

Lab 3

Deploy and Run the Campaigns Flow

Overview

Continuing on from Lab 2, let's test our new flow end-end and examine the various capabilities that Flow Designer provides.

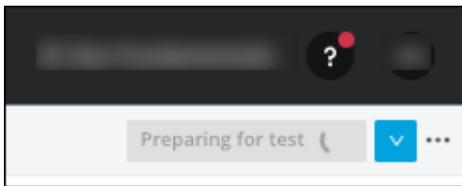
Step 1: Test the application

Like we did in Lab 1, we are going to test the API.

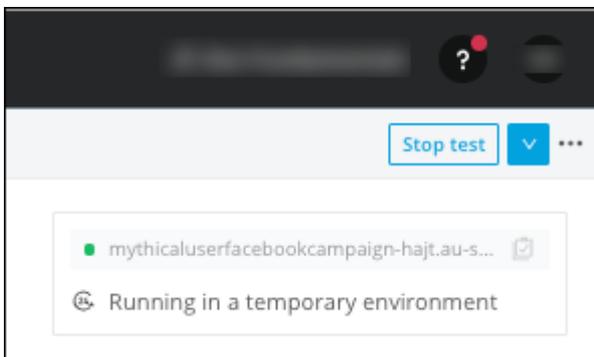
1. Click the Test Button situated on the right top part of the screen.



2. Once you click on the Test button, the message will change to *Preparing for test*

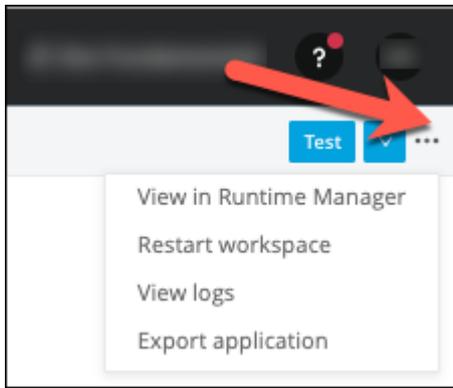


3. Finally you will see the message *Running in a temporary environment*



Step 2: Test the Application

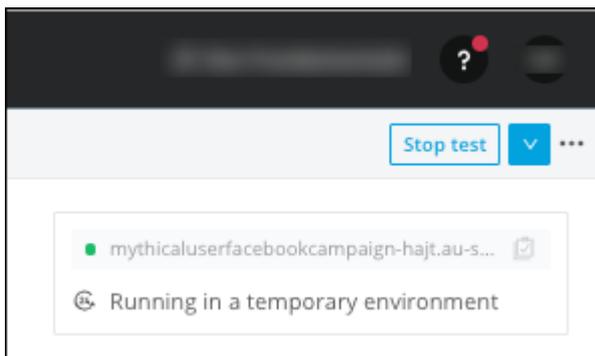
1. Clicking on the three dots to the right of the Test button will cause a Menu to be displayed



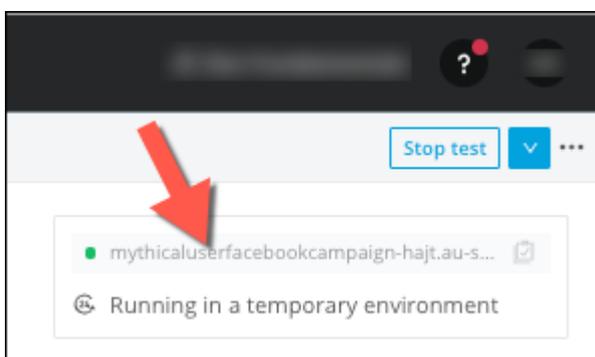
We will give a description of each item

- **View in Runtime Manager:** This option goes to the runtime manager and you can see the application deployed in Mule Runtime
- **Restart workspace:** This option will restart the deployed application in the same way as you will do in in the Runtime Manager section.
- **View logs:** This option shows the Runtime logs online.
- **Export application:** Export the application as as jar file.

2. Click on the three dots to the right of the Test button again to close the Menu



3. We need to know the URL of the API we are going to test, so we will click on the name of the API. This copies the URL to the clipboard.

