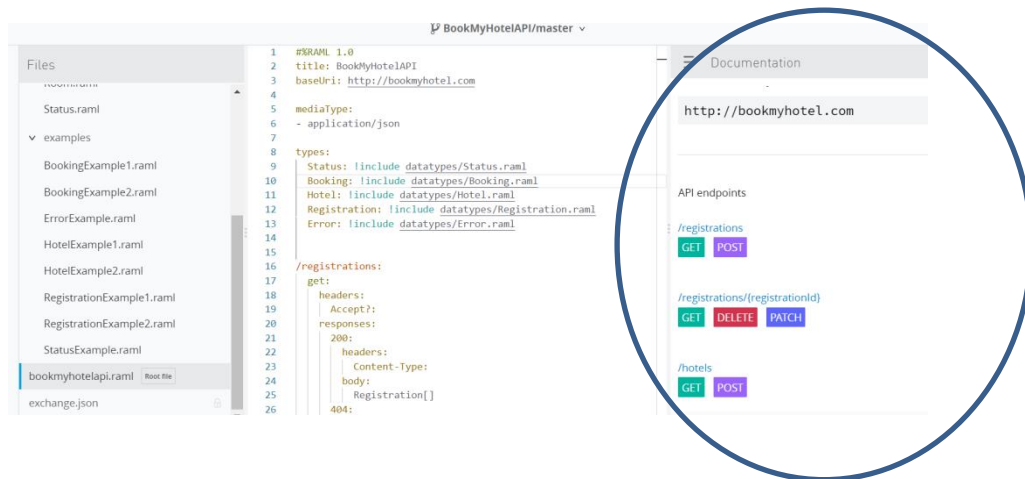## TOC

LAB : Test using Mocking Service

In this lab you will -

a) Test using Mocking Service from API Designer
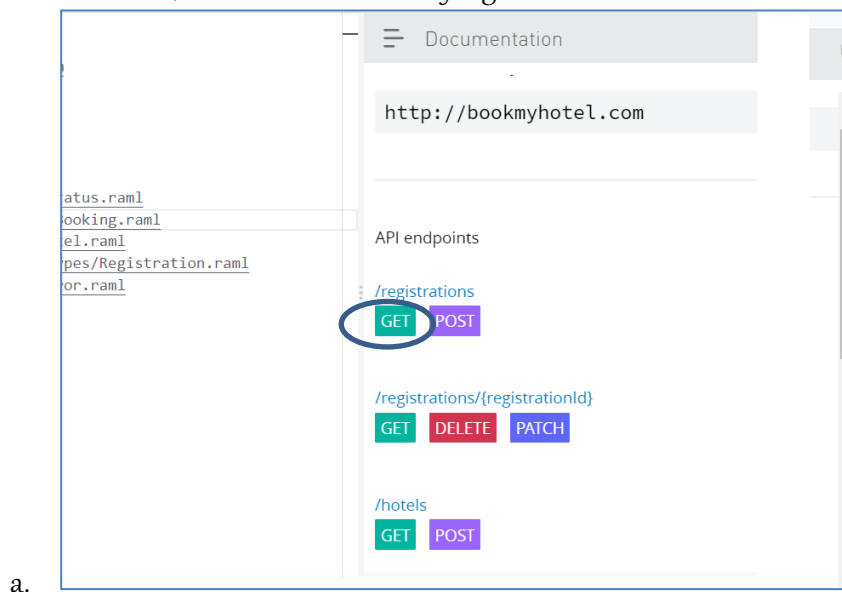
b) Test using Mocking Service from Postman

Step 1 – Test GET method of Registrations Resource (/registrations)

1. Mocking Service is enabled by default

2. When we test using Mocking Service, the response returns the HTTP status code defined in our RAML specs. The response body contains the example values defined using Example Fragments.

