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

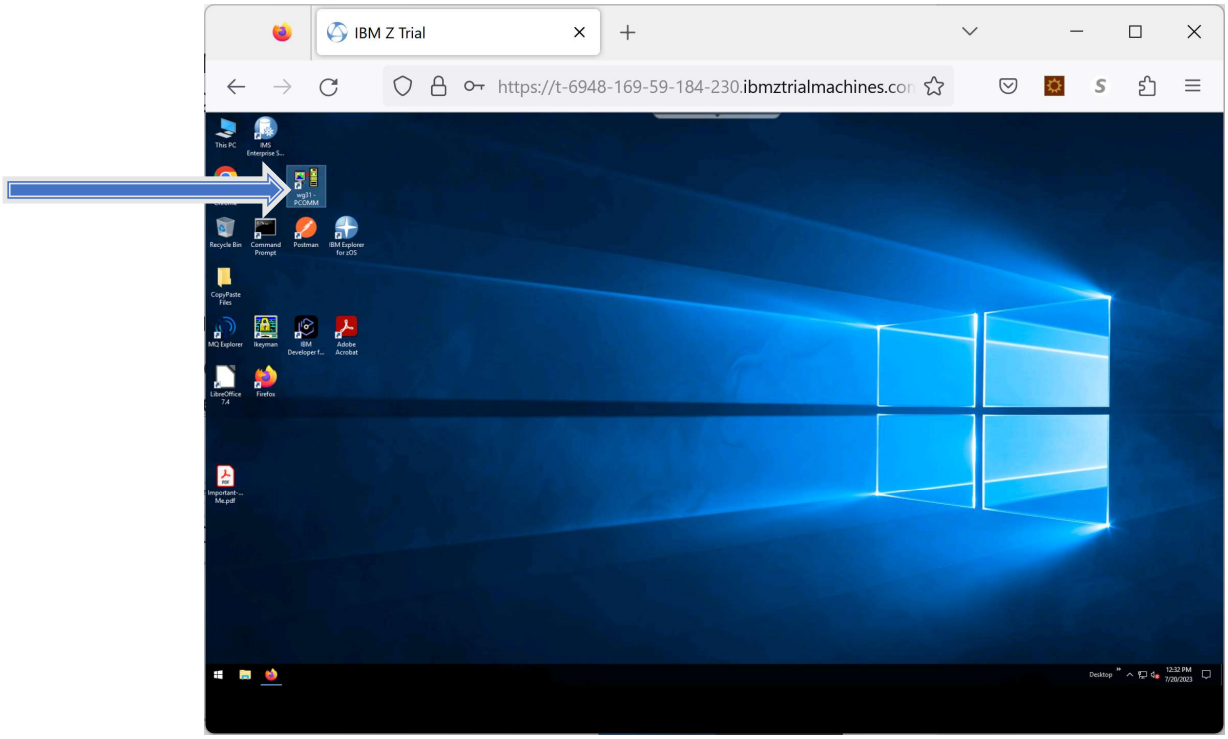
PART 3 – Create the API to UPDATE the contact you added

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.

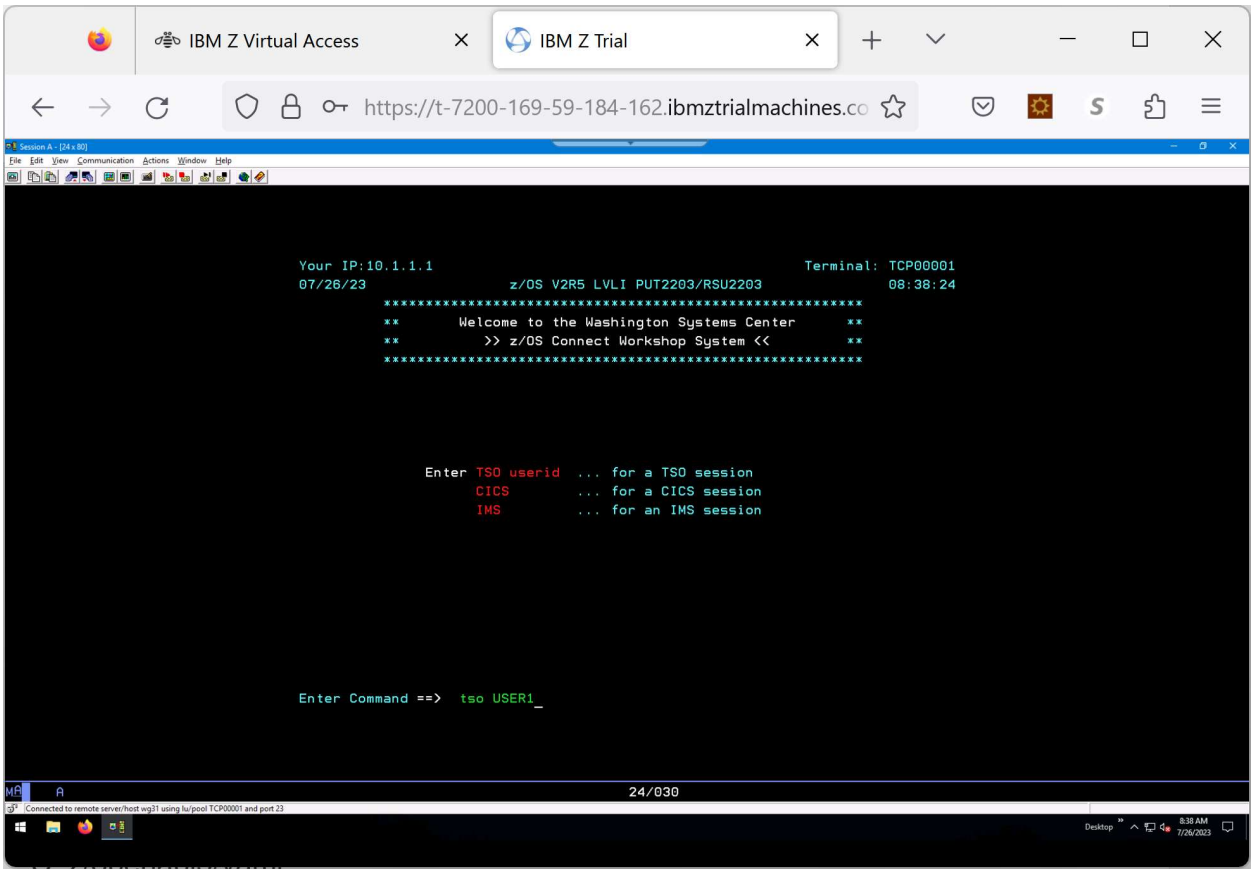
REVIEW the IMS Environment

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

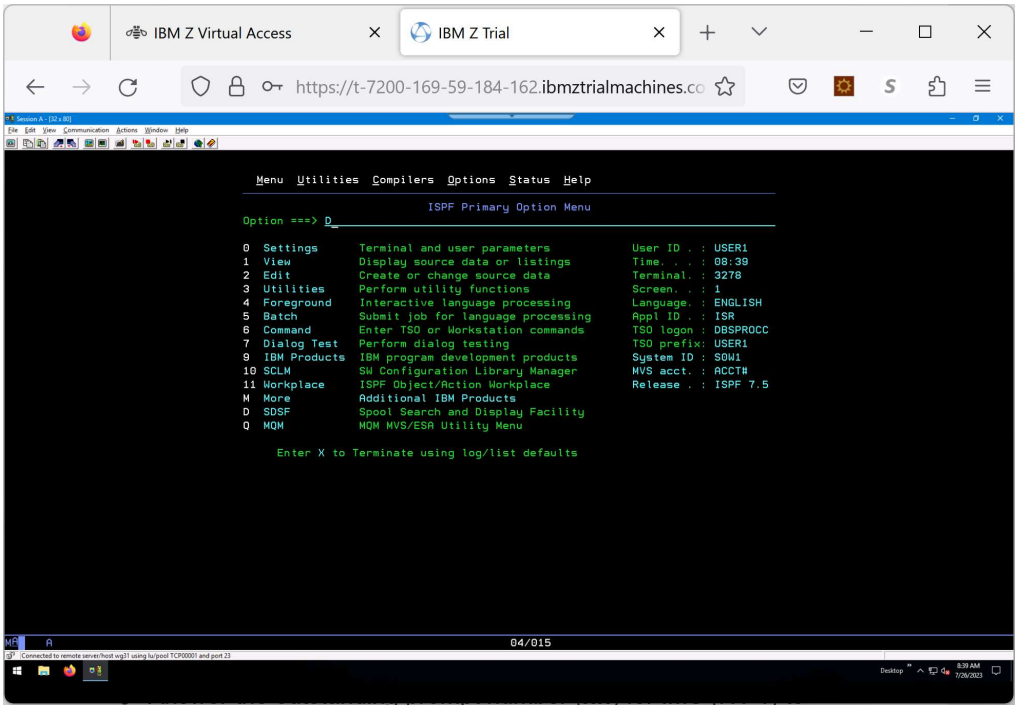


(note that shift- Enter is used as the enter function)

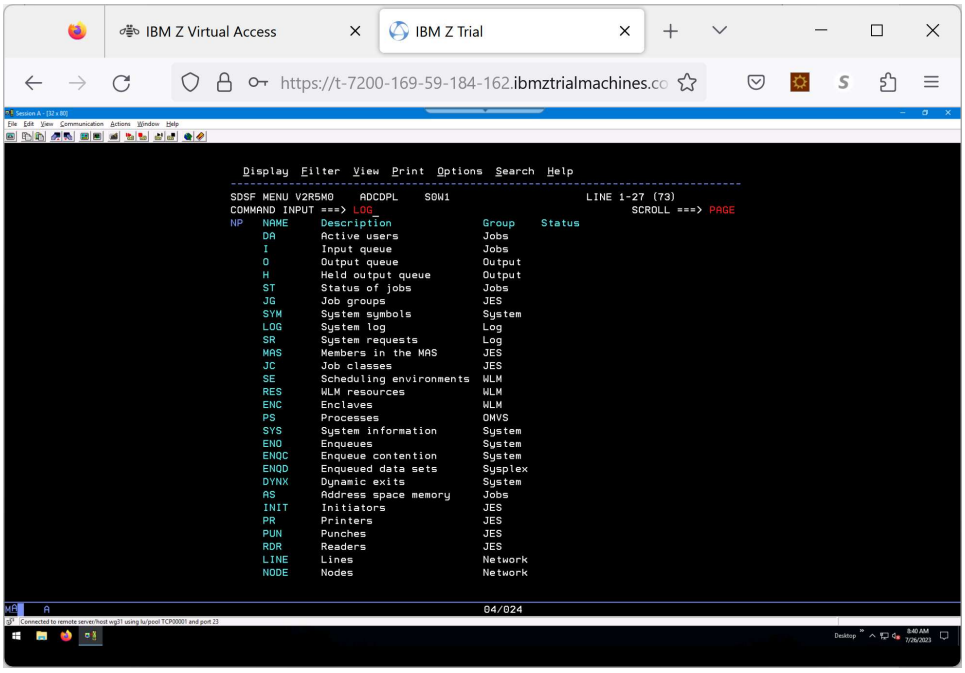
- Logon to TSO using USER1
  - Password is user1



- ISPF option D allows access to SDSF

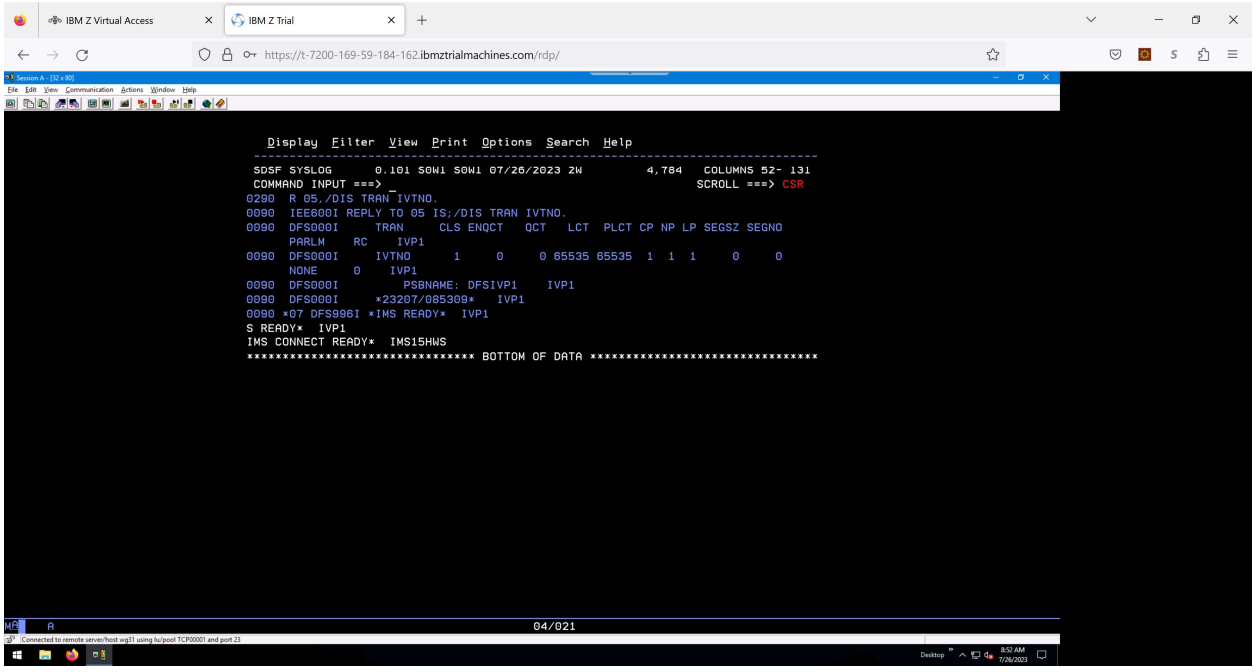


- Go to LOG

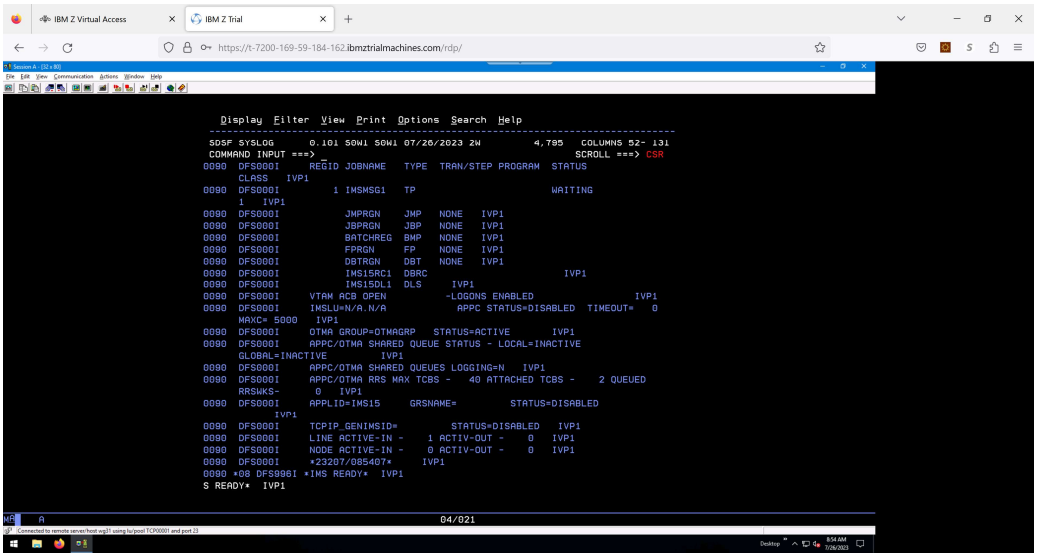


You will see the outstanding prompts for IMS (**IVP1**) and IMSConnect **IMS15HWS**)

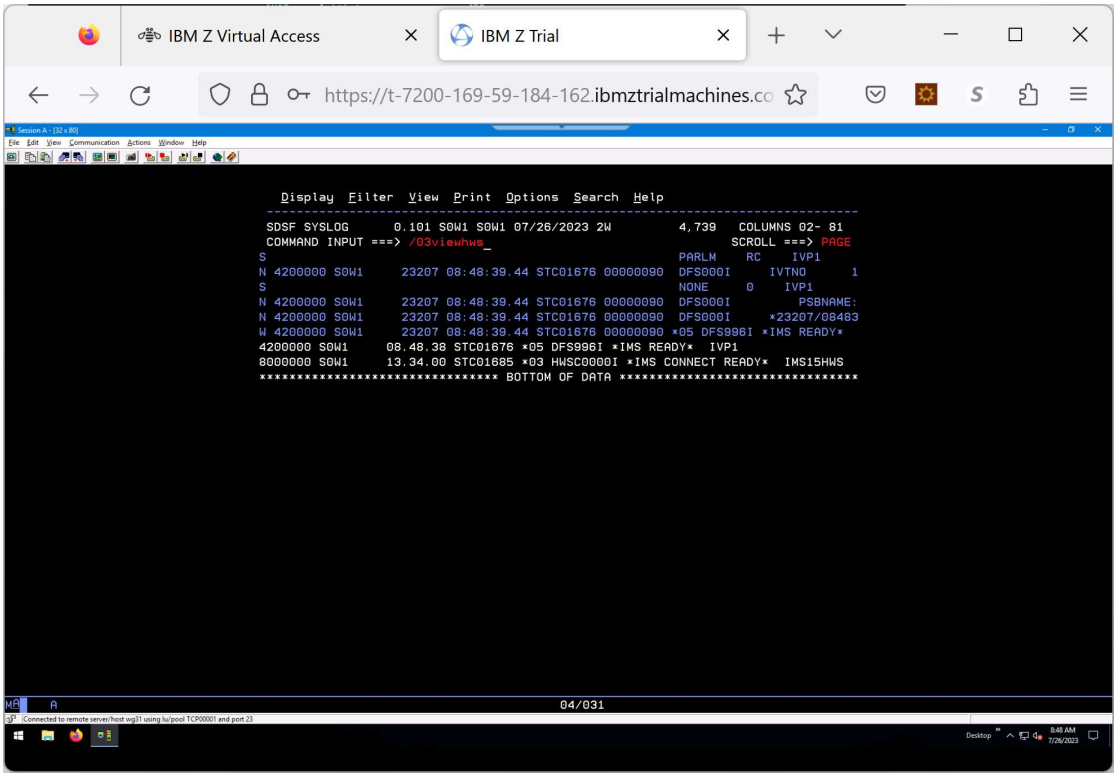
- Answer the outstanding prompt number (***nn***) for IMS (IVP1) to check on message region availability and the transaction ENQCT:
  - */nn*/**DIS TRAN IVTNO**.
- Make note of the value for ENQCT which will keep track of the number accesses of the IVTNO transaction

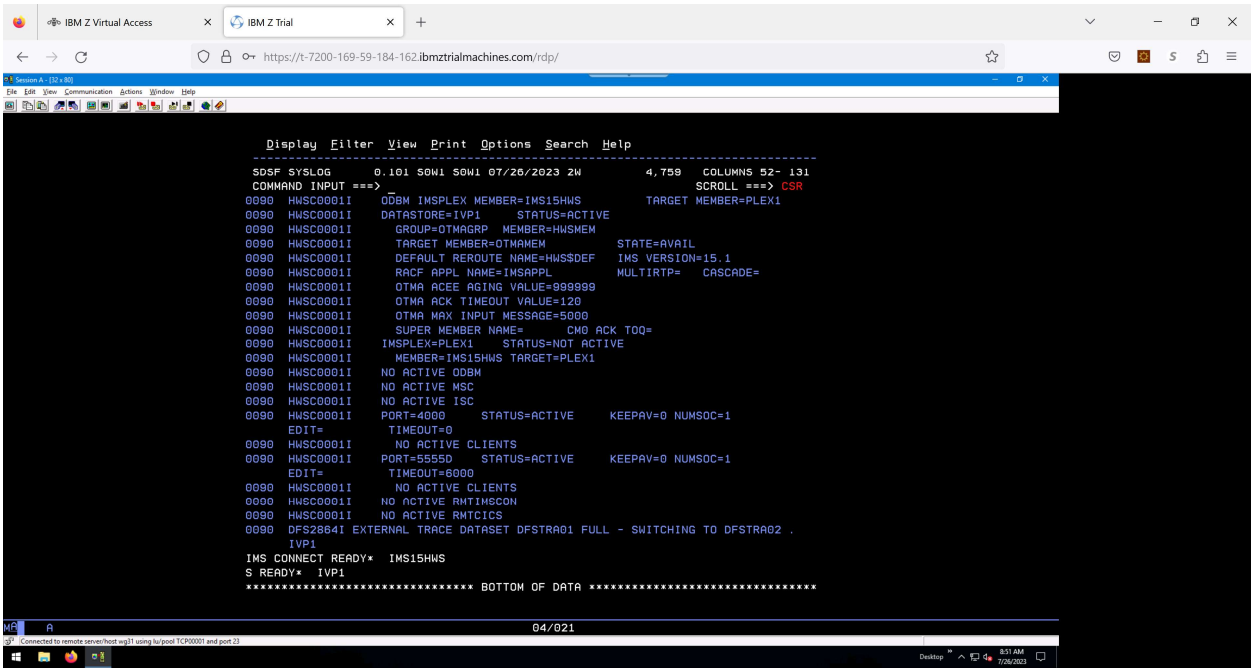


- You can also issue a display active command to verify that the dependent regions are activing
    - */nn*/**DIS A.**



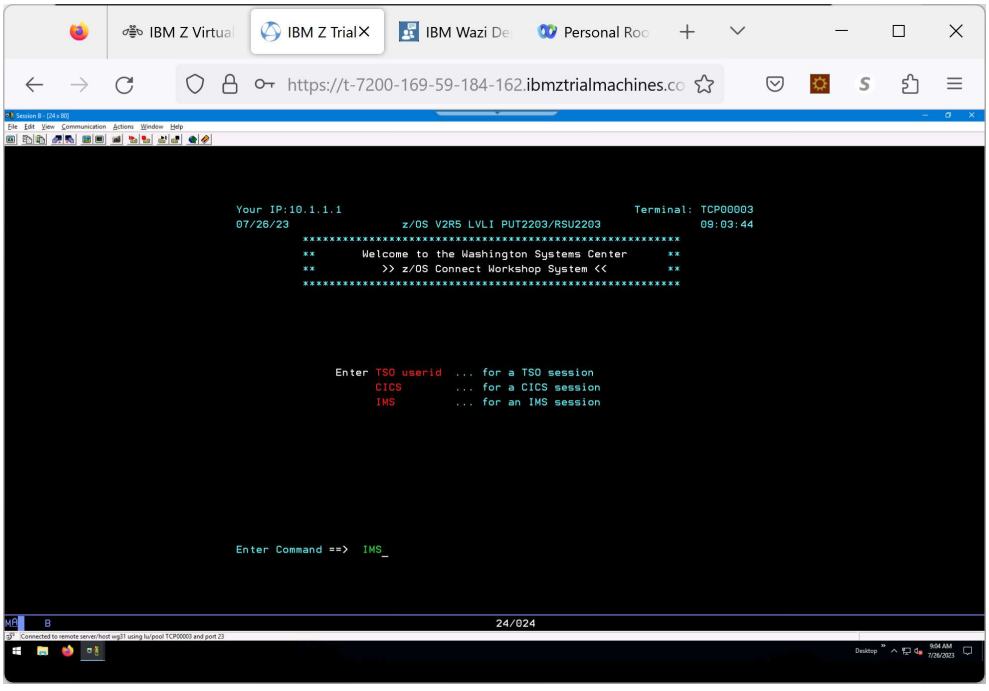
- Answer the outstanding prompt number (***zz***) for IMS Connect (IMS15HWS)
    - Respond with /*zz*/**VIEWHWS**
        - Make sure Datastore=IVP1 status is active