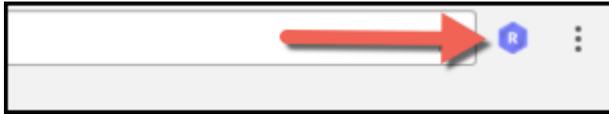


IMPORTANT

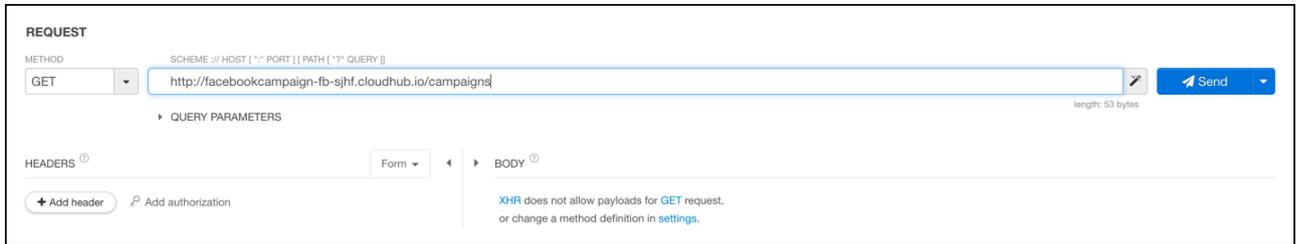
You can use Postman or any other REST Client. If you are unable to install a Chrome extension, try using a web-based tester like API Tester (<https://apitester.com>)

4. If you **DO NOT** have the **Restlet Client** API Tester installed, skip to step 7.

5. Otherwise, open the Chrome browser and open the **Restlet Client** extension.



6. Paste the URL you just copied and append **/campaigns** to the URL



7. Press 

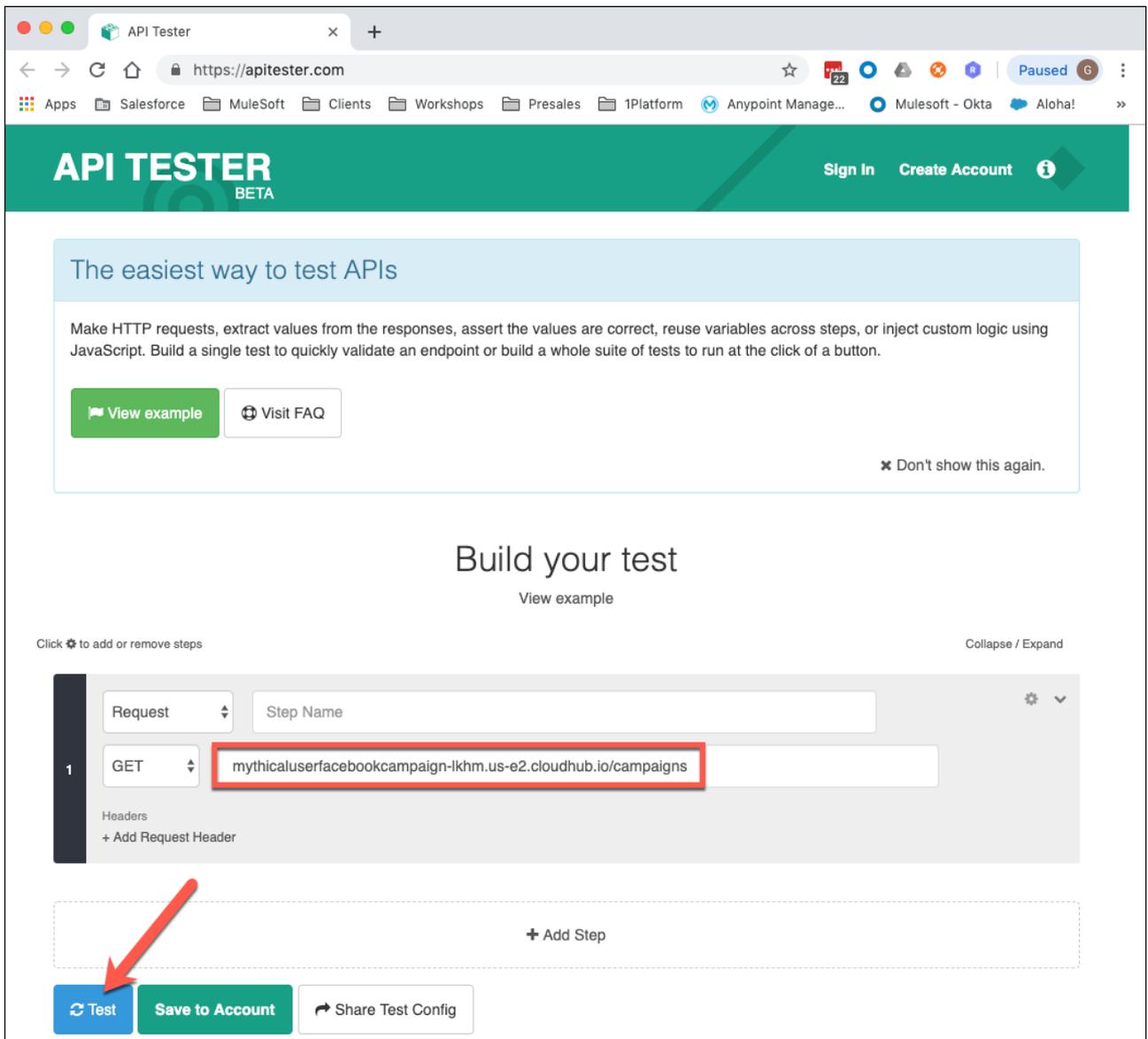
After doing click in the response, you will see the following response



