

## TOC

|  |    |
|--|----|
| TOC .....  | 1  |
| LAB : ResourceTypes.....   | 1  |
| Step 1 – Create Collection ResourceType Fragment .....   | 1  |
| Step 2 – Define ResourceType in Root RAML using Collection ResourceType Fragment .....                 | 5  |
| Step 3 – Create Member ResourceType Fragment .....   | 7  |
| Step 4 – Define ResourceType in Root RAML using Member ResourceType Fragment .....                     | 9  |
| Step 5 (Optional) – Set displayName and description in Collection & Member ResourceType fragment ..... | 11 |

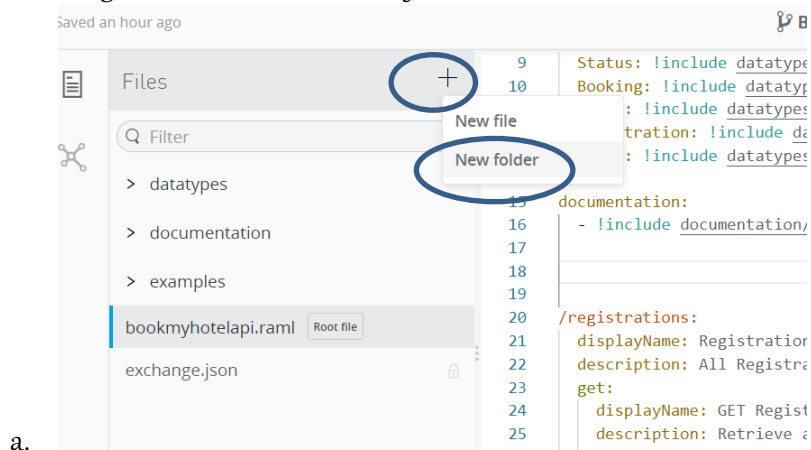
## LAB : ResourceTypes

In this lab you will -

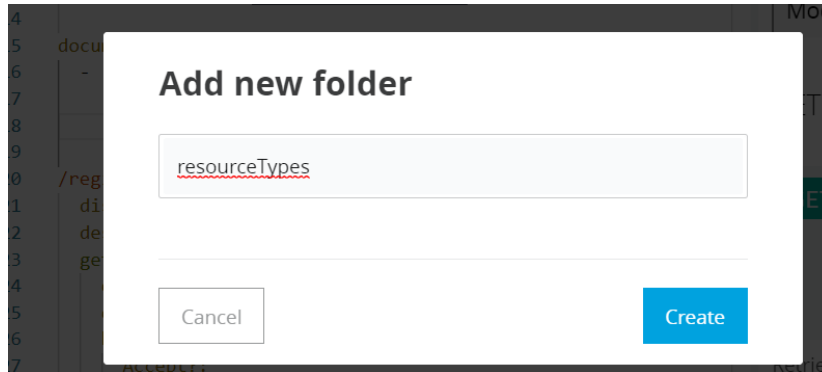
- Create Collection and Member ResourceType Fragment
- Use Collection and Member Resource Type fragment in Root RAML file

### Step 1 – Create Collection ResourceType Fragment

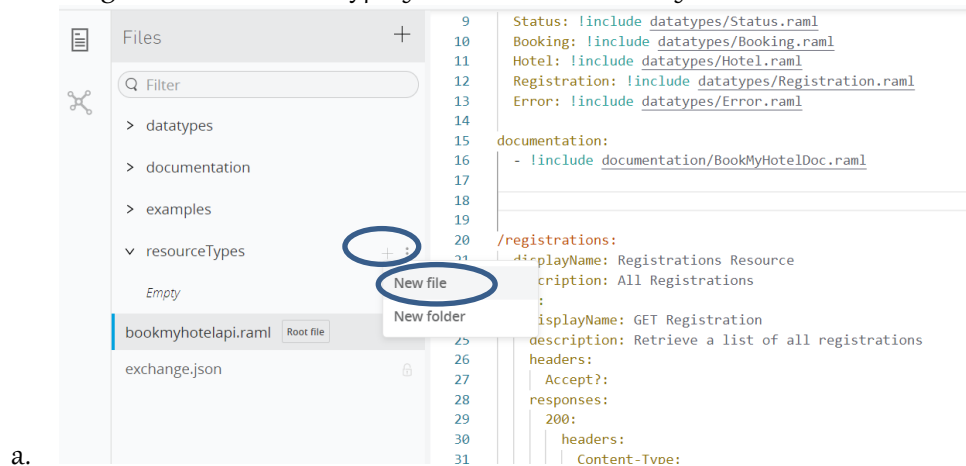
- Click on + sign and then Create Folder from the Menu as shown below :-



