

### TOC

TOC .....	1
LAB : Library .....	1
Step 1 – Create Library Fragments.....	1
Step 2 – Define Library in Root RAML file.....	6
Step 3 – Change Root RAML file and define items using library .....	7
Step 4 - Change Traits Library Fragment and define Error using datatypes library .....	8
Step 5– Change ResourceType Fragments and use library.....	9

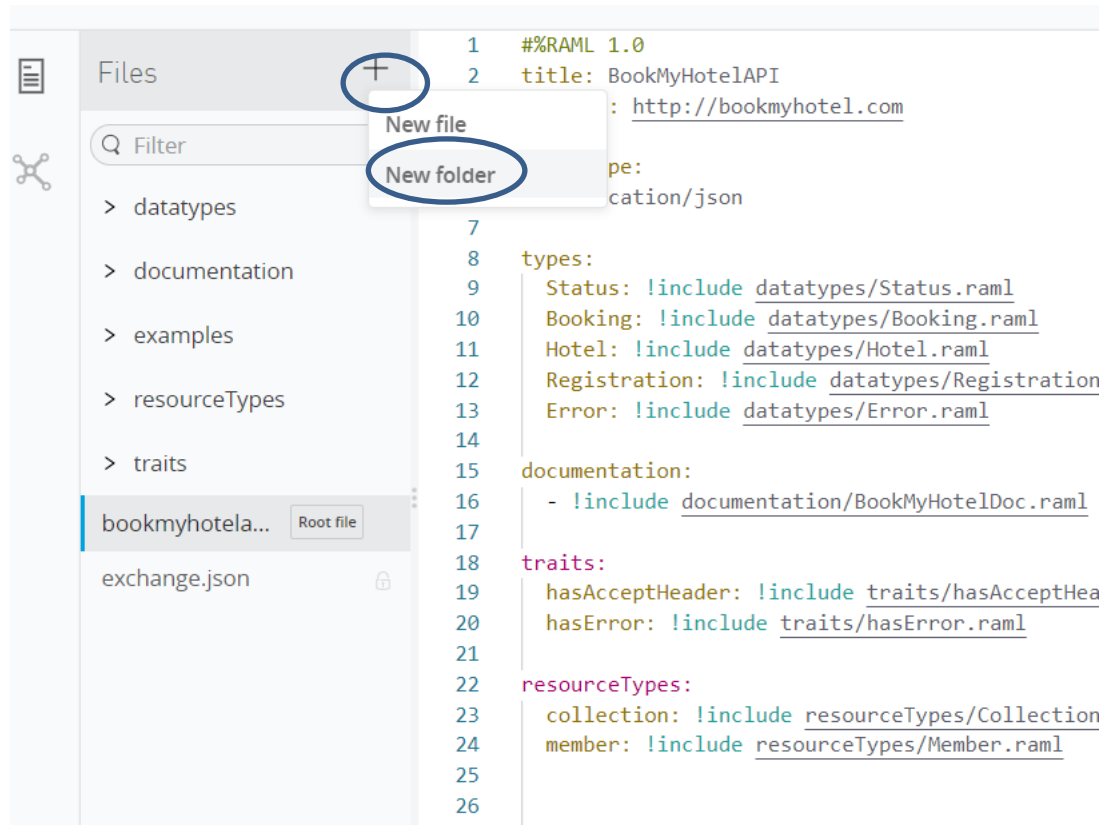
### LAB : Library

In this lab you will -

- a) Create Library fragments for traits, datatypes and ResourceTypes
- b) Define and Use Library

