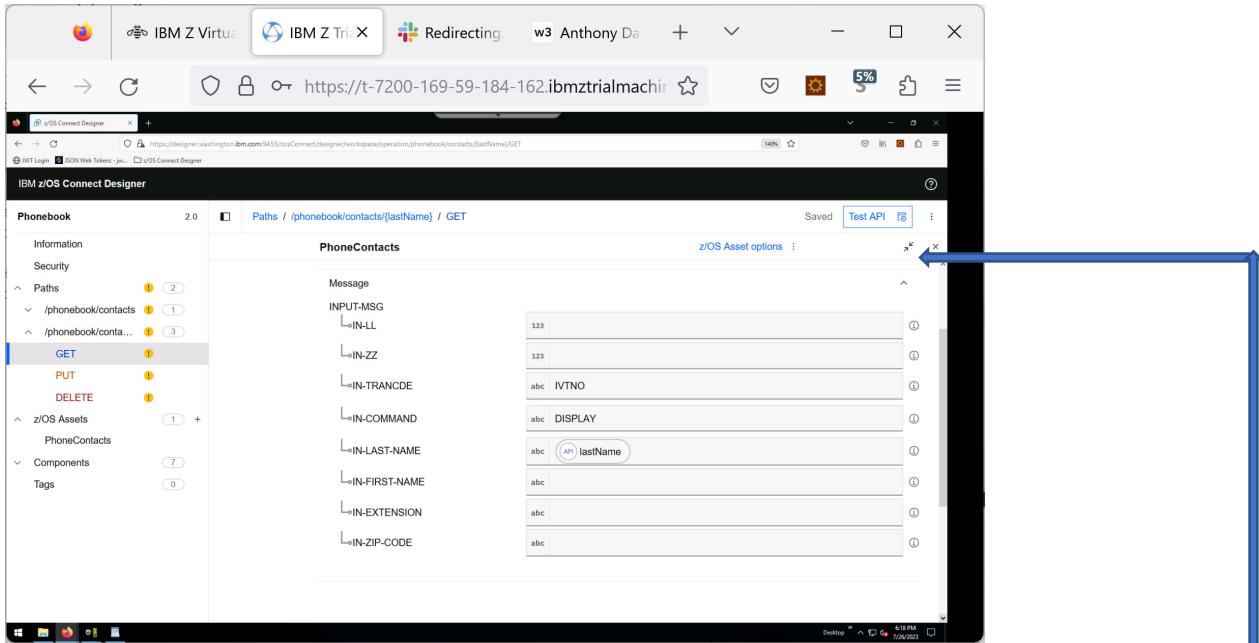


- Click on **Edit Mapping** under **Request**
- Expand the bottom pane by clicking on the two arrows on the top right



- 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.*

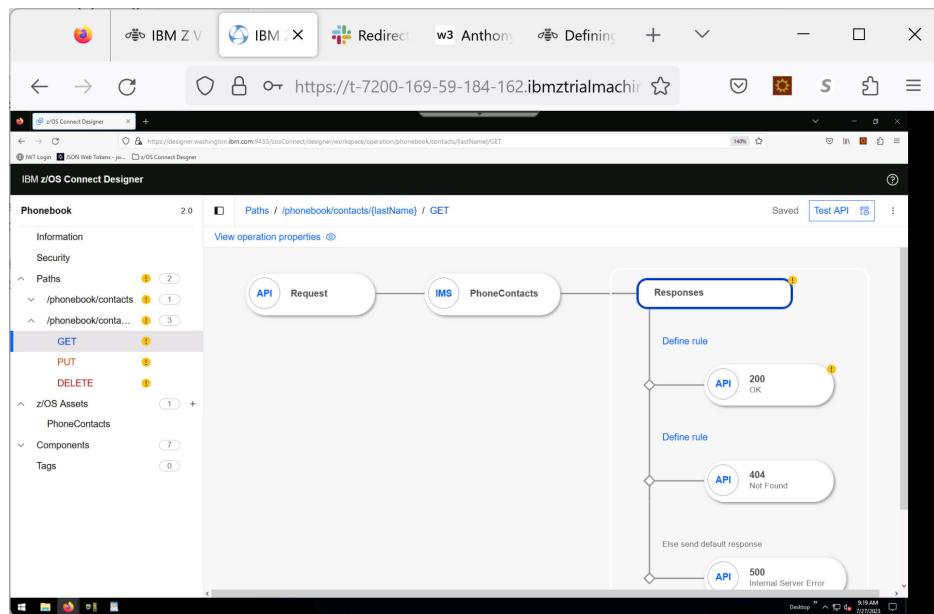
- You can now minimize this panel by clicking on the arrows on the top right. This will bring you back to the main operations flow diagram.

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.

