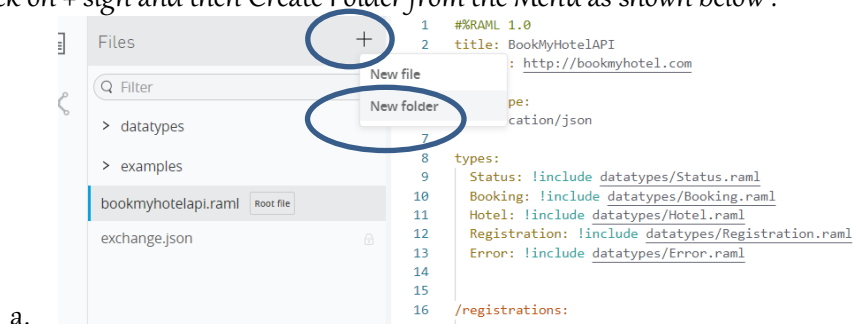## TOC

LAB : Documentation
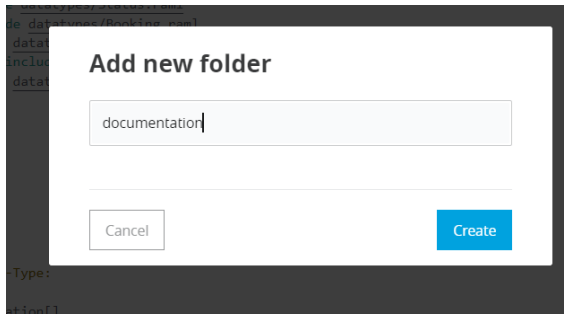
In this lab you will -

a) Create Documentation Fragment

b) Include Documentation Fragment in Root RAML file

c) Define DisplayName and Description
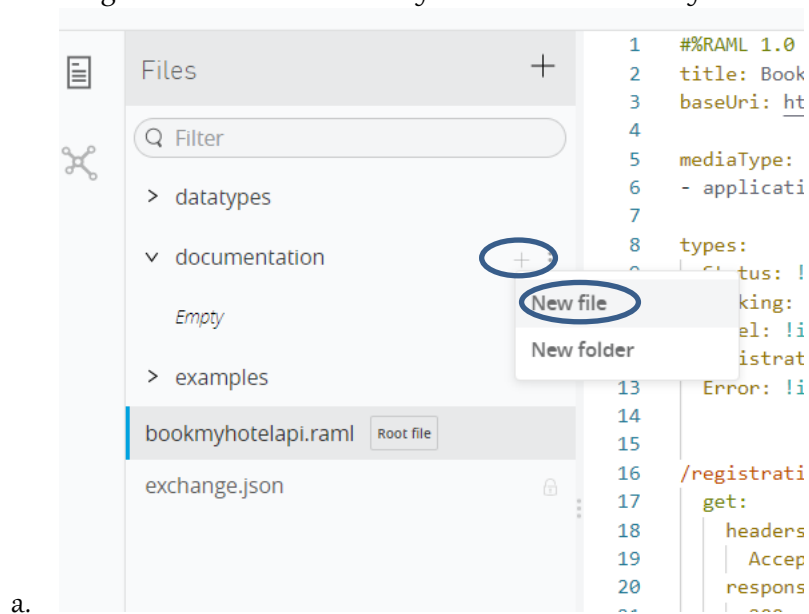
Step 1 – Create Documentation Fragment

1. Click on + sign and then Create Folder from the Menu as shown below :-

    a.

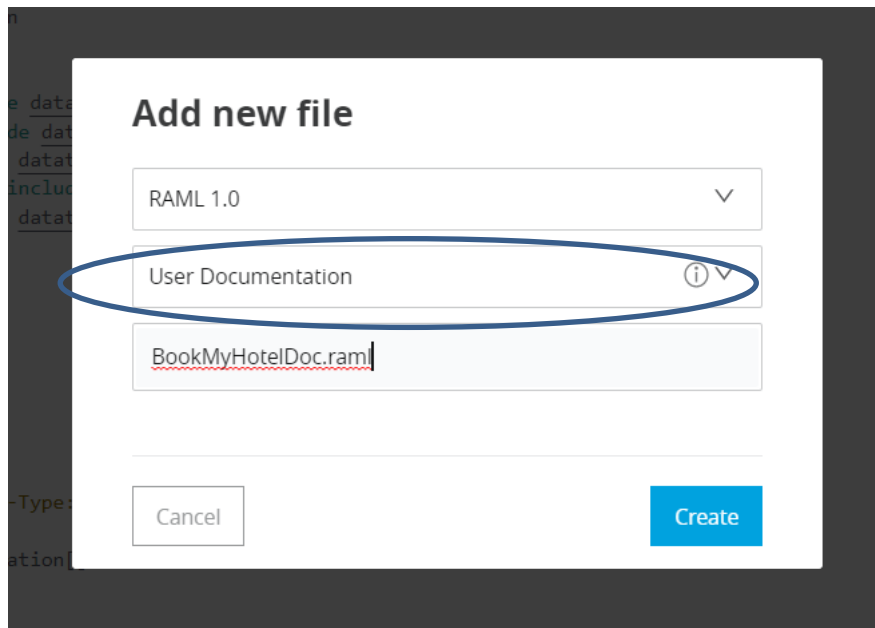2. In the dialogue box, specify the name of the folder as **documentation** and then click on Create button