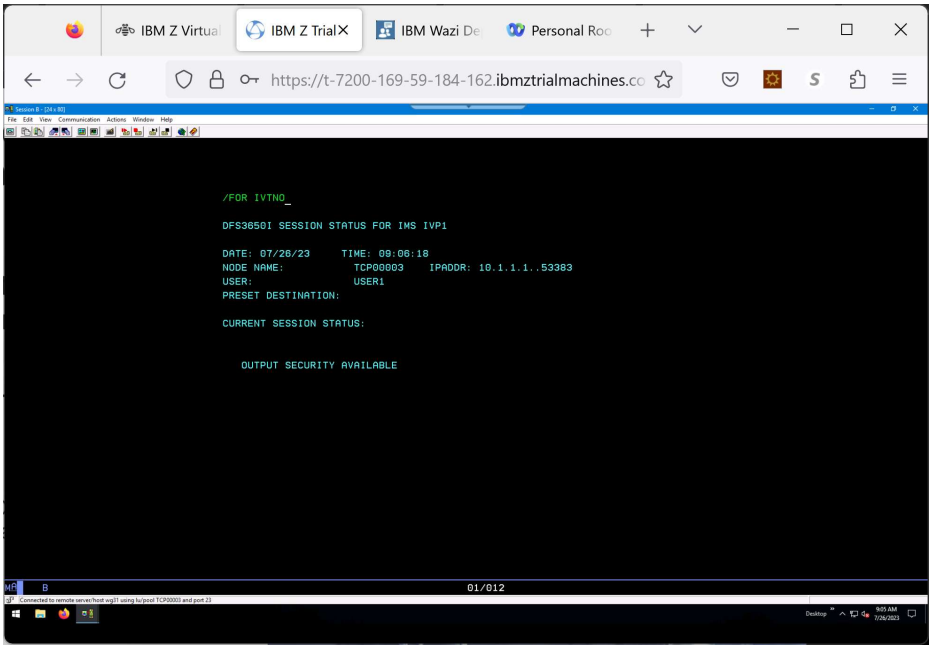
- If PORT=4000 shows an inactive status, issue the command */zz***STARTPT 4000**

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

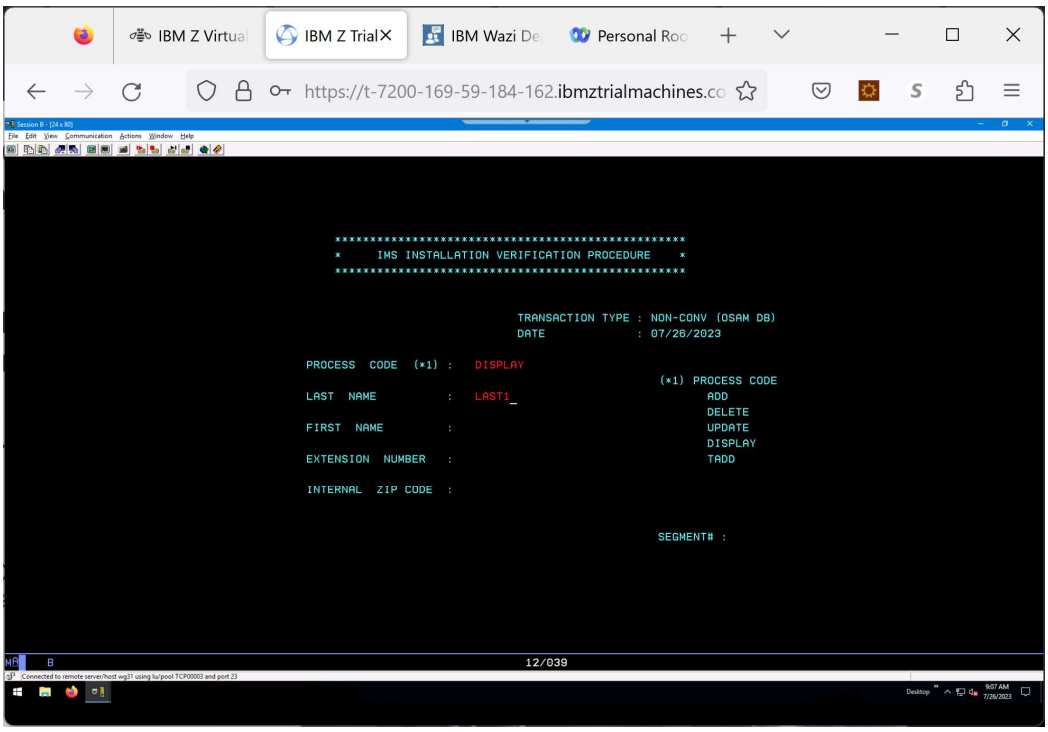


Userid and password are USER1/user1

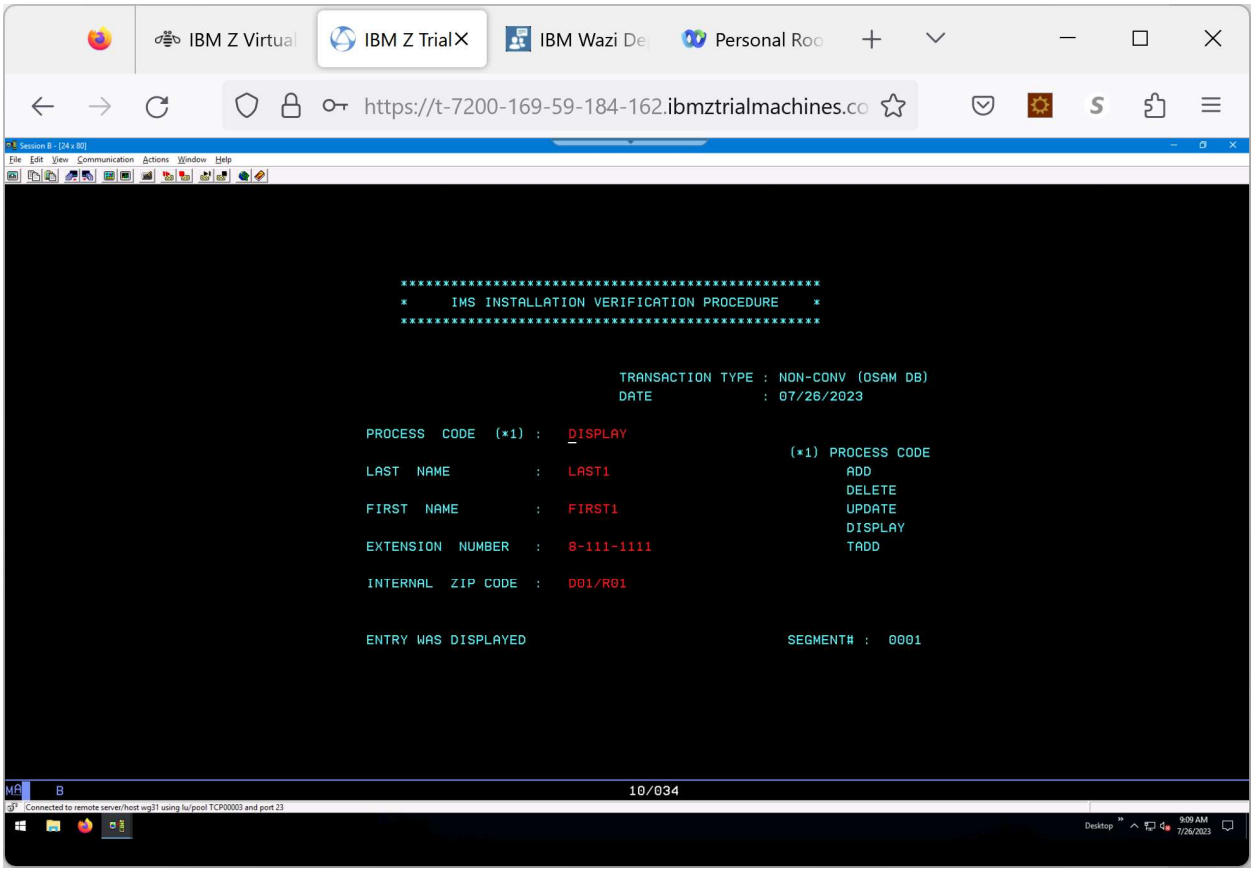
Once you are logged on, use format IVTNO (**/FOR IVTNO)**



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)



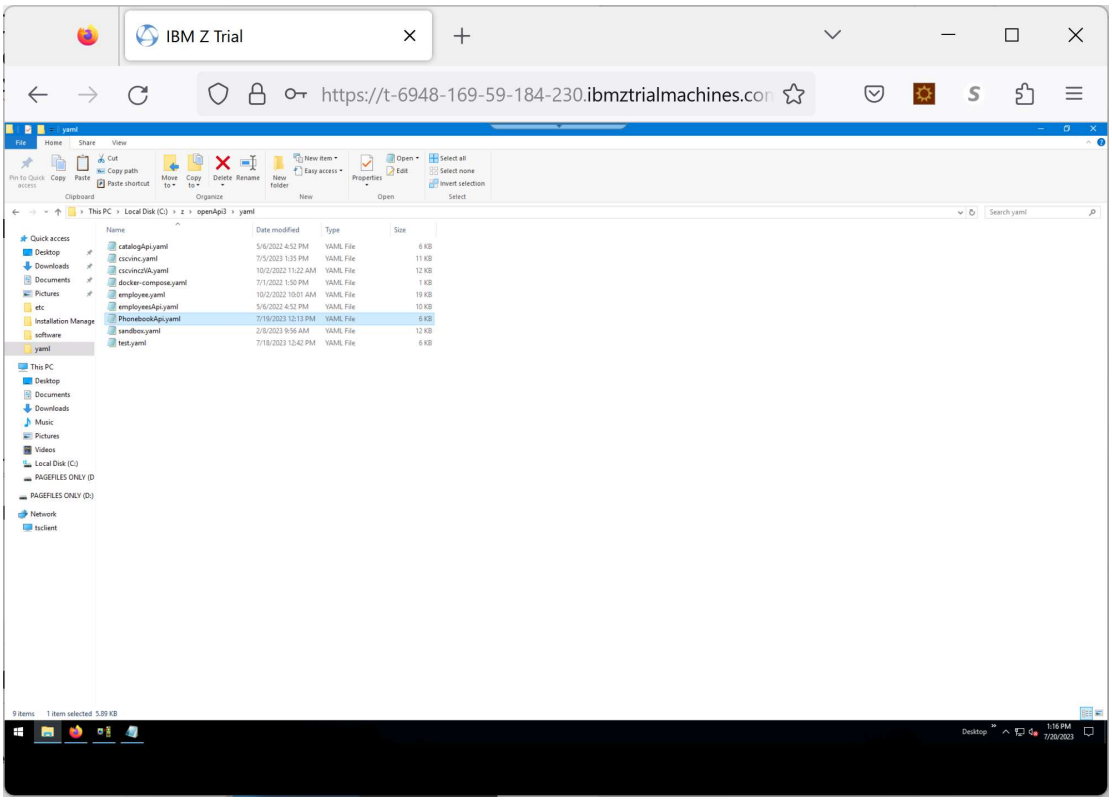
If everything on the IMS system is working properly, you will receive the results as follows

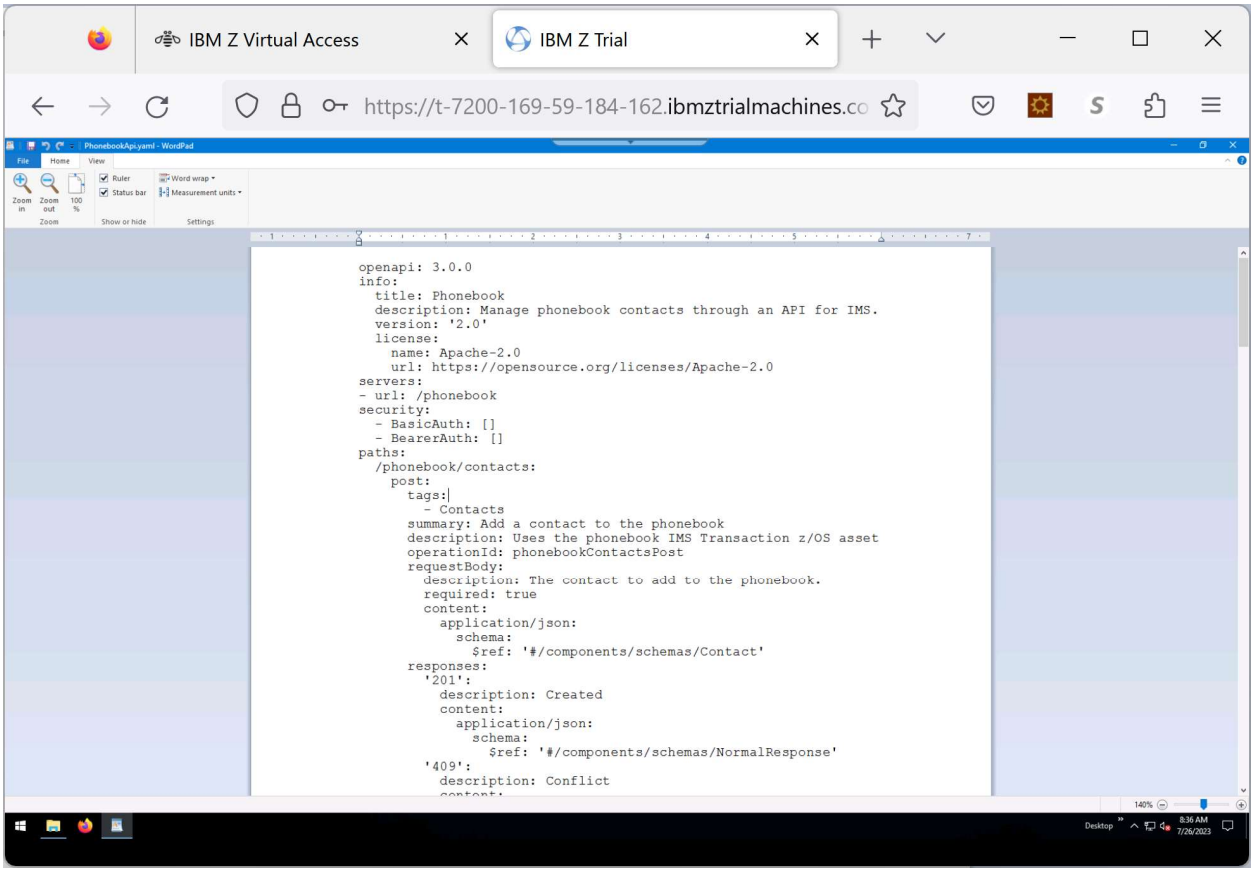


CREATE an IMS z/OS Connect API

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

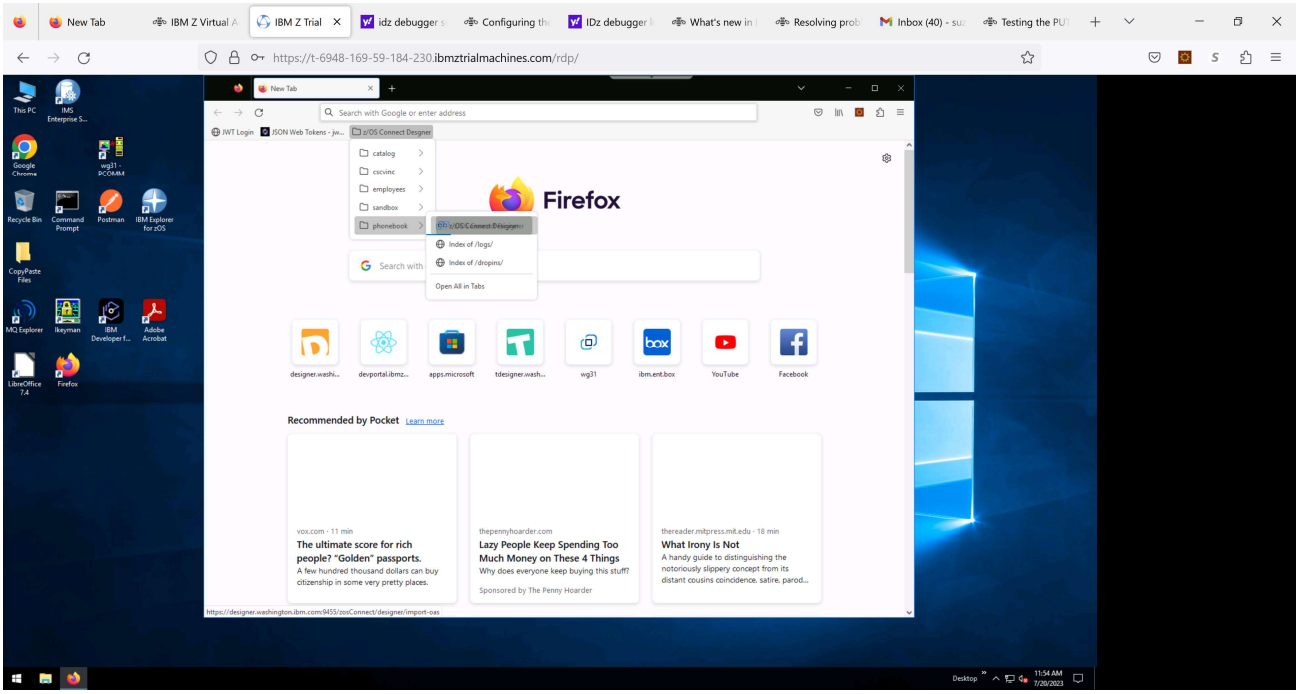
C: z/openapi3/yaml


IBM Z Trial
https://t-6948-169-59-184-230.ibmztrialmachines.com

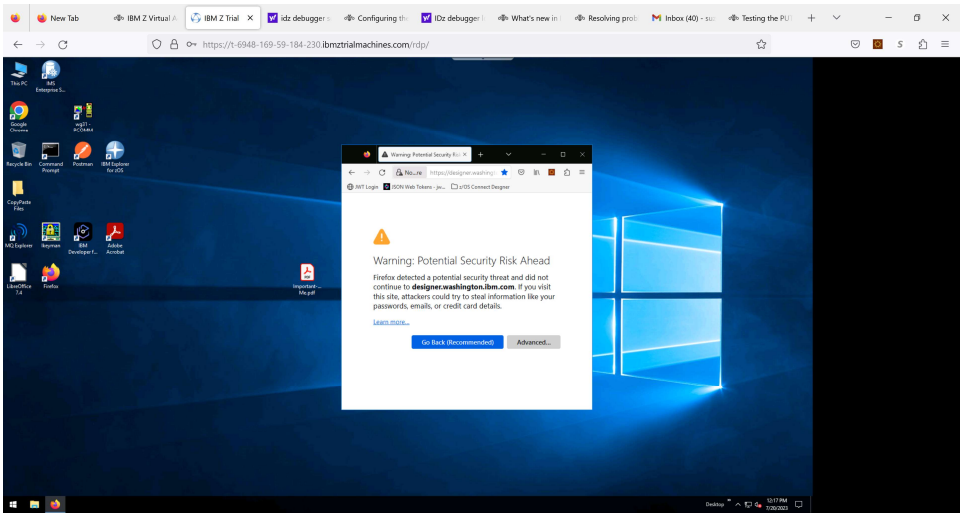

IBM Z Virtual Access
IBM Z Trial
https://t-7200-169-59-184-162.ibmztrialmachines.com
PhonebookApi.yaml - WordPad
openapi: 3.0.0
info:
title: Phonebook
description: Manage phonebook contacts through an API for IMS.
version: '2.0'
license:
name: Apache-2.0
url: https://opensource.org/licenses/Apache-2.0
servers:
- url: /phonebook
security:
- BasicAuth: []
- BearerAuth: []
paths:
/phonebook/contacts:
post:
tags:
- Contacts
summary: Add a contact to the phonebook
description: Uses the phonebook IMS Transaction z/OS asset
operationId: phonebookContactsPost
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

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

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

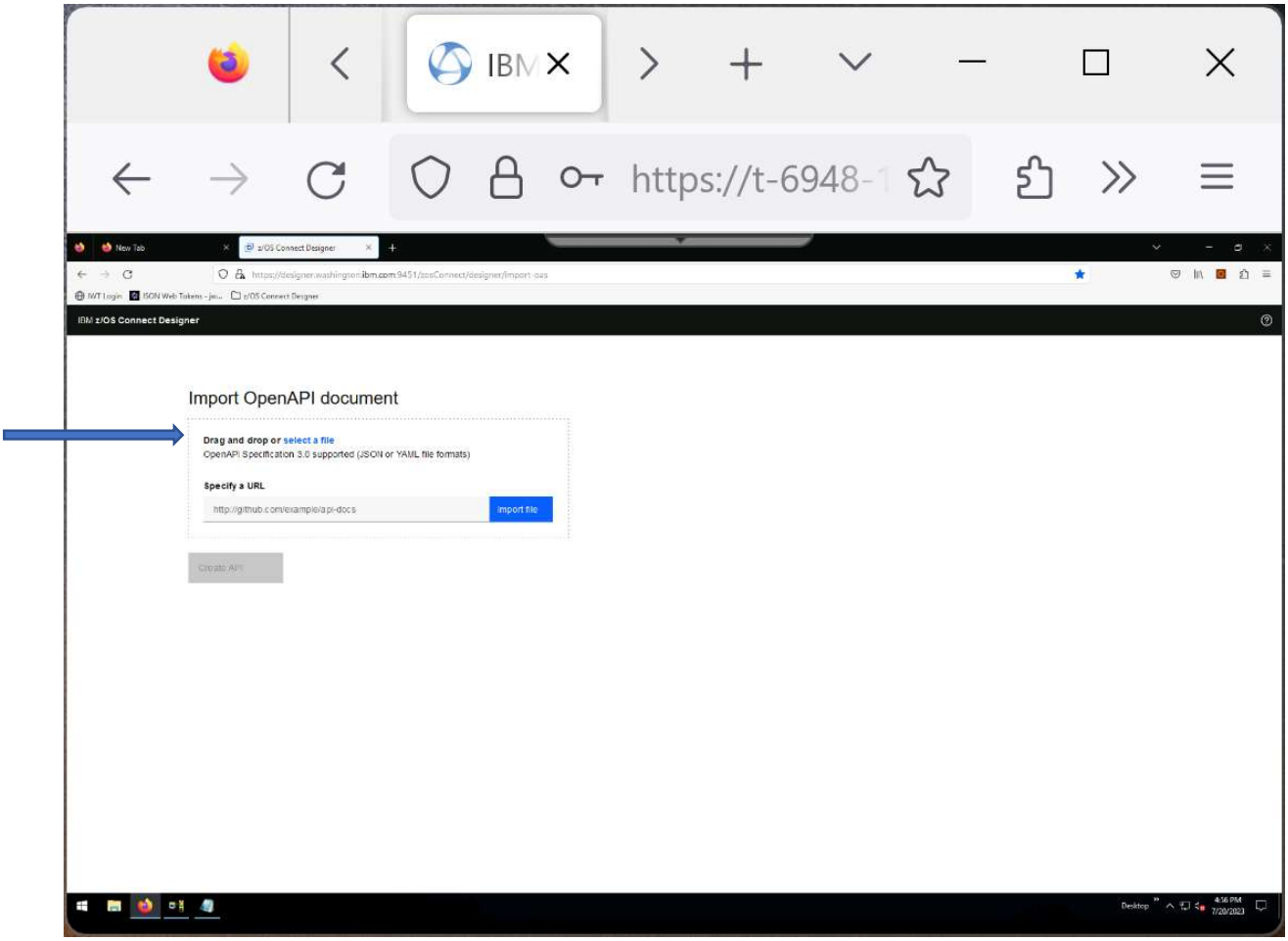


- Accept the risk (Advanced) and wait (it takes a while to set it up)