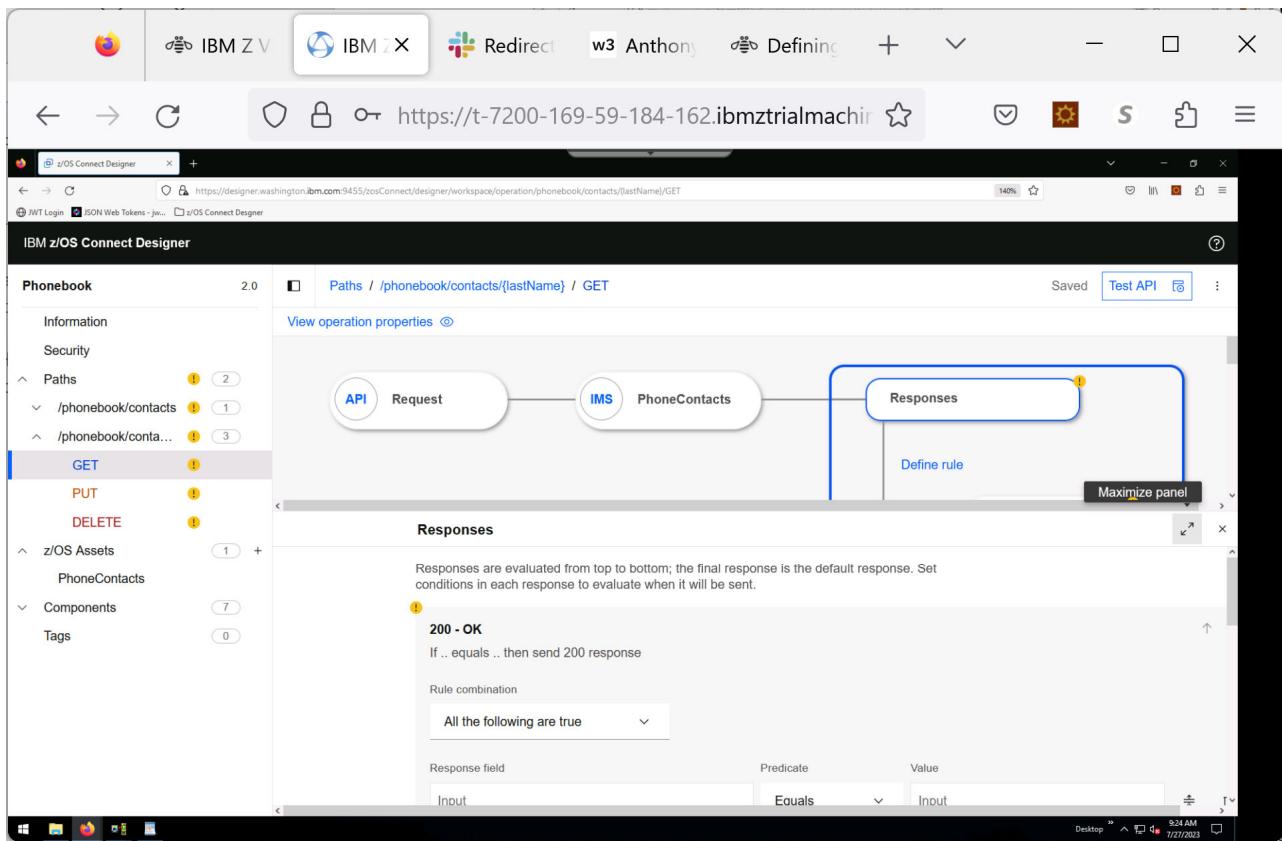
Each response has the following properties:

- 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.

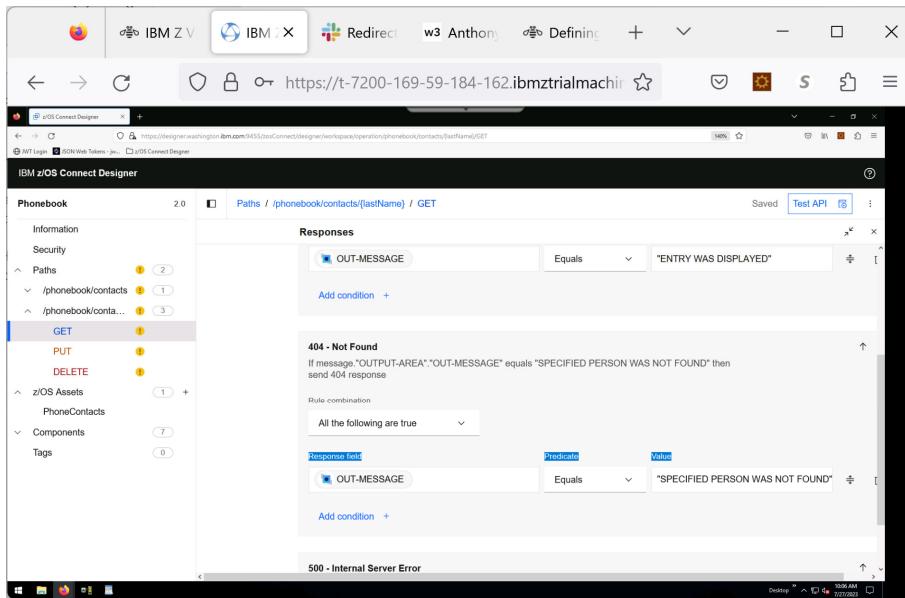
- Also key in “**ENTRY WAS DISPLAYED**” (note the double quotes) in the corresponding Value field.

The screenshot shows the IBM z/OS Connect Designer interface. On the left, there's a sidebar with sections like Information, Security, Paths, Components, and Tags. The main area is titled "Phonebook" and shows a "Paths / /phonebook/contacts/{lastName} / GET" entry. Under "Responses", it says "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." It lists a "200 - OK" response with a condition: "If message."OUTPUT-AREA".OUT-MESSAGE" equals "ENTRY WAS DISPLAYED" then send 200 response. Below that is a "404 - Not Found" response with a condition: "If .. equals .. then send 404 response". A yellow exclamation mark icon is positioned next to the 200-OK entry.

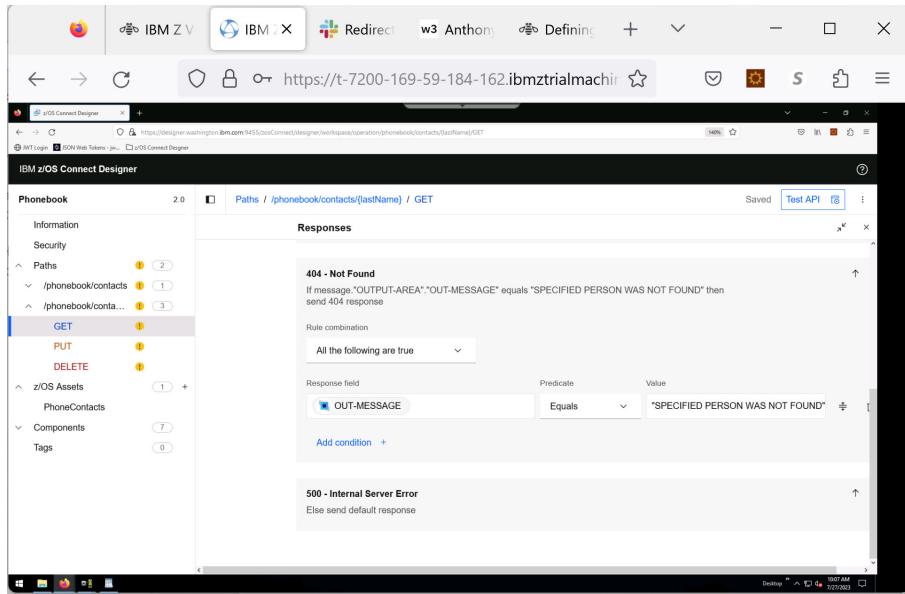
If done correctly, the ! mark associate with the 200- OK should disappear.