8. If **DO NOT** have the **Restlet Client** API Tester installed, open the Chrome browser and go to API Tester (<https://apitester.com>).

9. Paste the URL you just copied and append **/campaigns** to the URL

10. Press the **Test** button.

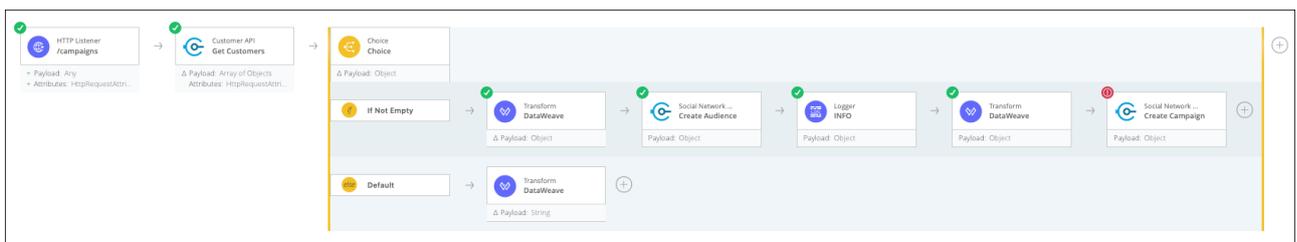


11. Ensure we have got a 500 response. There is an error in our flow.

Let's go back to the Flow Designer and see what happened

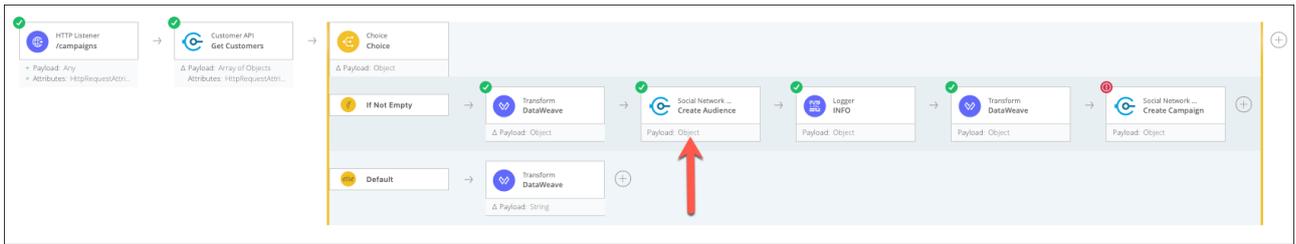
Step 3: Debug the Facebook Campaign API

1. Return to your project in Flow Designer



You can see a green circle with a 'tick' in many components of the flow. The green circles are steps that were successfully executed in the flow. There is one card that has a red circle with a '!', this means an error occurred. If you touch on any card, you can see the input and output.

2. Open the Create Audience Card



3. Go to the **Input** tab

Social Network API for Facebook
Create Audience

Configuration **Input** Output Notes

Show: Payload ▾ Data type: Object ▾ Add Edit

History	
Jul 29, 2019 03:42pm	<pre>{ "customers": [{ "country": "USA", "phone": "1-(951)768-8479", "city": "Riverside", "postalCode": "92513", "email": "jerome@ee.fr", "product": [] }, { "country": "USA", "phone": "1-(951)768-8479", "city": "Riverside", "postalCode": "92513", "email": "jeromess@ee.fr", "product": [] }, { "country": "USA", "phone": "1-(951)768-8479", "city": "Riverside", "postalCode": "92513", "email": "jeromddess@ee.fr", "product": [] }, { "country": "USA", </pre>