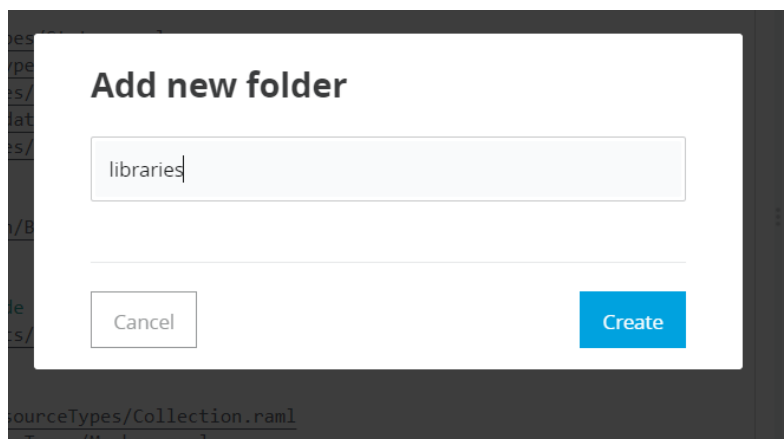
#### Step 1 – Create Library Fragments

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

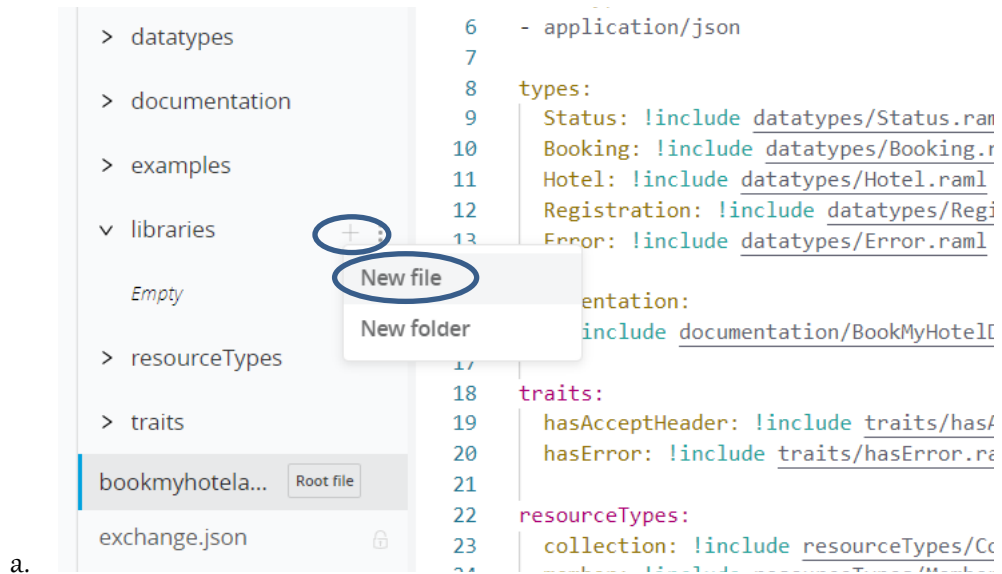


a.

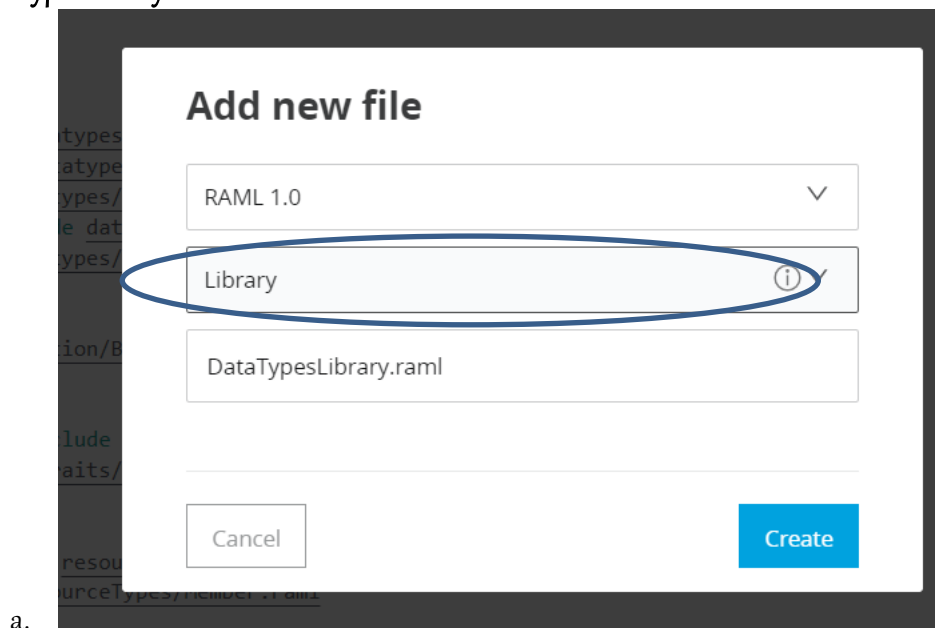
2. In the dialogue box, specify the name of the folder as **libraries** and then click on Create button



