

IMS Hands-On Lab

z/OS Connect and IMS OpenAPI 3 - PART 2

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

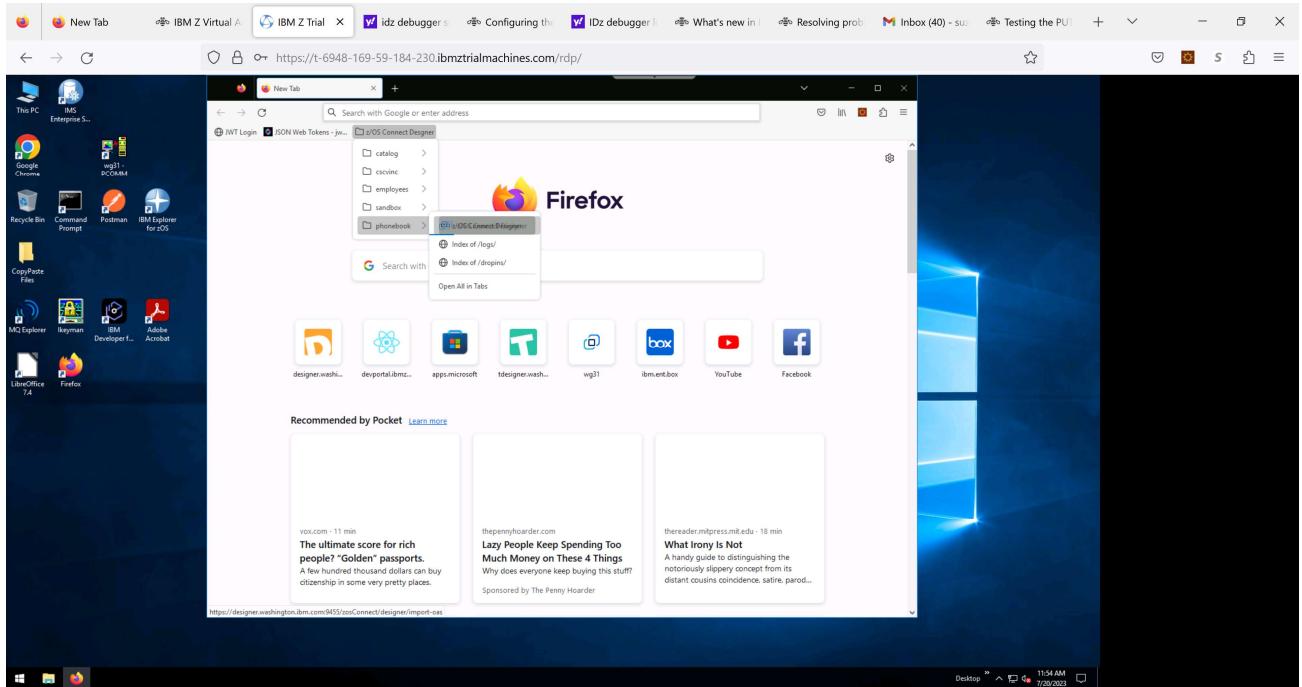
PART 3 – Create the API to UPDATE the contact you added

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

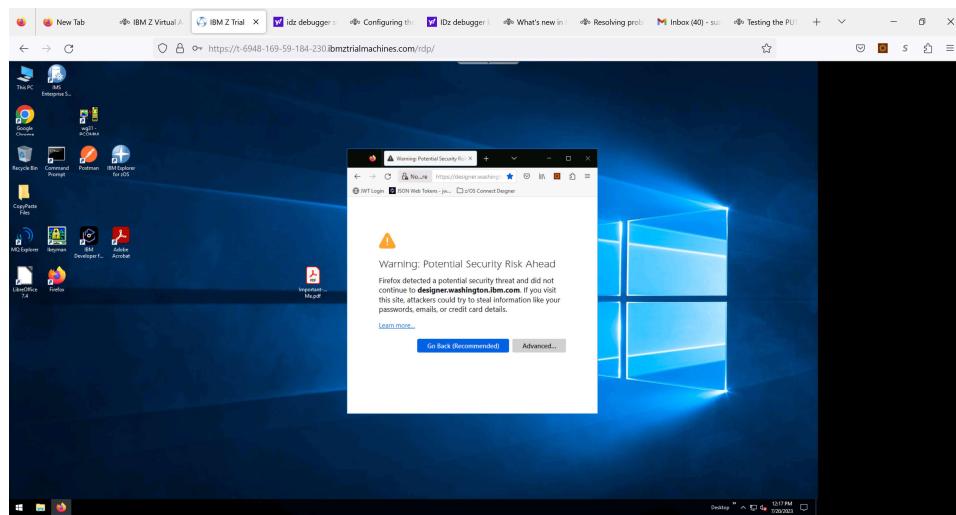
This lab is a continuation of the GET method lab for the IMS IVTNO transaction (Phonebook). It assumes you have completed Part 1 and have already imported the YAML file and created the z/OS asset.

Start the Designer...If not already done, 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)

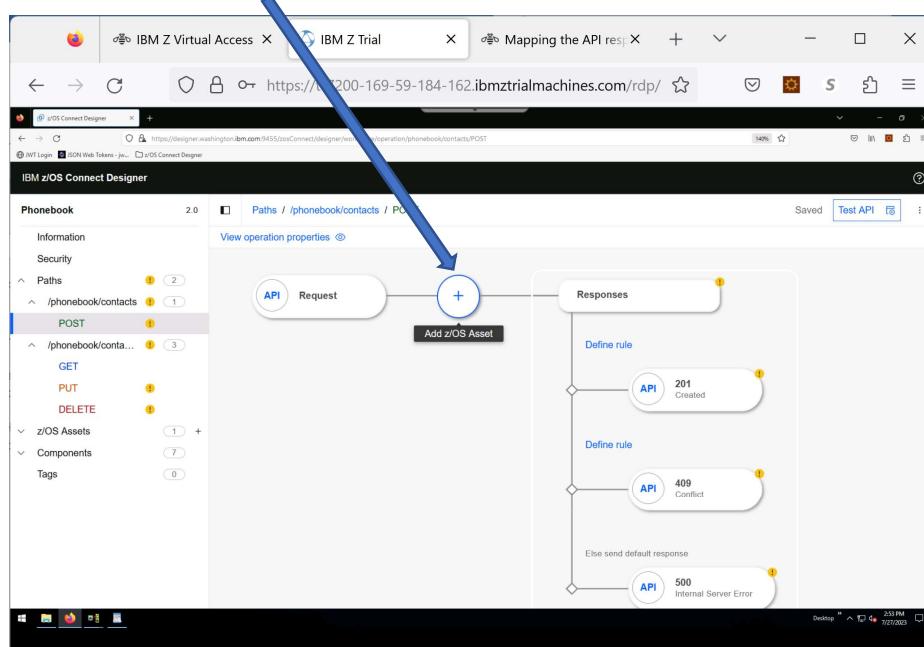


Since the **GET** method configuration was completed in the previous exercise, Note that the exclamation mark by the GET is no longer there.

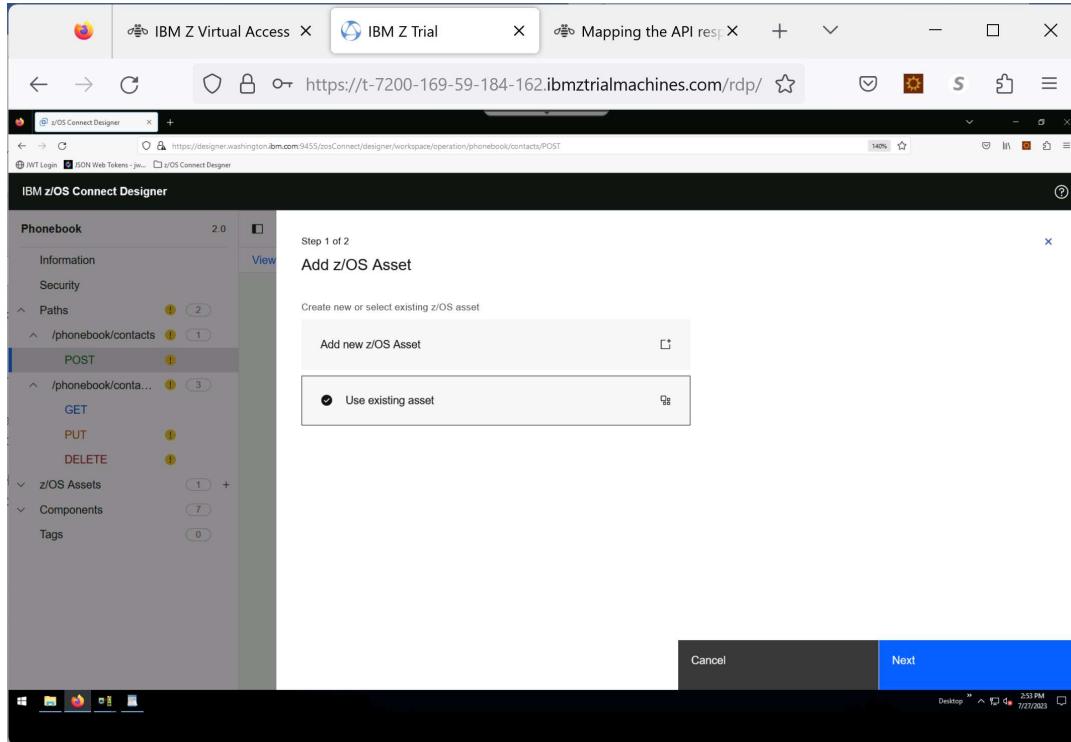
This exercise uses the **POST** method to add a new contact into the phonebook.

The screenshot shows the IBM z/OS Connect Designer interface. On the left, a sidebar lists API paths: Phonebook (2.0), Information, Security, Paths, /phonebook/contacts (1), POST (1), /phonebook/conta... (3), GET, PUT, DELETE, z/OS Assets, Components, Tags. A blue arrow points to the POST entry under /phonebook/contacts. The main panel displays the 'Information' tab for the POST method. It includes fields for Title (Phonebook), Version (2.0), Description (Manage phonebook contacts through an API for IMS), Contact (Name, Email, URL), License (Name: Apache-2.0, URL: https://opensource.org/licenses/Apache-2.0), and Servers (URL: /phonebook, Description). A 'Test API' button is at the top right.