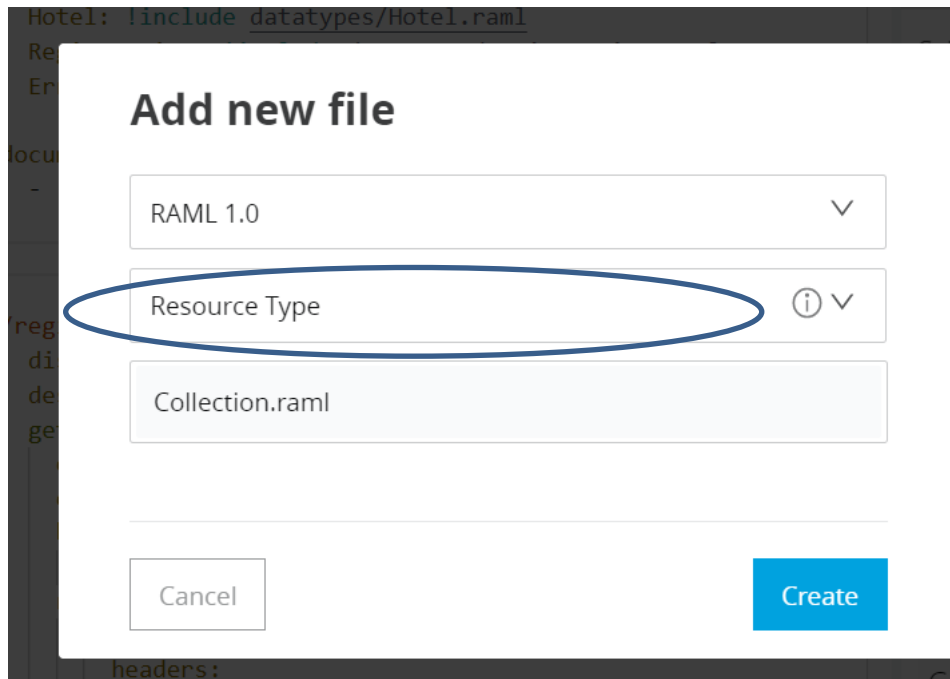
2. In the dialogue box, specify the name of the folder as **resourceTypes** and then click on Create button



3. In the left hand side you can see the resourceTypes folder created and it is empty as of now
4. Click on + sign near the resourceTypes folder and click on new file



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

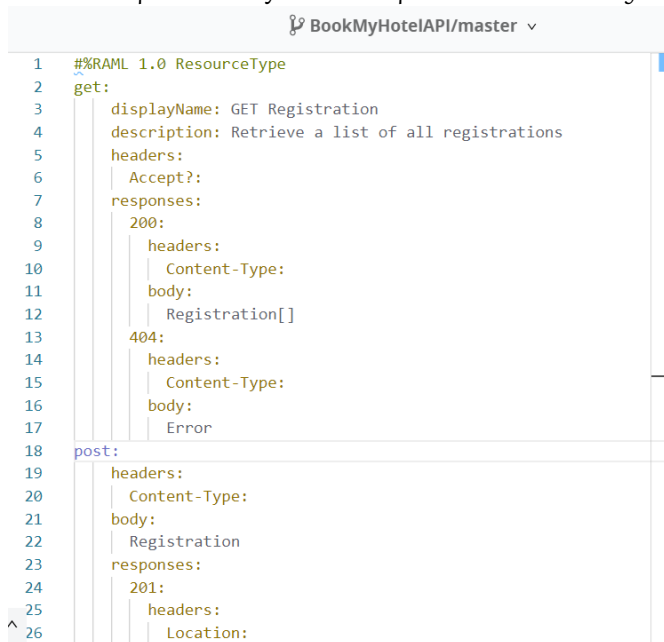


a.

b. This will create ResourceType fragment – Collection.raml

6. Go to Root RAML file and Copy the GET and POST method of registrations resource (/resgistrations)

7. In Collection.raml , paste what you have copied, it will look like following:-



a.

b. Make sure indentation is proper

8. Make GET and POST methods optional : use ? with method

```
1  #%RAML 1.0 ResourceType
2  get?:
3      displayName: GET Registration
4      description: Retrieve a list of all registrations
5      headers:
6          Accept?:
7      responses:
8          200:
9              headers:
10                 Content-Type:
11             body:
12                 Registration[]
13          404:
14              headers:
15                 Content-Type:
16             body:
17                 Error
18  post?:
19      headers:
20          Content-Type:
21      body:
22          Registration
23      responses:
24          201:
25              headers:
26                 Location:
```

a.