This screenshot shows the same interface as the previous one, but the yellow exclamation mark icon that was previously next to the 200-OK entry in the "Responses" section has disappeared, indicating that the configuration has been successfully saved or updated.

- 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).

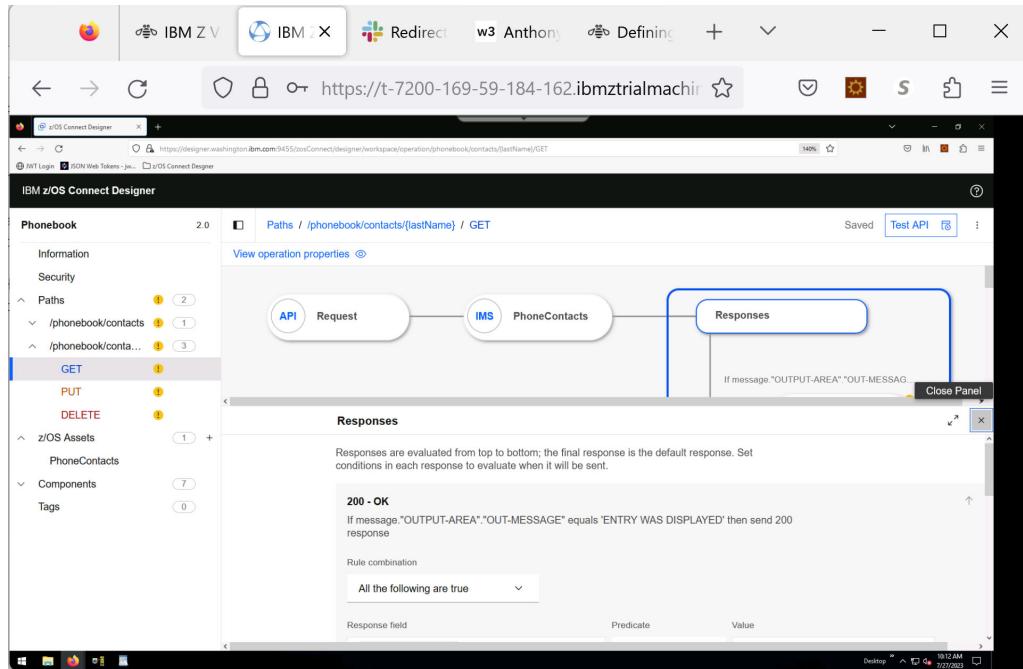


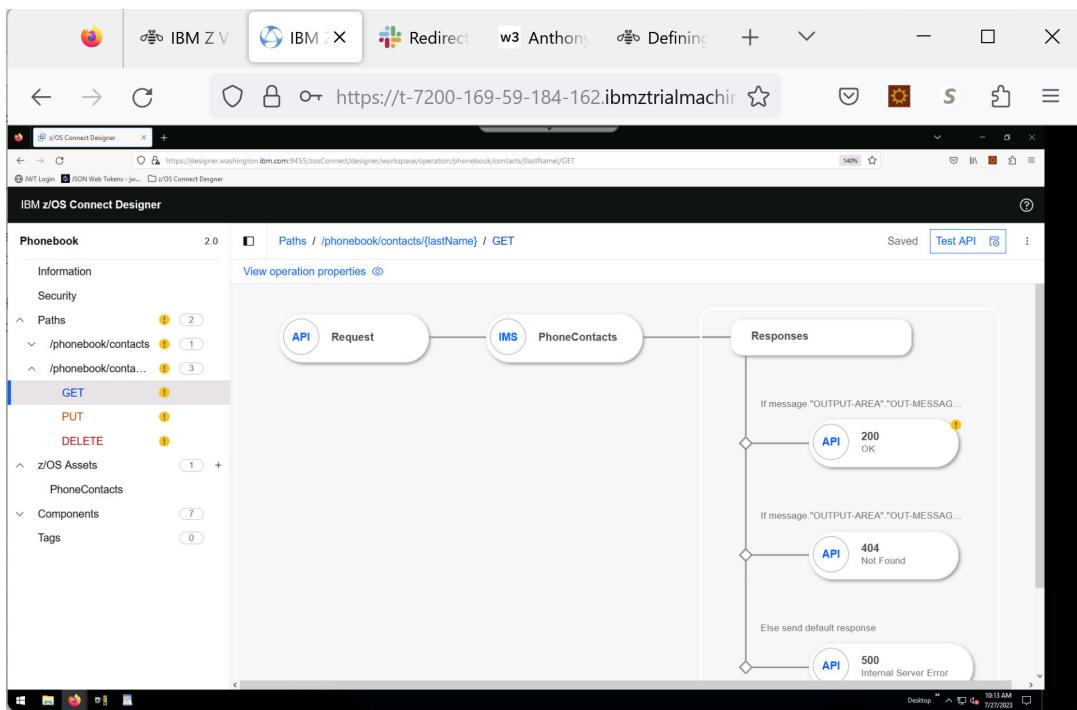
The 500 -Internal server error response is the default, so it has no conditions and must be the last entry in the table.



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.