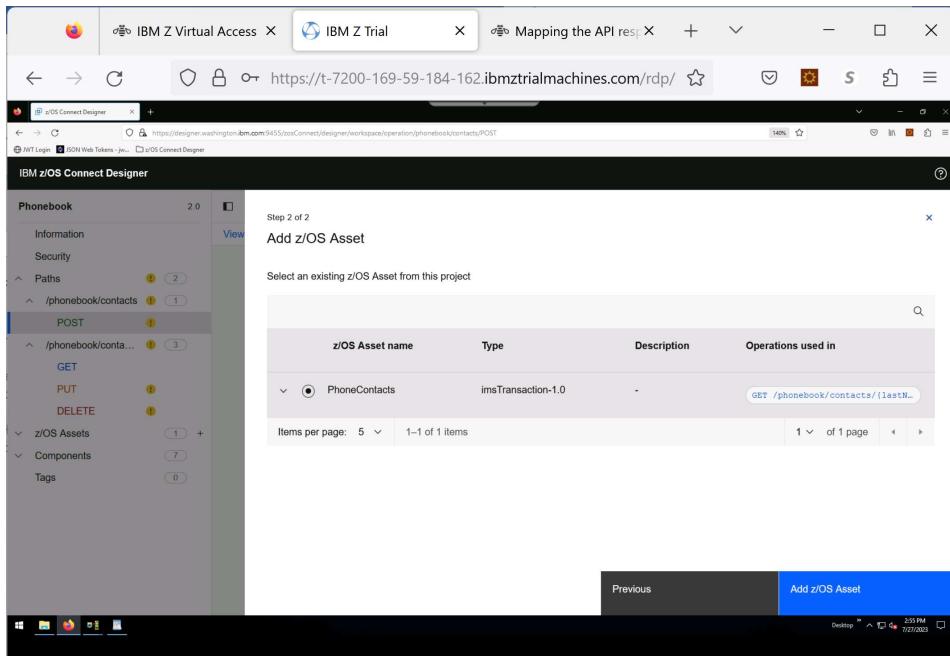
- Click on the **POST** method to start the process and bring up the Operations Flow Diagram.
- Click on the **+** (plus) to add a z/OS asset



Add the z/OS Asset and map the request fields

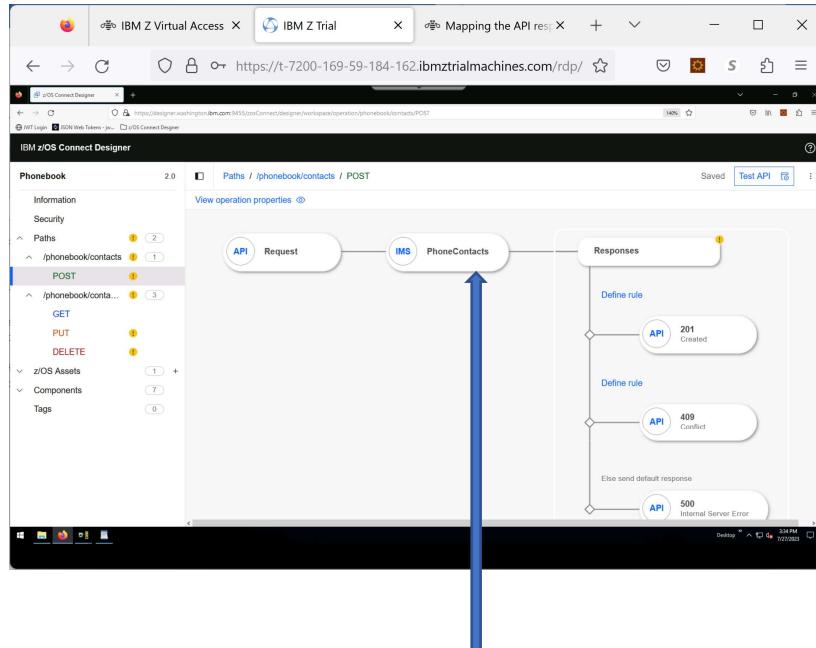


- Select **Use existing asset** and click **Next**.

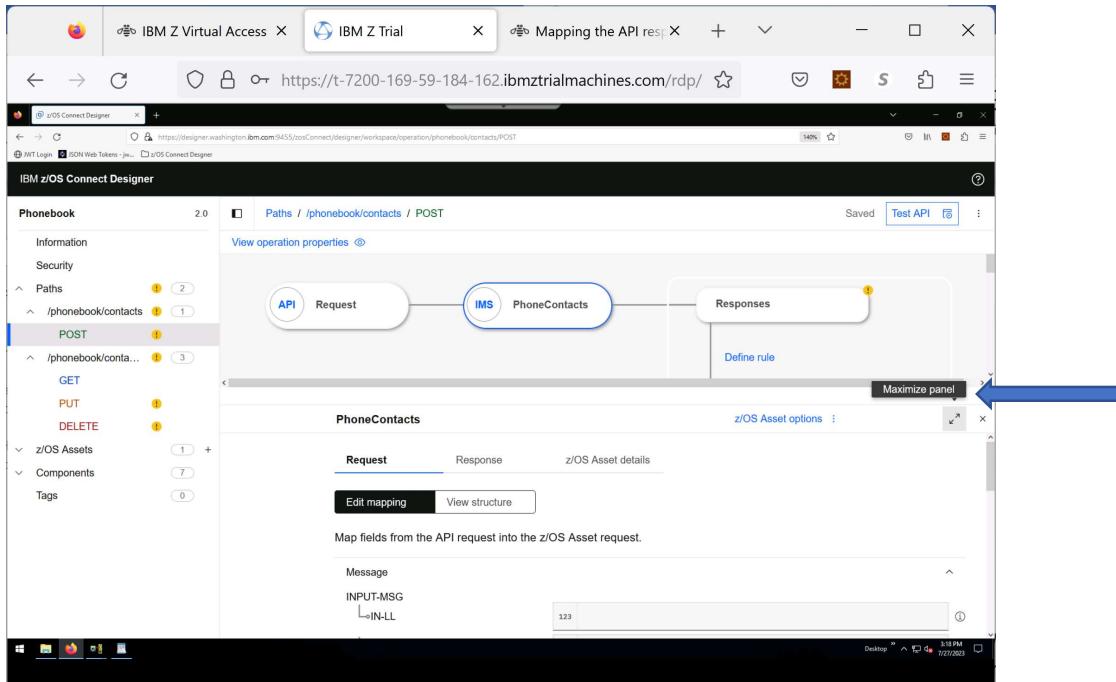


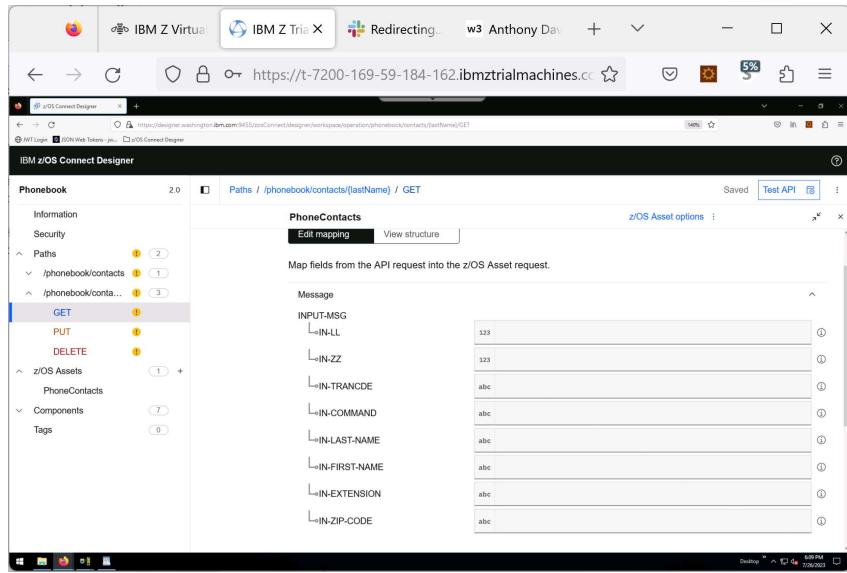
- Select **PhoneContacts** which was created in the first exercise.

- Click Add z/OS Asset

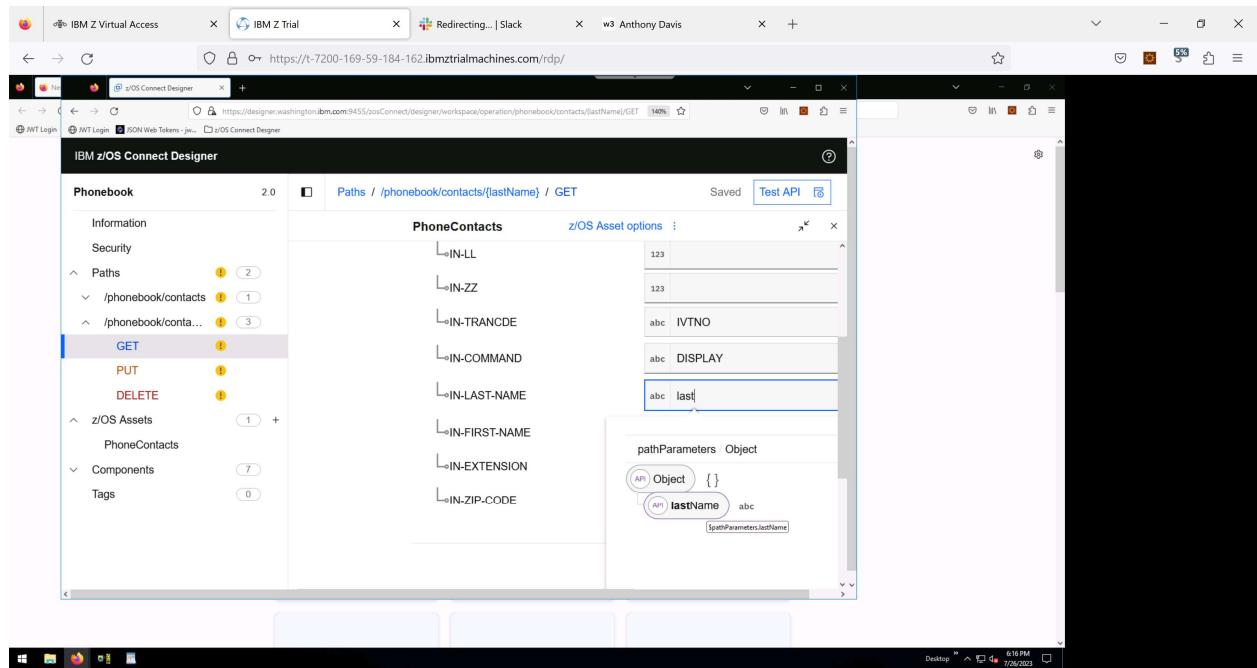


- When the mapping panel opens up
- Maximize the panel using the two arrows on the upper right