3. In the API designer, you will see the Interactive API console at the right hand side -
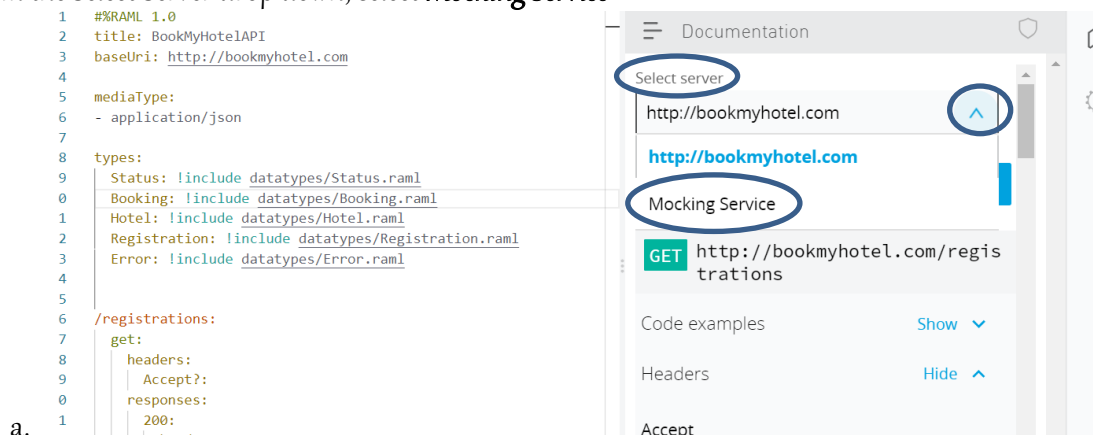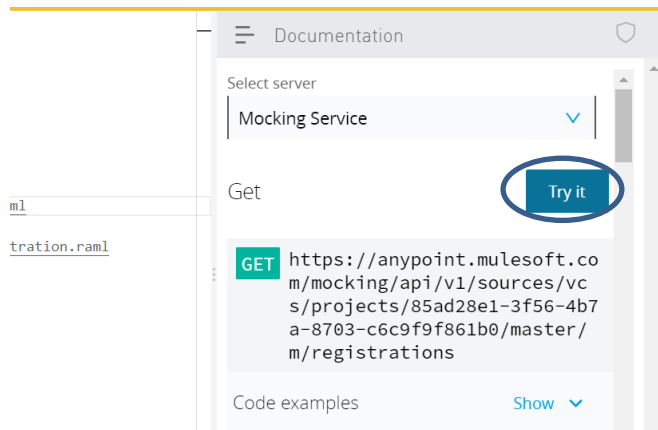
4.

```
1   #%RAML 1.0
2   title: BookMyHotelAPI
3   baseUri: http://bookmyhotel.com
4
5   mediaType:
6   - application/json
7
8   types:
9     Status: !include datatypes/Status.raml
10    Booking: !include datatypes/Booking.raml
11    Hotel: !include datatypes/Hotel.raml
12    Registration: !include datatypes/Registration.raml
13    Error: !include datatypes/Error.raml
14
15
16  /registrations:
17    get:
18      headers:
19        Accept?:
20      responses:
21        200:
22          headers:
23            Content-Type:
24          body:
25            Registration[]
26        404:
```

5. In the API Console,Click on GET method of registrations resource

a.

