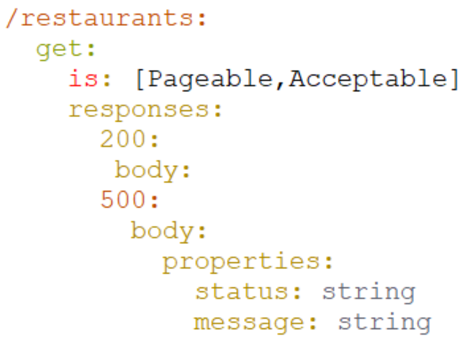
LAB- Api Design

Defining Data types

After baseUri in the top of the file, add mediatype as application/json

Now we want to start describing the responses for each resource.

Under Get /restaurants, describe response status codes as shown below:

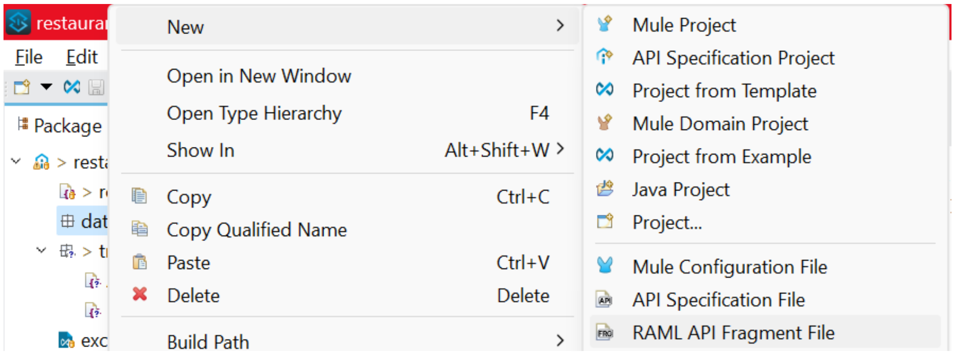


```
/restaurants:
  get:
    is: [Pageable,Acceptable]
    responses:
      200:
       body:
      500:
        body:
          properties:
            status: string
            message: string
```

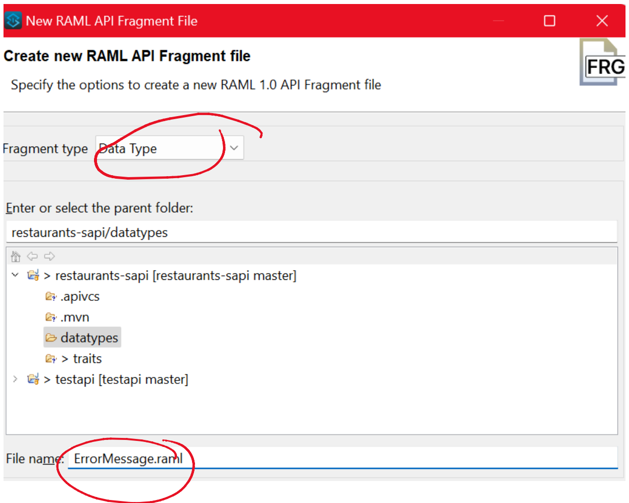
As we might want to describe about 500 status code for many other resources, better to create our own Datatype which describes the properties.

Create a new folder with name “datatypes” under the project folder

Right click on the datatypes folder, select new -> RAML API Fragment File as shown below



Select Fragment Type as “Data Type ” and give the name as ErrorMessage.raml


New RAML API Fragment File
FRG

Create new RAML API Fragment file

Specify the options to create a new RAML 1.0 API Fragment file

Fragment type: Data Type

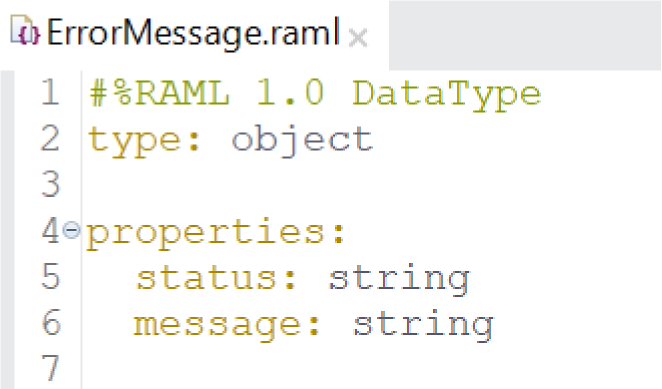
Enter or select the parent folder:

restaurants-sapi/datatypes

restaurants-sapi [restaurants-sapi master]
.apivcs
.mvn
datatypes
traits
testapi [testapi master]

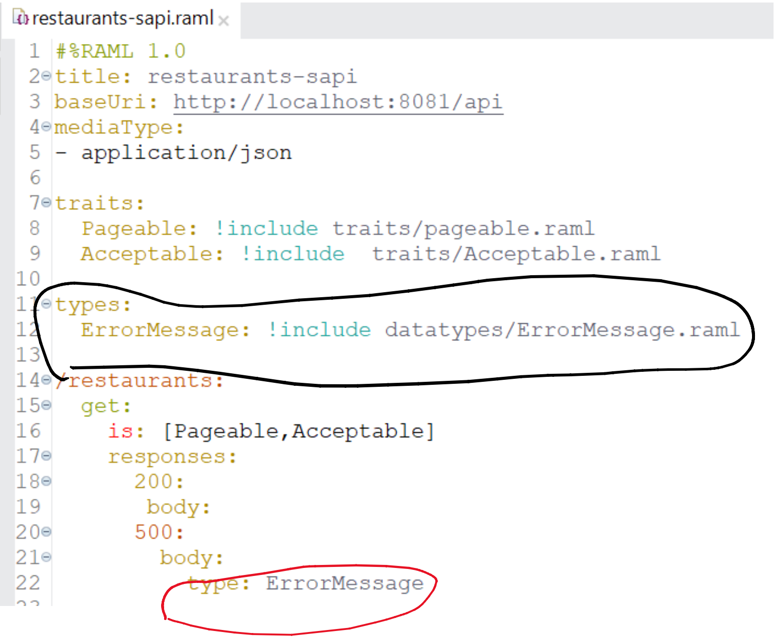
File name: ErrorMessage.raml

Write ErrorMessage.raml as shown below:



```
#%RAML 1.0 DataType
type: object

properties:
  status: string
  message: string

```

Now, we have to define this datatype in main raml as shown below:

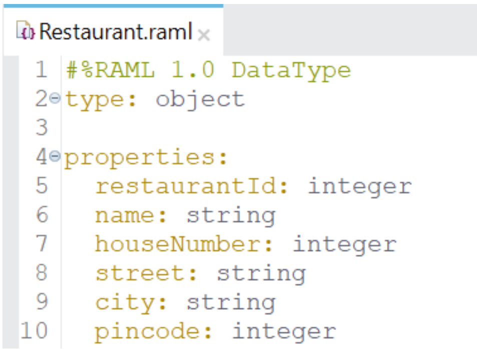


restaurants-sapi.raml

```
#%RAML 1.0
title: restaurants-sapi
baseUri: http://localhost:8081/api
mediaType:
- application/json

traits:
  Pageable: !include traits/pageable.raml
  Acceptable: !include  traits/Acceptable.raml

types:
  ErrorMessage: !include datatypes/ErrorMessage.raml

/restaurants:
  get:
    is: [Pageable,Acceptable]
    responses:
      200:
       body:
      500:
        body:
          type: ErrorMessage
```

Simlary, we want to describe a data type for Restaurant. Under datatypes folder, create another data type with name Restaurant.raml.