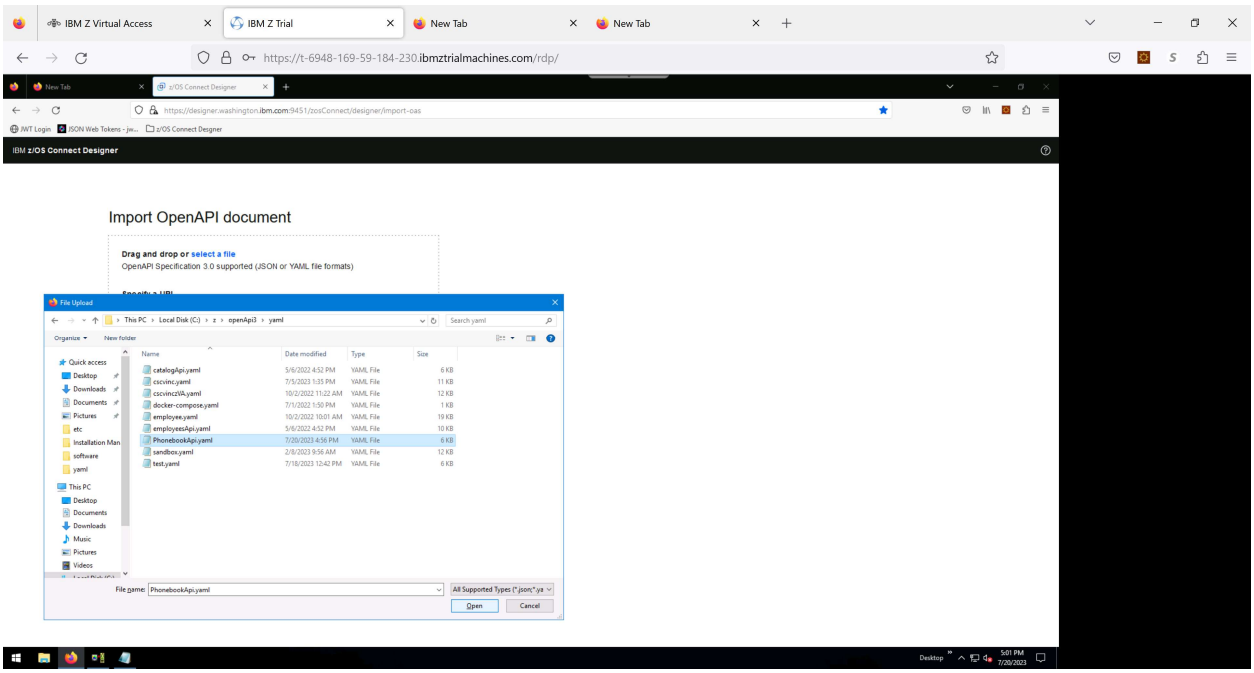


The tool will ask you to choose an OpenAPI document (yaml)

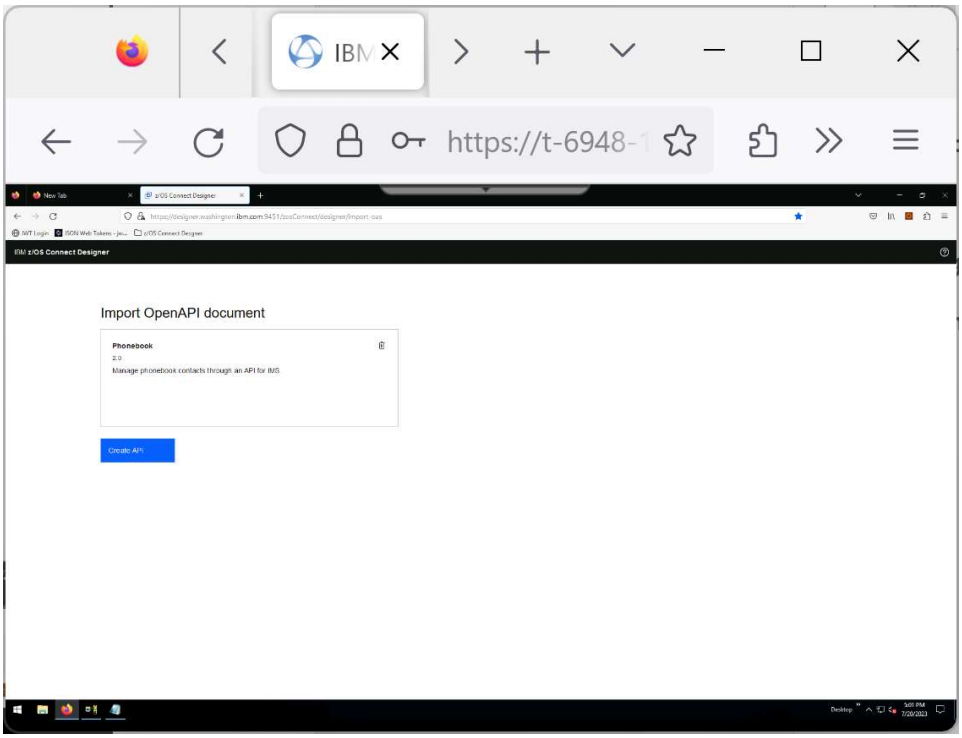
- Click on **select a file**



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



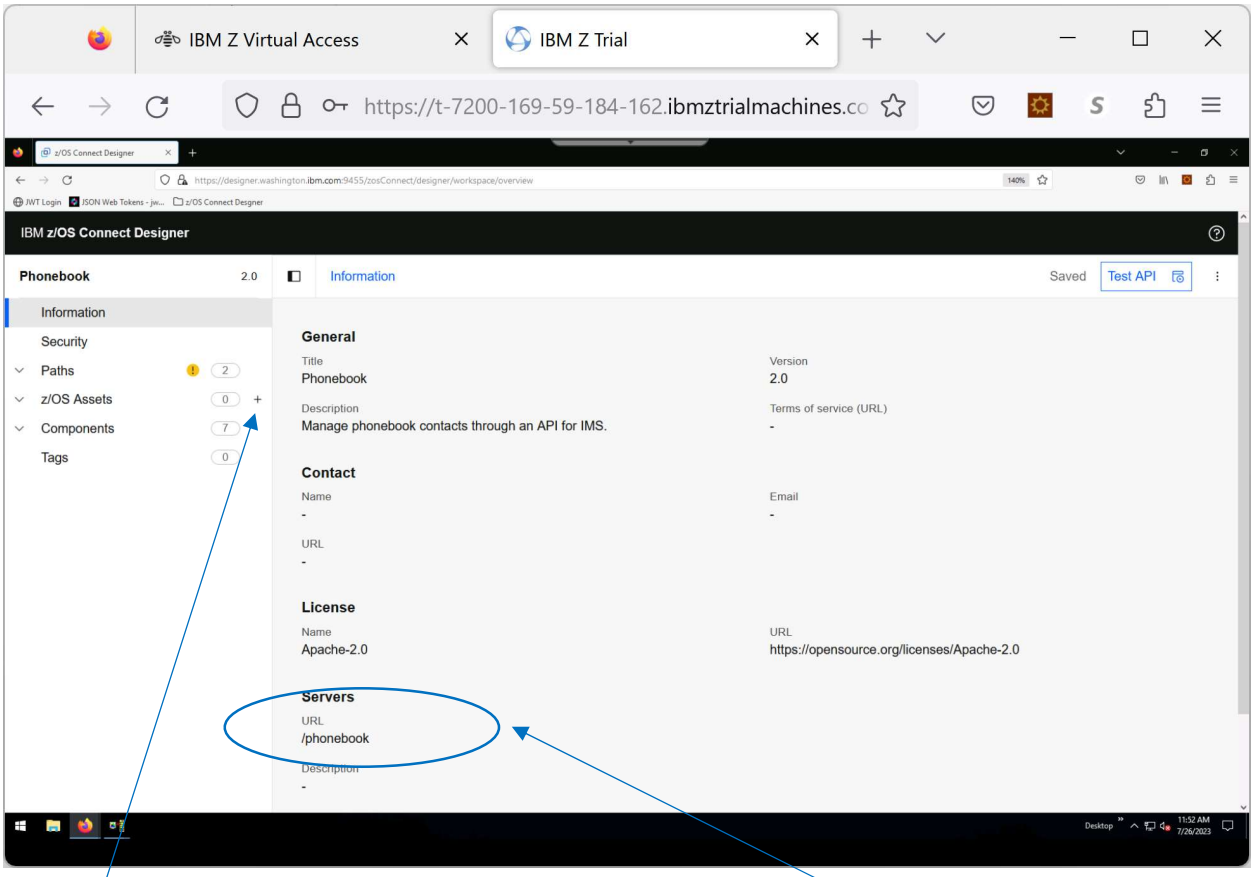
- Click on Create API



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

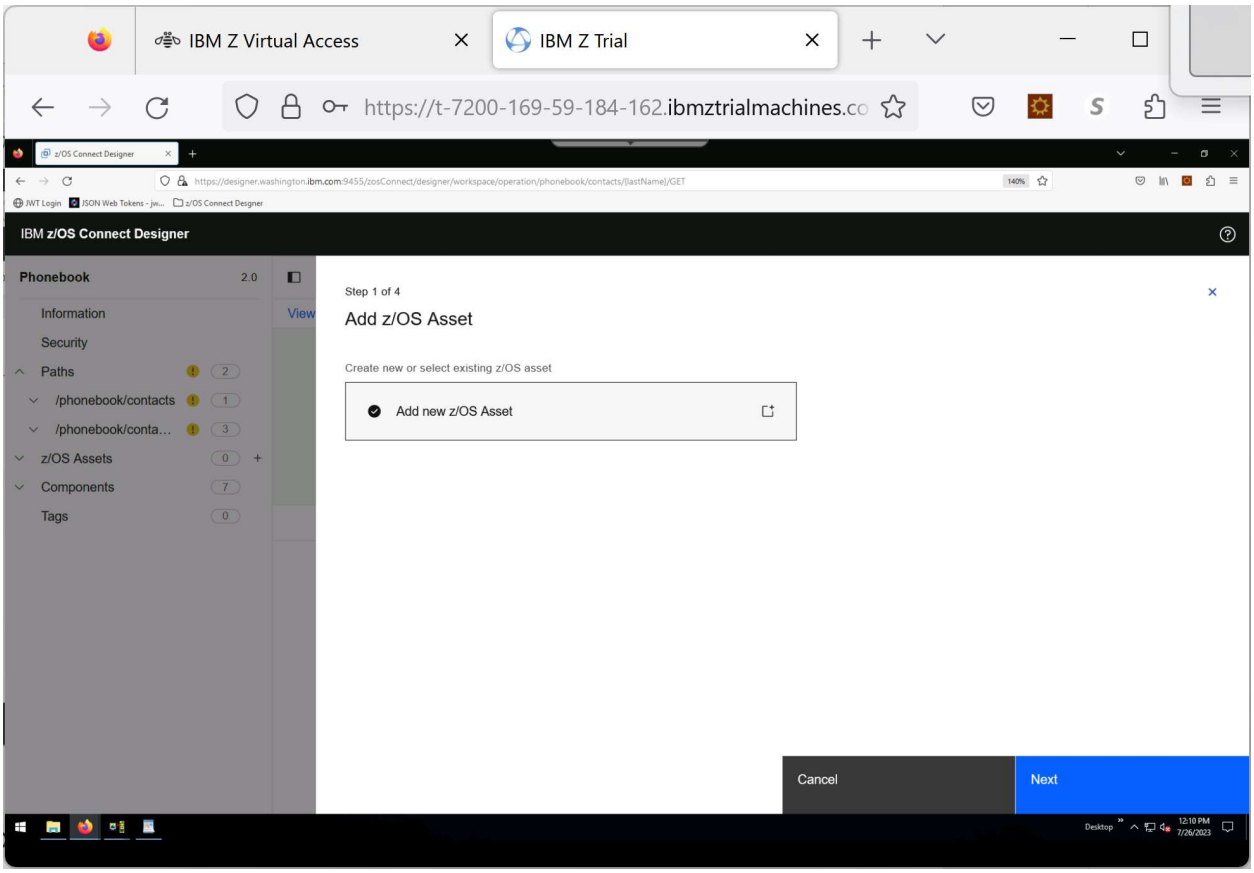


Click on the plus (**+**) sign by **z/OS Assets**

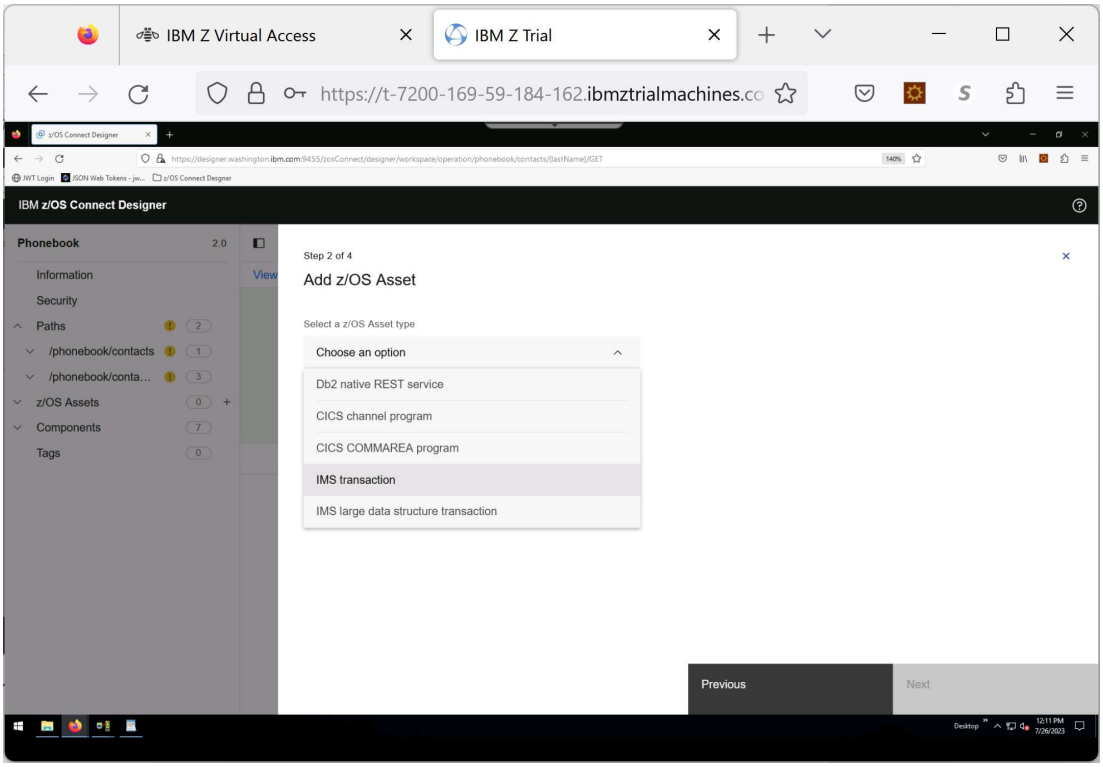
(Note: the URL of the server you will use to test later)

- 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.



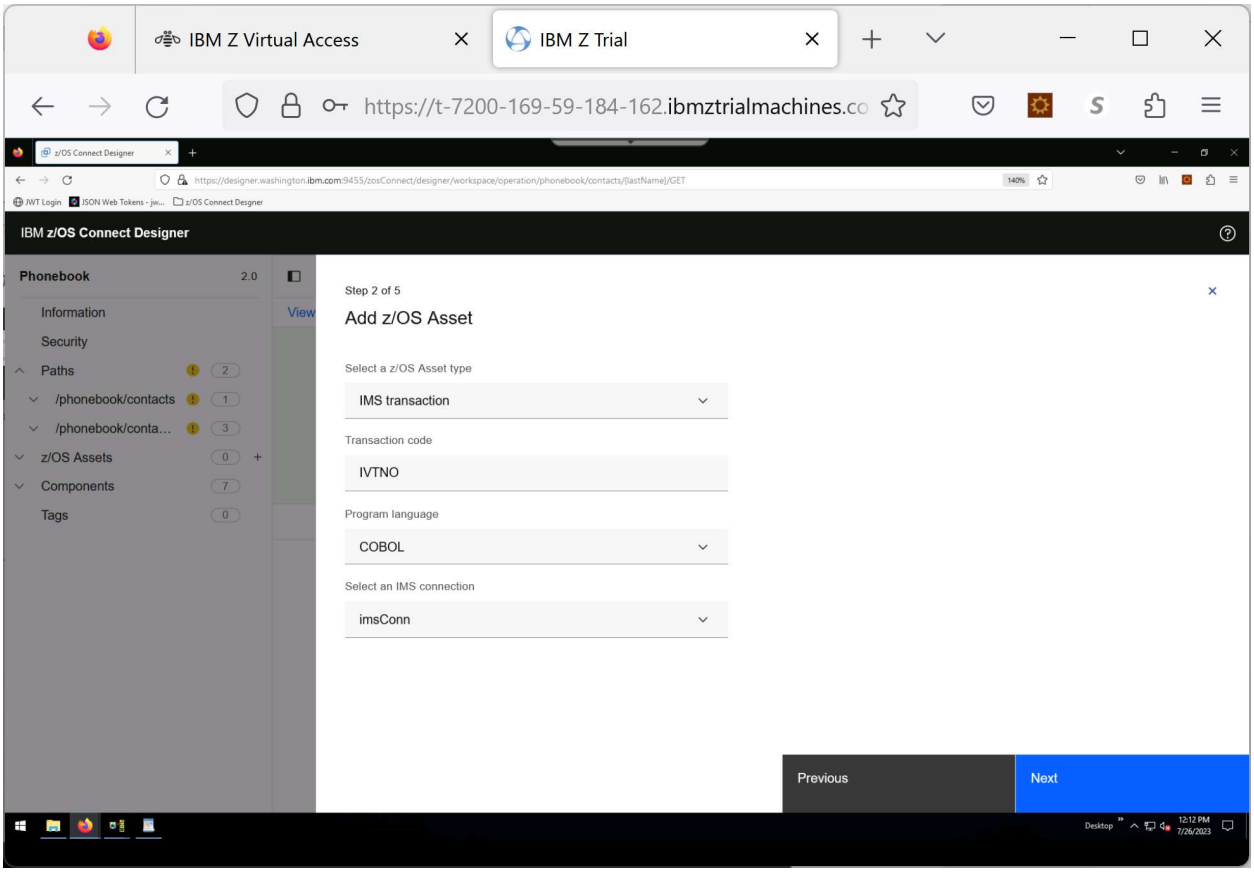
- Highlight “Add new z/OS Asset” and click Next.