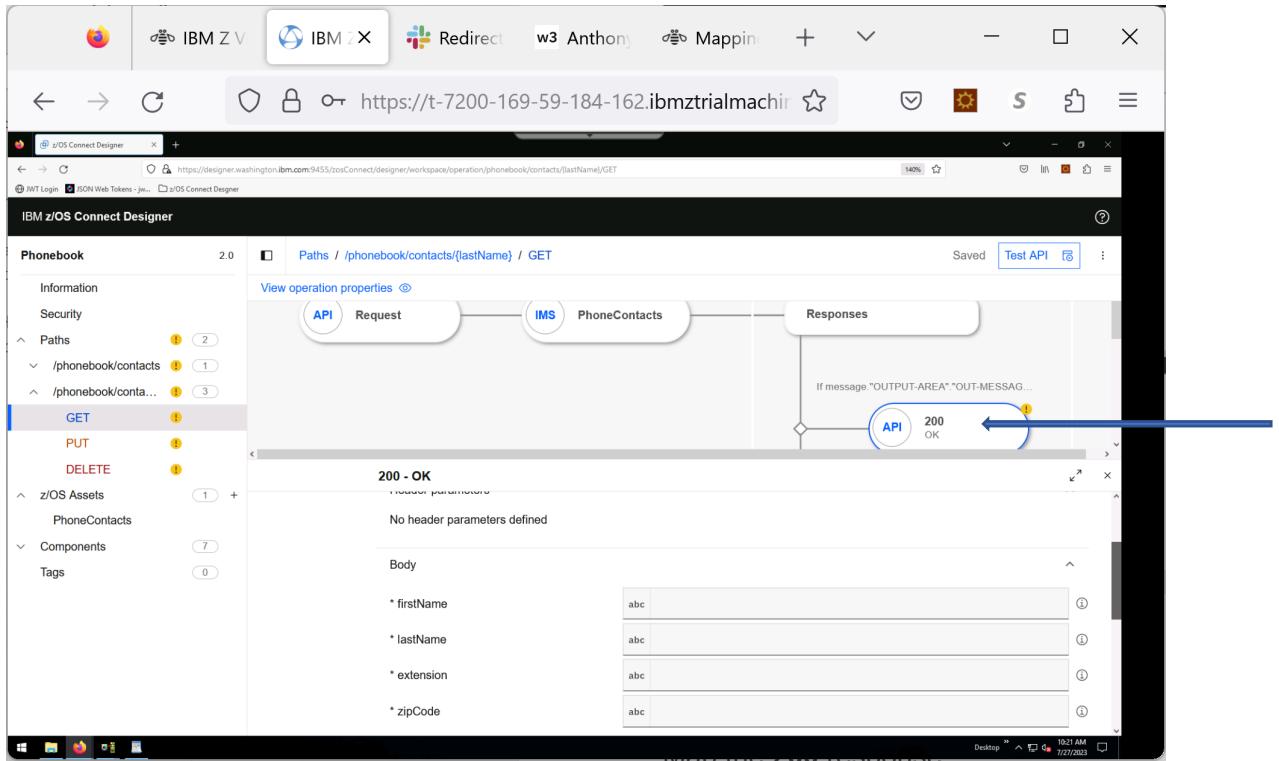




Map the 200 response.

- 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 sidebar with a tree view of paths and operations. The main panel shows a mapping configuration for a GET operation. It has sections for Header parameters (empty), Body (with fields: * firstName, * lastName, * extension, * zipCode), and a 'Test API' button.

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

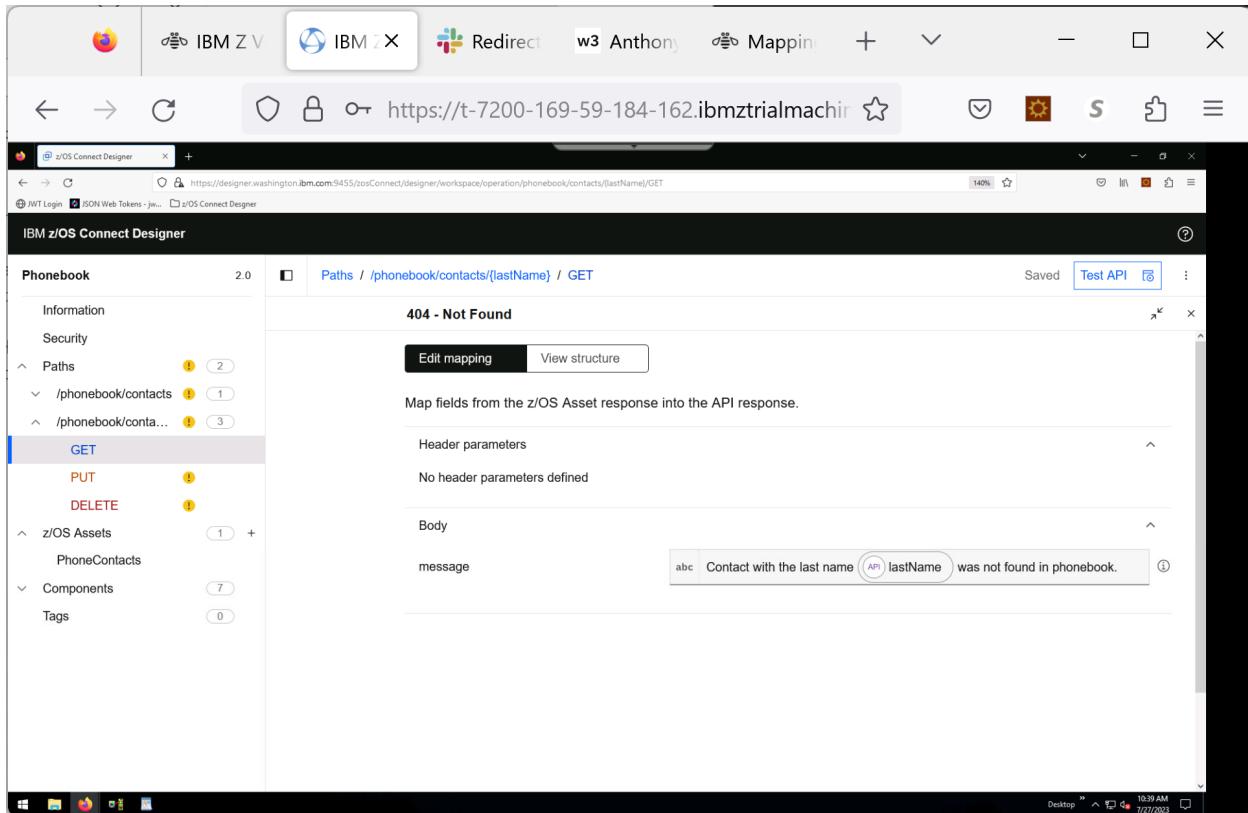
Map the 404 response.