3. In the left hand side you can see the **libraries** folder created and it is empty as of now
4. Lets first create Datatypes **Library** fragment
5. Click on + sign near the libraries folder and click on new file



6. In the Add New File dialogue box, select **Library** from the drop down and specify the name as **DataTypesLibrary.raml**.



- b. This will create Library fragment – DataTypesLibrary.raml
7. Define this library fragment as shown below :-

```
1  #%RAML 1.0 Library
2  usage: Library of DataTypes
3  types:
4    Error: !include ../datatypes/Error.raml
5    Hotel: !include ../datatypes/Hotel.raml
6    Registration: !include ../datatypes/Registration.raml
7    Booking: !include ../datatypes/Booking.raml
8    Status: !include ../datatypes/Status.raml
9
```

- a.
  - b. This library fragment is using **types** node and then specifying the key value pairs under it
  - c. It has all the Datatype names and including all the existing datatype fragments
  - d. This is same as defined in our root raml file. Going forward, we will delete this section from the root raml file.
8. Lets now create ResourceTypes **Library** fragment
9. Click on + sign near the libraries folder and click on new file
10. In the Add New File dialogue box, select **Library** from the drop down and specify the name as ResourceTypesLibrary.raml
11. Define this library fragment as shown below :-

```
1  #%RAML 1.0 Library
2  usage: Library of ResourceTypes
3  resourceTypes:
4    collection: !include ../resourceTypes/Collection.raml
5    member: !include ../resourceTypes/Member.raml
6
```

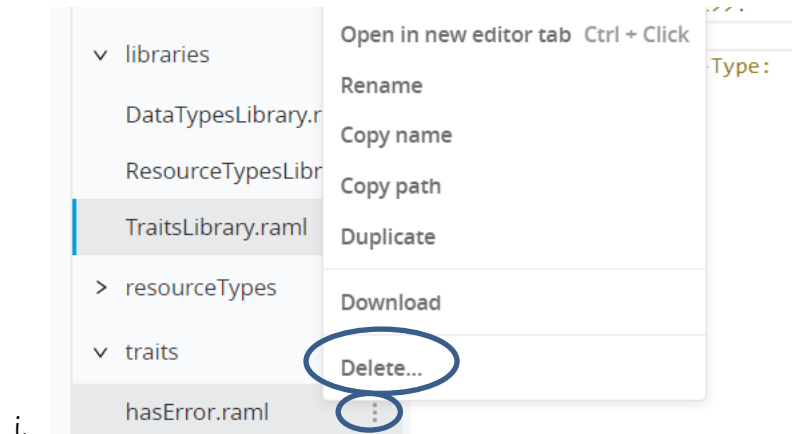
- a.
  - b. This library fragment is using **resourceTypes** node and then specifying the key value pairs under it
  - c. It has all the resource type names and including all the existing resource type fragments
  - d. This is same as defined in our root raml file. Going forward, we will delete this section from the root raml file.
12. Lets now create Traits **Library** fragment
13. Click on + sign near the libraries folder and click on new file