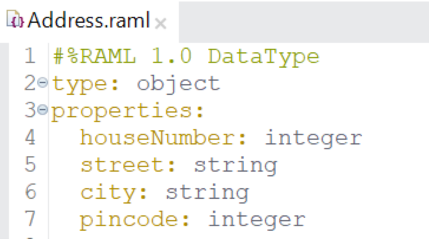
It should look like below:



Restaurant.raml

```
#%RAML 1.0 DataType
type: object

properties:
  restaurantId: integer
  name: string
  houseNumber: integer
  street: string
  city: string
  pincode: integer
```

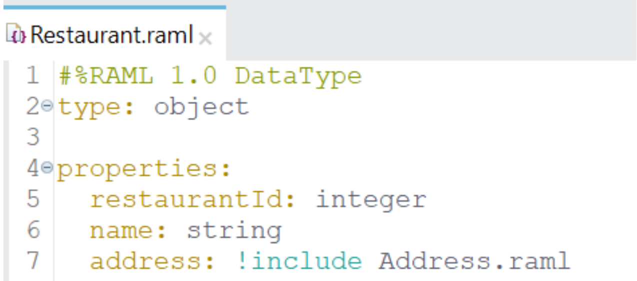
We want to describe all the address related properties in its own datatype with name Address. So, Create Address.raml under datatypes folder. It should look like below:



Address.raml

```
#%RAML 1.0 DataType
type: object
properties:
  houseNumber: integer
  street: string
  city: string
  pincode: integer
```

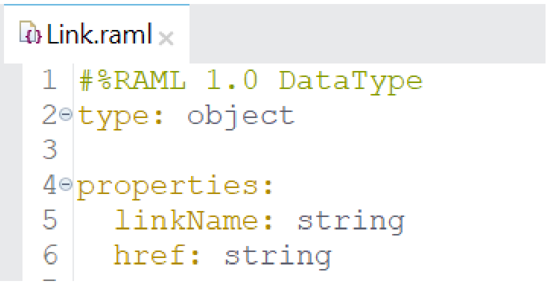
Now modify Restaurant.raml to use Address.raml as shown below:



```
#%RAML 1.0 DataType
type: object

properties:
  restaurantId: integer
  name: string
  address: !include Address.raml
```

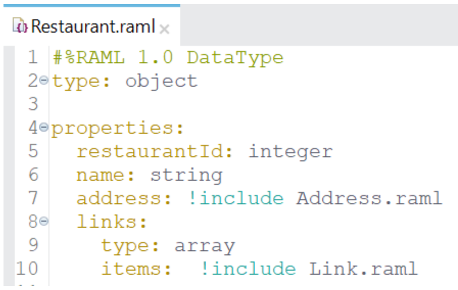
Now We want to add link retieve all menus of a restaurant

Create a Data type with name “Link.raml” as shown below:



```
#%RAML 1.0 DataType
type: object

properties:
  linkName: string
  href: string
```

Now update Restaurant.raml as shown below:



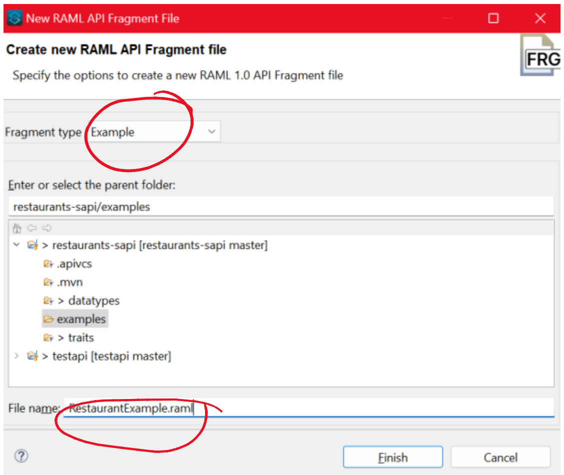
```
#%RAML 1.0 DataType
type: object

properties:
  restaurantId: integer
  name: string
  address: !include Address.raml
  links:
    type: array
    items:  !include Link.raml
```

We want to describe an example for Restaurant Datatype.

Create a new folder with name “examples” under the project folder.