6. From the Select Server drop down, select **Mocking Service**
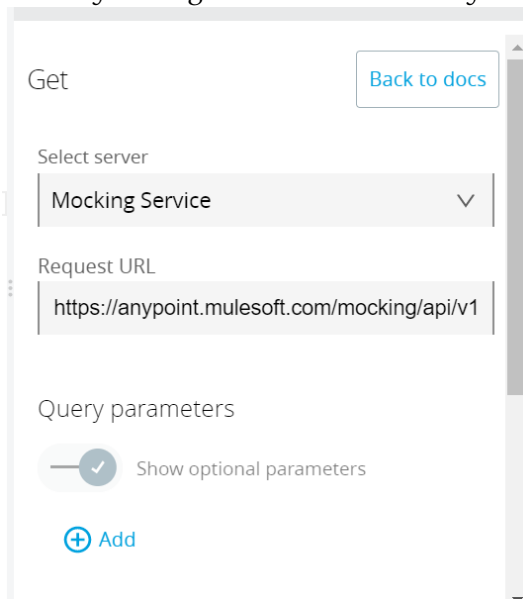
a.

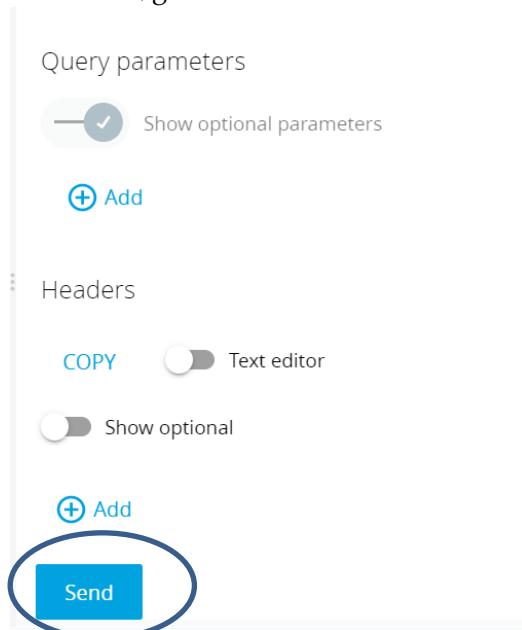7. Click on try it button

a.

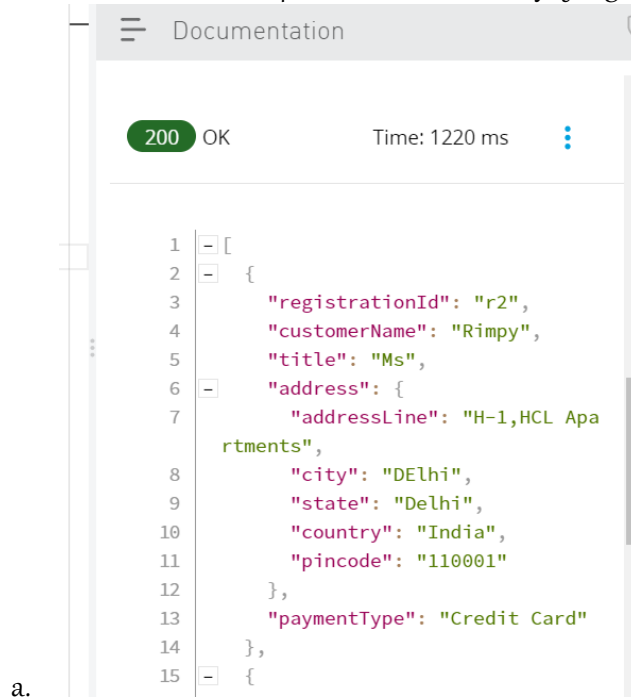8. Now you will see following details in the window of API Console—

a.

b. Here you will see Mocking Service selected and the request URL of mocking service

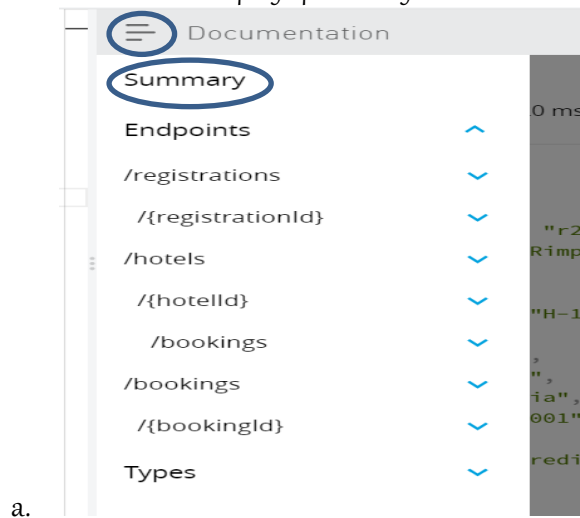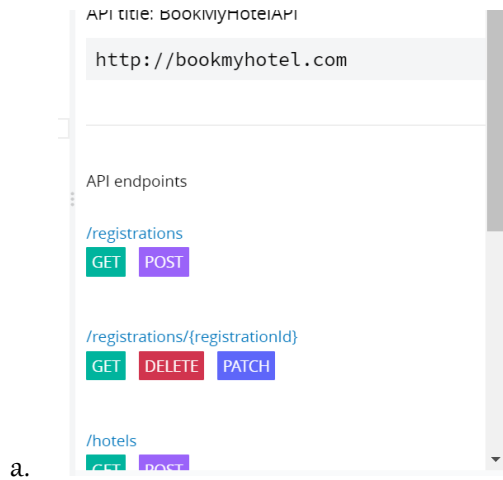9. In the same window, go down and click on send button