14. In the Add New File dialogue box, select **Library** from the drop down and specify the name as TraitsLibrary.raml

15. Define this library fragment as shown below :-

```
1  #%RAML 1.0 Library
2  ~
3  usage: Library of Traits
4  traits:
5    hasAcceptHeader:
6      headers:
7        Accept?:
8          example: application/json
9          description: MediaType of Expected Response
10   hasError:
11     responses:
12       <<errorCode>>:
13         headers:
14           Content-Type:
15             body:
16               Error
```

- a.
  - b. Here we have used traits node
  - c. Under this node , we have defined two traits
  - d. First one is named hasAcceptHeader and the value contains the whole definition of this trait (this is same as the one defined in hasAcceptHeader.raml – our trait fragment)
  - e. Second one is named hasError and the value contains the whole definition of this trait (this is same as the one defined in hasError.raml – our trait fragment)
  - f. In this case, we have complete definitions in the library fragment. We are not using the include tag to include the existing fragment.
16. Now delete the existing trait fragments as we don't need them
- a. Go to traits folder ,Click on three dots near the hasError trait fragment and click on delete



i.

ii. In the same way, delete other trait fragment also

**Congratulations, you have completed this step!!**

### Step 2 – Define Library in Root RAML file

1. In the root raml file, remove the declarations of traits, resource types and datatypes

```
1  #%RAML 1.0
2  title: BookMyHotelAPI
3  baseUrl: http://bookmyhotel.com
4
5  mediaType:
6  - application/json
7
8  documentation:
9  - !include documentation/BookMyHotelDoc.raml
10
11 types:
12   Status: !include datatypes/Status.raml
13   Booking: !include datatypes/Booking.raml
14   Hotel: !include datatypes/Hotel.raml
15   Registration: !include datatypes/Registration.raml
16   Error: !include datatypes/Error.raml
17
18
19 traits:
20   hasAcceptHeader: !include traits/hasAcceptHeader.raml
21   hasError: !include traits/hasError.raml
22
23 resourceTypes:
24   collection: !include resourceTypes/Collection.raml
25   member: !include resourceTypes/Member.raml
26
27
28
29
30 /registrations:
```

a.

Delete

2. Define **uses** node. Specify key as library name. Specify value as location to library fragments as shown below:-

```
#%RAML 1.0
title: BookMyHotelAPI
baseUri: http://bookmyhotel.com

mediaType:
- application/json

documentation:
- !include documentation/BookMyHotelDoc.raml

uses:
  traitsLib: libraries/TraitsLibrary.raml
  resTypesLib: libraries/ResourceTypesLibrary.raml
  dataTypesLib: libraries/DataTypesLibrary.raml
```

- a. `/registrations:`
- b. Notice that include tag is not required here

**Congratulations, you have completed this step!!**

### Step 3 – Change Root RAML file and define items using library

1. In the Root RAML file, Registrations resource (`/registrations`) and Registration (`/registrations/{registrationId}`) looks as follows as of now :-

```
/registrations:
  type: collection
  get:
  post:
  /{registrationId}:
    displayName: Registration
    description: Specific Registration
    type: member
    get:
    patch:
    delete:
```

- a.
  - b. Notice the red line under collection and member. It represents error and we are having this error because these are not found
2. Change it as follows :-

```
/registrations:  
  type: resTypesLib.collection  
  get:  
  post:  
    /{registrationId}:  
      displayName: Registration  
      description: Specific Registration  
      type: resTypesLib.member  
      get:  
      patch:  
      delete:
```

a.

b. Notice we have now used **resTypesLib.collection** and **resTypesLib.member** and it is picked successfully. We need to specify the `nameOfLibrary.itemInLibrary`

3. Make the same changes for all Resources in root raml file. Use **resTypesLib.collection** or **resTypesLib.member**.

**Congratulations, you have completed this step!!**

### Step 4 - Change Traits Library Fragment and define Error using datatypes library

1. Open Traits Library Fragment – `TraitsLibrary.raml`
2. As of now, your traits library looks like following –

```
1  #%RAML 1.0 Library  
2  usage: Library of Traits  
3  traits:  
4    hasAcceptHeader:  
5      headers:  
6        Accept?:  
7          example: application/json  
8          description: MediaType of Expected Response  
9    hasError:  
10     responses:  
11       <<errorCode>>:  
12         headers:  
13           Content-Type:  
14         body:  
15           Error
```

a.