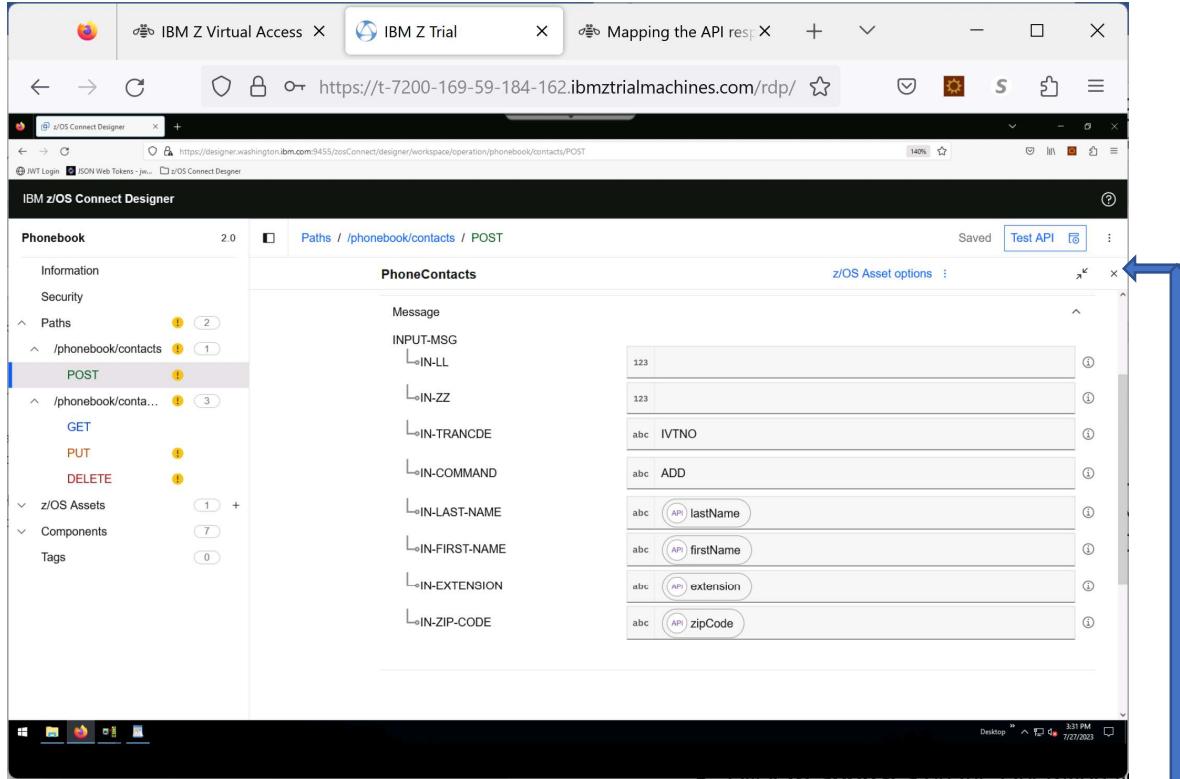




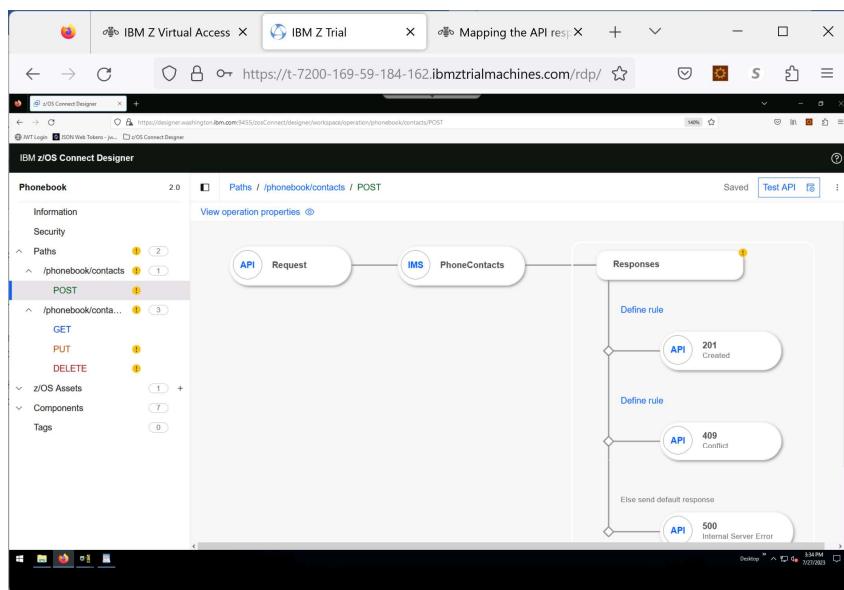
- Key in **IVTNO** (uppercase) in the **IN_TRANCDE** field
- Key in **ADD** (uppercase) in the **IN-COMMAND** field
- Map the API Request parameter **lastName** into the **IN-LAST-NAME** z/OS Asset Request field.
 - Key in **lastName** in the **IN-LAST-NAME** field. Note that when you start typing, the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



- Key in **firstName** in the IN-FIRST-NAME field.
- Key in **extension** in the IN-EXTENSION field.
- Key in **zipCode** in the IN-ZIP-CODE field.



- Minimize this panel (double arrows) or close it (x)on the upper right which displays the operation flow diagram.

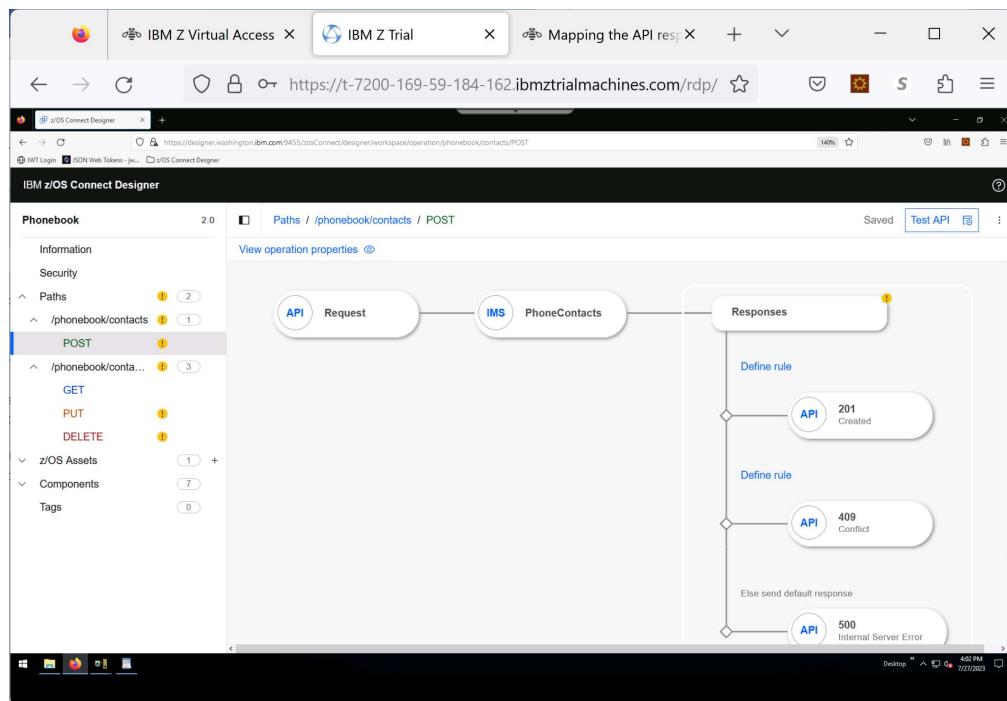


Map the API Response fields

- 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.

As a reminder, each response has the following properties:

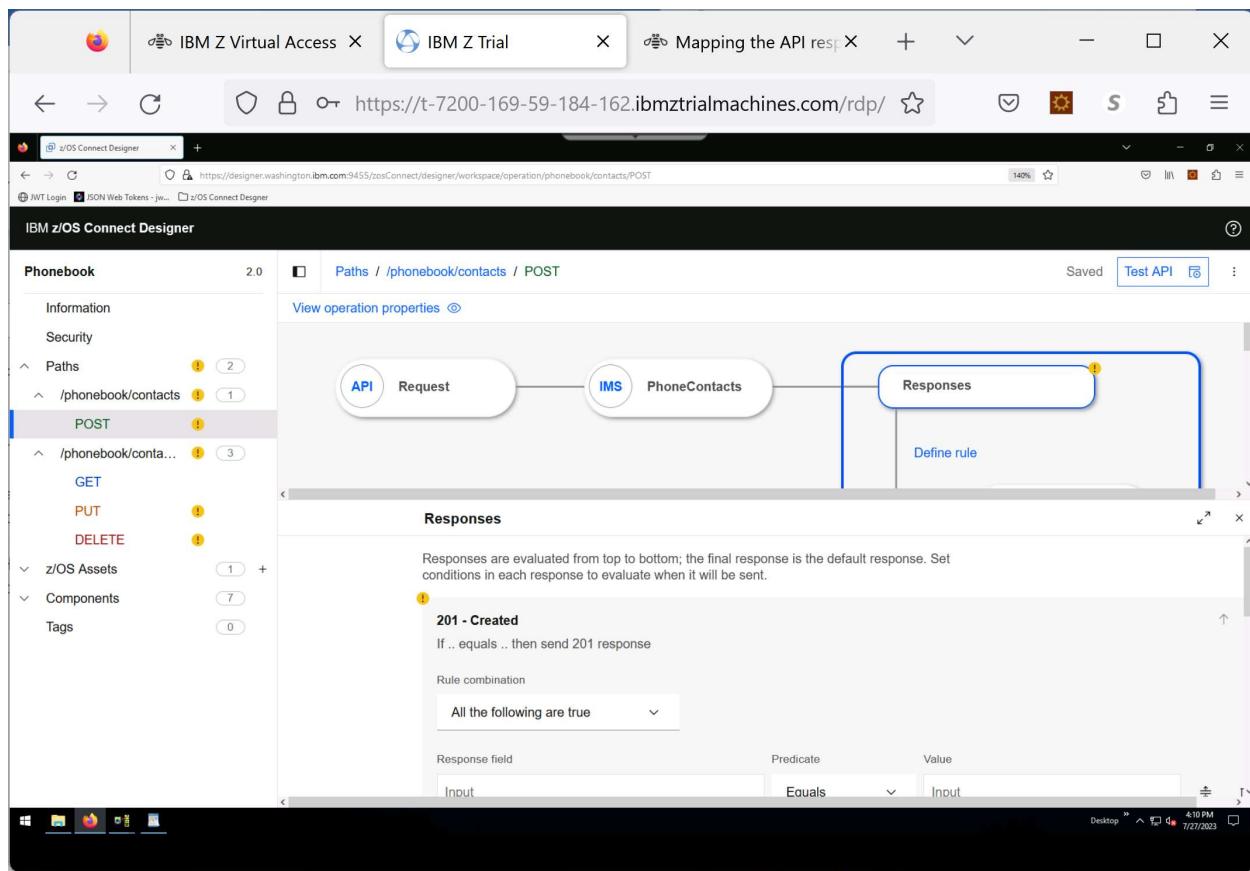
- A condition with three fields, *response*, *predicate*, and *value*.
- One or more *conditions*.
- You can change the order of the responses by using the ↑ and ↓ buttons next to each response case.
- The sequence of the conditions within a response can be changed. Click ↕ to change the position in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



The default order of the responses is such that **201 - Created** 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 **201 Created** response code condition
 - This code indicates that the requested contact and associated information were added to the phonebook database.
- Maximize the lower panel.



Set the conditions for the 201 Created 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.

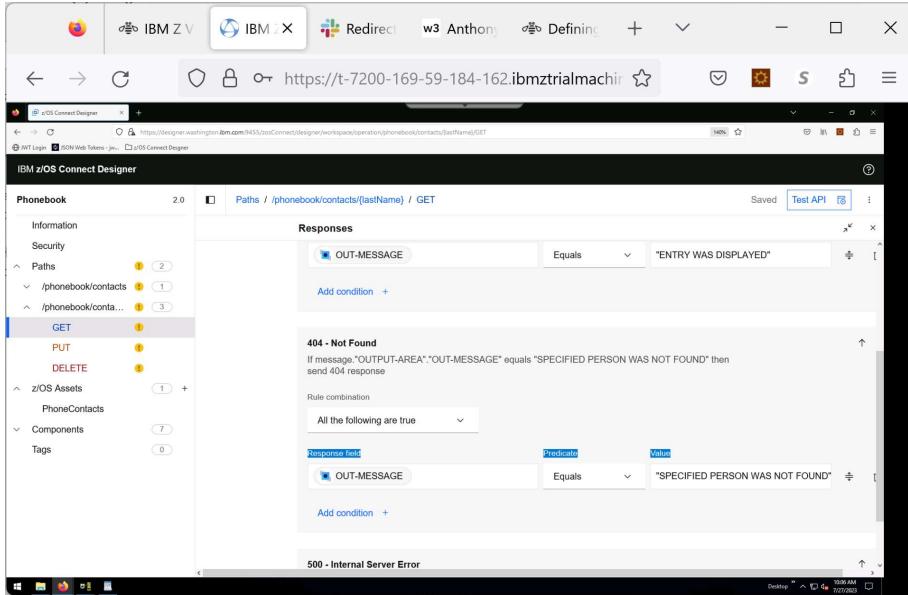
- Also key in “**ENTRY WAS ADDED**” (note the double quotes) in the corresponding Value field.

The screenshot shows the IBM z/OS Connect Designer interface. On the left, there's a sidebar with navigation links like 'Information', 'Security', 'Paths', and 'Components'. Under 'Paths', there's a section for 'POST' operations under '/phonebook/contacts'. The main area displays two response rules:

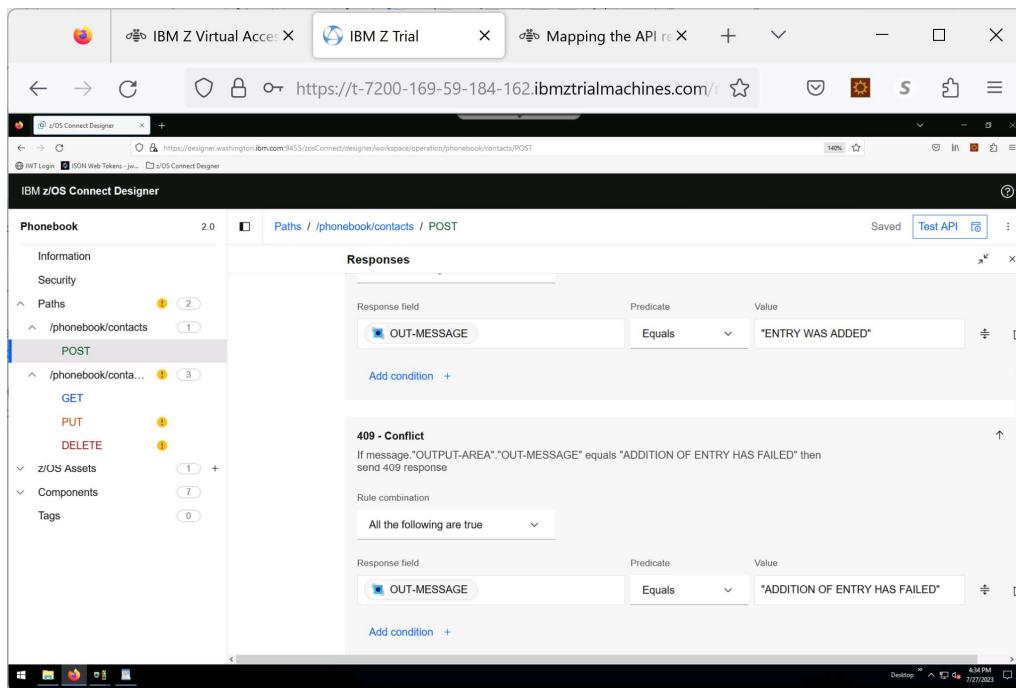
- 201 - Created**: A condition 'All the following are true' is set for the 'OUT-MESSAGE' field, which is checked for 'EQUALS' to the value '"ENTRY WAS ADDED"'.
- 409 - Conflict**: A condition 'All the following are true' is set for the 'OUT-MESSAGE' field, which is checked for 'EQUALS' to the value '"ADDITION OF ENTRY HAS FAILED"'.

If done correctly, the mark associate with the 201- Created should disappear.

- Set the **409 - Conflict** response code condition
- Either:
 - Copy and paste the condition from the 201 code
 - Or, once again type in:
\$zosAssetResponse.message."OUTPUT-AREA"."OUT-MESSAGE"
- Also key in “**ADDITION OF ENTRY HAS FAILED**” in the **Value** field.

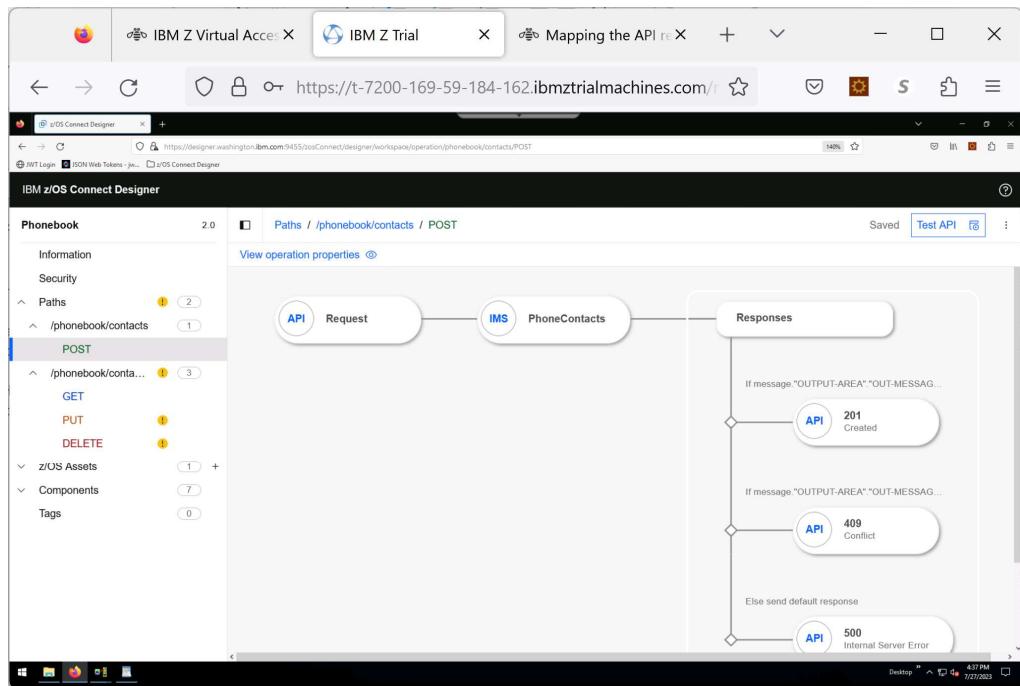
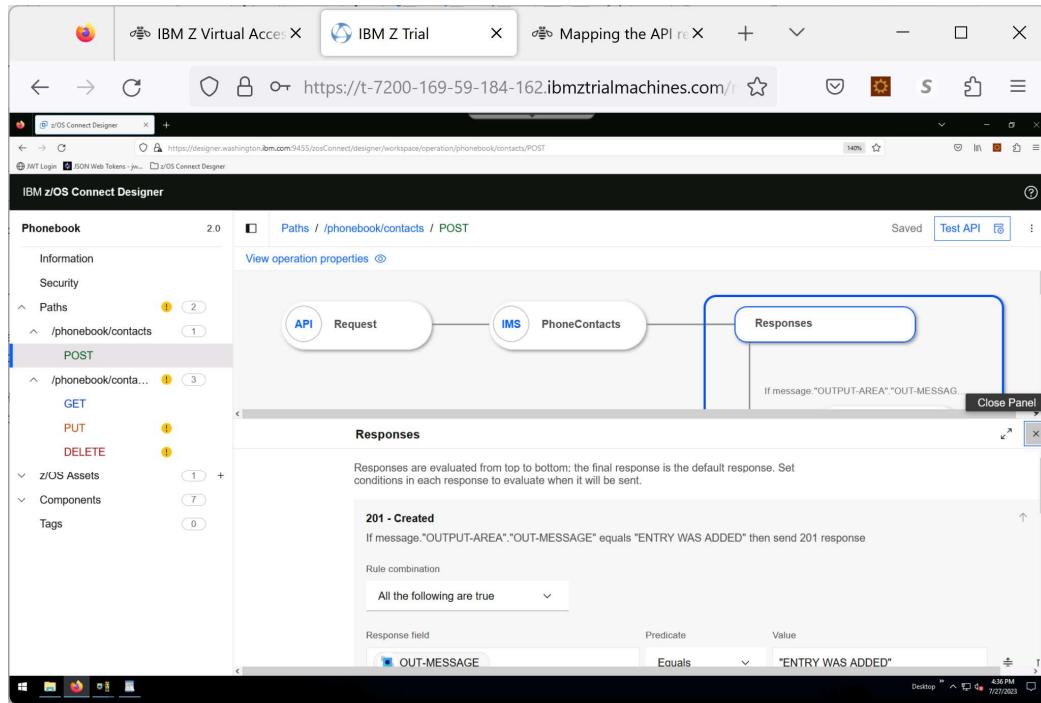