The screenshot shows the operation flow diagram for the GET /phonebook/contacts/{lastName} path. The flow starts with an API node labeled '200 OK'. From this node, a blue arrow points down to a decision diamond. The condition for this diamond is 'If message.OUTPUT-AREA.OUTPUT-MESSAGE...'. If the condition is false, the flow continues to another API node labeled '404 Not Found'. If the condition is true, the flow continues to a third API node labeled '500 Internal Server Error'. A blue arrow also points from the '404 Not Found' node back up towards the decision diamond.

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.

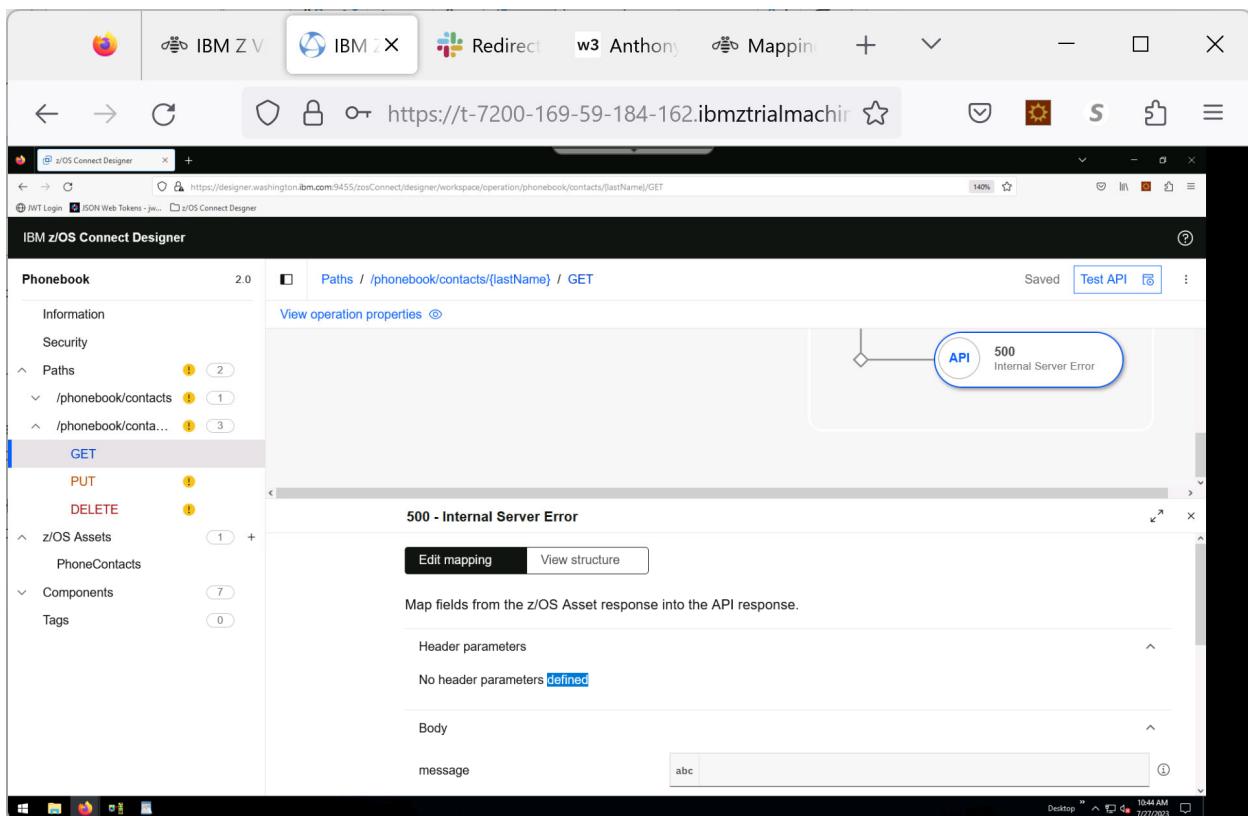


Note how the tooling pulls the lastName from the structure.

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

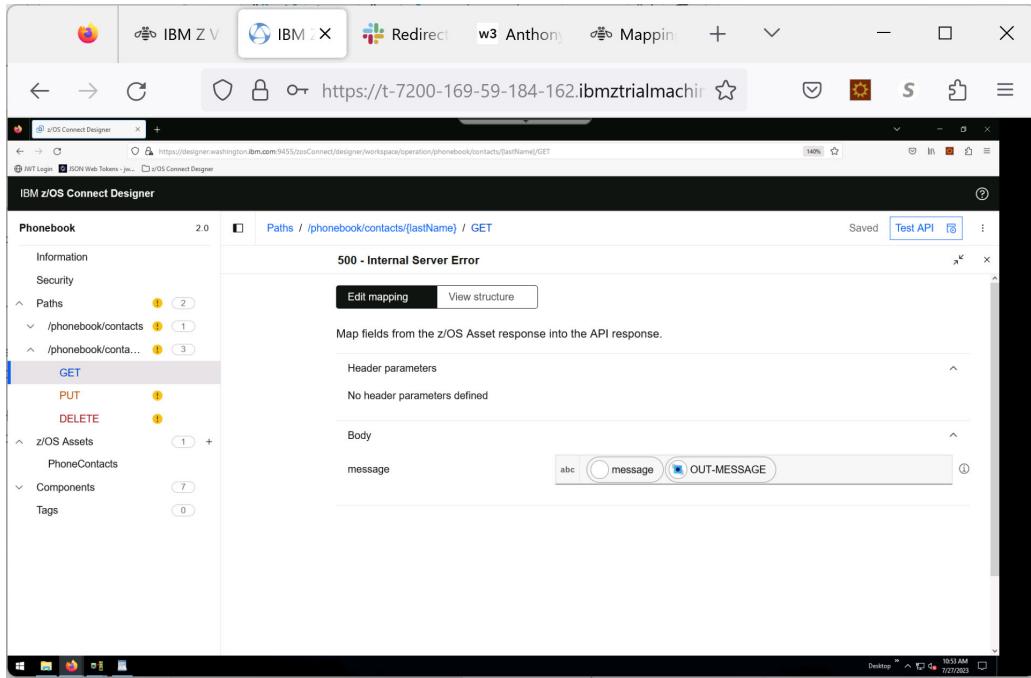
Map the 500 response.