3. In the left hand side you can see the documentation folder created and it is empty as of now

4. Click on + sign near the documentation folder and click on new file

   a.

5. In the Add New File dialogue box, select **User Documentation** from the drop down and specify the name as **BookMyHotelDoc.raml**

a.

b. This will create documentation fragment – BookMyHotelDoc.raml

6. In BookMyHotelDoc.raml , define as follows :-

a.

b. The documentation title and content are defined above
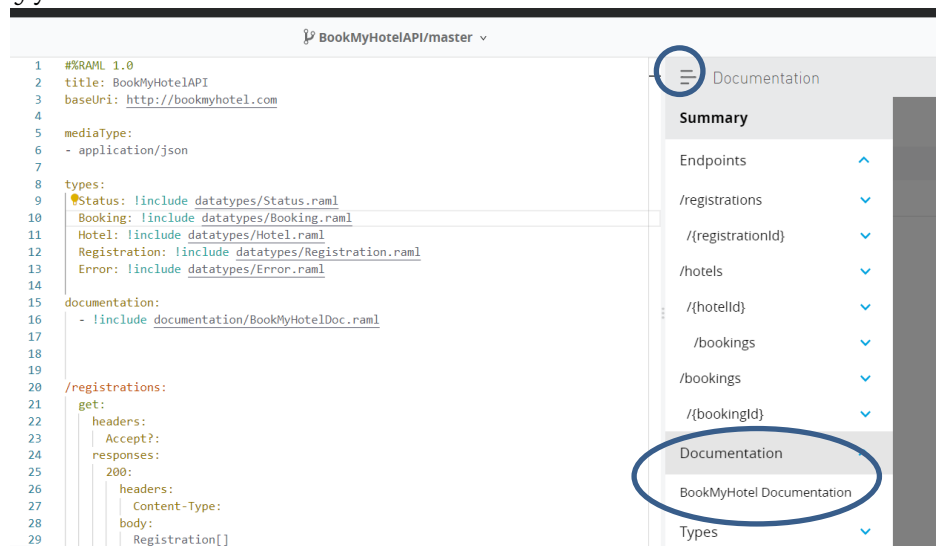
Congratulations, you have completed this step!!

Step 2 – Include Documentation Fragment in Root RAML file

1. Open Root RAML file – bookMyHotelApi.raml

2. Define **documentation** node and include documentation fragment as shown below –

```
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   documentation:
16     - !include documentation/BookMyHotelDoc.raml
17
18
19
20   /registrations:
21     get:
22       headers:
```
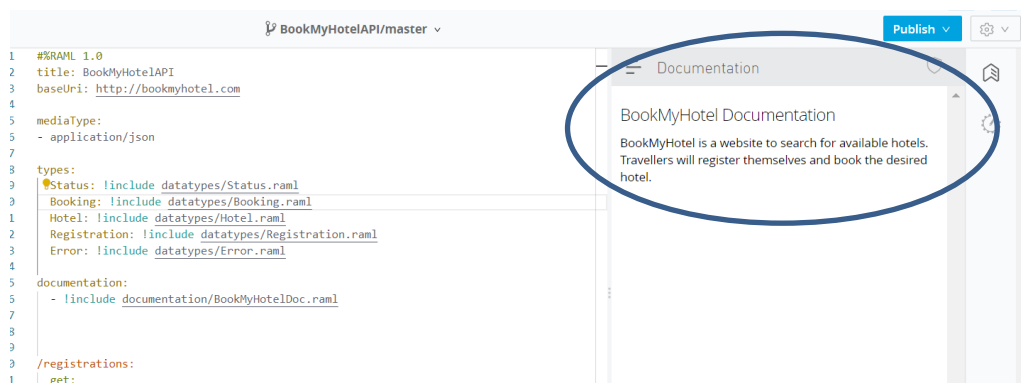
a.

3. Click on the three bars at the top left of API console, click on documentation and then click on the title of your documentation

a.