a.

10. In the same window of API console, you will get the response back

11. Go down and notice 200 http status code and array of registrations as the response

a.

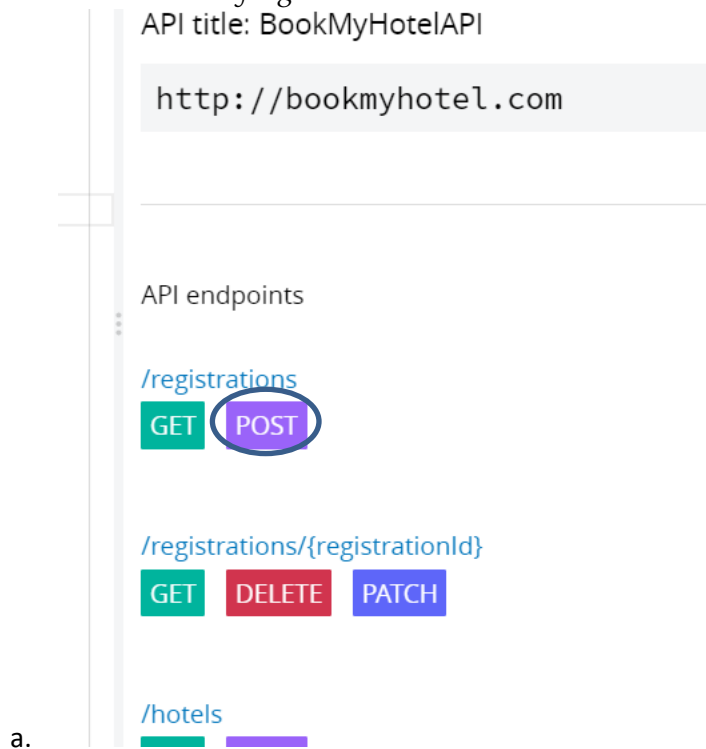12. Click on three bars at top left portion of API Console and then click on Summary

a.

13. You will be back to the summary page of API Console

a.

Congratulations, you have completed this step

Step 2 – Test POST method of Registrations Resource (/registrations)

1. Click on POST method of registrations resource

a.

2. From the Select Server drop down, select **Mocking Service**

3. Click on Try-It button

4. In the window, specify the value of Content-Type as application/json as shown below -

a.

b. Notice the * sign near Content-Type indicating that this is a required field

5. In the same window, if you go down then you can see that POST body is already set with the
    example fragment value of registration

a.