The 500 -Internal server error response is the default, so it has no conditions and must be the last entry in the table.



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.

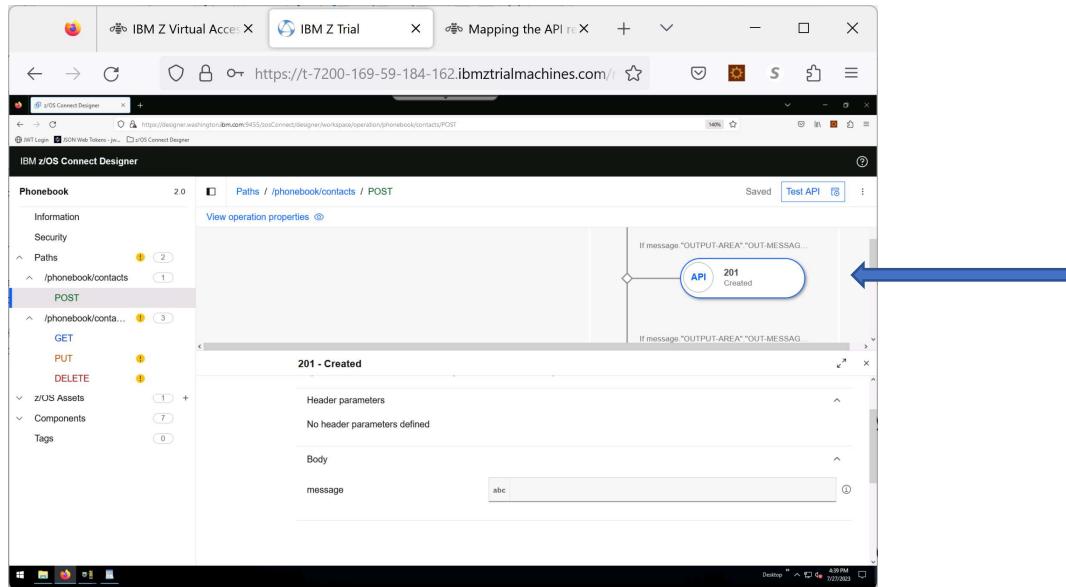
- Minimize the panel you have been working on by clicking the double arrows at the top right to get back to the primary window. You can also close the panel.



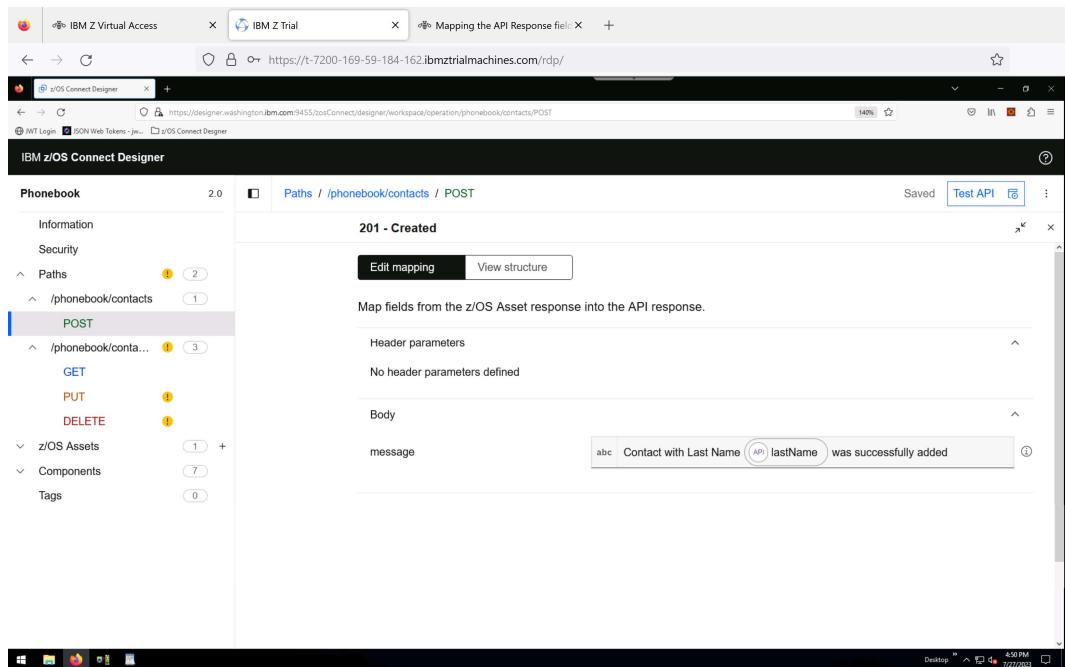
Map the 201 response.

- In the Operation flow diagram, click the 201 response node.

A 201 response code indicates that the addition of the requested phonebook contact was successfully completed.

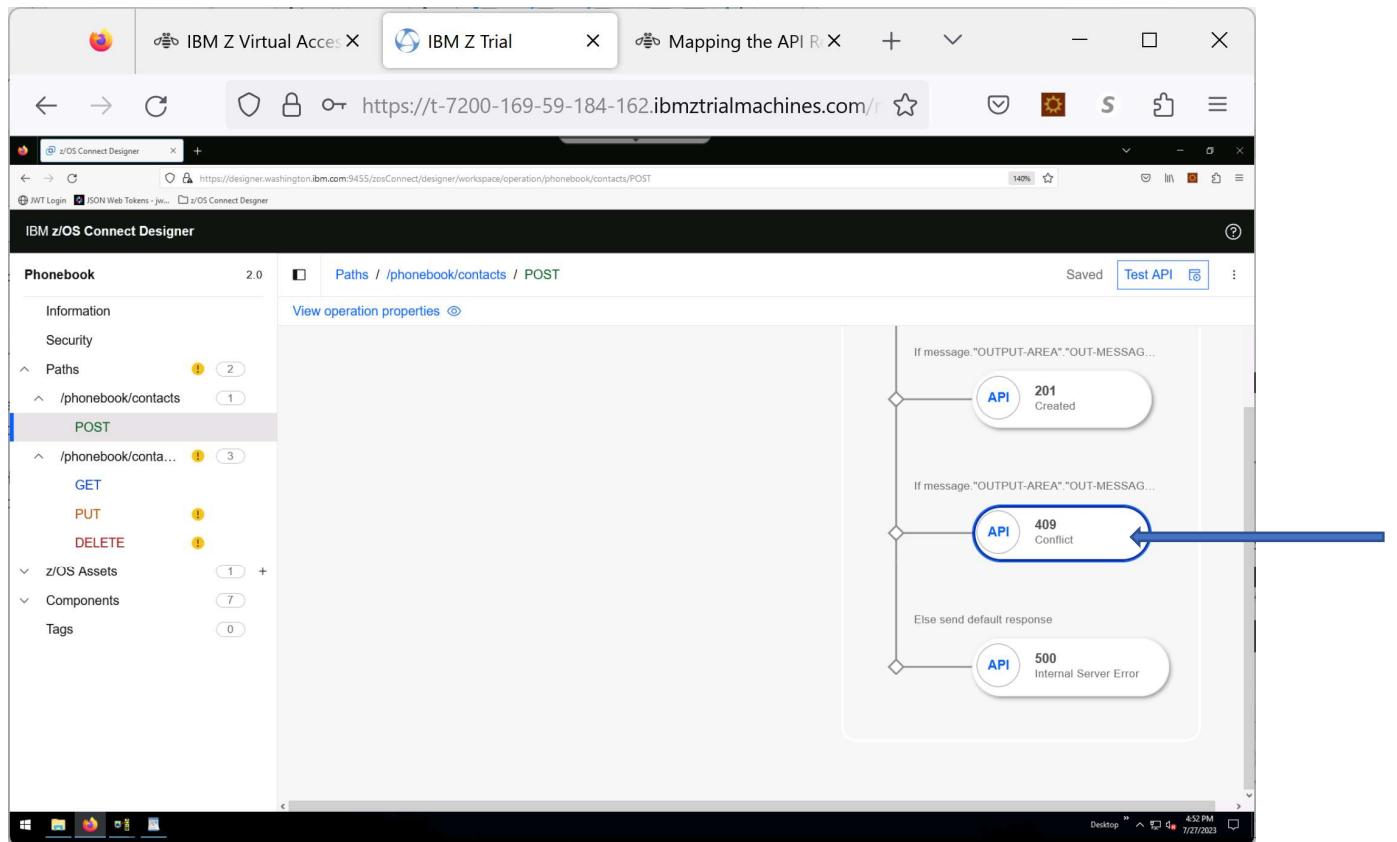


- For a successful operation, all that is returned by the application is a message.
 - Key in:
 - **Contact with Last Name {{apiRequest.pathParameters.lastName}} was successfully added.**