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

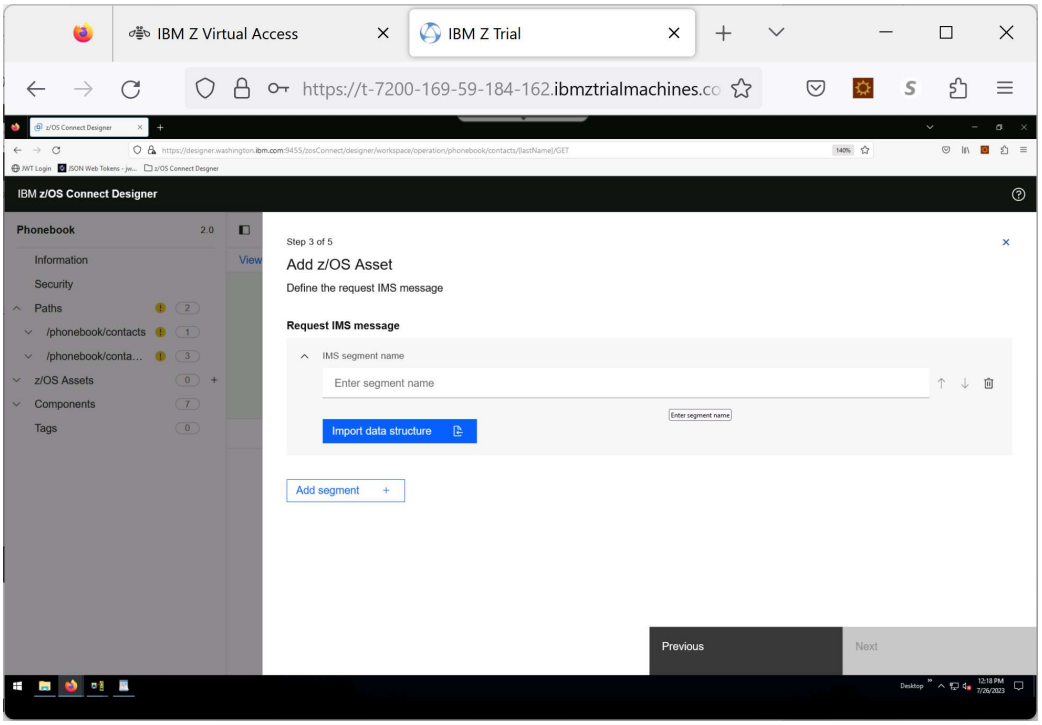
When the window opens,

  - 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

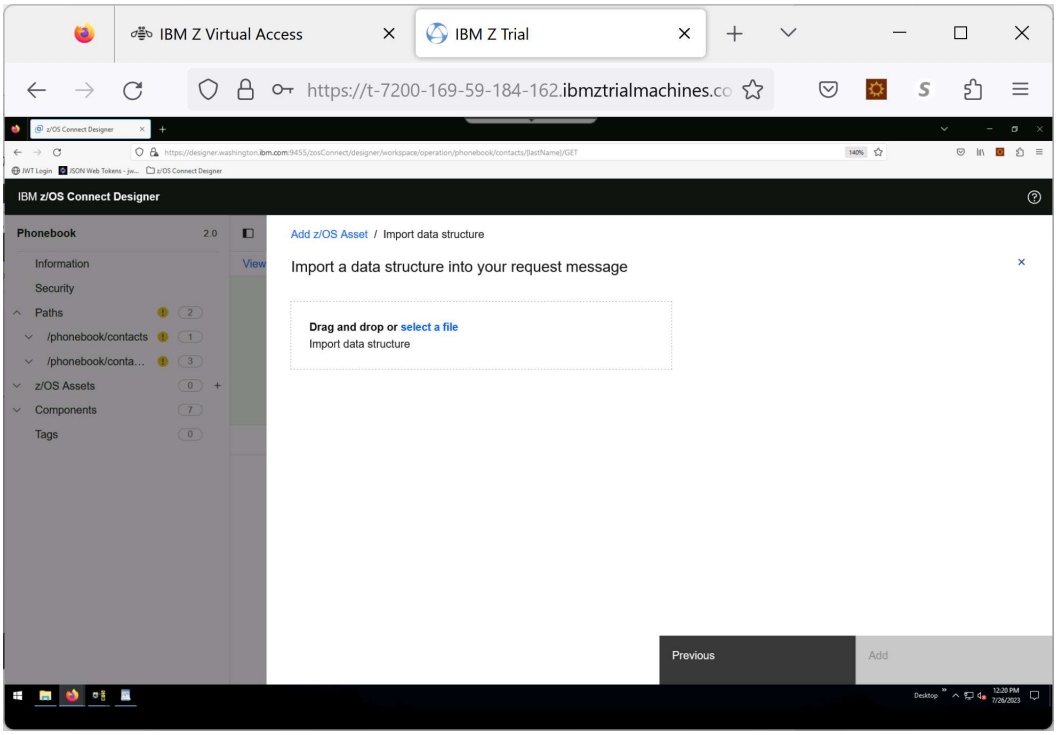


Click **Next.**

Define the request IMS message

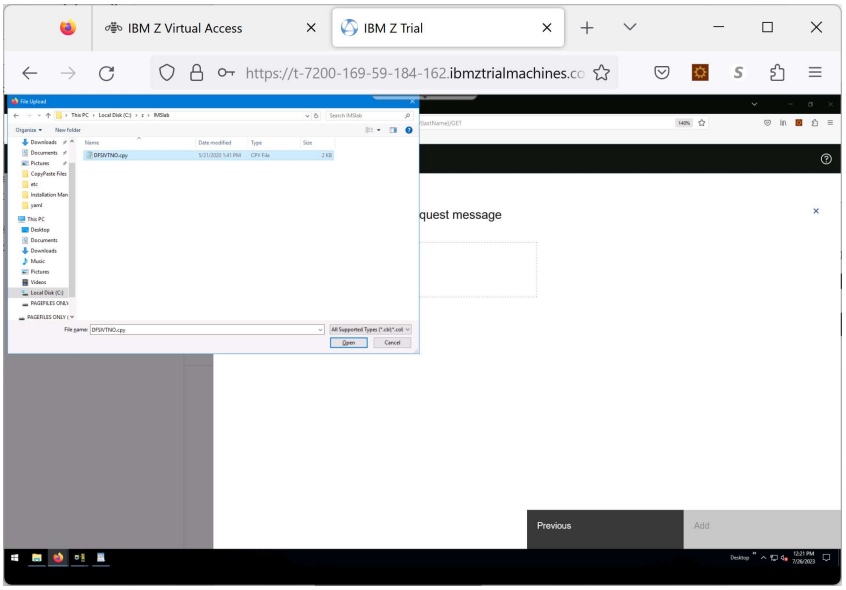


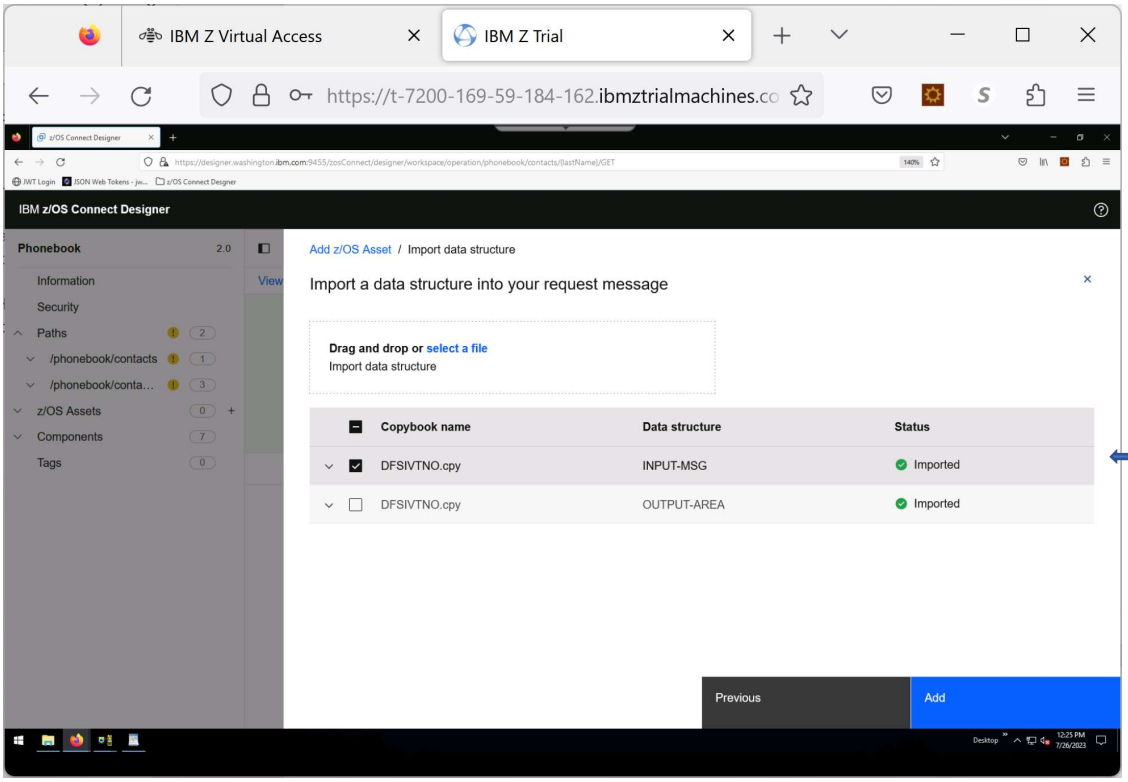
- Key in **INPUT-MESSAGE** as the segment name and click on **Import data structure**



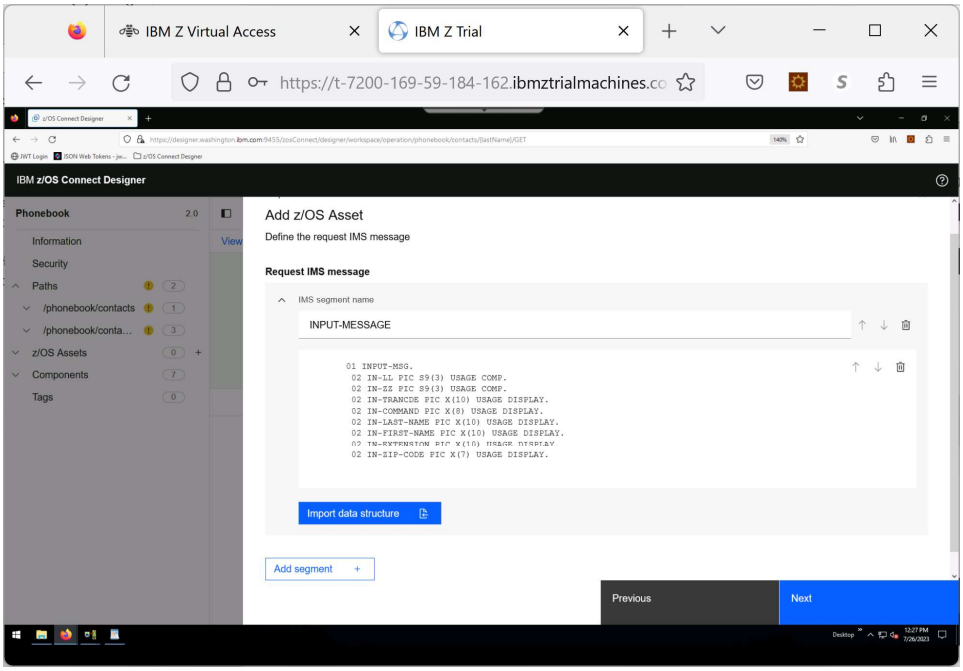
Click on **select a file**

- 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





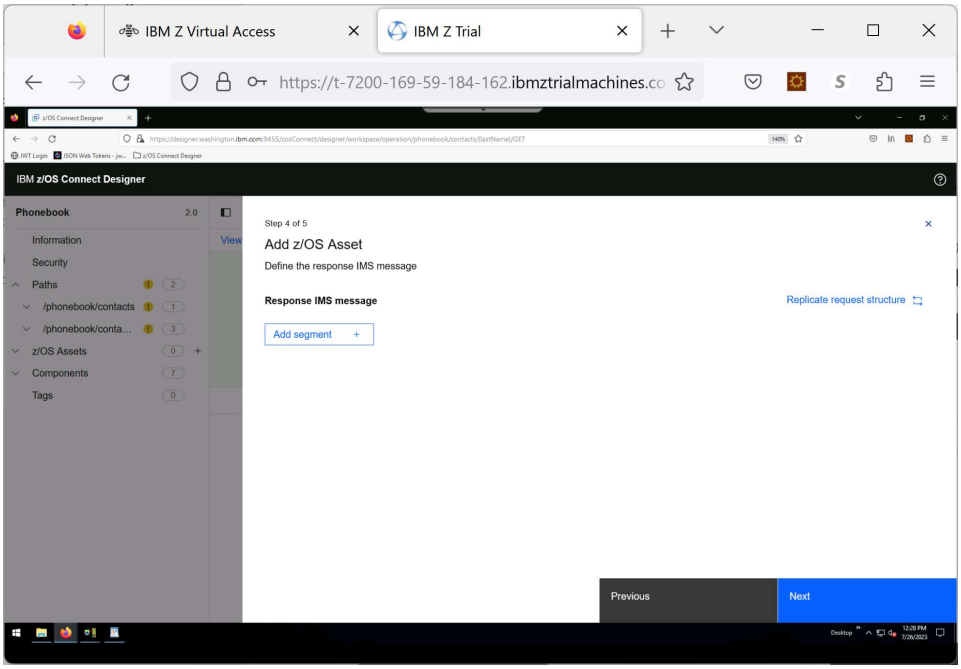
To include the request message, ONLY select the INPUT-MSG data structure. Click **Add.**



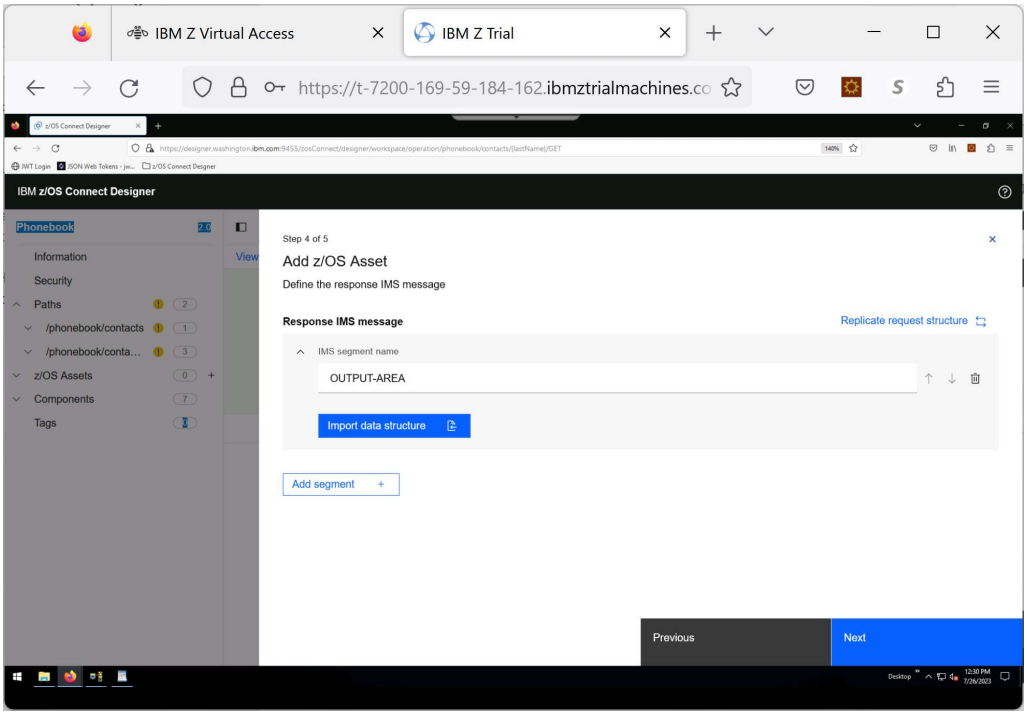
This is the COBOL copybook associated with the input message for IVTNO.

Click **Next**.

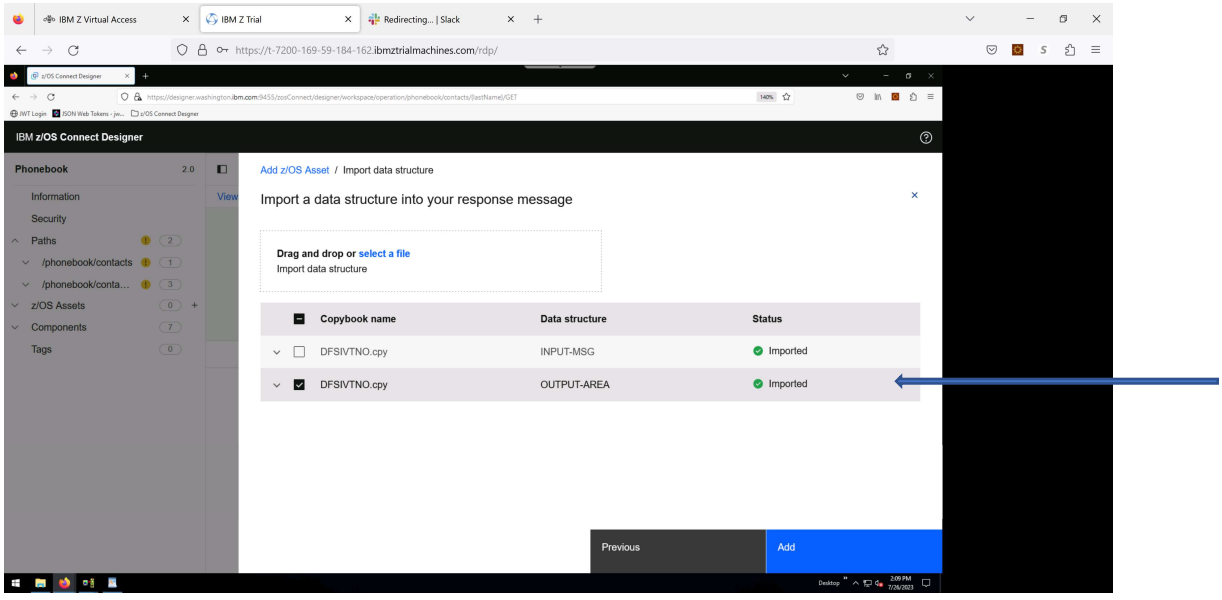
Add the segment for the **response** message.



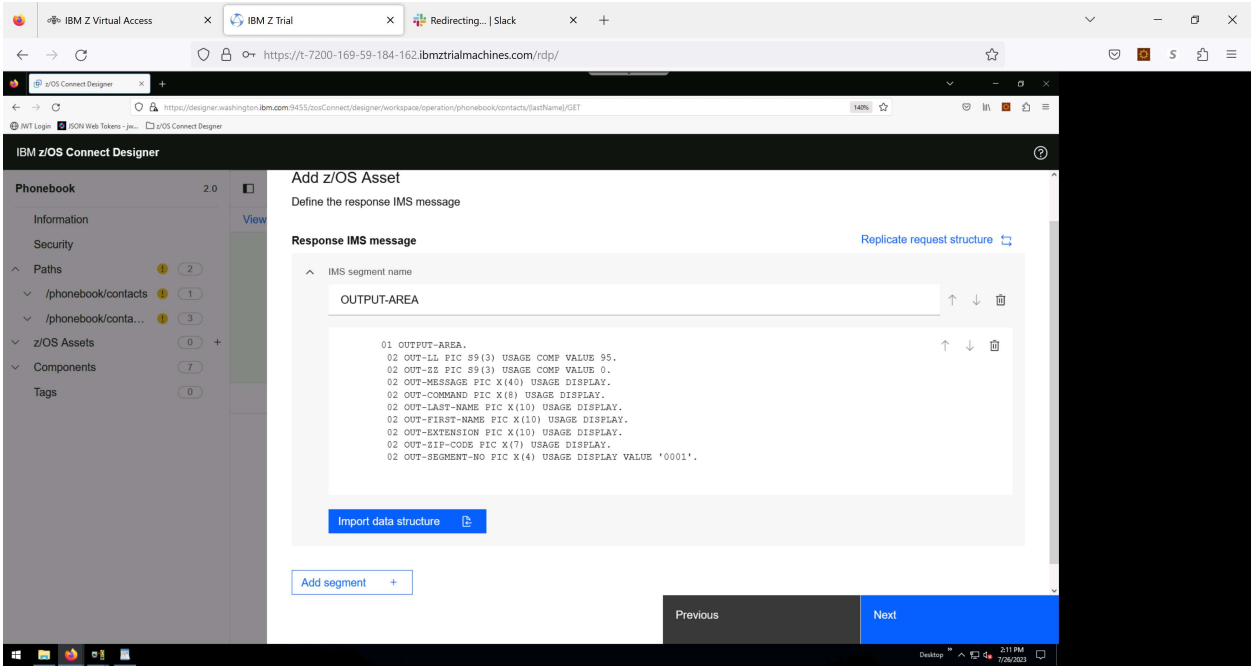
For segment name, specify **OUTPUT-AREA**



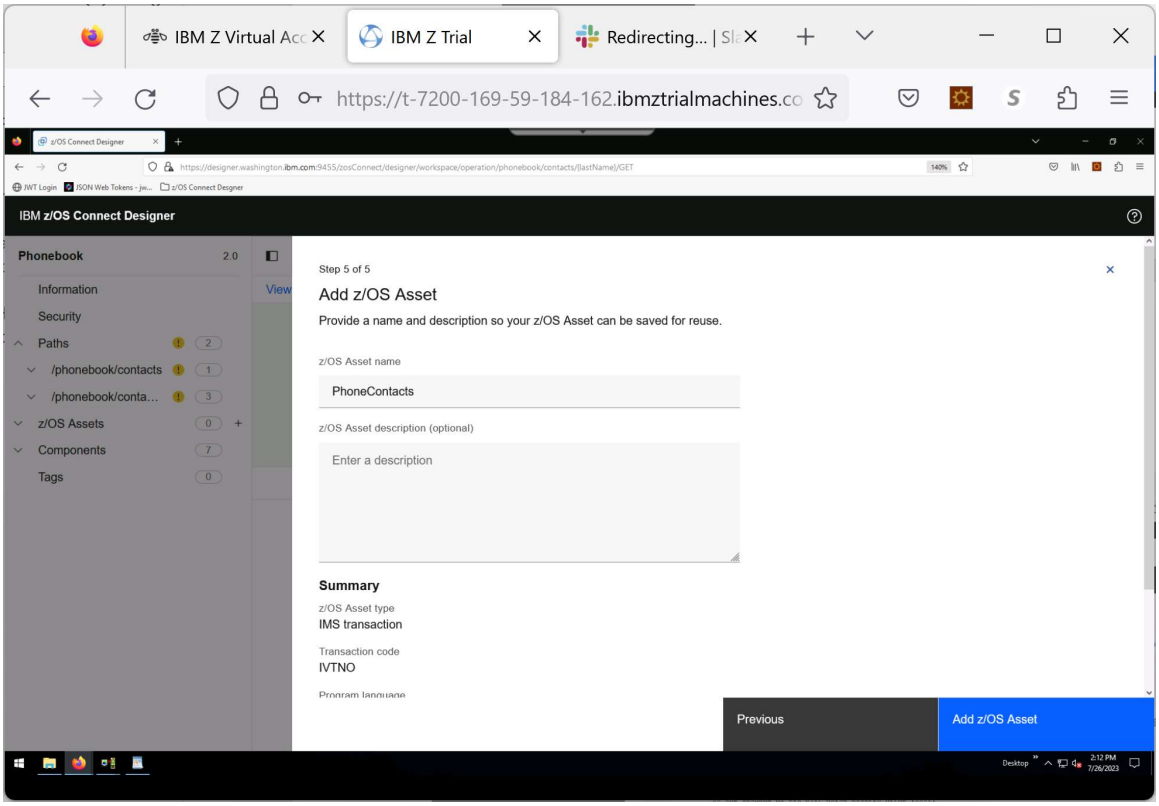
- Click **Import data structure** and follow the steps, as above, to retrieve the DFSIVTNO.CPY file