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



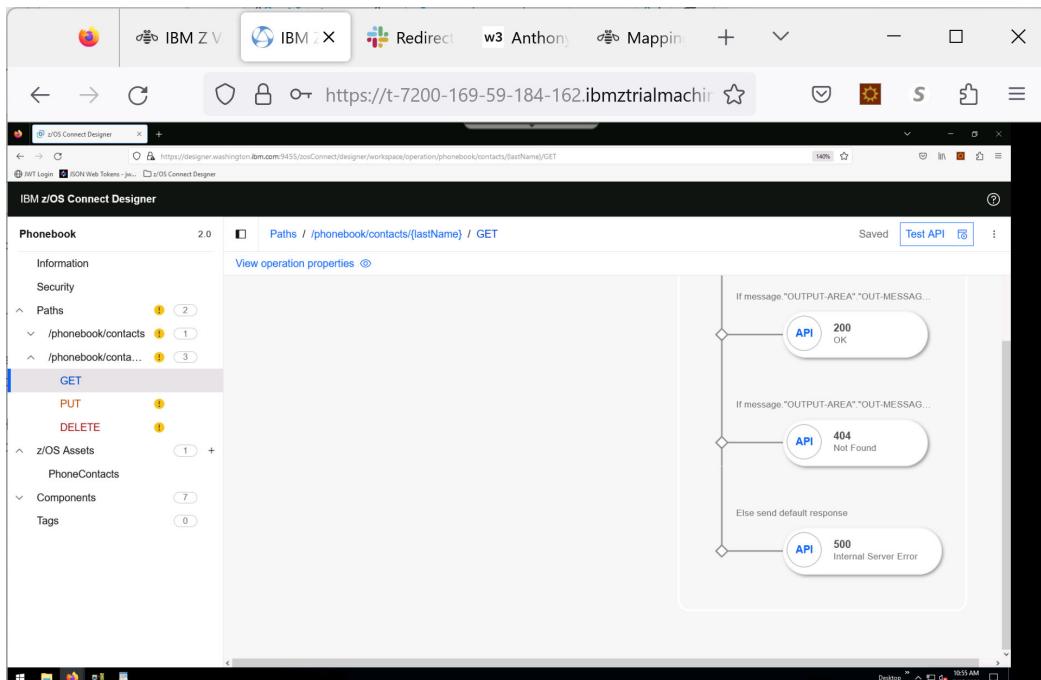
A 500 response code indicates an internal server error. Configure the 500 response to return the z/OS Connect error message by typing the following into the message field (Note the case, brackets, and quotes):

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



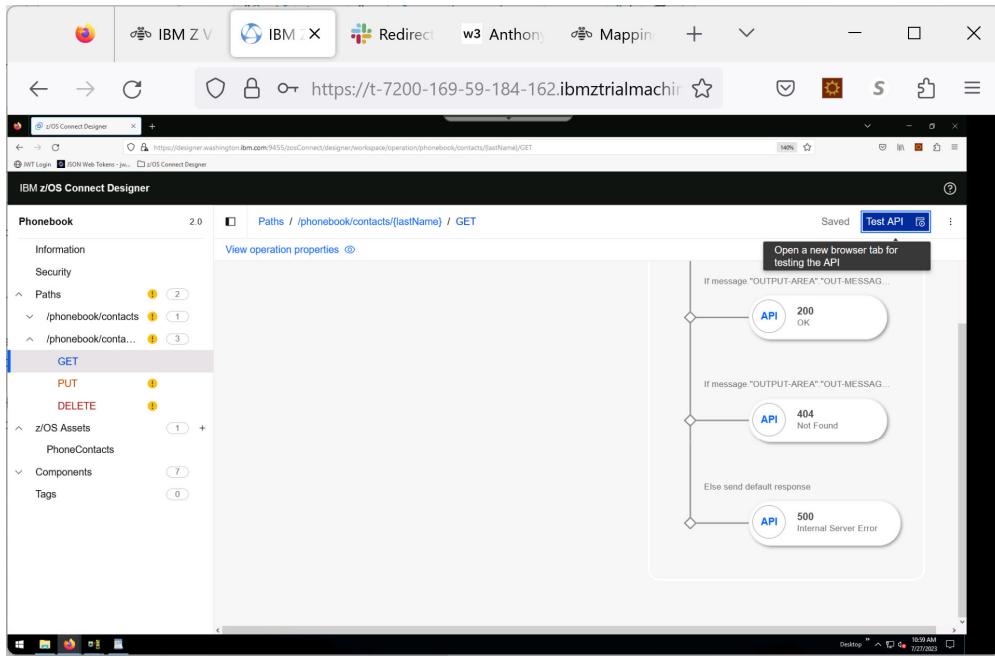
- Click **X** at the top right of this panel to close it and return to the Operations flow diagram.