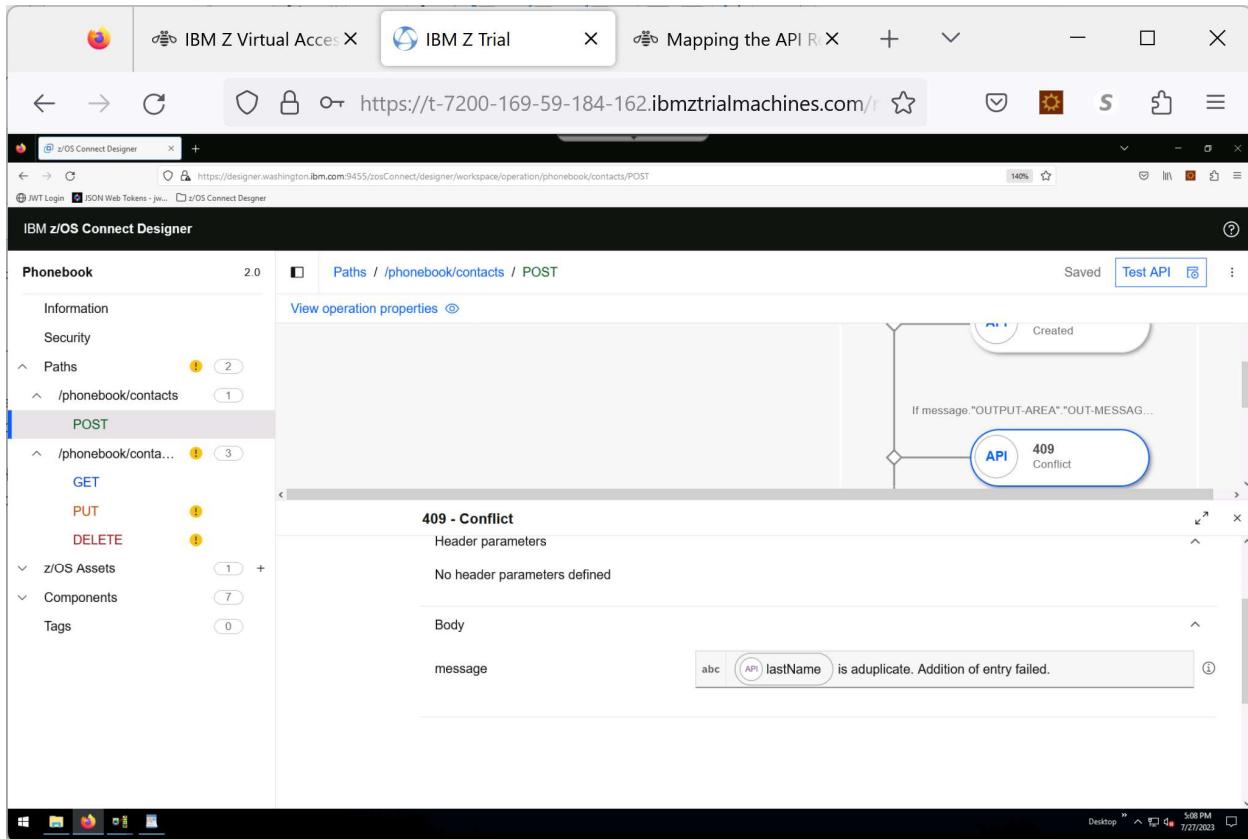
- Minimize the panel to go back to the Operation flow diagram

Map the 409 response.



Configure the 409 response to return a message to explain that the contact could not be added. The conflict is because the entry is already there.

- Key the following into the **message** field (be aware of case, and brackets):
 - **{{apiRequest.pathParameters.lastName}}** is a duplicate. Addition of entry failed.

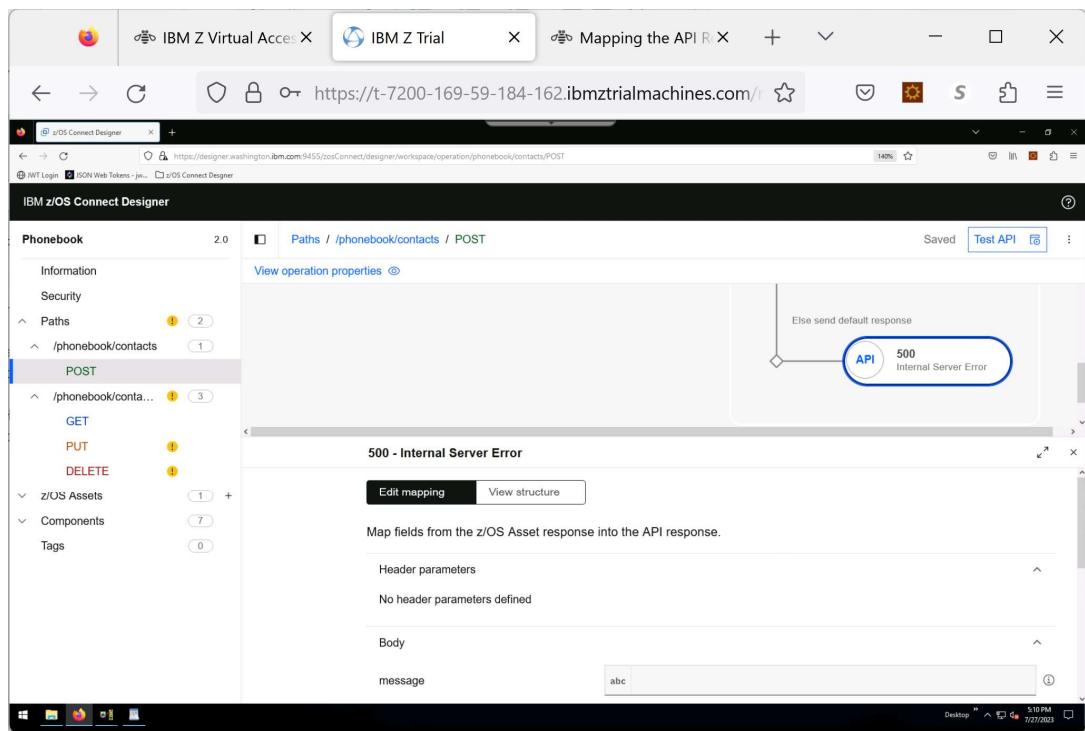


Note how the tooling pulls the lastName from the structure.

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

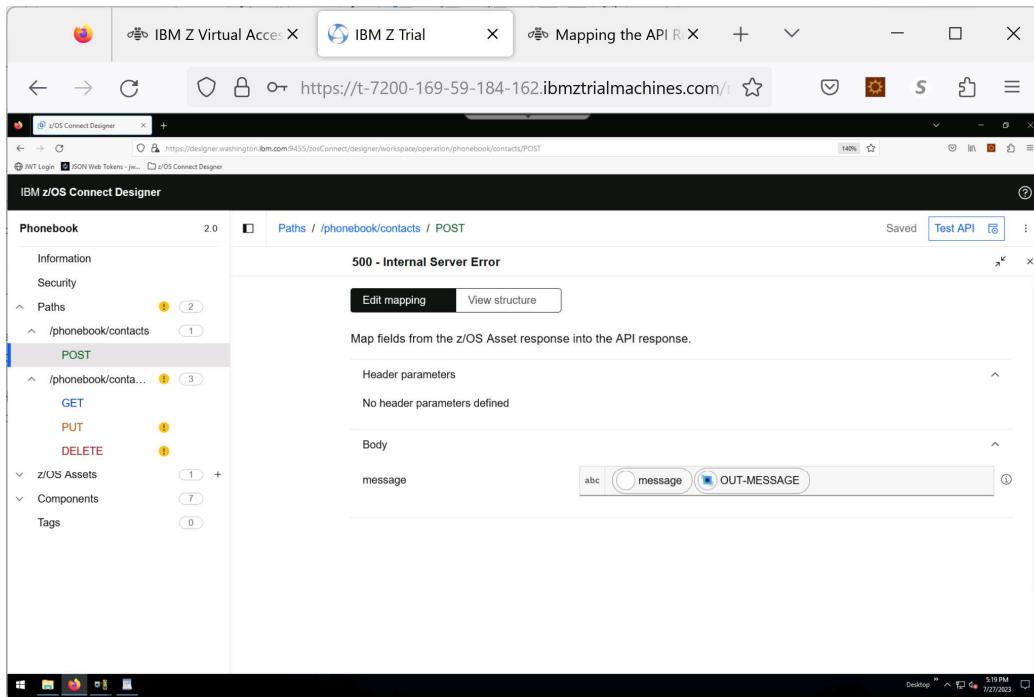
Map the 500 response.