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

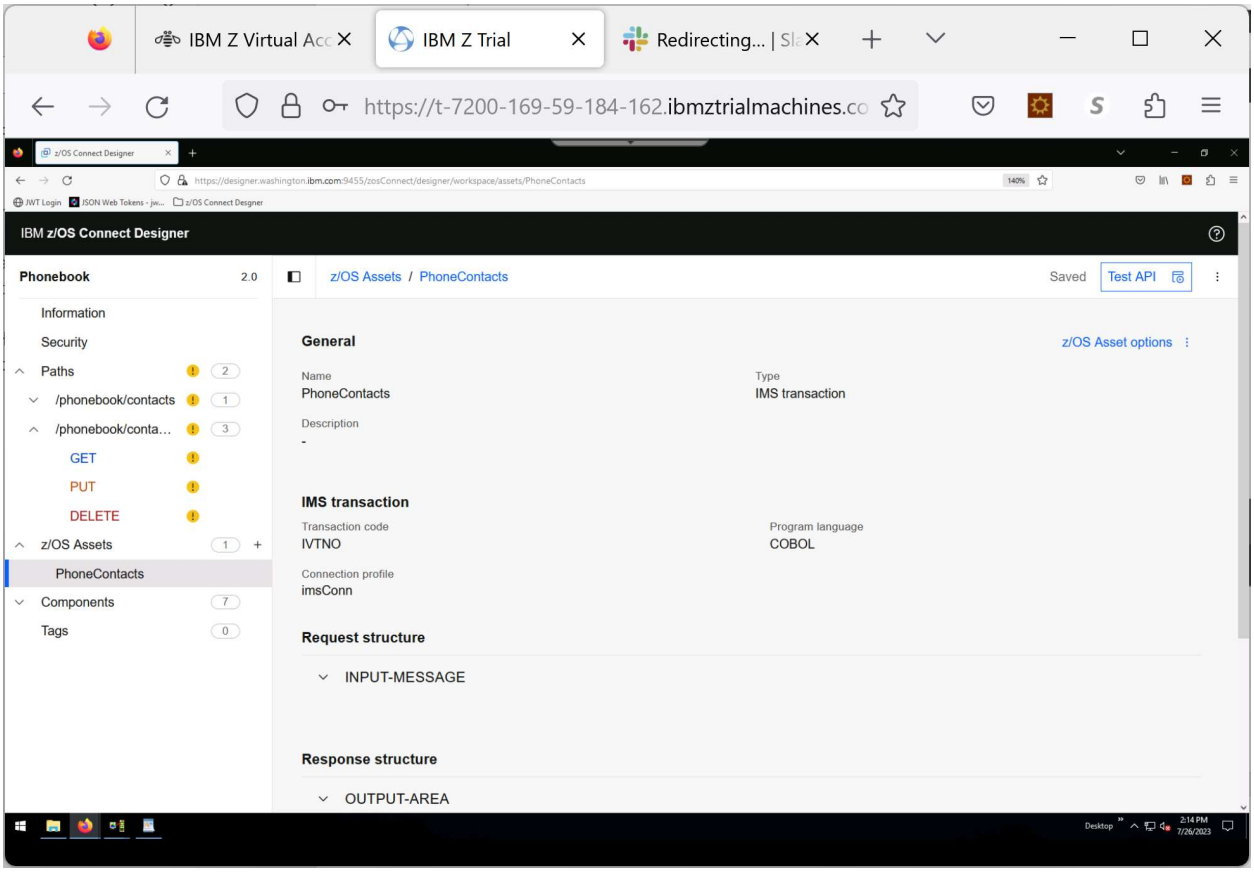


Click **Next.**



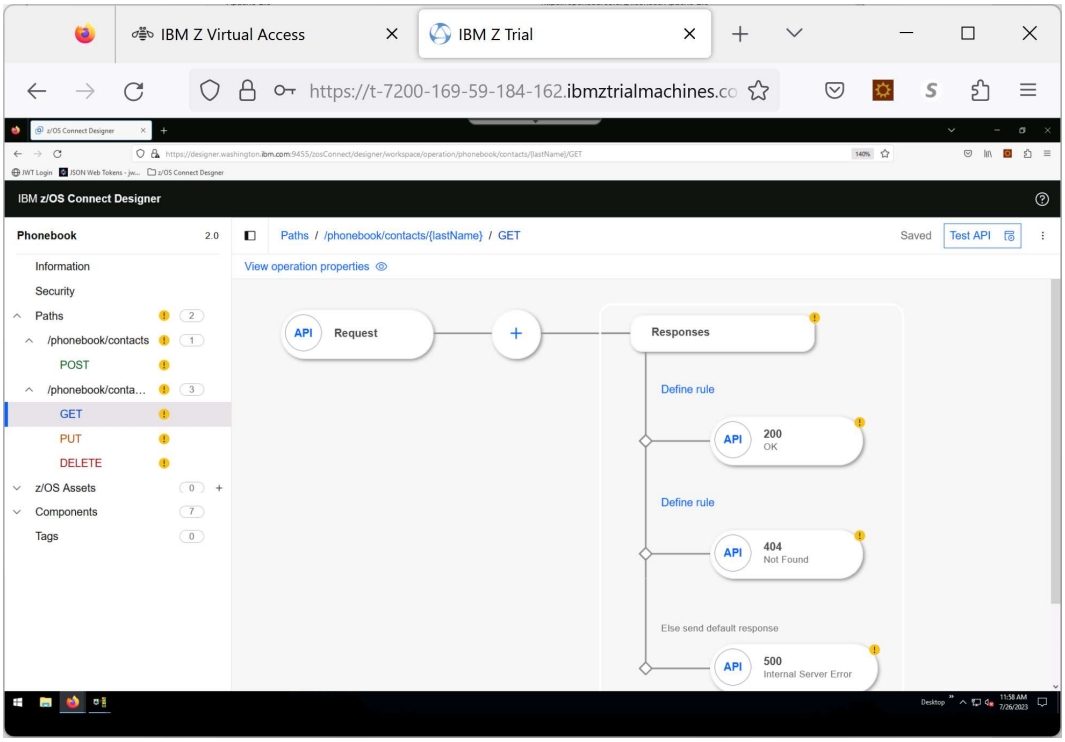
- Name the z/OS Asset – e.g., **PhoneContacts**
- Click on “Add z/OS Asset”

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

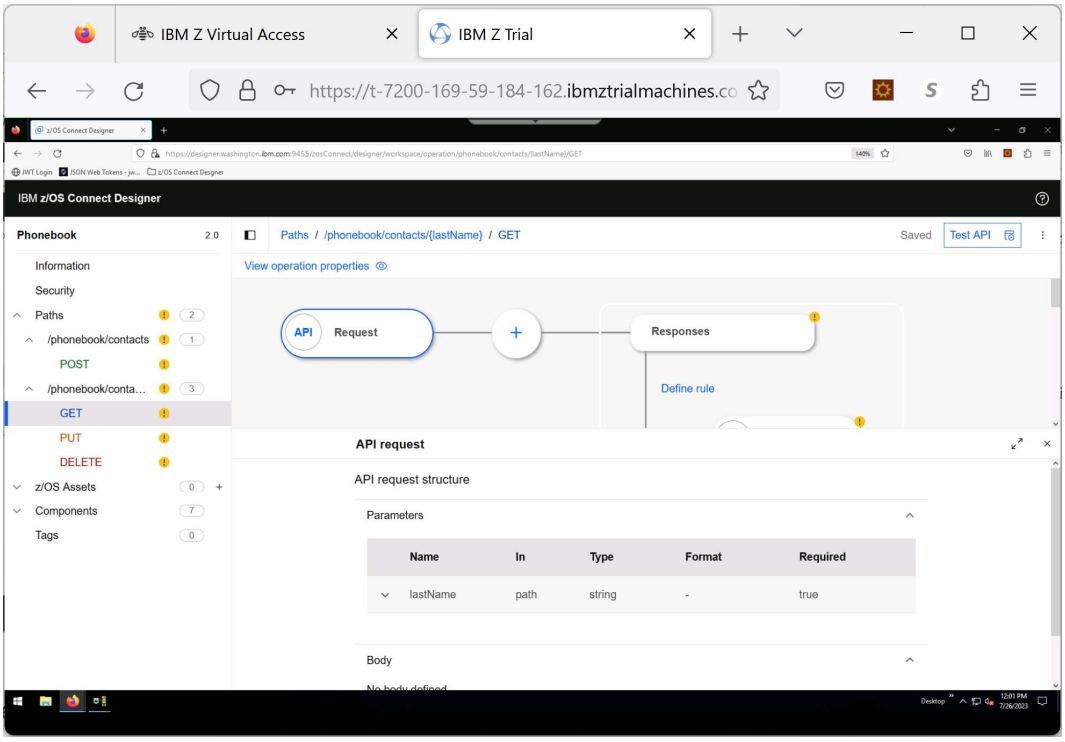


Create the API – GET method

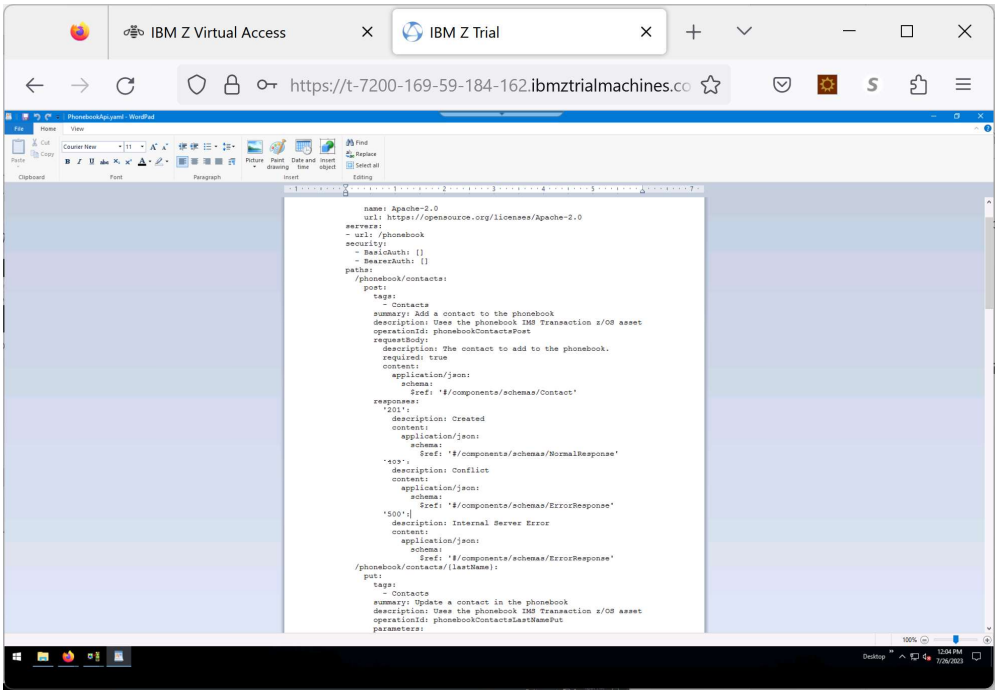
Click on the GET method to view the operation properties. This opens the GET /phonebook/contacts/{lastName} method



Click on (API) request

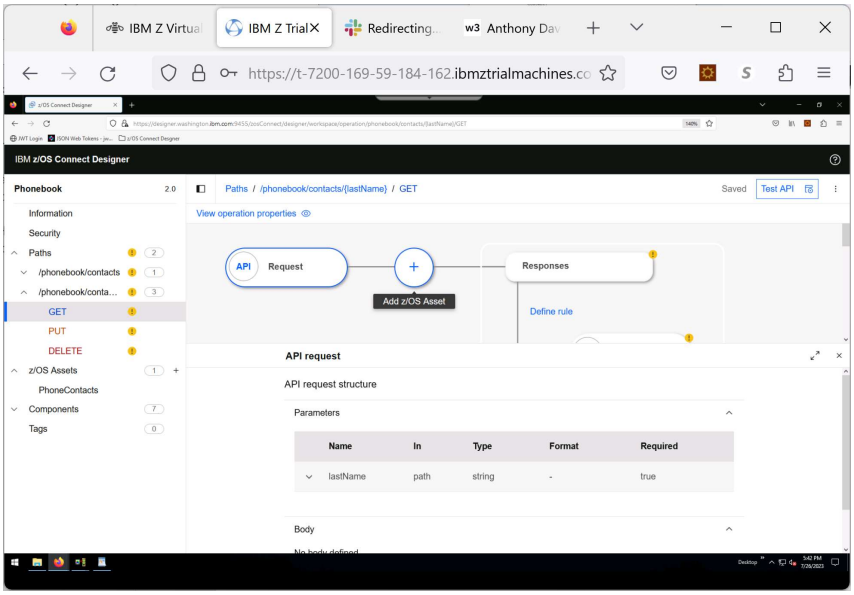


Notice that the lastName parameter is part of the request – this parameter was defined in the phonebook.yaml

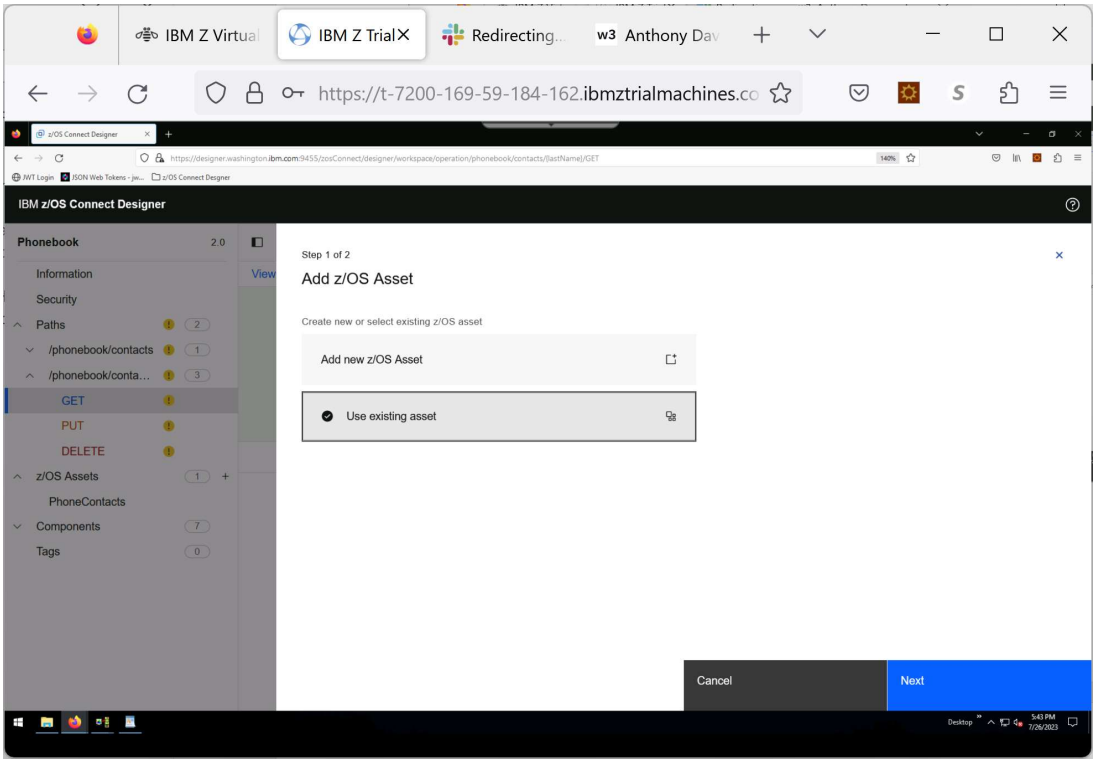


Map the API Request to the z/OS asset

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

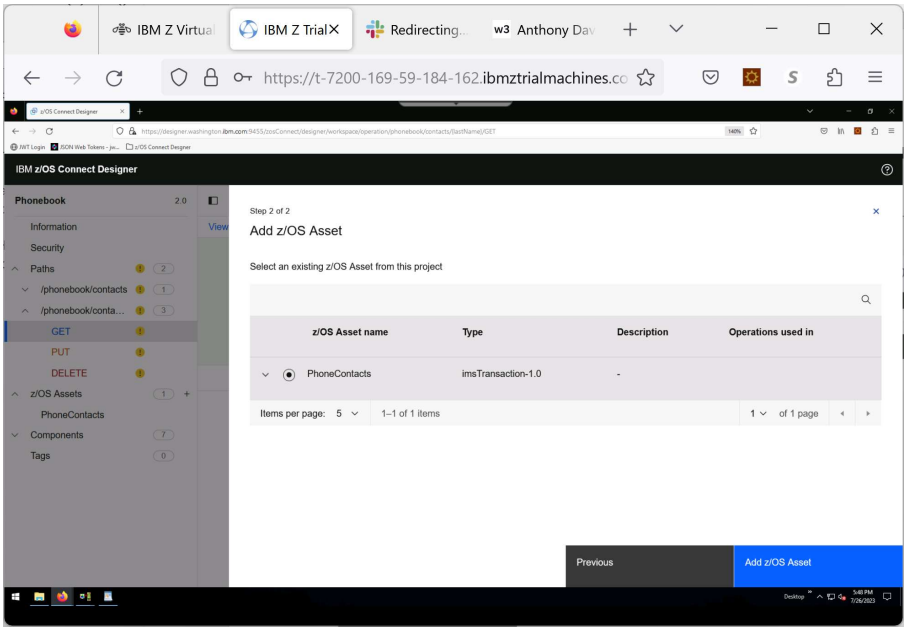


- Click on **Use existing asset** since we will use the one you just created.



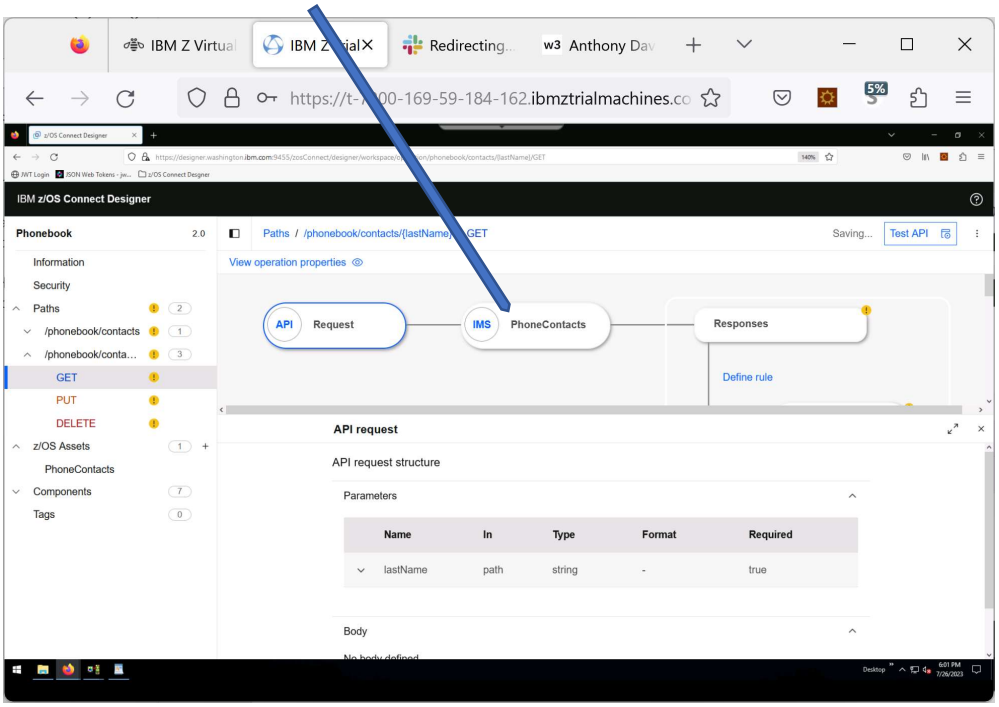
- Click **Next**.

The z/os Asset previously created is displayed

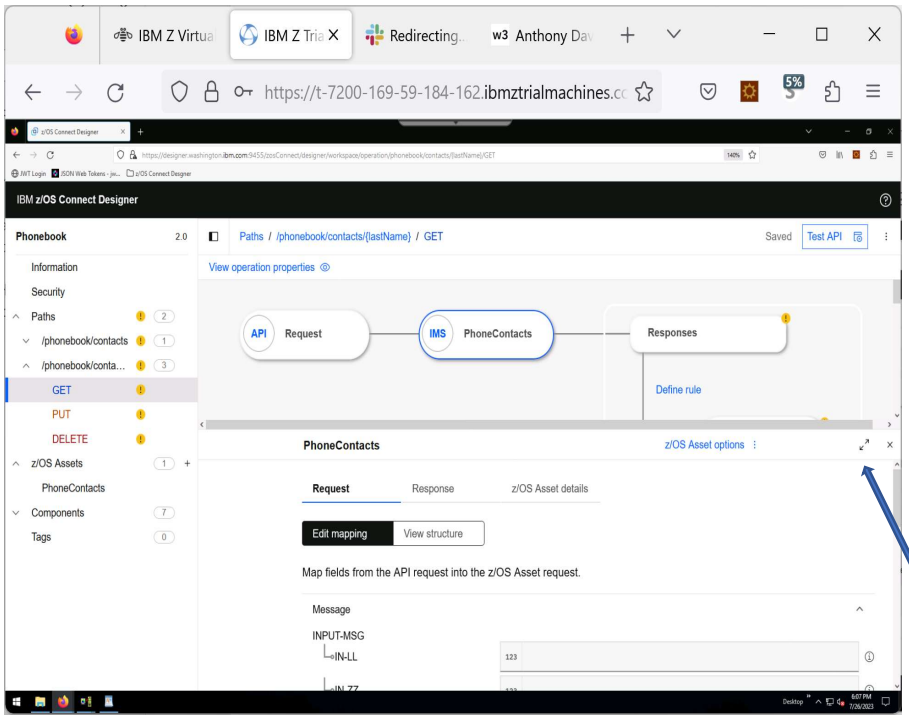


- Click **Add z/OS Asset**

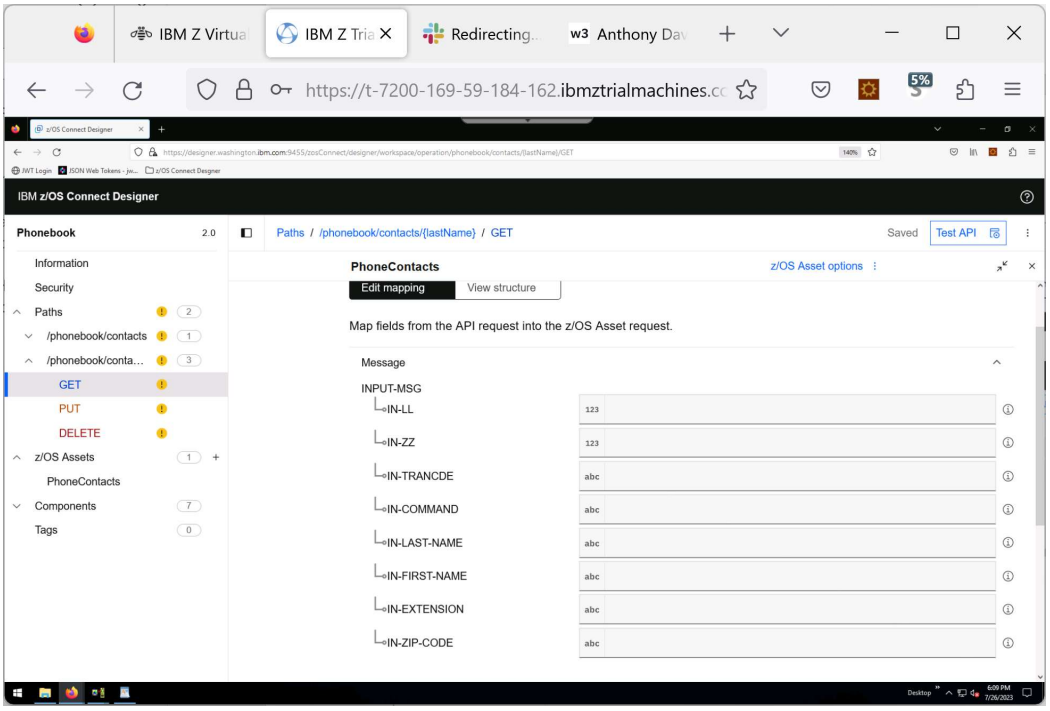
Note that the asset is now listed in the Operations flow diagram.