4. Go to the **Output** tab

Social Network API for Facebook
Create Audience

Configuration Input **Output** Notes

Show: Payload ▾ Data type: Object ▾ Add Edit

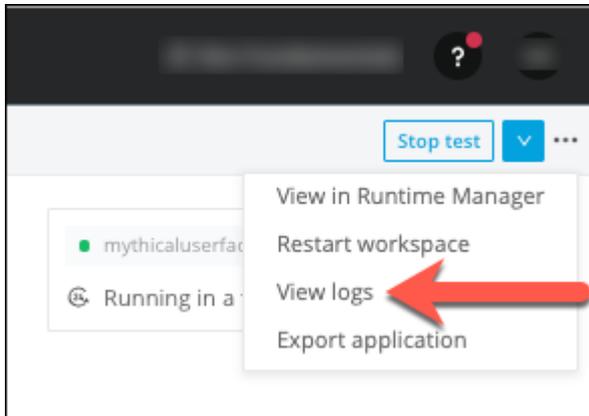
History	
Jul 29, 2019 03:42pm	<pre>{ "id": "FBAu38u9ru-0001" }</pre>

NOTE

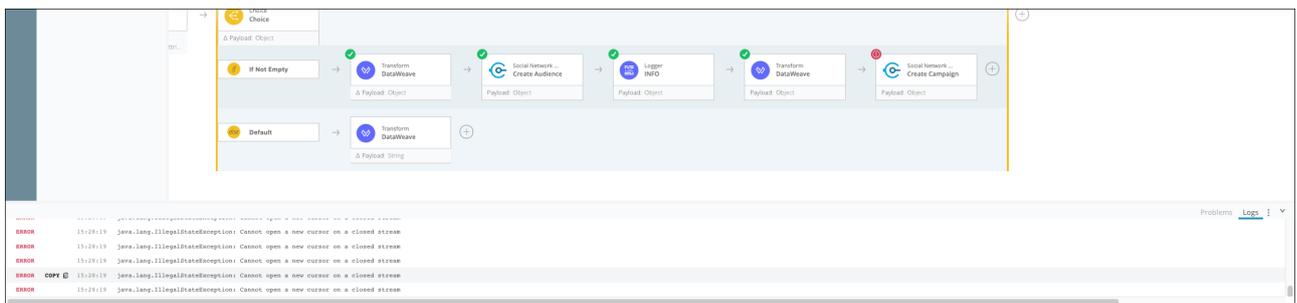
If the step was executed without any error it is possible to see the input and output.

Now let's see the logs

5. Go to the Menu and select **View logs**



A new window will open at the bottom of the flow with the logging information.



6. Scroll up and you will find an error similar to this

```

*****
Message           : HTTP POST on resource 'http://social-network-
api.cloudhub.io:80/api/campaign' failed: bad request (400).
Element           : new_flow/processors/1/route/0/processors/4 @
mythicaluserfacebookcampaign-lkhm:mule-config.xml:54 (Create Campaign)
Element XML       : <social-network-ap-ifer-facebook:create-campaign client-
id="11def1b704e24d87a5ea5769772c90a7" client-
secret="88845E529f1F42E4a5aC96Fd504c3e01" config-
ref="social_network_api_for_facebook" doc:id="358907a7-c7c3-4491-a535-7527605fe209"
doc:name="Create Campaign" doc:timestamp="1564373789423"></social-network-ap-ifer-
facebook:create-campaign>
Error type        : SOCIAL-NETWORK-AP-IFOR-FACEBOOK:BAD_REQUEST
FlowStack         : at social-network-ap-ifer-facebook:create-campaign 76 ms
                   at new_flow(new_flow/processors/1/route/0/processors/4 @
mythicaluserfacebookcampaign-lkhm:mule-config.xml:54 (Create Campaign)) 77 ms
Payload           : null
Payload Type      : null
-----
Root Exception stack trace:
org.mule.runtime.api.exception.MuleRuntimeException: HTTP POST on resource
'http://social-network-api.cloudhub.io:80/api/campaign' failed: bad request (400).
    at
org.mule.runtime.extension.api.exception.ModuleException.<init>(ModuleException.jav
a:66)
    at
org.mule.runtime.extension.api.exception.ModuleException.<init>(ModuleException.jav
a:76)
    at
org.mule.extension.http.api.request.validator.ResponseValidatorTypedException.<init
>(ResponseValidatorTypedException.java:31)
    at
org.mule.extension.http.api.request.validator.ResponseValidatorTypedException.<init
>(ResponseValidatorTypedException.java:58)
    at
org.mule.extension.http.api.request.validator.RangeStatusCodeValidator.lambda$throw
ValidationException$3(RangeStatusCodeValidator.java:107)
    at java.util.Optional.map(Optional.java:215)

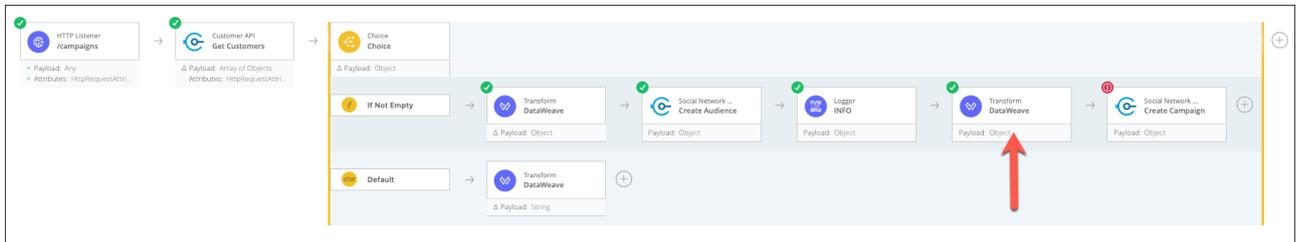
```