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



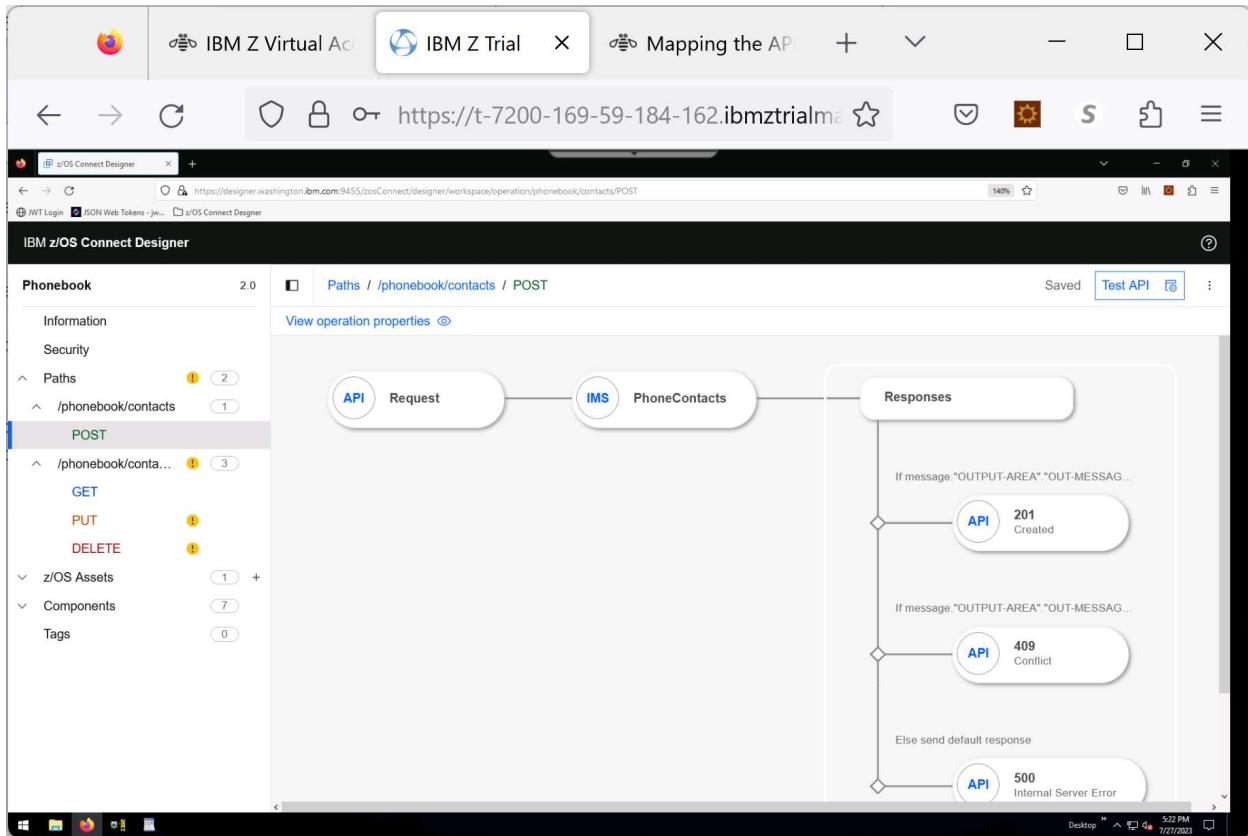
A 500 response code indicates an internal server error. Configure the 500 response to return the z/OS Connect error message by typing the following into the message field (Note the case, brackets, and quotes):

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



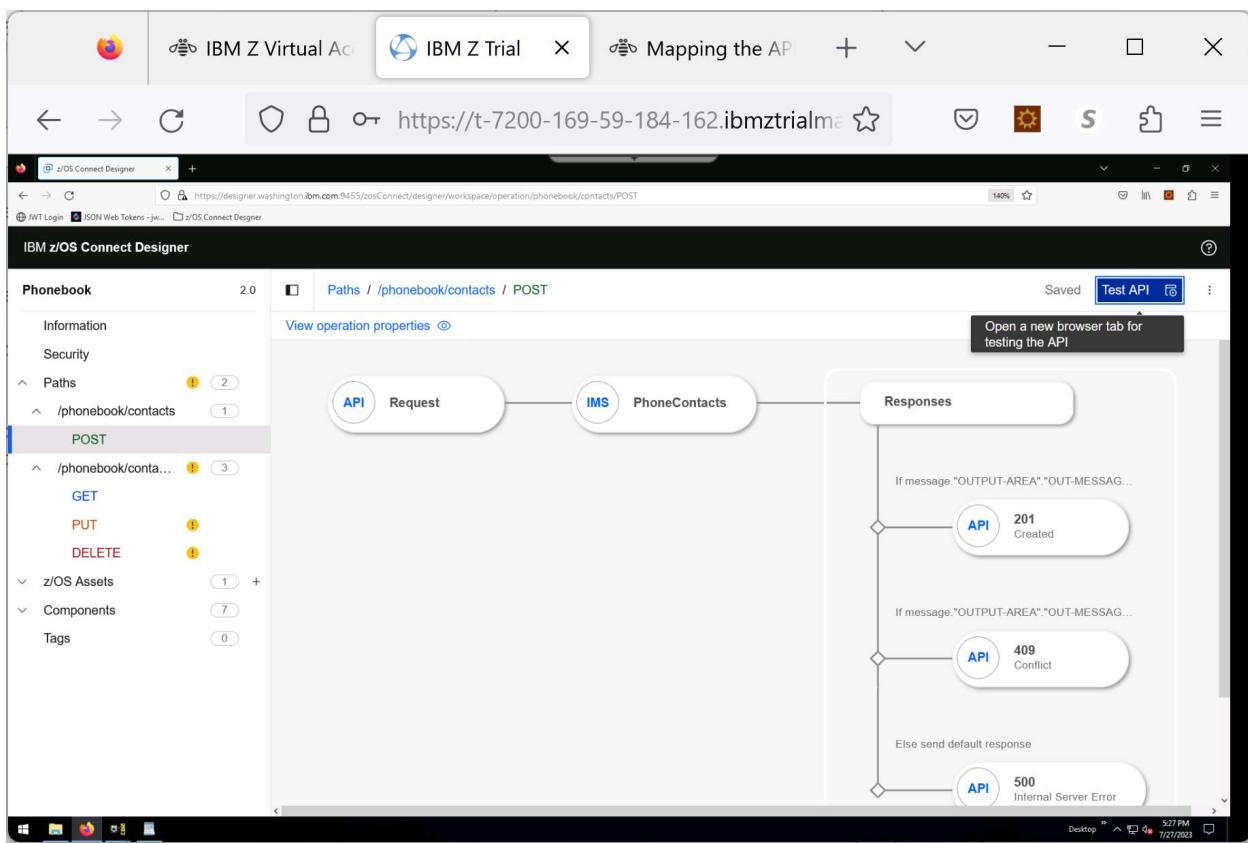
- Click **X** at the top right of this panel to close it and return to the Operations flow diagram.

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 POSTmethod.

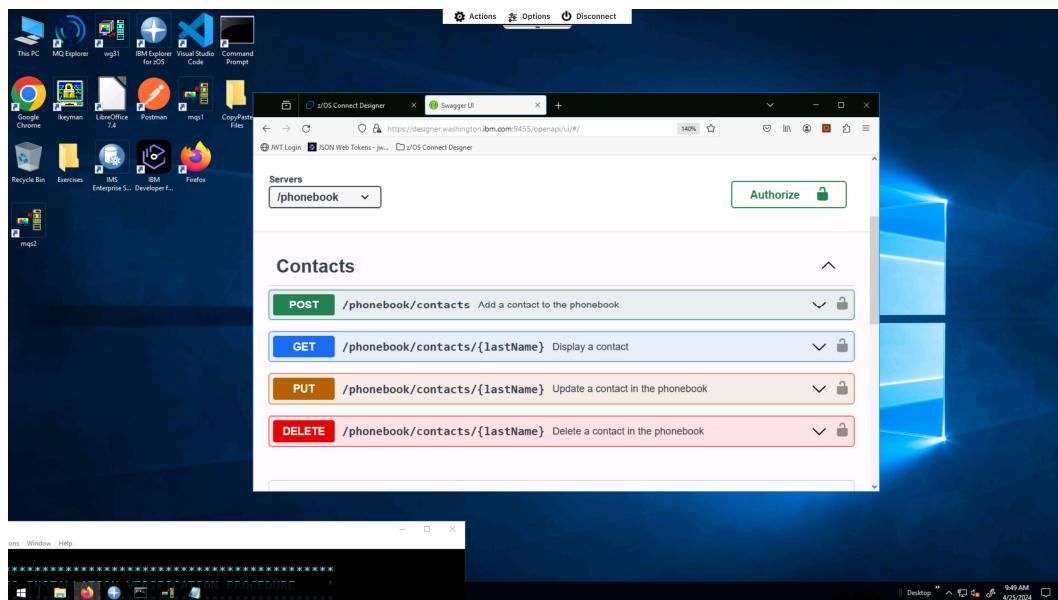


Test the API POST method.

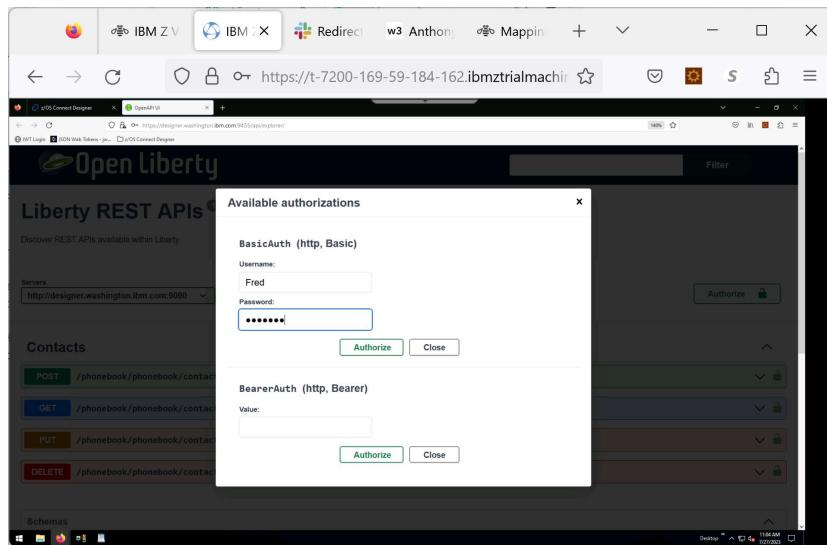
- Click on the **TEST API** button on the top right of the operations diagram.