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 <https://designer.washington.ibm.com:9455>

Liberty REST APIs 1.0.0 OAS3

Discover REST APIs available within Liberty

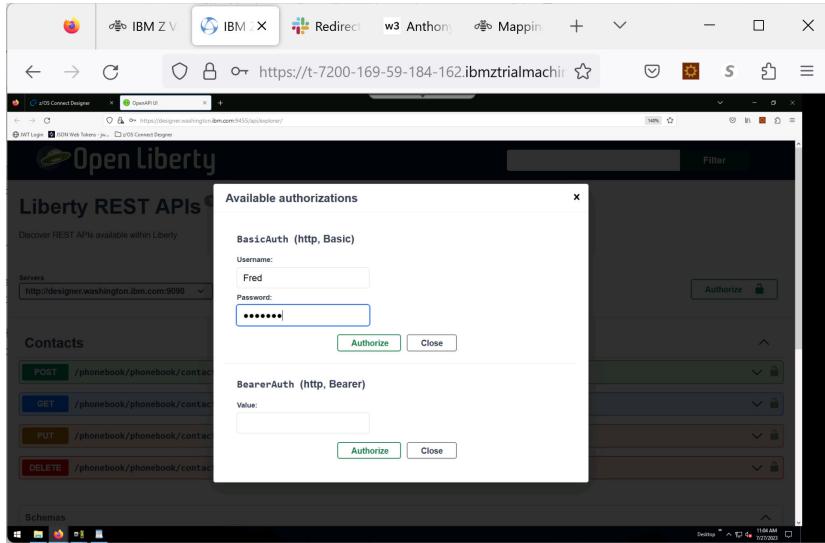
Servers

- http://designer.washington.ibm.com:9090
- http://designer.washington.ibm.com:9090
- https://designer.washington.ibm.com:9455

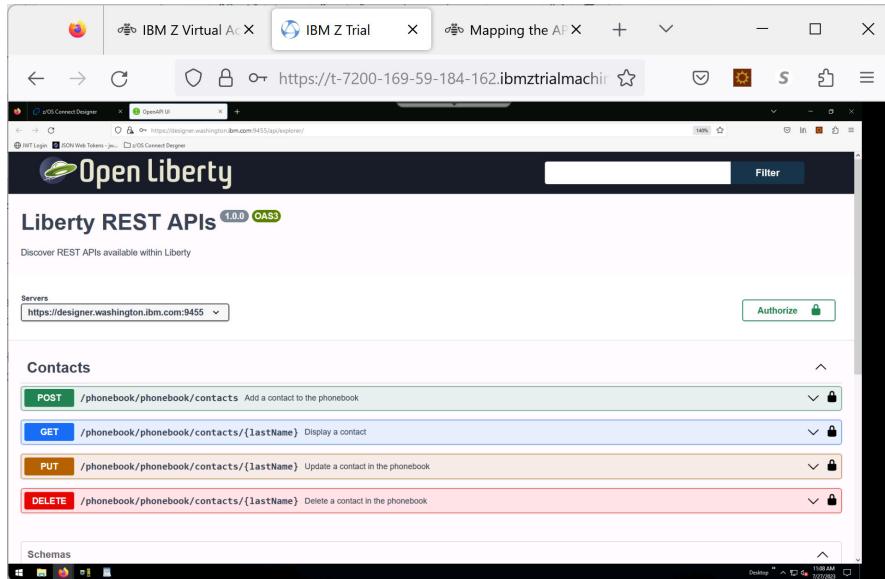
Contacts

- POST /phonebook/phonebook/contacts Add a contact to the phonebook
- GET /phonebook/phonebook/contacts/{lastName} Display a contact
- PUT /phonebook/phonebook/contacts/{lastName} Update a contact in the phonebook
- DELETE /phonebook/phonebook/contacts/{lastName} Delete a contact in the phonebook

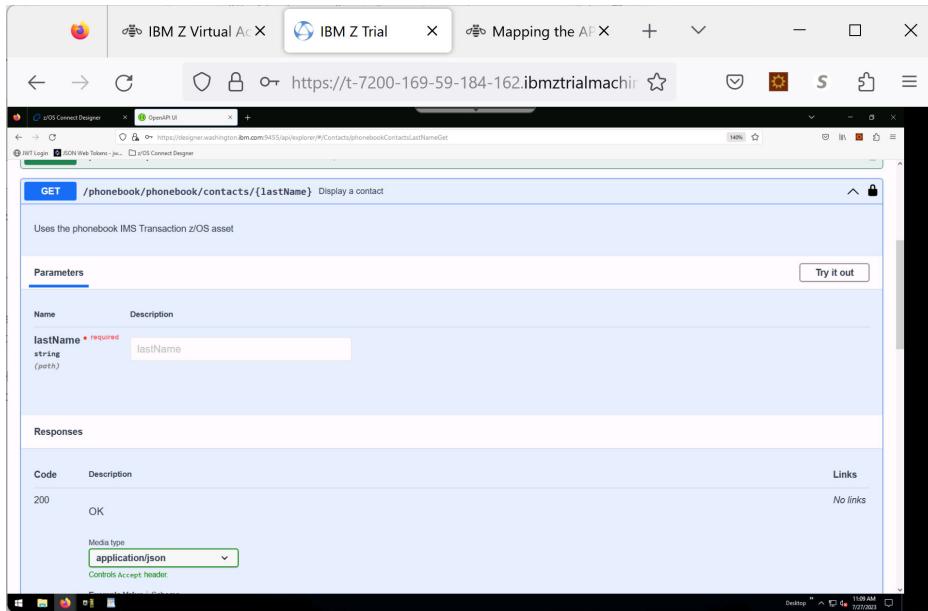
- Click the **Authorize** button
 - Key in **Fred** (note the capital F) for the Username
 - Key in **fredpwd** (Note all lowercase) for the password
- Click **Authorize.**