3. Make changes and use `datatypes` library as shown below –



BookM

```
1  %%RAML 1.0 Library
2  usage: Library of Traits
3  traits:
4    hasAcceptHeader:
5      headers:
6        Accept?: |
7          example: application/json
8          description: MediaType of Expected Response
9    hasError:
10     responses:
11       <<errorCode>>:
12         headers:
13           Content-Type:
14           body:
15             dataTypesLib.Error
```

- a.
- b. We have used dataTypesLib.Error

**Congratulations, you have completed this step!!**

### Step 5– Change ResourceType Fragments and use library

1. Open Collection ResourceType Fragment – Collection.raml
2. Make changes in get and post methods as shown below :-

```
get?:
  displayName: GET <<resourcePathName | !uppercamelcase>>
  description: Retrieve a list of all <<resourcePathName | !uppercamelcase>>
  is:
    - traitsLib.hasAcceptHeader
    - traitsLib.hasError: {errorCode: 404}
  responses:
    200:
      headers:
        Content-Type:
      body:
        dataTypesLib.<<resourcePathName | !uppercamelcase | !singularize>>[]

post?:
  displayName: POST <<resourcePathName | !uppercamelcase | !singularize>>
  description: Add a new <<resourcePathName | !uppercamelcase | !singularize>>
  is:
    - traitsLib.hasError: {errorCode: 500}
  headers:
    Content-Type:
  body:
    dataTypesLib.<<resourcePathName | !uppercamelcase | !singularize>>
  responses:
    201:
      headers:
        Location:
      example: http://bookmyhotel.com/<<resourcePathName>>/r1
```

- a.
- b. For the traits, we have used traitsLib to access the trait
- c. For the datatypes, we have used dataTypesLib to access the datatype

3. Open Member ResourceType Fragment – Member.raml

4. The get method should be changed to look like following :-

```
get?:
  displayName: GET <<resourcePathName | !uppercamelcase | !singularize>>
  description: Retrieve a specific <<resourcePathName | !uppercamelcase | !singularize>>
  is:
    - traitsLib.hasAcceptHeader
    - traitsLib.hasError: {errorCode: 404}
  responses:
    200:
      headers:
        Content-Type:
      body:
        dataTypesLib.<<resourcePathName | !uppercamelcase | !singularize>>
```

a.

5. The patch method should be changed to look like following :-

```
patch?:
  displayName: PATCH <<resourcePathName | !uppercamelcase | !singularize>>
  description: Update a specific <<resourcePathName | !uppercamelcase | !singularize>>
  is:
    - traitsLib.hasError: {errorCode: 500}
  body:
    dataTypesLib.<<resourcePathName | !uppercamelcase | !singularize>>
  responses:
    204:
```

a.

6. The delete method should be changed to look like following :-

```
delete?:
  displayName: DELETE <<resourcePathName | !uppercamelcase | !singularize>>
  description: Delete a specific <<resourcePathName | !uppercamelcase | !singularize>>
  is:
    - traitsLib.hasAcceptHeader
    traitsLib.hasError: {errorCode: 500}
  responses:
    200:
      headers:
        Content-Type:
      body:
        dataTypesLib.Status
```

- a.
- b. For the traits, we have used *traitsLib* to access the trait
- c. For the datatypes, we have used *dataTypesLib* to access the datatype

***Congratulations, you have completed this step!!***

**Congratulations!! You have completed this exercise.**