The log is saying that the **Create Campaign** method is returning *Bad Request* Also you can see that the green circles end at that card and there is a red circle instead.

Step 4: Fix the Request Message

At this point it seems to be obvious that we need to change the transform step that is before this card.

1. Open the Transform card

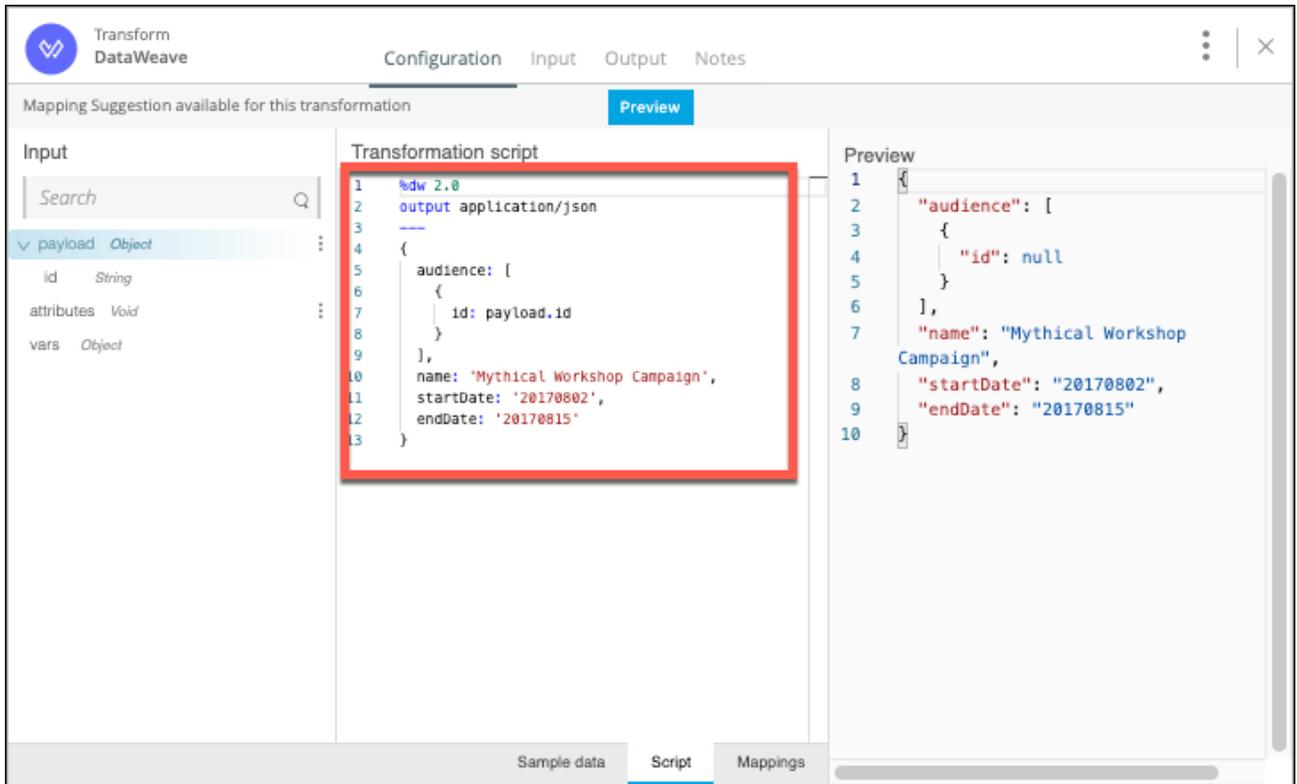


2. Go to the script tab. We are going to add the name, startDate and endDate.

3. Copy and paste the following text:

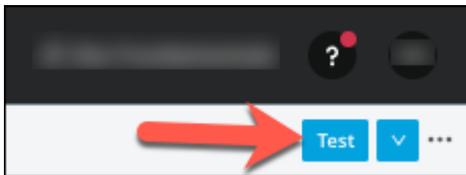
```
%dw 2.0
output application/json
---
{
  audience: [
    {
      id: payload.id
    }
  ],
  name: 'Mythical Workshop Campaign',
  startDate: '20170802',
  endDate: '20170815'
}
```

4. You will see the following after pasting in the new script:

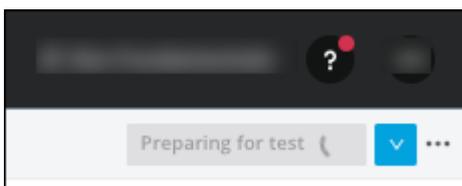


5. Close  card.

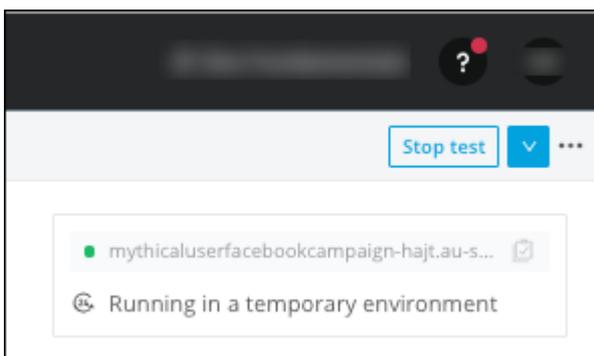
6. Press the **Test** button.



7. You should see the status change to *Preparing for test*.



8. Wait until you see the message *Running in a temporary environment*.



9. Go back to the Rest Client that was opened in Chrome (or Postman/REST API Tester)

10.

Press



Cache Detected - Elapsed Time: 2.3s

200

HEADERS [Ⓞ] pretty ▾

Connection: keep-alive
Content-Type: application/json
Date: 2017 Aug 2 10:05:15
Server: nginx
transfer-encoding: chunked

COMPLETE REQUEST HEADERS Pretty ▾

Accept: /*
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.8,es;q=0.6
Cookie: ajs_group_id=null; ajs_user_id=%22ca946152-2d6c-4800-9b94-2c2725327a31%22; ajs_anonymo... id=%22a5ea46-a4ee-4542-93f4-8fh430f928d7"