6. Click on Send button

7. HTTP status code :- 201 created , is returned in the same window
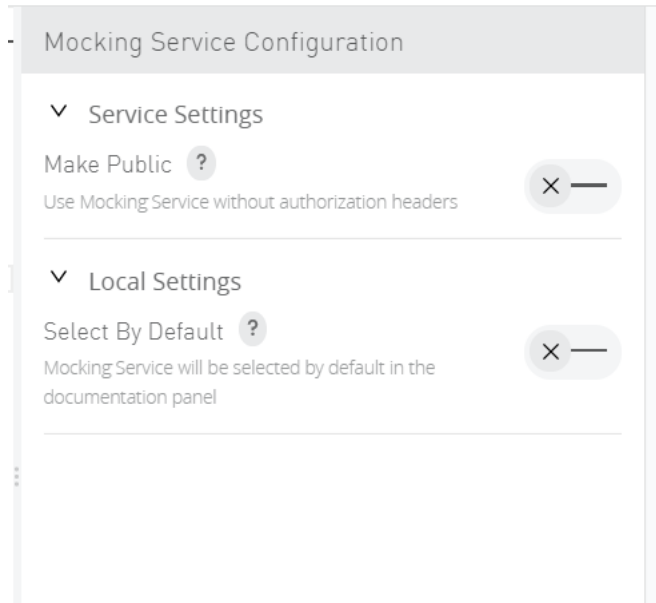
Congratulations, you have completed this step

Step 3 – Select by default

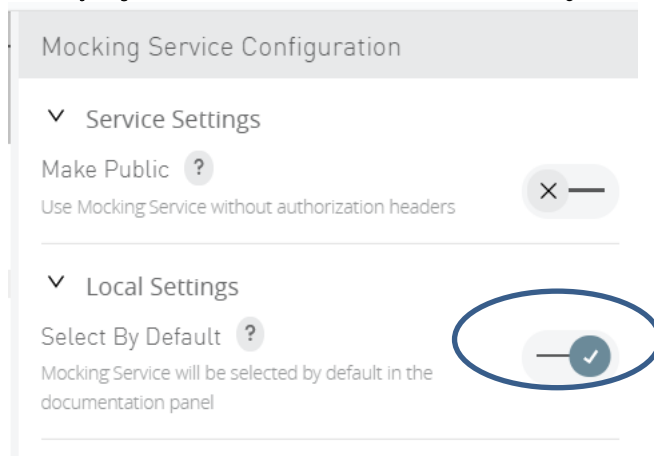1. Click on the Mocking Service Configuration icon -
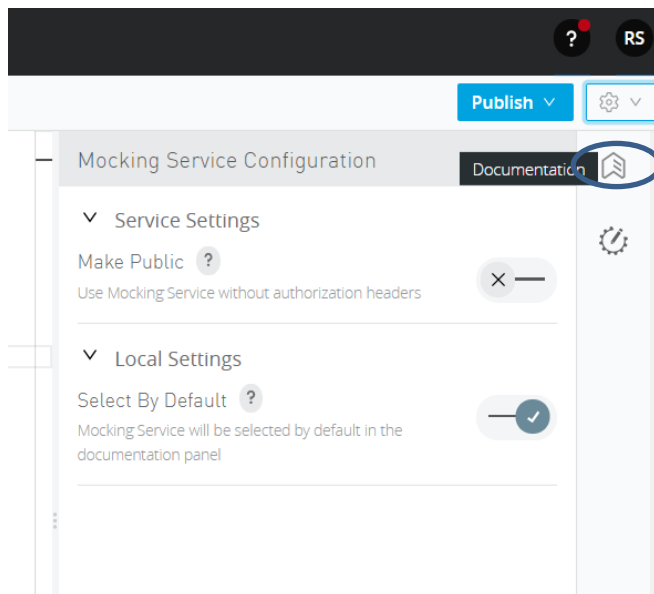

a.

b. You will see following –

c.