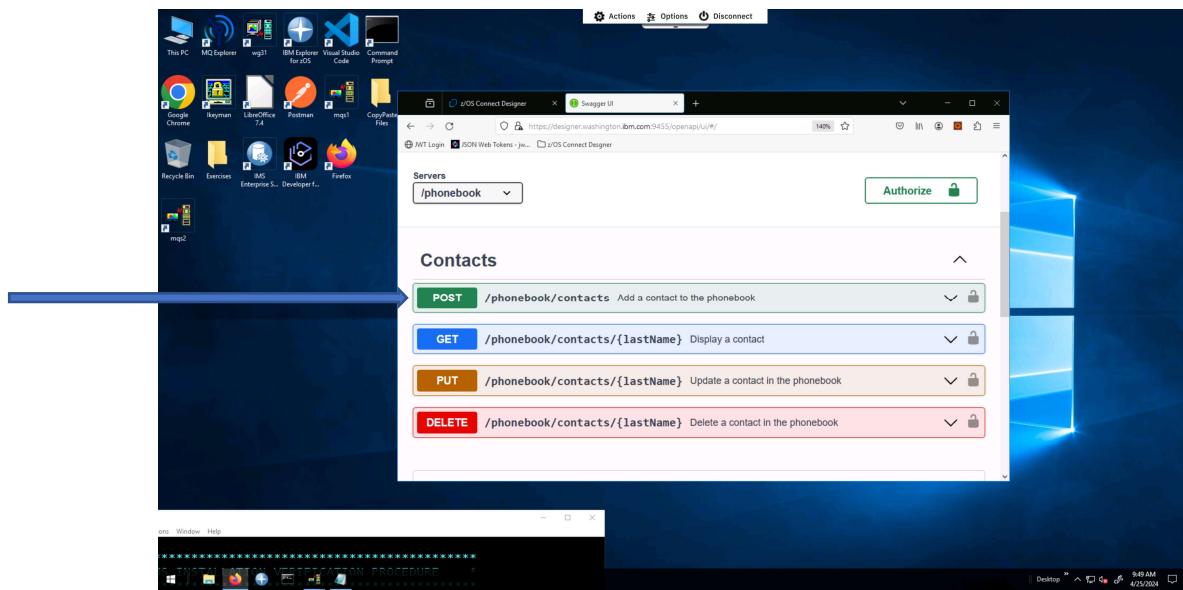
- On the **Servers** drop down, select **/Phonebook**



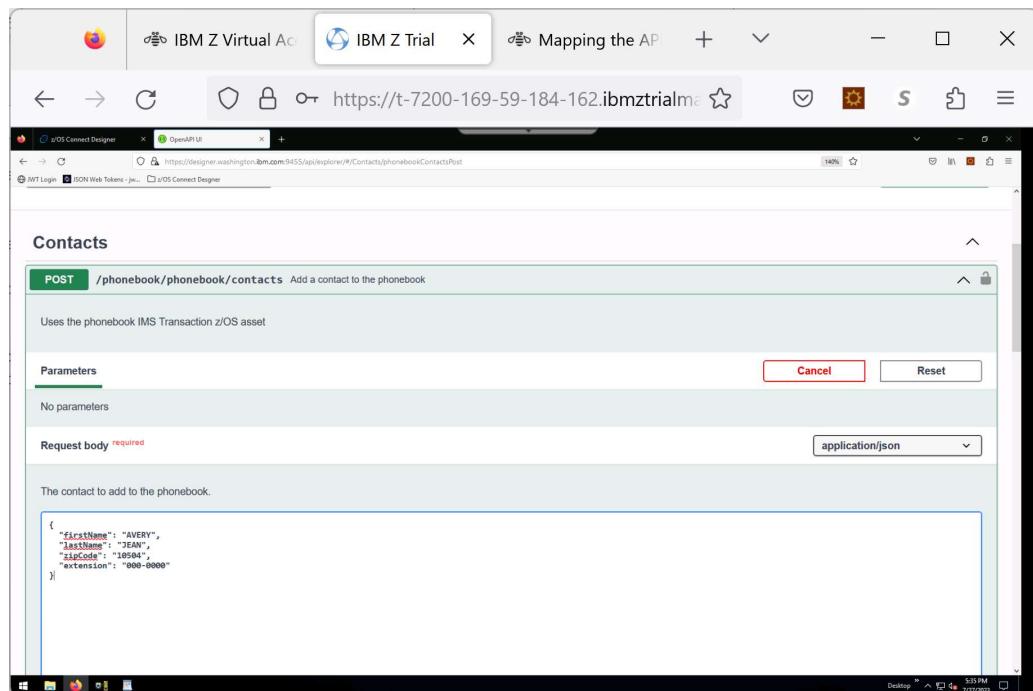
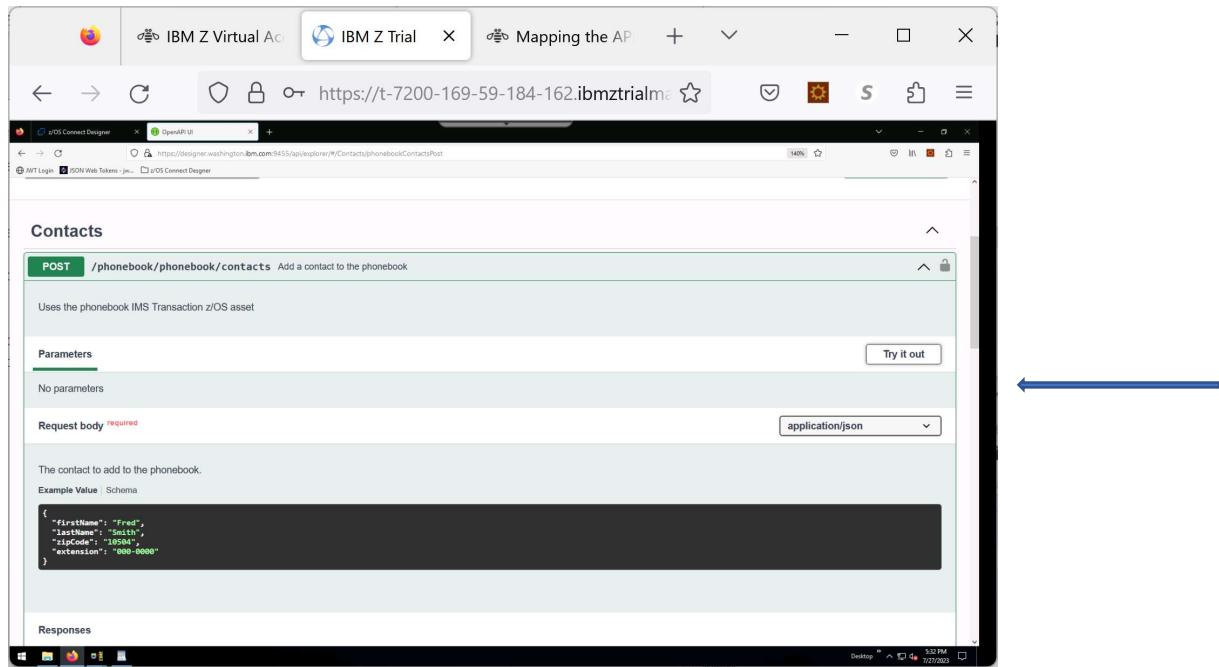
- Click the **Authorize** button and choose BasicAuth
 - Select either of the available usernames user1 or Fred
 - Otherwise, if the username is blank
 - Key in **Fred** (note the capital F) for the Username
 - Key in **fredpwd** (Note all lowercase) for the password
- Click **Authorize**.



Back on the main panel, you will see all the possible methods that can be used for this API. Select the **POST** method.



- Click **Try it out**



- Key in any values of your choice for `firstName`, `lastName`, `zipCode`, and `extension`.
- Click **Execute**.

- You should see the following:

Curl

```
curl -X 'POST' \
'https://designer.washington.ibm.com:9455/phonebook/phonebook/contacts' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "firstName": "AVERY",
  "lastName": "JEAN",
  "middleName": "M",
  "extension": "8000-0000"
}'
```

Request URL

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

Server response

Code	Details
201	<p>Response body</p> <pre>{ "message": "Contact with Last Name JEAN was successfully added" }</pre> <p>Download</p> <p>Response headers</p> <pre>content-encoding: gzip content-language: en-US content-length: 83 content-type: application/json date: Thu,27 Jul 2023 22:49:00 GMT vary: Accept-Encoding x-powered-by: Servlet/4.0</pre>

Responses

Code	Description
------	-------------

Links

Note that the Response body of the 201 code shows a message include the Last Name that was added into the phonebook.

- Try to add the same information that you just added to see if the 409 code is returned
 - Click **Clear**

Parameters

No parameters

Request body **required**

application/json

The contact to add to the phonebook.

```
{
  "firstName": "AVERY",
  "lastName": "JEAN",
  "middleName": "M",
  "extension": "8000-0000"
}
```

Execute Clear

Responses

- The values you entered previously should still be in the request body. Click **Execute**.

The screenshot shows a browser window with the URL <https://t-7200-169-59-184-162.ibmtrialma>. The page displays the results of a POST request to `/phonebook/contacts`. The curl command in the terminal shows a conflict error:

```
curl -X 'POST' \
  'https://designer.washington.ibm.com:9455/phonebook/phonebook/contacts' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{"firstName": "EVERY", "lastName": "JEAN", "zipCode": "10504", "extension": "000-0000"}'
```

The Request URL is `https://designer.washington.ibm.com:9455/phonebook/phonebook/contacts`. The Server response shows a 409 Error: Conflict with the message: "JEAN is a duplicate. Addition of entry failed." The Response headers include connection: Close, content-encoding: gzip, content-language: en-US, and content-length: 79.

- To test the **500** error code, the transaction IVTNO in IMS can be stopped. When that is done, executing the POST will result in

The screenshot shows a browser window with the URL <https://t-7200-169-59-184-162.ibmtrialma>. The page displays the results of a POST request to `/phonebook/contactsPost`. The Server response shows a 500 Error: Internal Server Error with the message: "BAQRI862E: Internal INS z/OS Asset error." The Response headers include connection: Close, content-encoding: gzip, content-language: en-US, content-length: 79, content-type: application/json, date: Thu, 27 Jul 2023 23:12:54 GMT, vary: Accept-Encoding, and x-powered-by: Servlet/4.0.

The Responses section shows a 201 Created response with application/json media type. The Example Value and Schema buttons are visible at the bottom.

Congratulations! You have completed the exercise for the **POST** method.

(You can try the GET method you created earlier to see the entry you added with this POST method)