9. Change the GET method in Collection ResourceType fragment as shown below :-

```
1  #%RAML 1.0 ResourceType
2  get?:
3      displayName: GET <<resourcePathName | !uppercamelcase>>
4      description: Retrieve a list of all <<resourcePathName |
5      !uppercamelcase>>
6      headers:
7          Accept?:
8      responses:
9          200:
10              headers:
11                 Content-Type:
12             body:
13                 <<resourcePathName | !uppercamelcase |
14                 !singularize>>[]
15          404:
16              headers:
17                 Content-Type:
18             body:
19                 Error
```

a.

10. Change the POST method in Collection ResourceType fragment as shown below:

```
18  post?:
19      displayName: POST <<resourcePathName | !uppercamelcase | !singularize>>
20      description: Add a new <<resourcePathName | !uppercamelcase |
21      !singularize>>
22      headers:
23          Content-Type:
24      body:
25          <<resourcePathName | !uppercamelcase | !singularize>>
26      responses:
27          201:
28              headers:
29                 Location:
30                 example: http://bookmyhotel.com/<<resourcePathName>>/r1
31          500:
32              headers:
33                 Content-Type:
34             body:
35                 Error
```

a.

11. Lets look into the expressions of GET and POST method

- a. For /registrations,
  - i. <<resourcePathName | !uppercamelcase>> will result in **Registrations**
  - ii. <<resourcePathName | !uppercamelcase | !singularize>> will result in **Registration**
- b. For /hotels
  - i. <<resourcePathName | !uppercamelcase>> will result in **Hotels**
  - ii. <<resourcePathName | !uppercamelcase | !singularize>> will result in **Hotel**
- c. Same for other collection resources

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

### Step 2 – Define ResourceType in Root RAML using Collection ResourceType Fragment

1. In the root raml file, use **resourceTypes** node as shown below:-

```
1  #%RAML 1.0
2  title: BookMyHotelAPI
3  baseUrl: 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  resourceTypes:
19    collection: !include resourceTypes/Collection.raml
20
21
22
```

- a.
- b. Under resourceTypes: node, key value pair is specified

- c. We have specified the key name as **collection** (any name of your choice can be specified) and in the value, we have included the Collection ResourceType fragment
2. In the root raml file, remove everything from get and post method of Registrations resource (/registrations)
3. Then using collection resource type, define as follows :-

```
/registrations:
  displayName: Registrations Resource
  description: All Registrations
  type: collection
  get:
  post:
  /{registrationId}:
    displayName: Registration
    description: Specific Registration
    get:
      displayName: GET Registration
      description: Retrieve a specific Registration
    headers:
      Accept?:
    responses:
```

a.

4. For Hotels resource (/hotels), first copy the query parameters –

```
0 /hotels:
1   get:
2     headers:
3       Accept?:
4     queryParameters:
5       city:
6         required: false
7       state:
8         required: false
9       country:
10        required: false
11     responses:
12       200:
13         headers:
14           Content-Type:
15         body:
```

a.

5. Now remove everything from the get and post method of Hotels resource (/hotels)
6. Define as follows :-

```
/hotels:
  displayName: Hotels
  description: All Hotels
  type: collection
  get:
    queryParameters:
      city:
        required: false
      state:
        required: false
      country:
        required: false
  post:
```

a.

7. We have specified type as collection and then specified get and post methods
8. For get method, we need to paste the query parameters. These query parameters are not part of resource type fragment and is only specific to hotels resource. In root raml, we can add nodes which are not available in resource type fragments.
9. For `/hotels/{hotelId}/bookings`, we only have get method so define as follows :-

```
/bookings:
  displayName: Bookings
  description: Bookings of specific Hotel
  type: collection
  get:
```

a.