2. Enable **Select by default** as shown below (it is disabled as of now) –
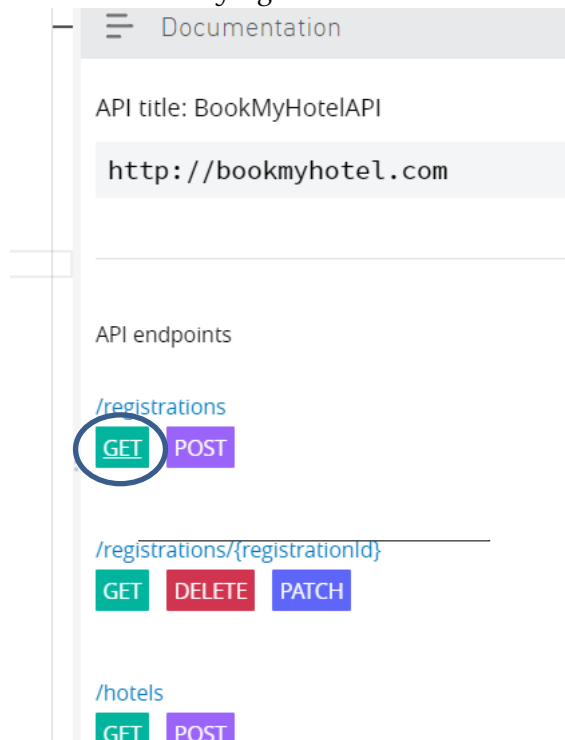

a.

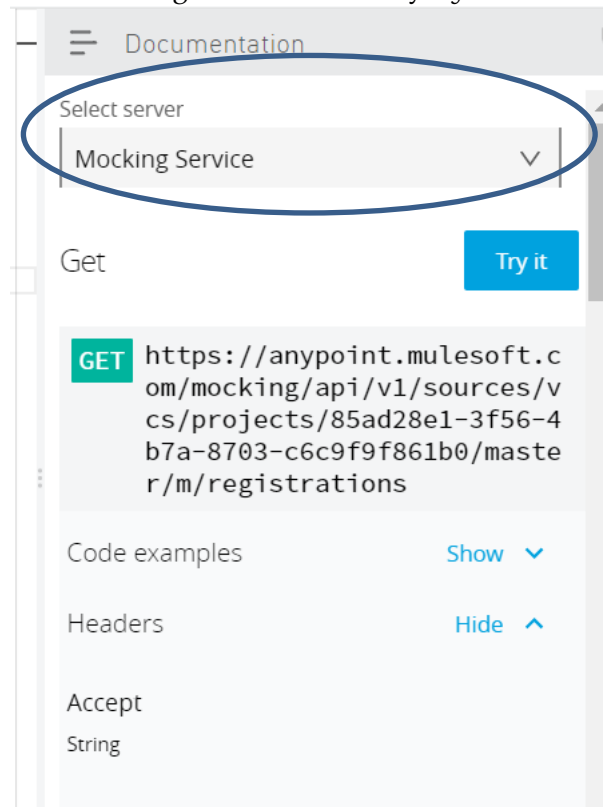3. Now click on documentation icon –

a.

4. Click on the GET method of registrations resource –

   a.

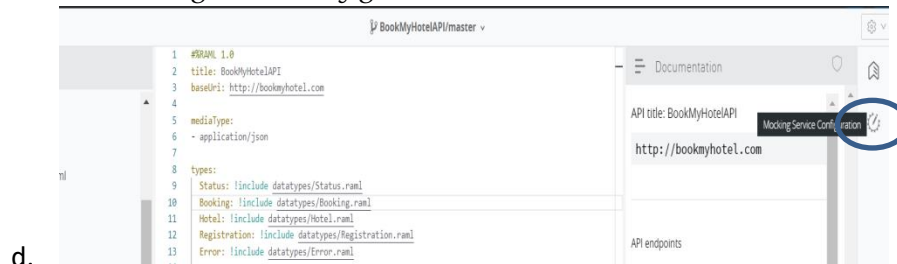5. Notice that the mocking service is selected by default in the **Select Server** drop down

   a.