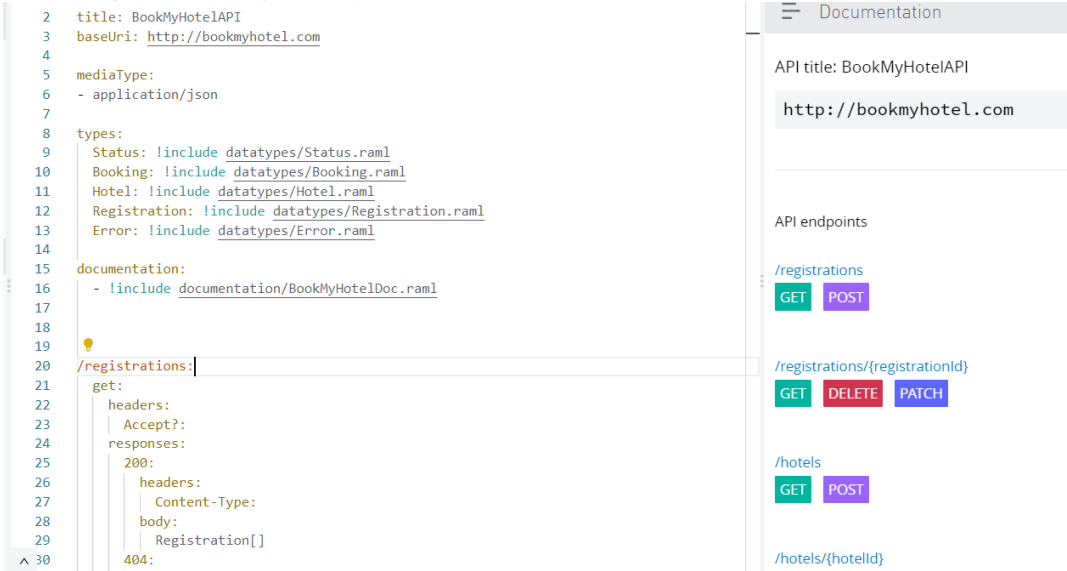4. You can now see the documentation on the API Console—

a.

Congratulations, you have completed this step

Step 3 – Define displayName and description for resources and methods in Root RAML

1. Open Root RAML file – bookMyHotelApi.raml

   a.
   ```
    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   documentation:
   16     - !include documentation/BookMyHotelDoc.raml
   17
   18
   19
   20   /registrations:
   21     get:
   22       headers:
   23         Accept?:
   24       responses:
   25         200:
   26           headers:
   27             Content-Type:
   28           body:
   29             Registration[]
   30         404:
   ```
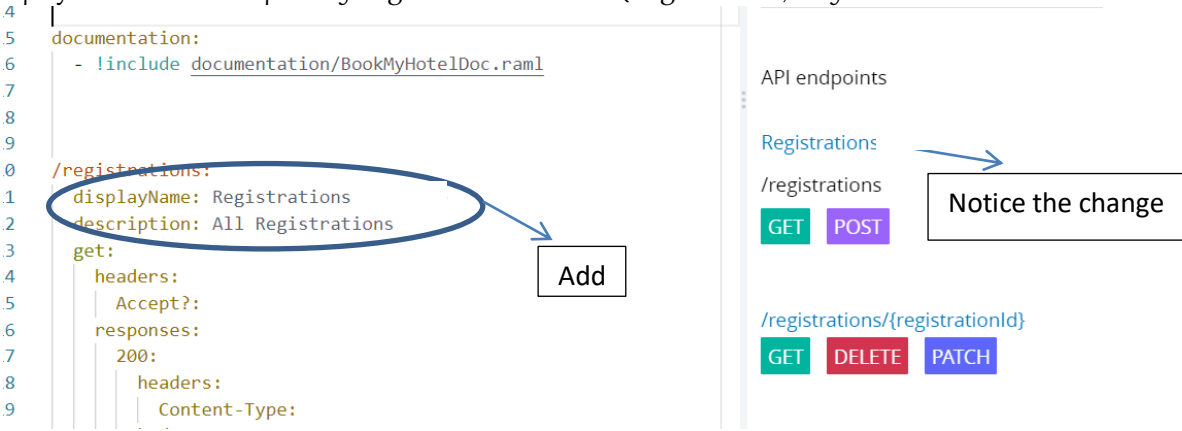
   Documentation

   API title: BookMyHotelAPI

   `http://bookmyhotel.com`

   API endpoints

   /registrations
   GET POST

   /registrations/{registrationId}
   GET DELETE PATCH

   /hotels
   GET POST

   /hotels/{hotelId}

   b. Notice the API console which has resources and methods. The resources are specified using the resource path