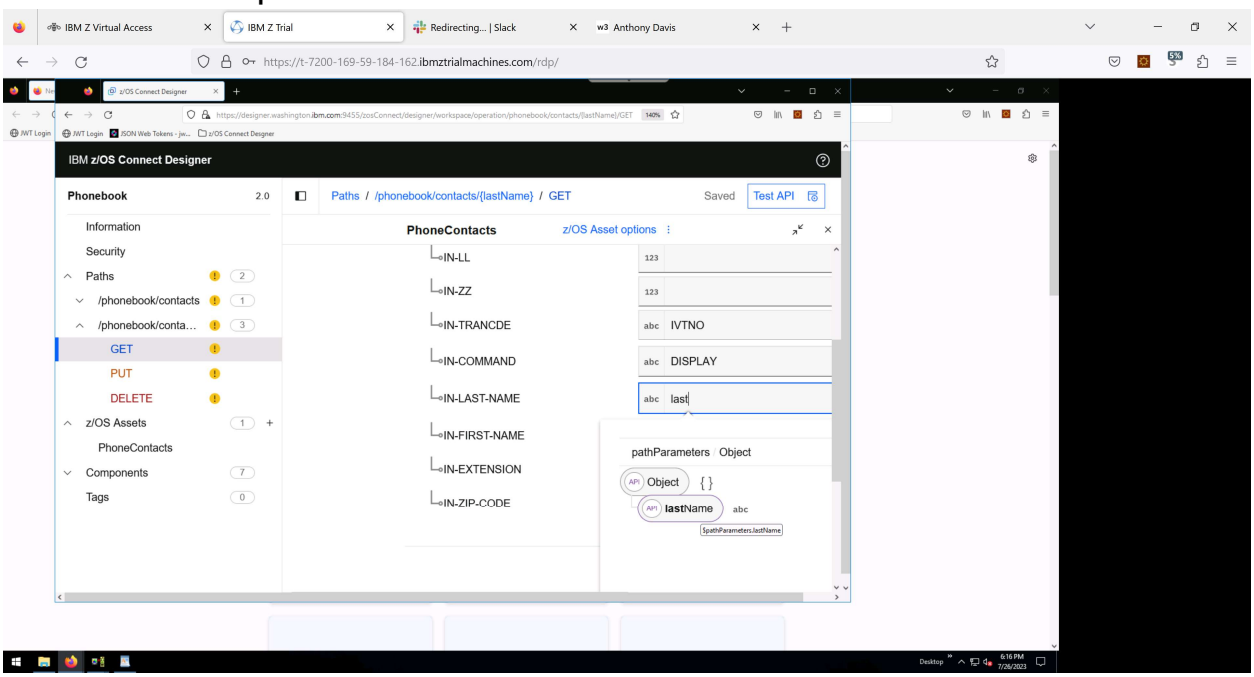
- Click the asset your created e.g., “PhoneContacts”

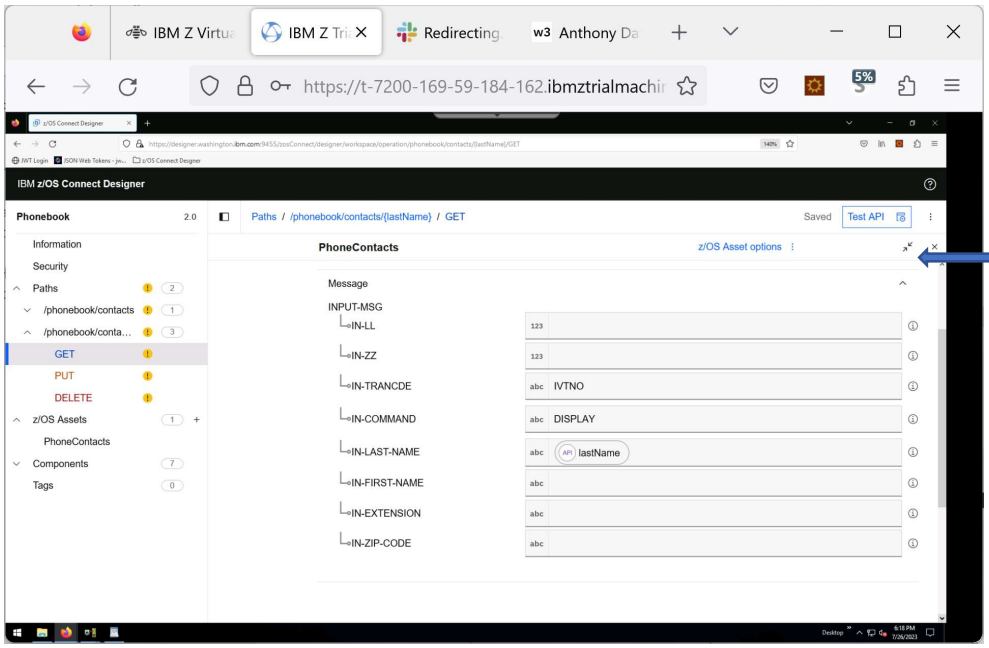


- Click on **Edit Mapping** under **Request**
- Expand the bottom pane by clicking on the two arrows on the top right



- 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*.

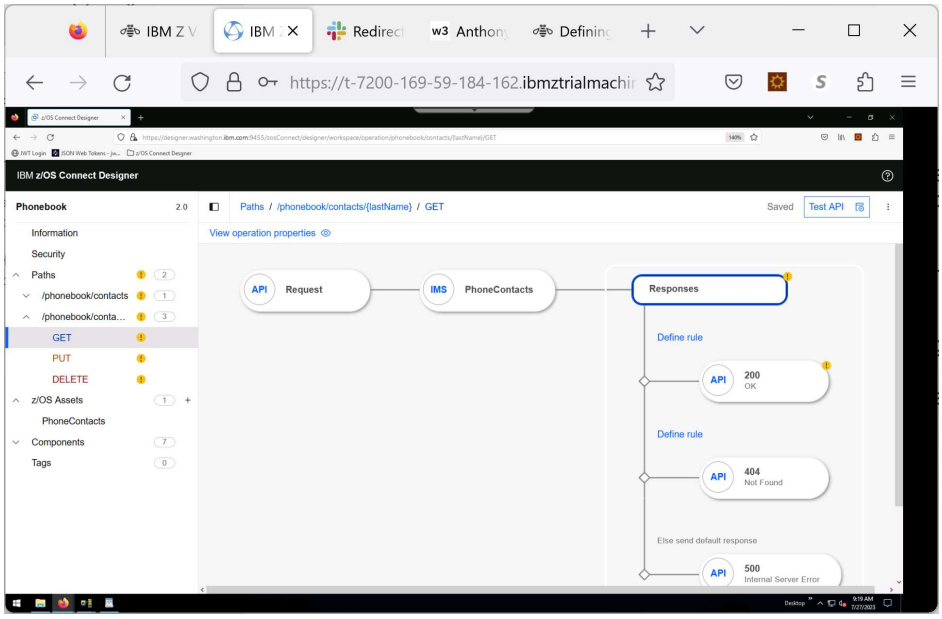
- You can now minimize this panel by clicking on the arrows on the top right. This will bring you back to the main operations flow diagram.

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:

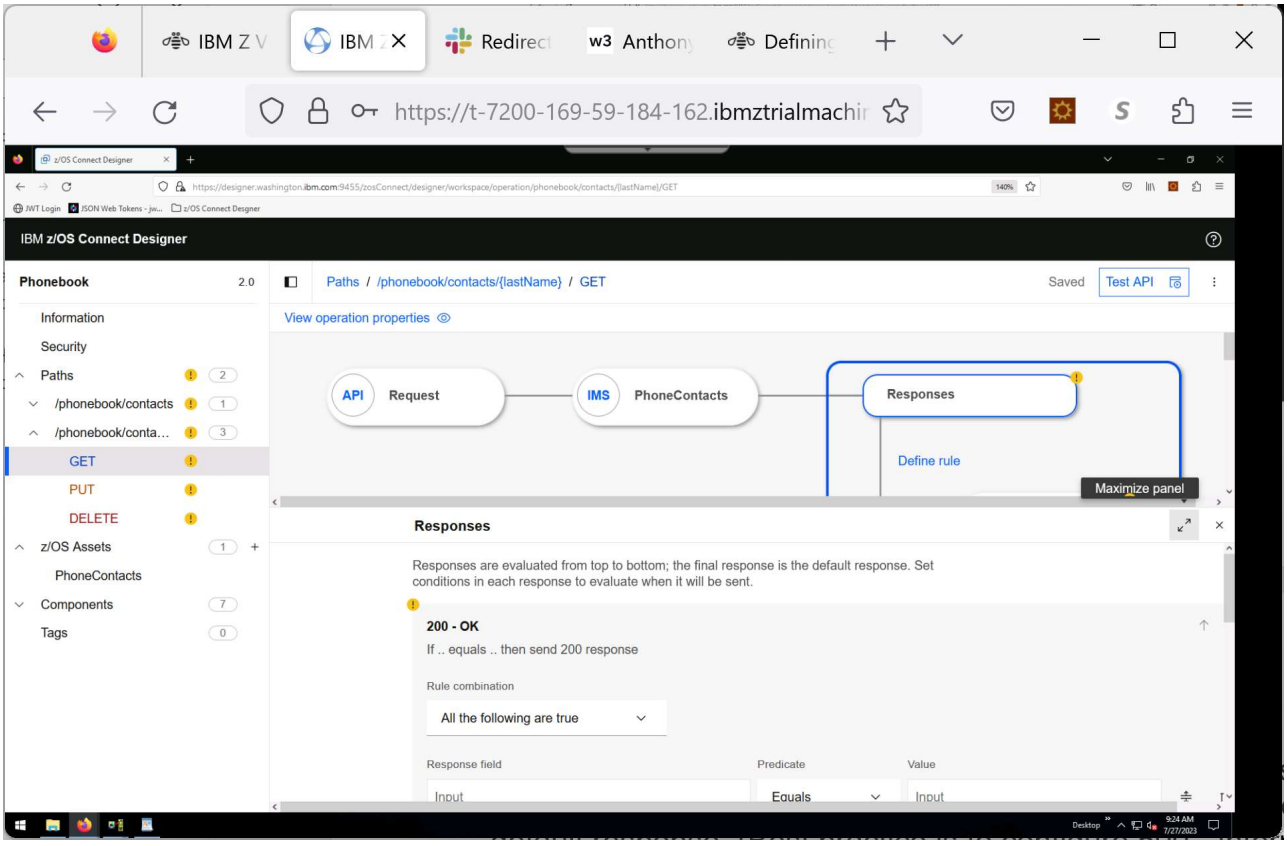
- A condition with three fields, *response*, *predicate*, and *value*.
- One or more *conditions*.
- You can change the order of the responses by using the ↑ and ↓ buttons next to each response case.
- The sequence of the conditions within a response can be changed. Click ⇕ to change the position in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

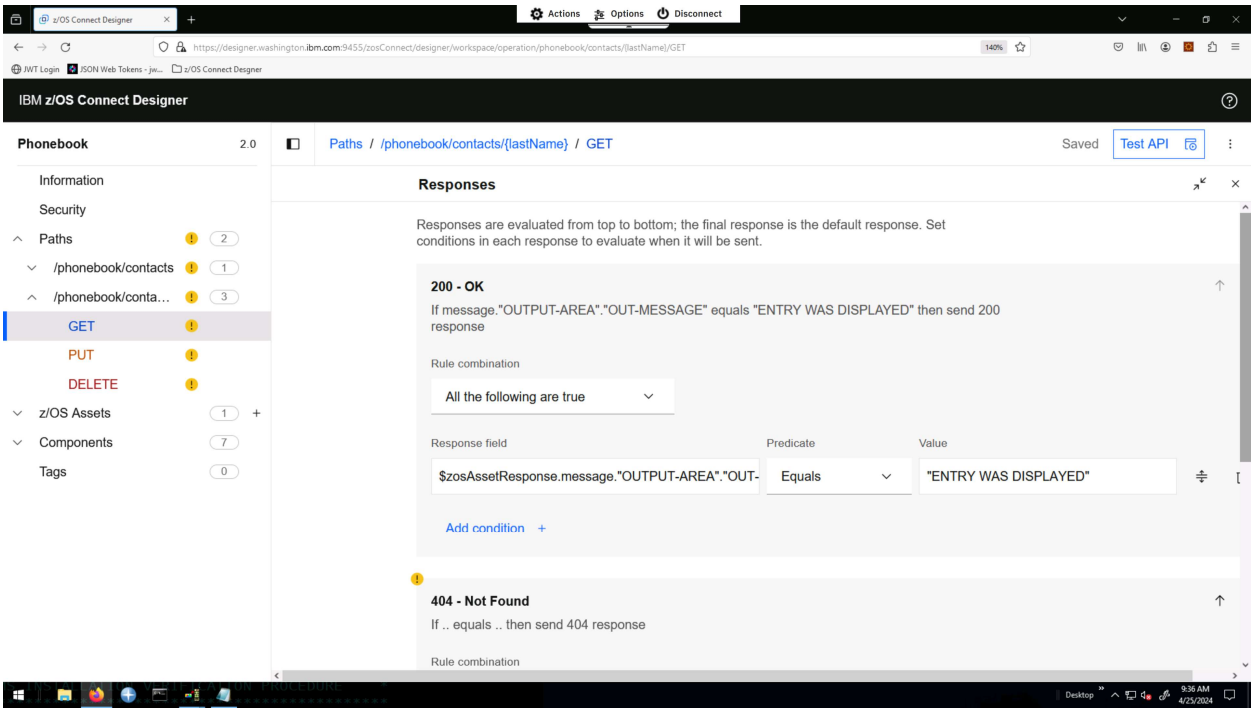


Set the conditions for the 200 OK response condition.

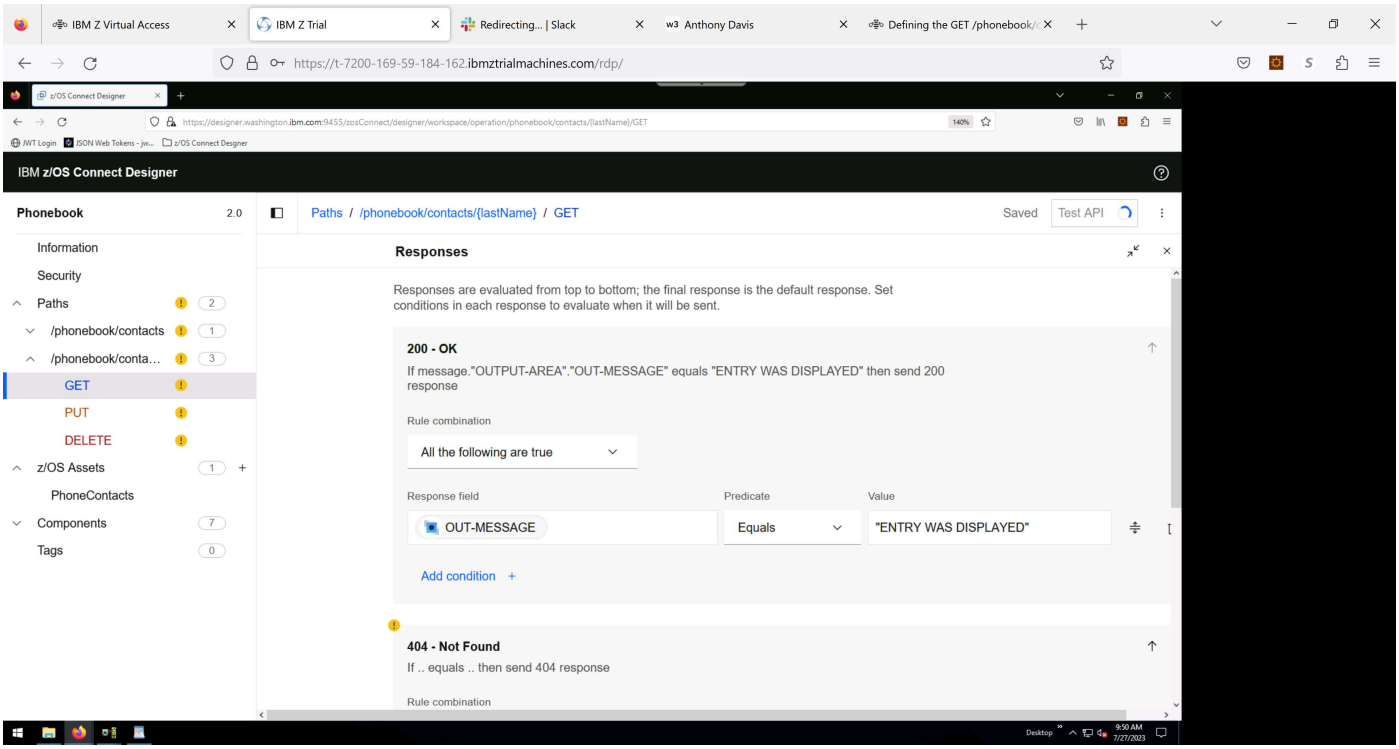
- Key in the following into the **Response field** – ***note*** the case and quotes
  $zosAssetResponse.message.”OUTPUT-AREA”.”OUT-MESSAGE”

  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.



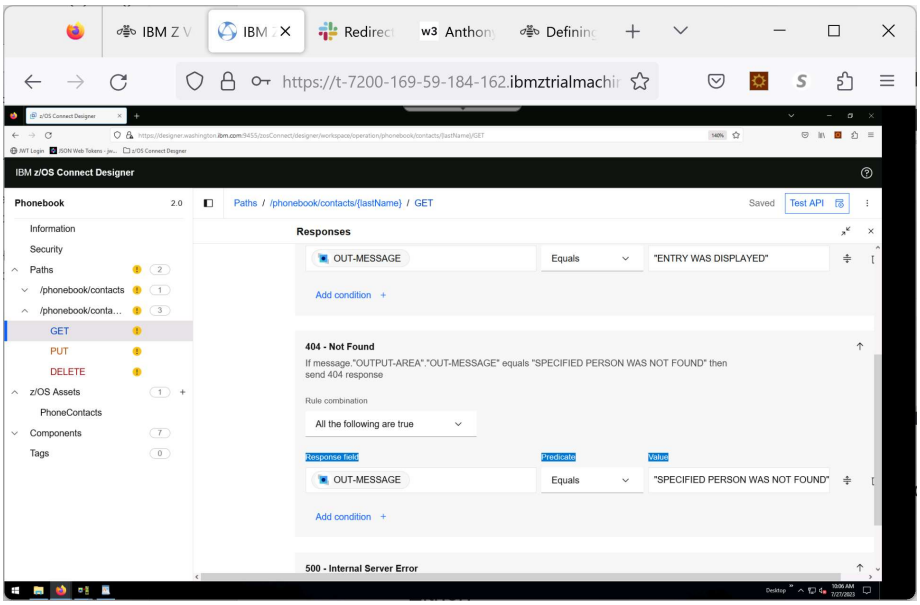
If done correctly, the ! mark associate with the 200- OK should disappear.



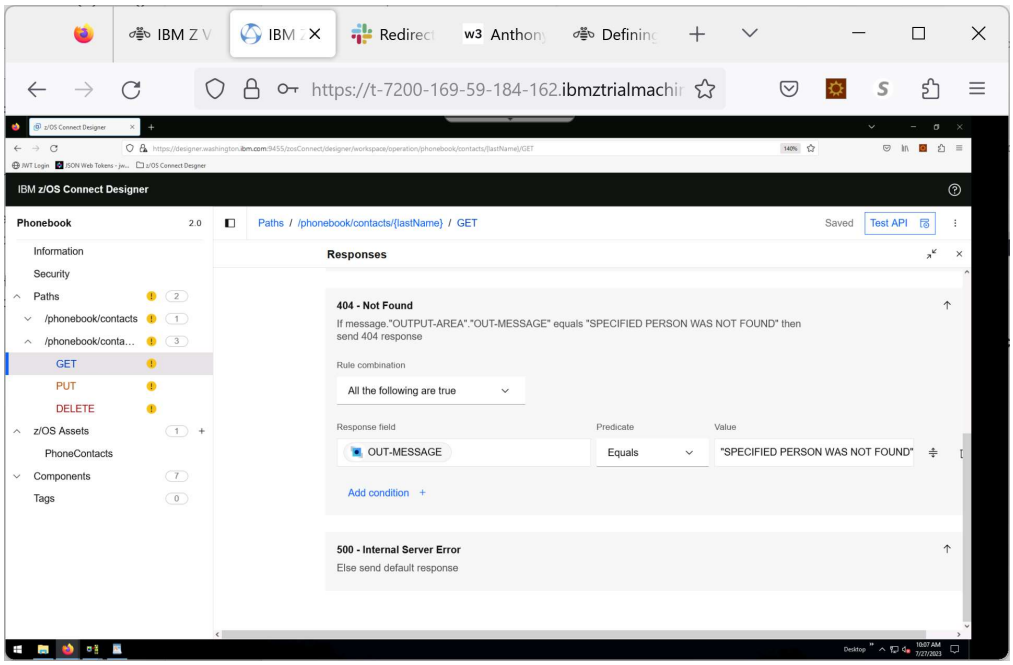
- Set the **404** response code condition – Not Found response.
- Either:
  - Copy and paste the condition from the 200 code
  - Or, once again type in:

$zosAssetResponse.message.”OUTPUT-AREA”.”OUT-MESSAGE”

- Also key in **“SPECIFIED PERSON WAS NOT FOUND”**in the **Value** field (note the double quotes).