6. The setting **Select by default** which we have enabled in Mocking Service configuration makes sure that Mocking service is selected by default in the drop down(for all methods/resources). You don't need to select it again and again when you try to do testing.
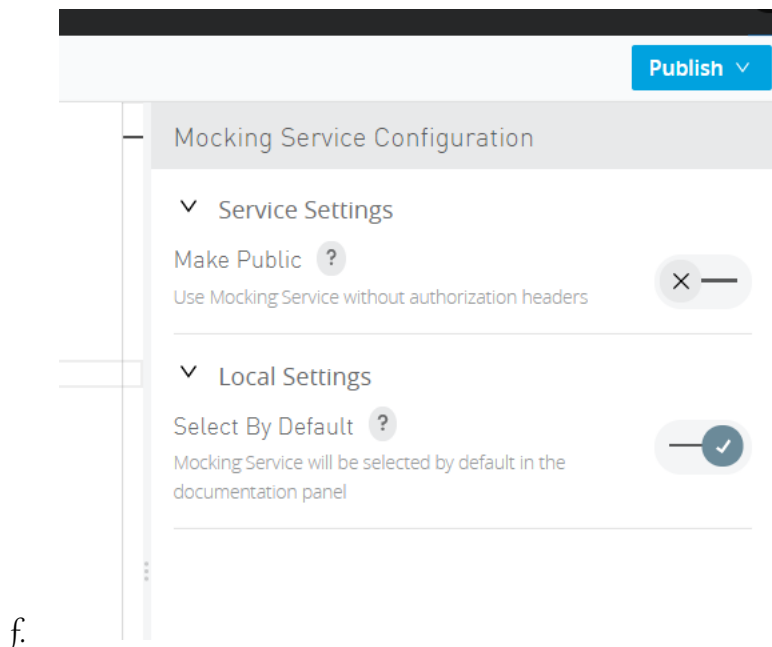
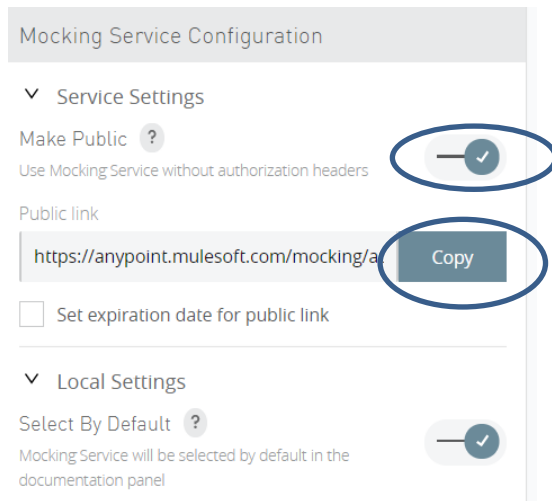**Congratulations, you have completed this step**

Step 4 – Test on POSTMAN

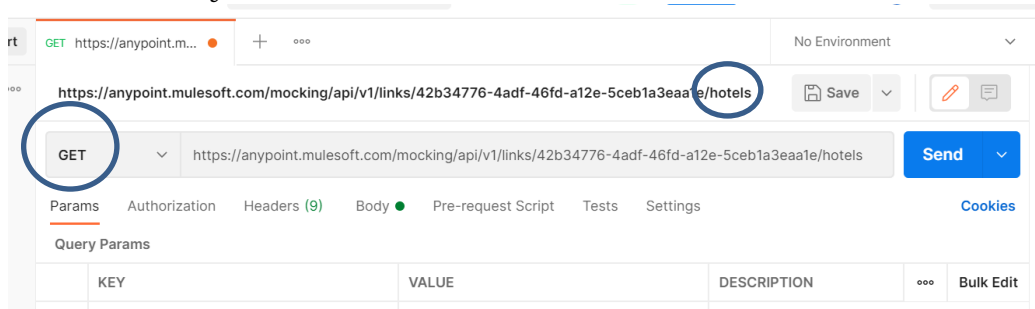1. Click on the Mocking Service Configuration icon -

   d.

   e. You will see following after previous step (step 3) changes –

   f.

2. Enable **Make Public** option as shown below :-

a.

b. Notice the **public link appears**

c. Click on the **Copy** button to Copy the url

3. Open postman and paste this URL

4. Specify **/hotels** at the end of this URL and select **GET** method

a.

5. Click send button

6. We will get the list of hotels

Congratulations, you have completed this step

Congratulations!! You have completed this exercise.