BODY [Ⓞ] pretty ▾

```
{  
  "id": "20a5ea46-a4ee-4542-93f4-8fh430f928d7"  
}
```

length: 49 bytes

lines nums

⌕ Top ⌕ Bottom Collapse Open ↻ 2Request Copy Download

That is the Campaign ID of the Facebook Marketing campaign you've just deployed for your target Customer Audience.

Summary

In this lab, we:

- Tested using a REST client
- Tested our Application
- Debugged the Facebook Campaign API
- Fixed the Request Message

We easily created and designed a Campaigns application for our social media marketing campaign, providing Sheryl with the ability to upload and correlate customer data to Facebook. We leveraged Flow Designer for an **ad-hoc integration approach**.

We saw how by abstracting our back-end systems as APIs, the line of business was able to reuse the **Customer** and **Social Network** APIs we created in other modules in a self-service way to create Process layer applications to streamline business processes.

Sheryl in Marketing is happy because she was able to get access to the data she needed for her initiative in a quick, self-service way, and Central IT is happy because they're not provisioning any new assets, yet still maintaining governance of their underlying data.

- See link [Run Application in a Design Center](#) doc for more information.
- See link [Troubleshoot in Flow Designer](#) doc for more information.
- See link [To Promote an Application to a Production Environment](#) doc for more information.

Congratulations! You have completed Module 4.

Please proceed to [Module 5](#)

[Take me to the TOP](#)