10. For Bookings Resource (`/bookings`), define as follows :-

```
/bookings:
  displayName: Bookings
  description: All Bookings
  type: collection
  get:
  post:
```

a.

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

### Step 3 – Create Member ResourceType Fragment

1. Click on + sign near the resourceTypes folder and click on new file
2. In the Add New File dialogue box, select **ResourceType** from the drop down and specify the name as **Member.raml**. This will create ResourceType fragment – Member.raml

3. Go to Root RAML file and Copy the GET ,DELETE and PATCH method of registration resource (/resgistrations/{registrationId}).
4. In Member.raml , paste what you have copied, it will look like following:-

BookMyHotelAPI/master v

```
1  #%RAML 1.0 ResourceType
2  displayName: Registration
3  description: Specific Registration
4  get:
5    displayName: GET Registration
5    description: Retrieve a specific Registration
7    headers:
3    Accept?:
3    responses:
3    200:
1      headers:
2      Content-Type:
3      body:
4    404:
5      headers:
5      Content-Type:
7      body:
3      Error
3  delete:
3    displayName: DELETE Registration
1    description: Delete a specific Registration
2    headers:
3    Accept?:
4    responses:
5    200:
```

a.

- b. Above screenshot only shows get and delete methods but all three methods are pasted (get,delete and patch)
  - c. Make sure indentation is proper otherwise you will get error
5. Make these methods optional (use ? with method)
  6. Change the GET method in Member ResourceType fragment as shown below :-



a. 

```
get?:
  displayName: GET <<resourcePathName | !uppercamelcase | !singularize>>
  description: Retrieve a specific <<resourcePathName | !uppercamelcase | !singularize>>
  headers:
    Accept?:
  responses:
    200:
      headers:
        Content-Type:
      body:
        <<resourcePathName | !uppercamelcase | !singularize>>
    404:
      headers:
        Content-Type:
      body:
        Error
```

7. Change the DELETE method in Member ResourceType fragment as shown below:

a. 

```
delete?:
  displayName: DELETE <<resourcePathName | !uppercamelcase | !singularize>>
  description: Delete a specific <<resourcePathName | !uppercamelcase | !singularize>>
  headers:
    Accept?:
  responses:
    200:
      headers:
        Content-Type:
      body:
        Status
    404:
      headers:
        Content-Type:
      body:
        Error
```

8. Change the PATCH method in Member ResourceType fragment as shown below:

a. 

```
patch?:
  displayName: PATCH <<resourcePathName | !uppercamelcase | !singularize>>
  description: Update a specific <<resourcePathName | !uppercamelcase | !singularize>>
  body:
    <<resourcePathName | !uppercamelcase | !singularize>>
  responses:
    204:
    500:
      headers:
        Content-Type:
      body:
        Error
```

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

### Step 4 – Define ResourceType in Root RAML using Member ResourceType Fragment

1. In the root raml file, for resourceTypes node – define and include Member resource type fragment as shown below:-

```
1  #%RAML 1.0
2  title: BookMyHotelAPI
3  baseUrl: 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  resourceTypes:
19    collection: !include resourceTypes/Collection.raml
20    member: !include resourceTypes/Member.raml
```

- a.
- b. Under resourceTypes: node, key value pair is specified
- c. We have specified the key name as **member** (any name of your choice can be specified) and in the value , we have included the Member ResourceType fragment

11. In the root raml file, remove everything from all methods of registration resource (/registrations/{registrationId})
12. Then use member resource type to define these methods as follows :-

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

- a.
13. For Hotel resource (/hotels/{hotelId}), remove everything from all methods
14. Now use resource type to define these methods as follows :-

```
/hotels:
  displayName: Hotels
  description: All Hotels
  type: collection
  get:
    queryParameters:
      city:
        required: false
      state:
        required: false
      country:
        required: false
  post:
    /{hotelId}:
      displayName: Hotel
      description: Specific Hotel
      type: member
      get:
      patch:
      delete:
```

a.

15. For Booking resource (/bookings/{bookingId}), remove everything from all methods

16. Now use resource type to define these methods as follows :-

```
/bookings:
  displayName: Bookings
  description: All Bookings
  type: collection
  get:
  post:
    /{bookingId}:
      displayName: Booking
      description: Specific Booking
      type: member
      get:
      patch:
      delete:
```

a.