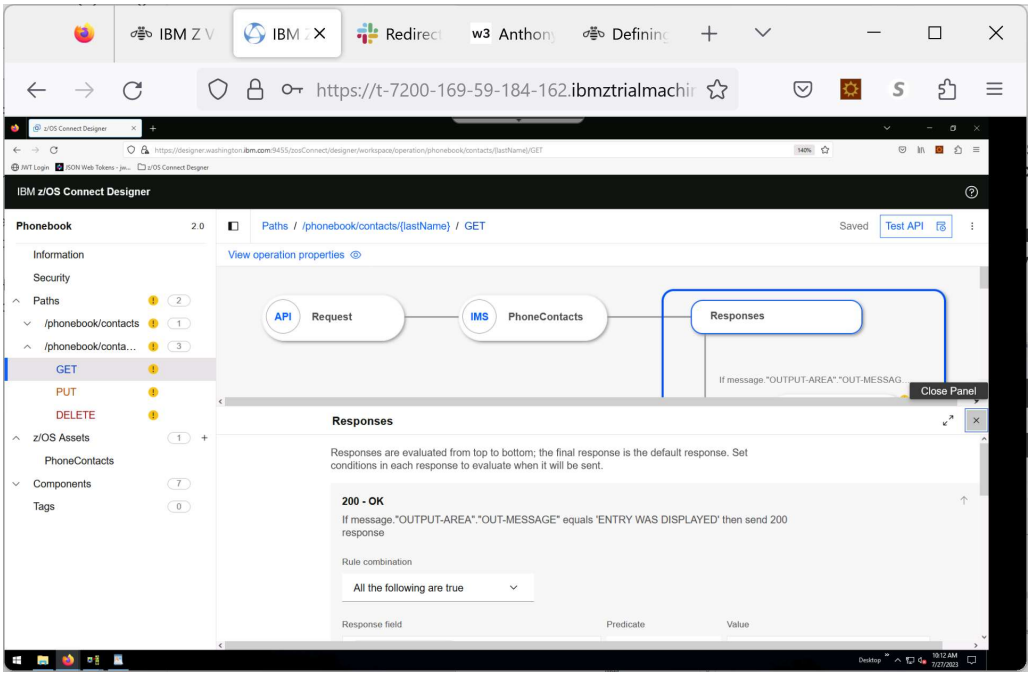


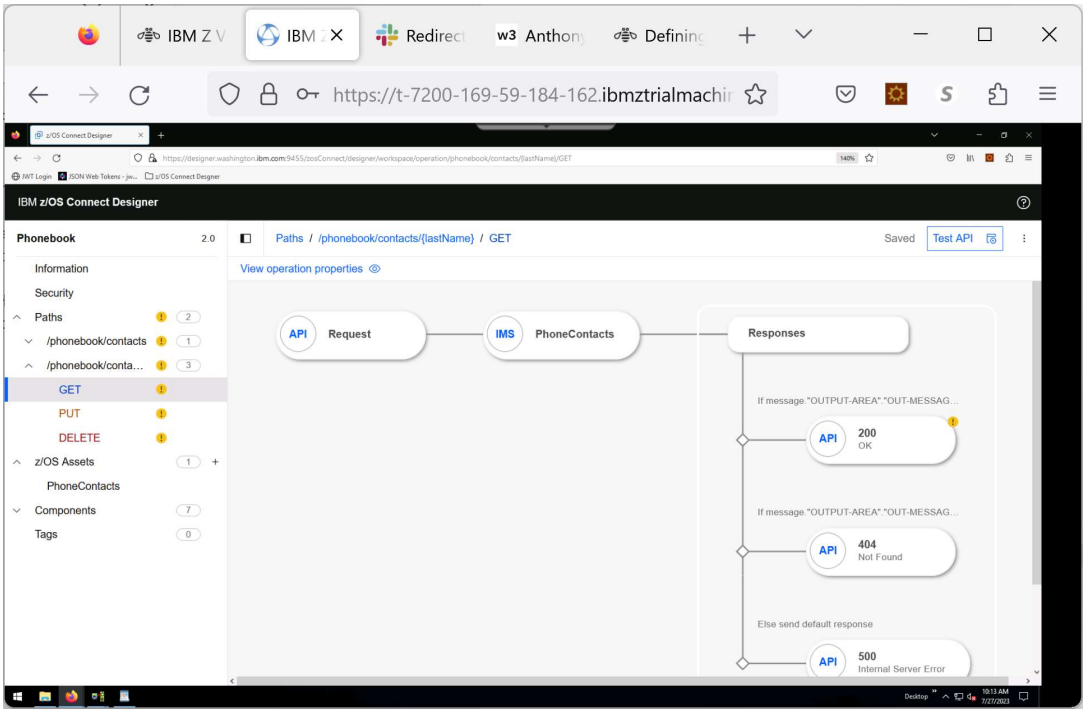
The 500 -Internal server error response is the default, so it has no conditions and must be the last entry in the table.



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.

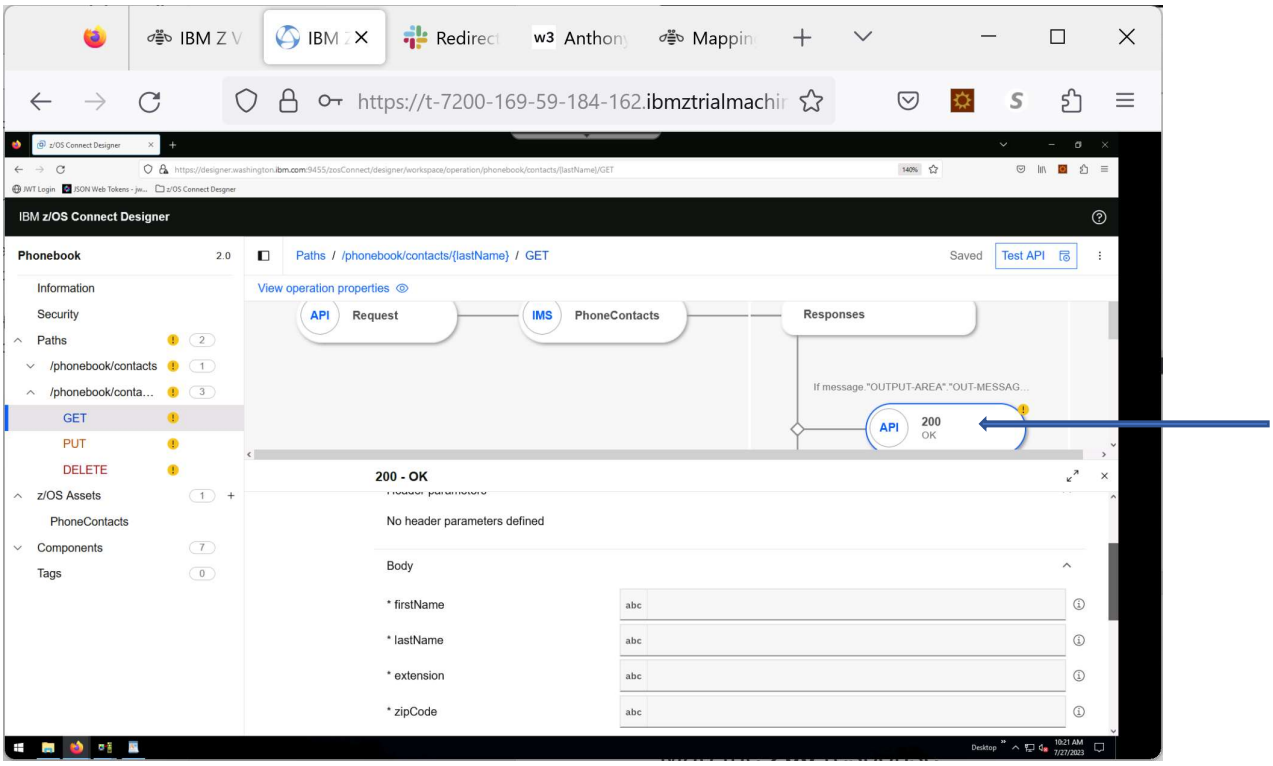




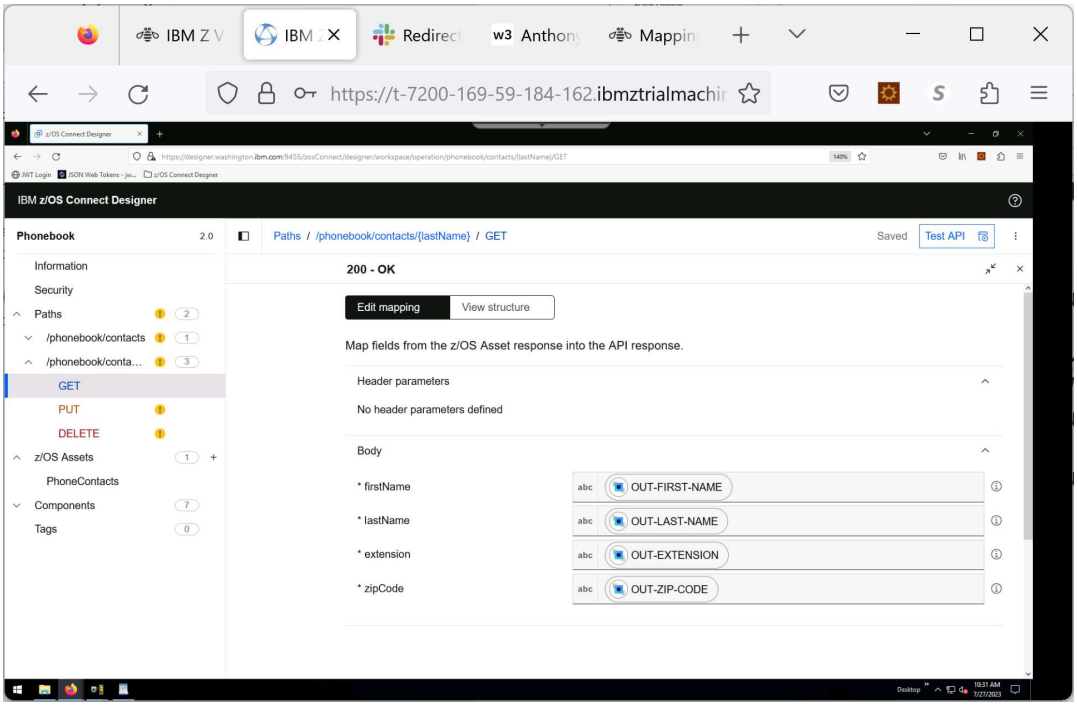
Map the 200 response.

- 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.

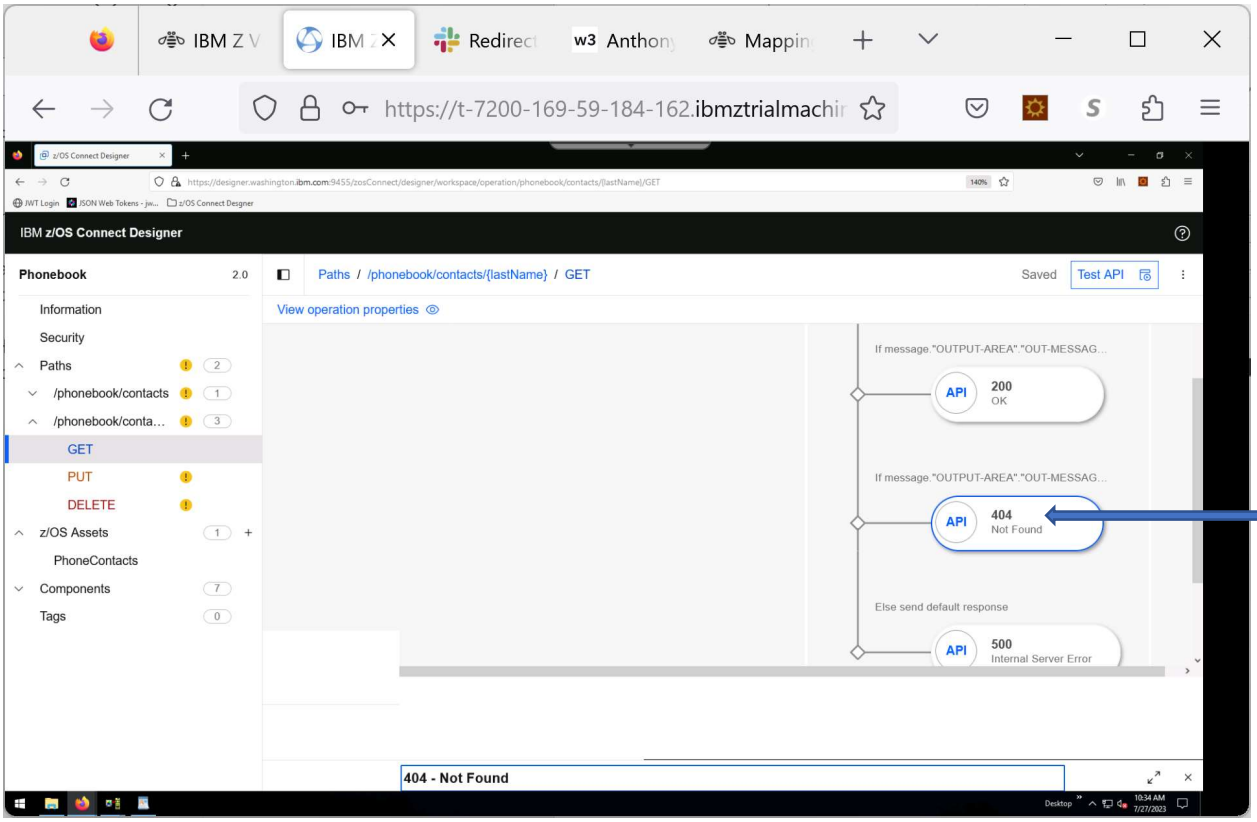


- Map the z/OS Asset response input **OUT-FIRST-NAME** to the firstName API response field.
    - ***Start keying in OUT-FIRST-NAME, the tool will allow you to select the appropriate field.***
- Map the z/OS Asset response input **OUT-LAST-NAME** to the lastName API response field.
- Map the z/OS Asset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



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

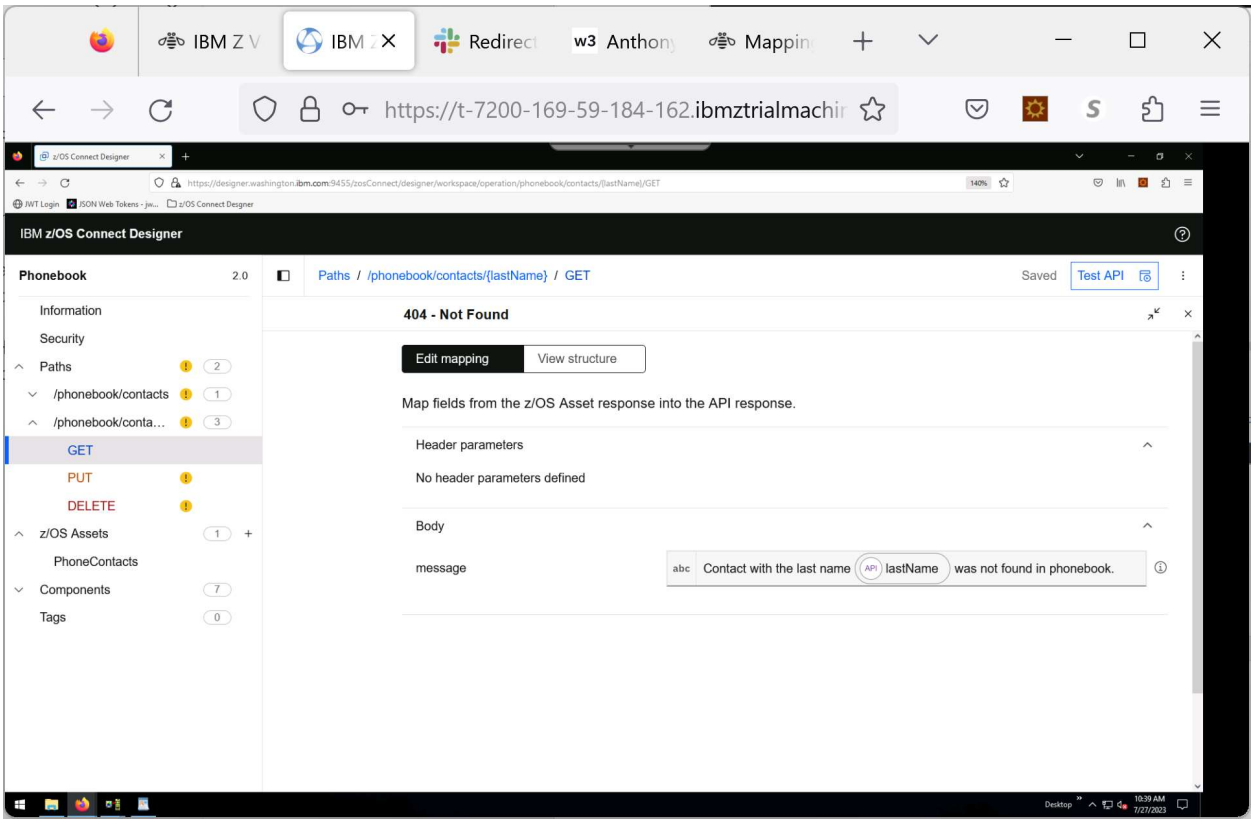
Map the 404 response.



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.

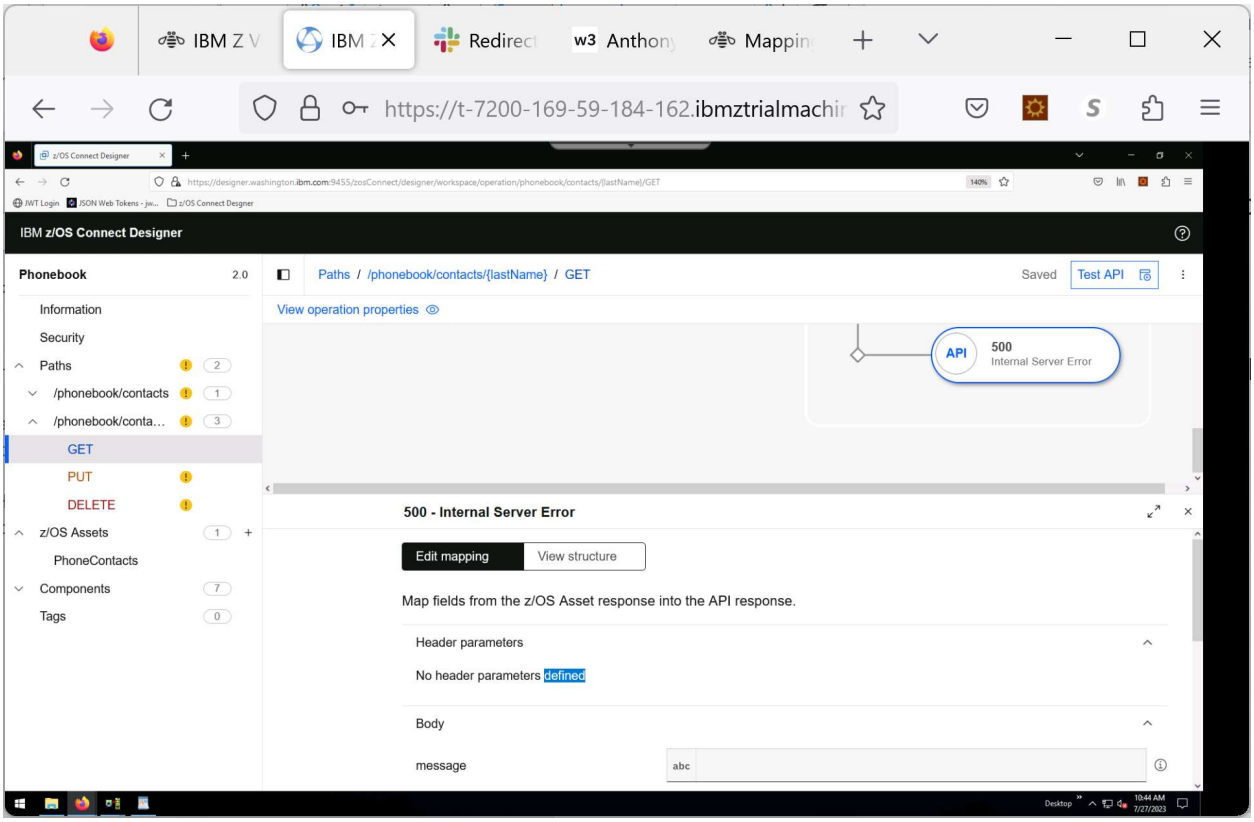


Note how the tooling pulls the lastName from the structure.

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

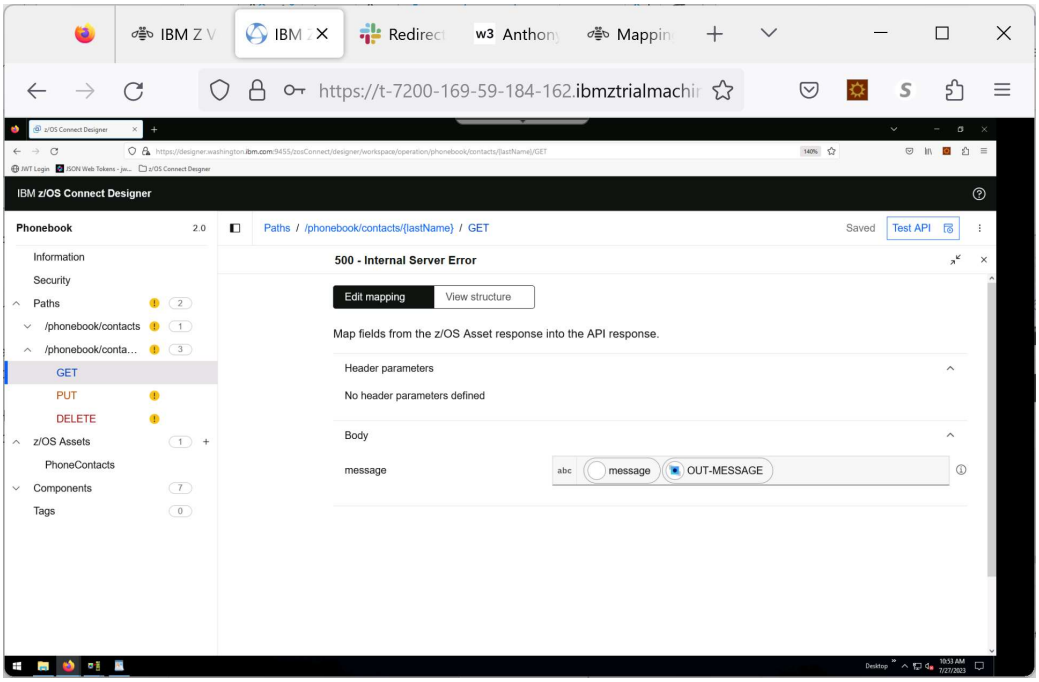
Map the 500 response.

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



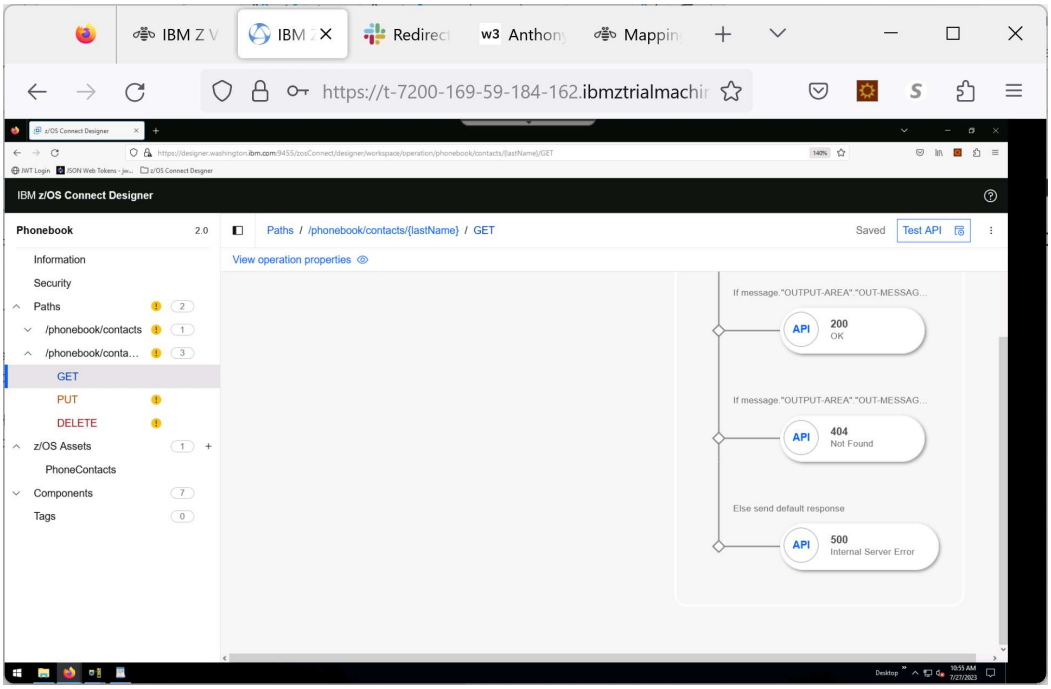
A 500 response code indicates an internal server error. Configure the 500 response to return the z/OS Connect error message by typing the following into the message field (Note the case, brackets, and quotes):

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



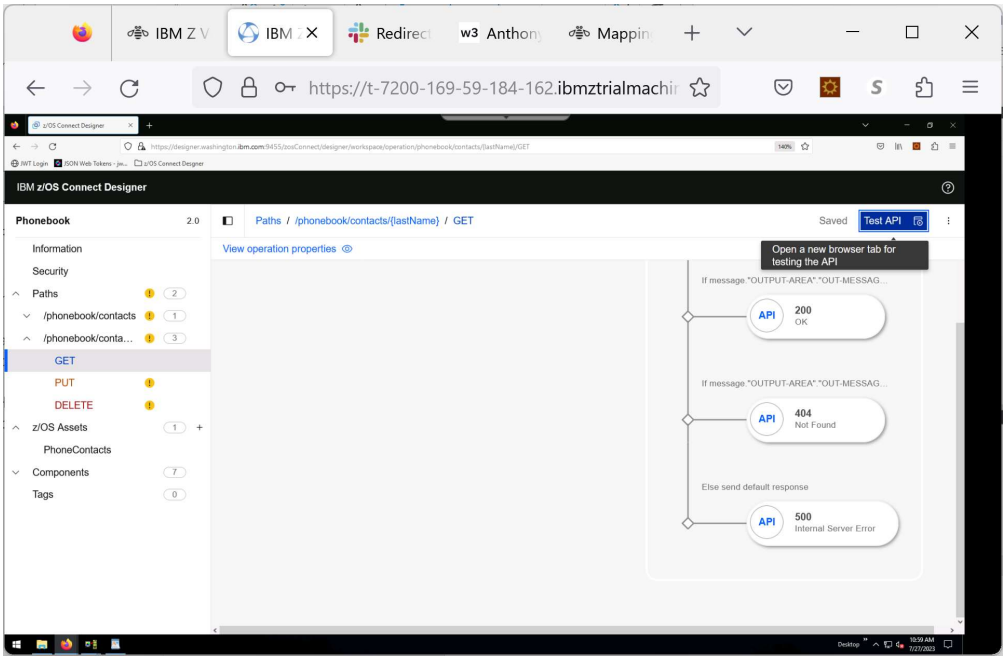
- Click **X** at the top right of this panel to close it and return to the Operations flow diagram.