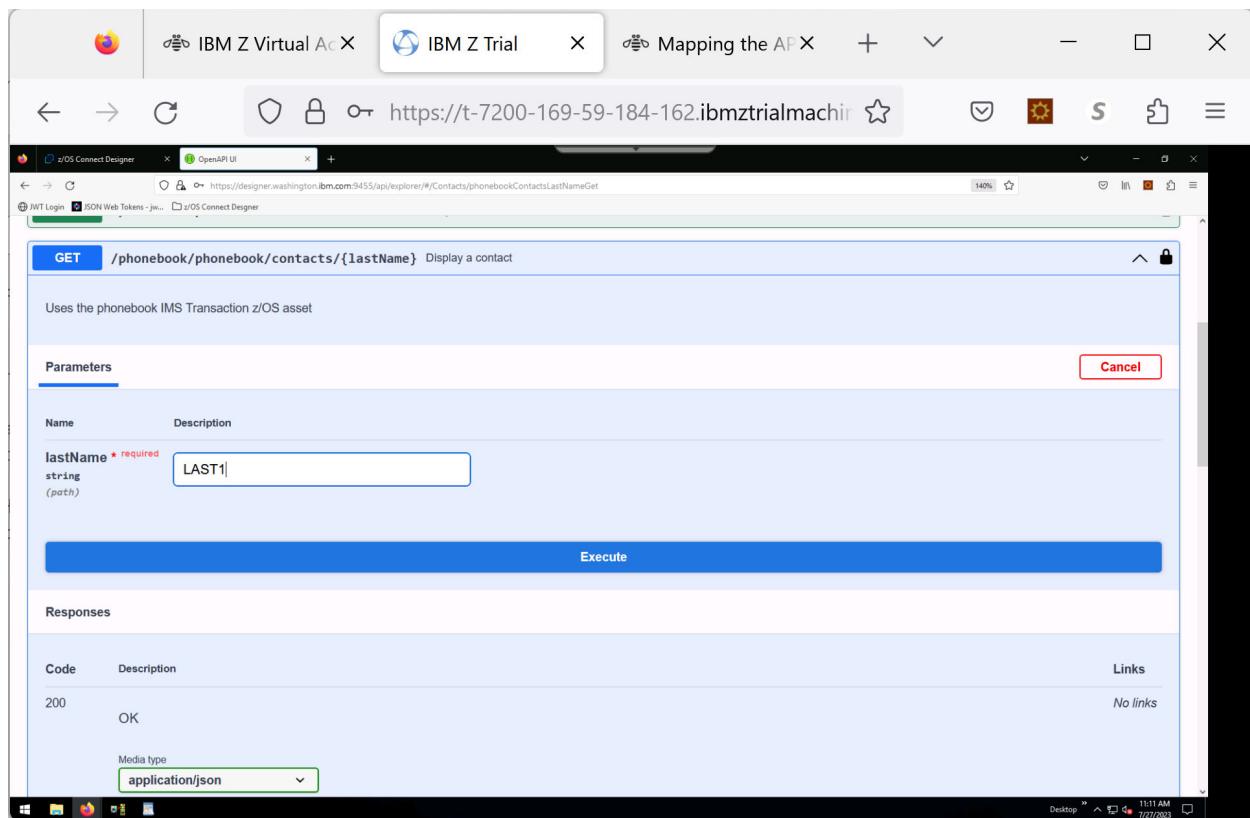
Back on the main panel, you will see all the possible methods that can be used for this API. The only one configured at this point is the **GET** method so it is the only one that can be tested.



- Click **GET**



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



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 RnJlZDpmcmVkcHdk'
```

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"
}
```

Download

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

- Try another one. Click on the **Clear** button.

Parameters

Name Description

lastName * required string (path)

LAST2

Execute Clear

Responses

Curl

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

Request URL

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

Server response

- Enter MOUSE as the last name.

The screenshot shows the z/OS Connect Designer OpenAPI UI in a Firefox browser window. The URL is <https://t-7200-169-59-184-162.ibmztrialmachir>. The page displays the 'Contacts' section with the following details:

- POST /phonebook/phonebook/contacts**: Add a contact to the phonebook.
- GET /phonebook/phonebook/contacts/{lastName}**: Display a contact. This is the selected endpoint.

Under Parameters, the 'lastName' parameter is defined as a required string path parameter with the value 'MOUSE' entered. The 'Execute' button is visible at the bottom of the form.

The screenshot shows the execution results for the GET /phonebook/phonebook/contacts/{lastName} endpoint with lastName set to 'MOUSE'. The results are displayed in the following sections:

- Curl**: A terminal command to execute the API call.
- Request URL**: The full URL for the API call: <https://designer.washington.ibm.com:9455/phonebook/phonebook/contacts/MOUSE>.
- Server response**:
 - Code**: 200
 - Details**: Response body and Response headers.

The Response body contains the following JSON data:

```
{
  "firstName": "WINNIE",
  "lastName": "MOUSE",
  "extension": "8-234-4567",
  "zipCode": "33333"
}
```

The Response headers include:

```
content-encoding: gzip
content-language: en-US
content-length: 37
content-type: application/json
```

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

The screenshot shows a browser window with three tabs:

- IBM Z Virtual Access
- IBM Z Trial (active tab)
- Mapping the API resp

The main content area displays an API request in the z/OS Connect Designer interface. The URL is <https://designer.washington.ibm.com:9455/api/explorer/#/Contacts/phonebookContactsLastNameGet>. A parameter `lastName` is set to `ZZZZZ`.

Below the request, the "Responses" section shows the following details:

- Curl:**

```
curl -X 'GET' \
'https://designer.washington.ibm.com:9455/phonebook/phonebook/contacts/ZZZZZ' \
-H 'accept: application/json' \
-H 'Authorization: Basic RnJ1ZDpmcmVkcHdk'
```
- Request URL:** <https://designer.washington.ibm.com:9455/phonebook/phonebook/contacts/ZZZZZ>
- Server response:**

| Code | Details |
|------|------------------|
| 404 | Error: Not Found |

Response body:

```
{
  "message": "Contact with the last name ZZZZZ was not found in phonebook."
}
```

Download button is present.

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: "IBM Z Virtual Access", "IBM Z Trial", and "Mapping the API resp". The main content area displays the "z/OS Connect Designer" interface. A search bar at the top has the URL <https://t-7200-169-59-184-162.ibmztrialmachines.com/rdp/>. Below the URL, it says "https://designer.washington.ibm.com:9455/api/explorer/#/Contacts/phonebookContactsLastNameGet". A parameter "lastName" is set to "LAST1". Below the parameter input is a "Responses" section. Under "Responses", there is a "Curl" block containing:

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

There is also a "Request URL" block with the URL <https://designer.washington.ibm.com:9455/phonebook/phonebook/contacts/LAST1>. The "Server response" section shows a "Code" of 500 and a "Details" of "Error: Internal Server Error". The "Response body" is a JSON object:

```
{
  "message": "BAQR1062E: Internal IMS z/OS Asset error."
}
```

A "Download" button is available for the response body. At the bottom of the browser window, the taskbar shows icons for various applications, and the system tray indicates the date and time as 7/27/2023.

Congratulations! You have completed the exercise for GET method.