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

### Step 5 (Optional) – Set displayName and description in Collection & Member ResourceType fragment

1. Copy displayName and description nodes of Collection Resource Registrations (/registrations) –

```
/registrations:
  displayName: Registrations
  description: All Registrations
  type: collection
  get:
  post:
```

2.

3. Paste these in collection resource type fragment as shown below :-

```
1  #%RAML 1.0 ResourceType
2  displayName: Registrations
3  description: All Registrations
4  get?:
5      displayName: GET <<resourcePathName | !uppercamelcase>>
6      description: Retrieve a list of all <<resourcePathName | !uppercamelcase>>
7      headers:
8      Accept?:
9      responses:
10     200:
11         headers:
12         Content-Type:
13         body:
14             <<resourcePathName | !uppercamelcase | !singularize>>[]
15     404:
16         headers:
17         Content-Type:
18         body:
19             Error
20 post?:
21     displayName: POST <<resourcePathName | !uppercamelcase | !singularize>>
22     description: Add a new <<resourcePathName | !uppercamelcase | !singularize>>
```

- 4.
5. Change it using reserved parameter and function as shown below –

```
1  #%RAML 1.0 ResourceType
2  displayName: <<resourcePathName | !uppercamelcase>>
3  description: All <<resourcePathName | !uppercamelcase>>
4  get?:
5      displayName: GET <<resourcePathName | !uppercamelcase>>
6      description: Retrieve a list of all <<resourcePathName | !uppercamelcase>>
7      headers:
8      Accept?:
9      responses:
10     200:
11         headers:
12         Content-Type:
13         body:
14             <<resourcePathName | !uppercamelcase | !singularize>>[]
15     404:
16         headers:
17         Content-Type:
18         body:
19             Error
20 post?:
21     displayName: POST <<resourcePathName | !uppercamelcase | !singularize>>
22     description: Add a new <<resourcePathName | !uppercamelcase | !singularize>>
23     headers:
24     Content-Type:
25     body:
26         <<resourcePathName | !uppercamelcase | !singularize>>
27     responses:
```

- 6.
7. In **root raml**, remove displayName and description for all collection resources – Registrations (/registrations), Hotels (/hotels) and Bookings (/bookings)
8. Keep the displayName and description of /hotels/{hotelId}/bookings as it is
9. Copy displayName and description nodes of Member Resource Registration (/registrations/{registrationId}) –

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

a.

10. Paste these in Member resource type fragment as shown below :-

```
1  #RAML 1.0 ResourceType
2  displayName: Registration
3  description: Specific Registration
4  get?:
5    displayName: GET <<resourcePathName | !uppercamelcase | !singularize>>
6    description: Retrieve a specific <<resourcePathName | !uppercamelcase | !singularize>>
7    headers:
8    Accept?:
9    responses:
10   200:
11     headers:
12     Content-Type:
13     body:
14       <<resourcePathName | !uppercamelcase | !singularize>>
15   404:
16     headers:
17     Content-Type:
18     body:
19       Error
20 patch?:
21   displayName: PATCH <<resourcePathName | !uppercamelcase | !singularize>>
22   description: Update a specific <<resourcePathName | !uppercamelcase | !singularize>>
```

11.

12. Change it using reserved parameter and function as shown below –

```
1  #RAML 1.0 ResourceType
2  displayName: <<resourcePathName | !uppercamelcase | !singularize>>
3  description: Specific <<resourcePathName | !uppercamelcase | !singularize>>
4  get?:
5    displayName: GET <<resourcePathName | !uppercamelcase | !singularize>>
6    description: Retrieve a specific <<resourcePathName | !uppercamelcase | !singularize>>
7    headers:
8    Accept?:
9    responses:
10   200:
11     headers:
12     Content-Type:
13     body:
14       <<resourcePathName | !uppercamelcase | !singularize>>
15   404:
16     headers:
17     Content-Type:
18     body:
19       Error
20 patch?:
21   displayName: PATCH <<resourcePathName | !uppercamelcase | !singularize>>
22   description: Update a specific <<resourcePathName | !uppercamelcase | !singularize>>
```

13.

14. In **root.raml**, remove *displayName* and *description* of all Member resources – Registration (/registrations/{registrationId}) , Hotels (/hotels/{hotelId}) and Booking (/bookings/{bookingId})
15. No further changes are required in **root.raml** file as all resources are already specifying type as collection or member

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

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