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.

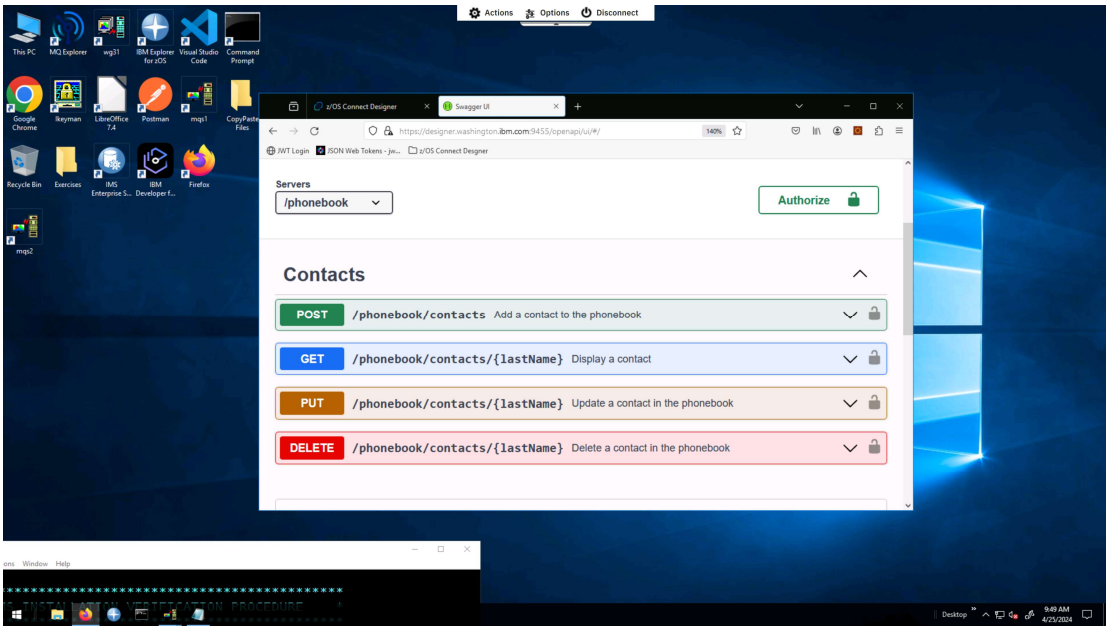


Test the API GET method.

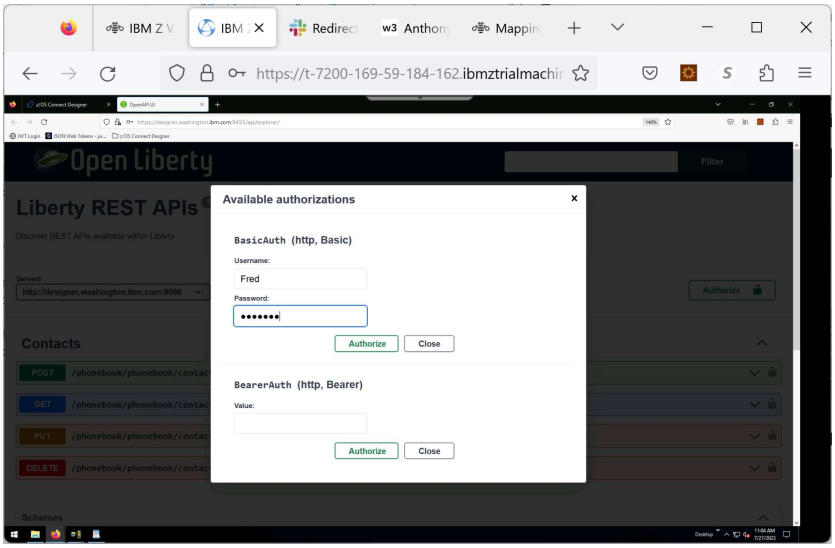
- Click on the **TEST API** button on the top right of the operations diagram.



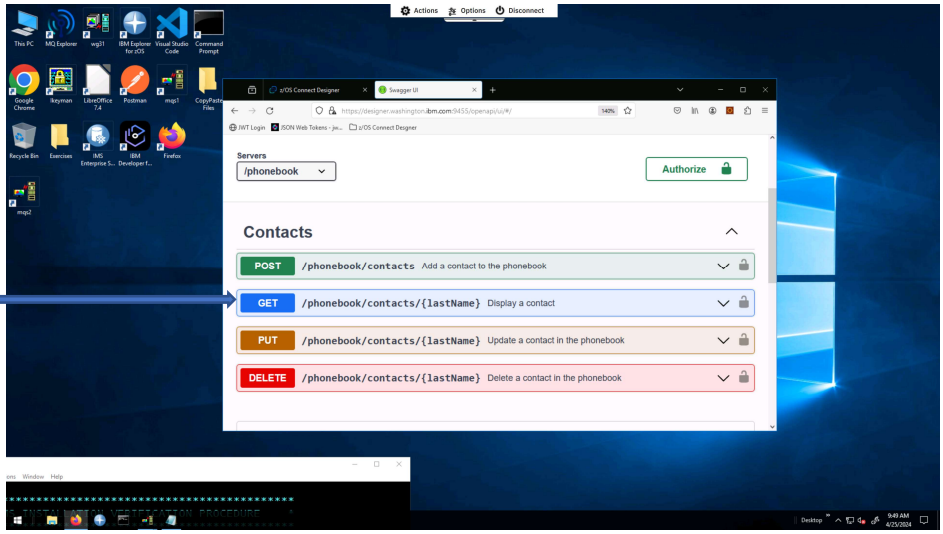
- On the **Servers** drop down, select **/phonebook**
- Click on **Authorize**



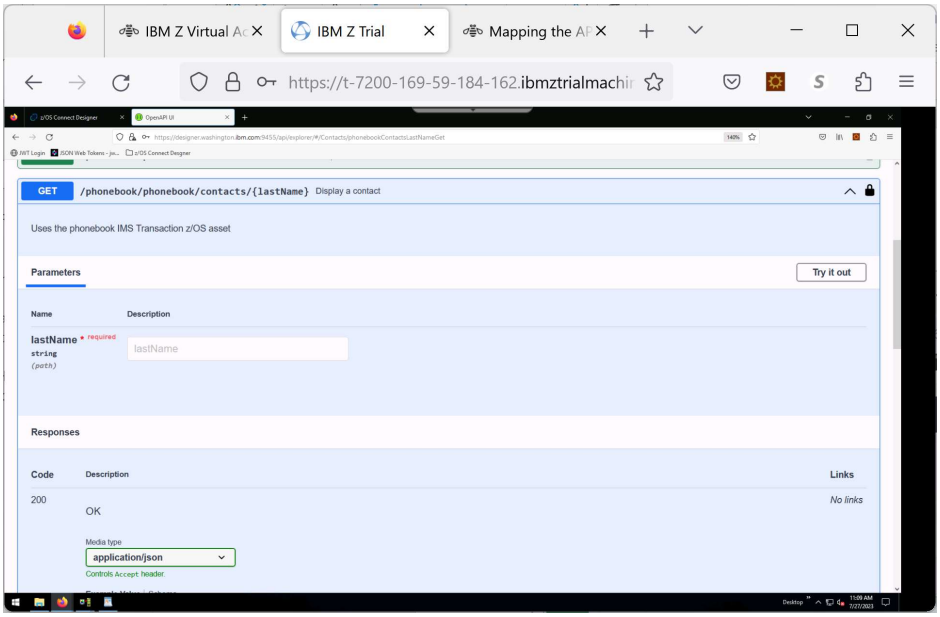
- Click the **Authorize** button and choose BasicAuth.
    - Two usernames (user1 and Fred) are available. You can use either one
    - Otherwise, if the username is blank
        - Key in **Fred** (note the capital **F**) for the Username
        - Key in **fredpwd** (Note all lowercase) for the password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



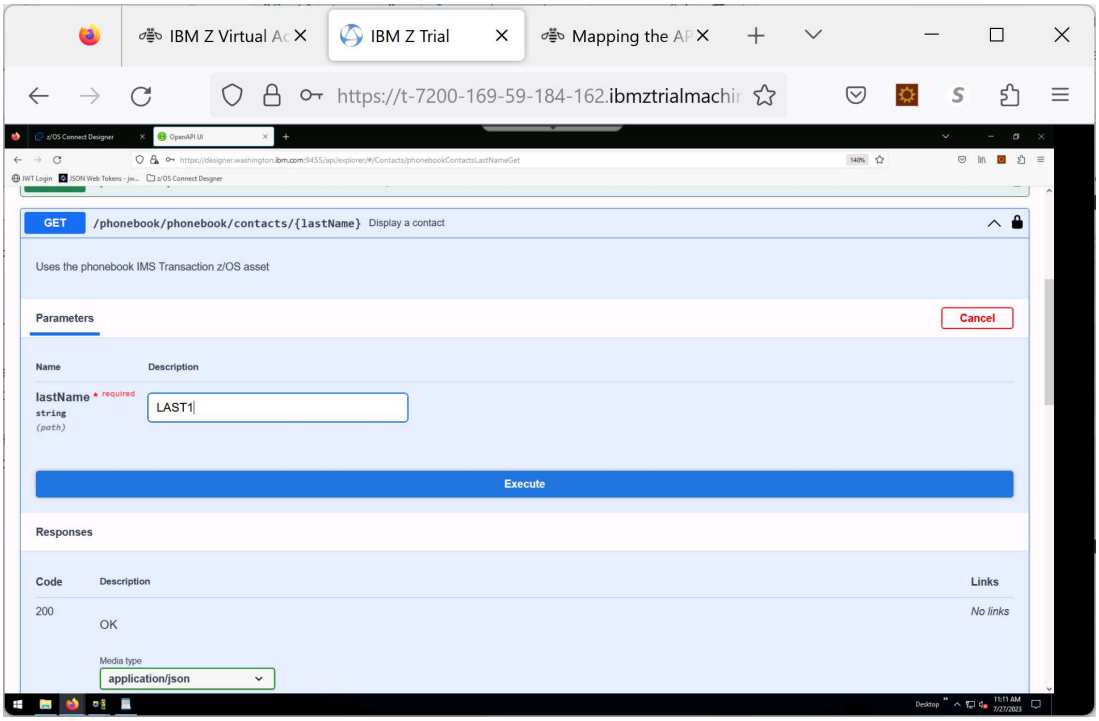
Back on the main panel, you will see all the possible methods that can be used for this API. The only one configured at this point is the **GET** method so it is the only one that can be tested.



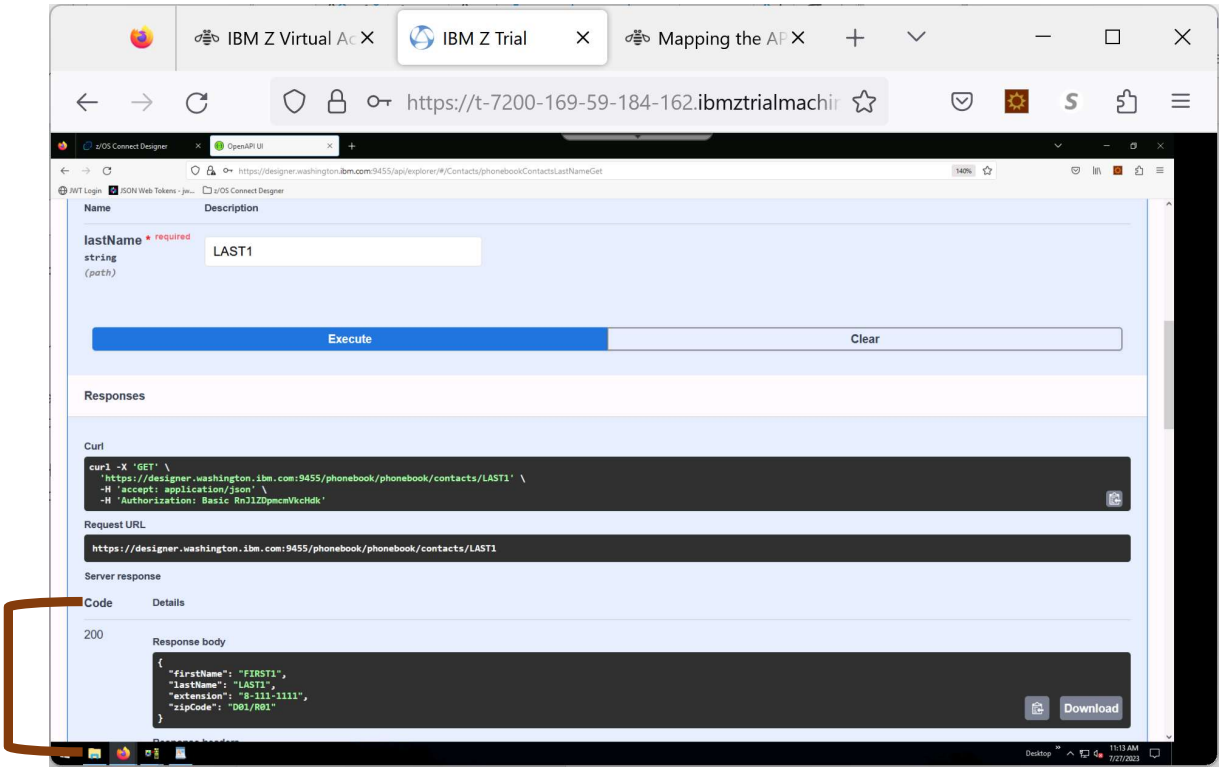
- Click **GET**



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



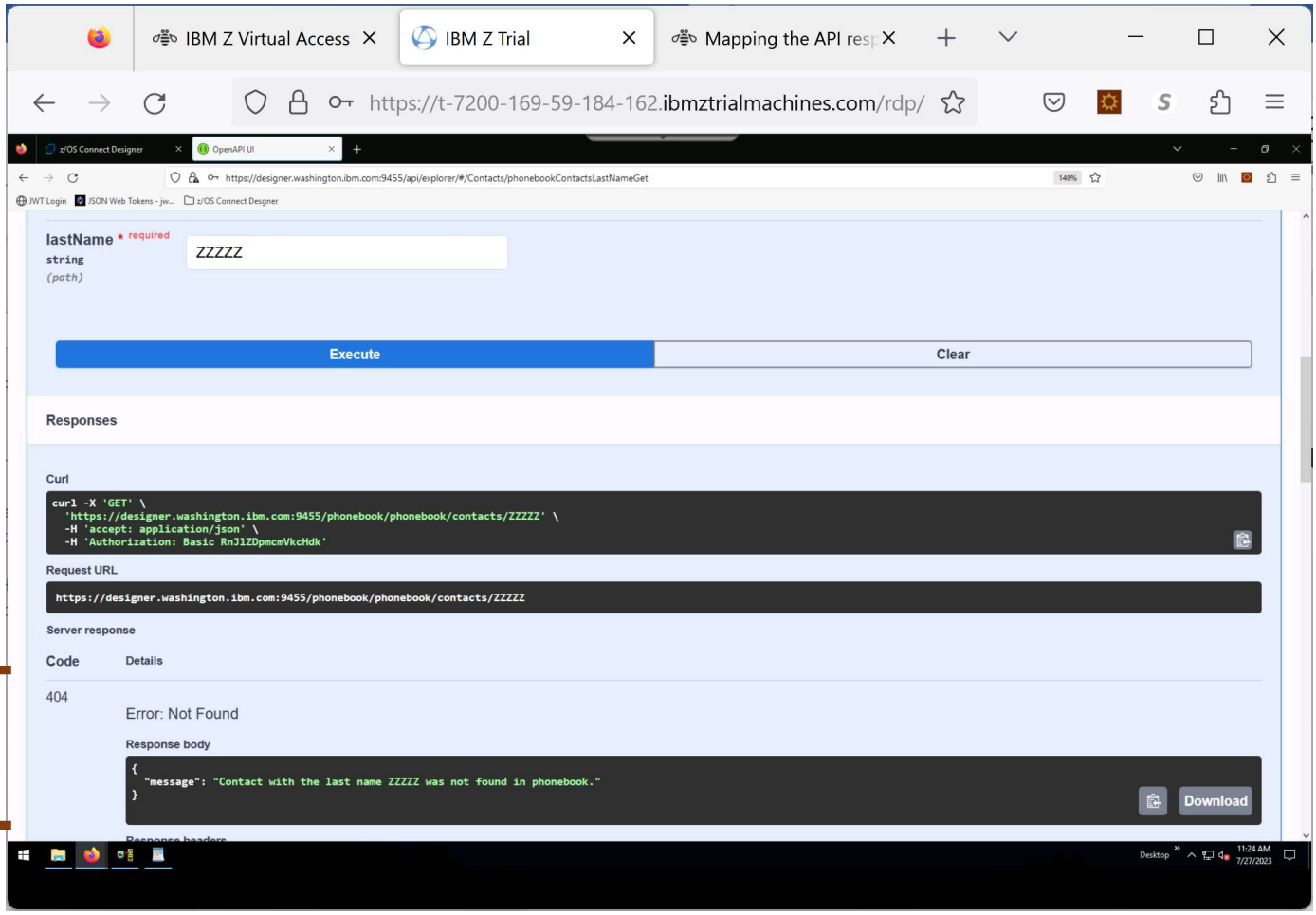
You should see the following:



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

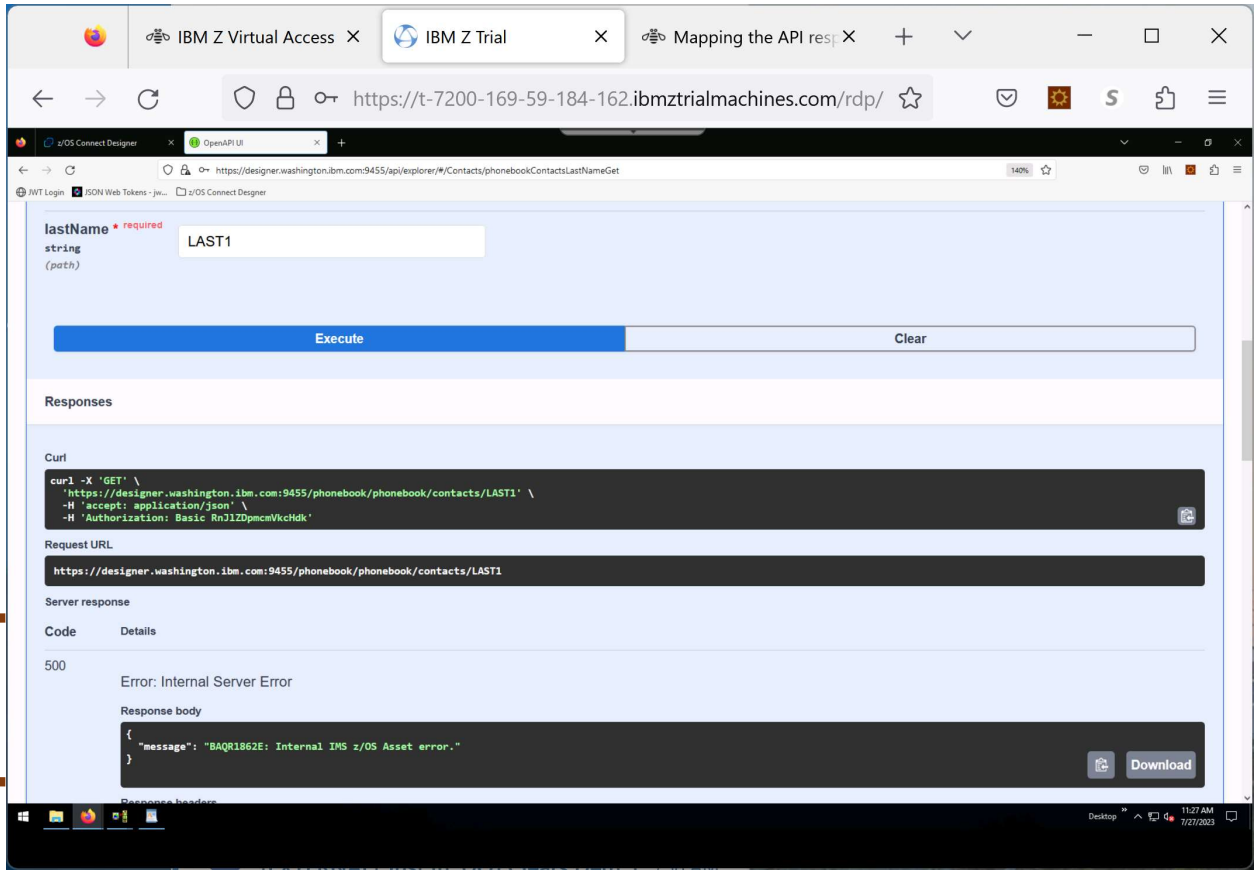
Note: after you complete Part 2 (POST) where you add your name into the phone book, you can return to this GET method to see what you successfully added.

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



You should see the **404** error message

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



Congratulations! You have completed the exercise for GET method.