2. Define displayName and description of Registrations resource (/registrations) as follows –

   a.
   ```
    4   |
    5   documentation:
    6     - !include documentation/BookMyHotelDoc.raml
    7
    8
    9
    0   /registrations:
    1     displayName: Registrations
    2     description: All Registrations
    3     get:
    4       headers:
    5         Accept?:
    6       responses:
    7         200:
    8           headers:
    9             Content-Type:
   ```
   Add

   API endpoints

   Registrations

   /registrations
   GET POST

   Notice the change

   /registrations/{registrationId}
   GET DELETE PATCH

   b. Notice the changes in API Console. It contains the displayName of the registrations resource

3. Click on **Registrations** in the API console

```
4   |
5   documentation:
6     - !include documentation/BookMyHotelDoc.raml
7
8
9
0   /registrations:
1     displayName: Registrations
2     description: All Registrations
3     get:
4       headers:
5         Accept?:
6       responses:
7         200:
8           headers:
9             Content-Type:
```

API endpoints

Registrations
/registrations
GET POST

/registrations/{registrationId}
GET DELETE PATCH

a.

4. Notice the description –

Documentation

Select server

Mocking Service

Registrations Resource

https://anypoint.mulesoft.com/moc
king/api/v1/sources/vcs/projects/
85ad28e1-3f56-4b7a-8703-c6c9f9f86
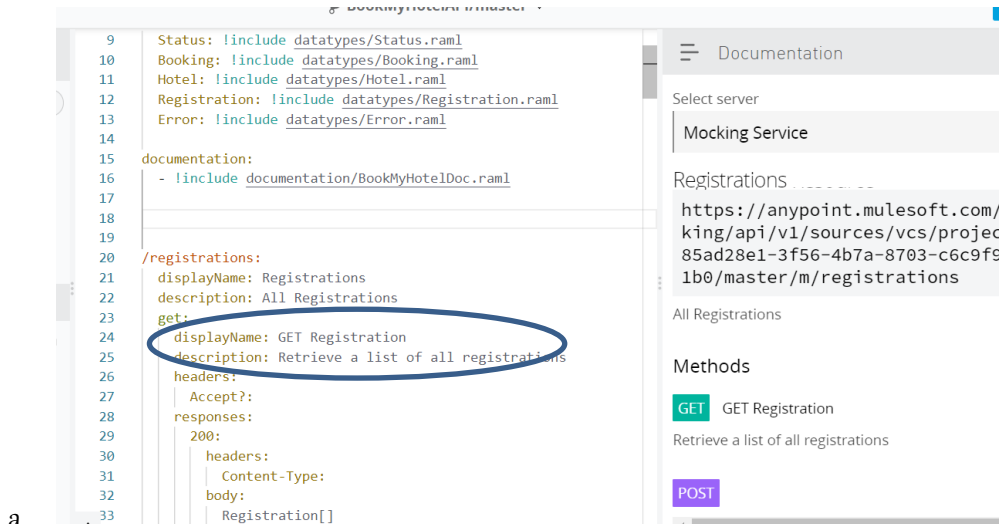1b0/master/m/registrations
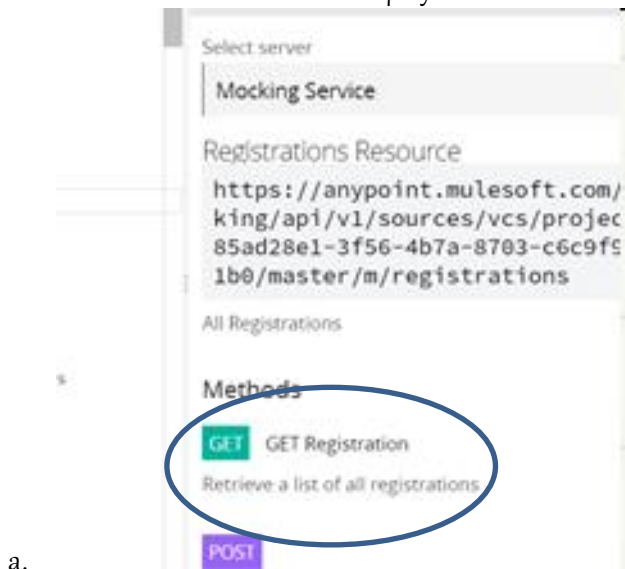
All Registrations

Methods

GET

POST

5. Define displayName and description for **GET method** of registrations resource as shown below :-

a.

6. Check API Console and notice the displayName and description for the GET method –

a.

7. If you click on GET method, you will see the details again

8. Define displayName and description for all resources/methods/datatype fragments or import zip file attached in the resources section

Congratulations, you have completed this step

Congratulations!! You have completed this exercise.