Right click on examples folder and slect new -> Raml API Fragment File

Select the Fragment type as “Example” and give name as “RestaurantExample.raml” as shown below:



Create new RAML API Fragment file

Specify the options to create a new RAML 1.0 API Fragment file

Fragment type: Example

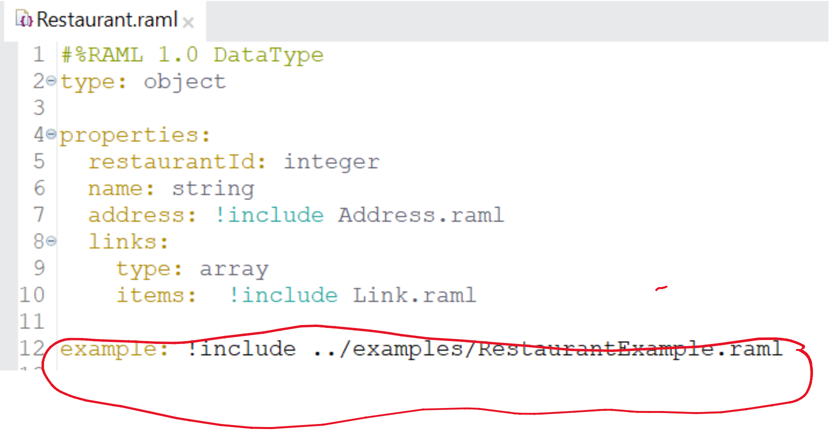
Enter or select the parent folder: restaurants-sapi/examples

File name: RestaurantExample.raml

Copy the below content into the file

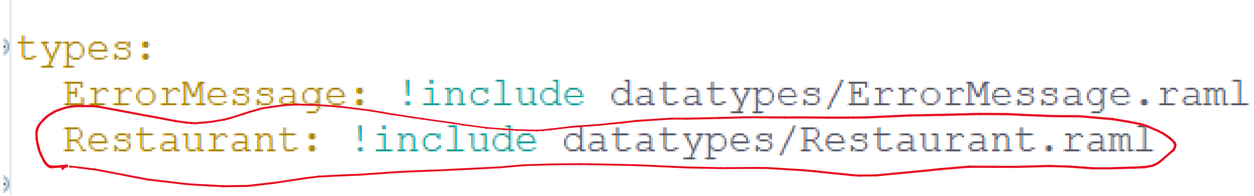
```
#%RAML 1.0 NamedExample
value:
  restaurantId: 1
  name: "Udupi Garden"
  address:
    houseNumber: 202
    street: "Some Street"
    city: "Bangalore"
    pincode: 560066
  links:
  -
    linkName: "menus"
    href: "http://localhost:8081/restaurants/1/menus"
```

Now, refer to this example fragment from Restaturant.raml as shown below:



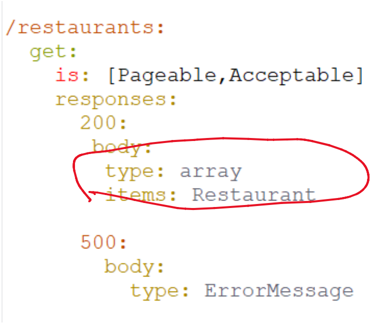
```
#%RAML 1.0 DataType
type: object

properties:
  restaurantId: integer
  name: string
  address: !include Address.raml
  links:
    type: array
    items:  !include Link.raml

example: !include ../examples/RestaurantExample.raml
```

Now , declare the Restaurant Datatype in main raml as shown below:



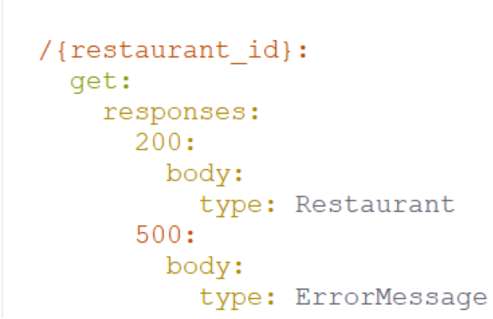
```
types:
  ErrorMessage: !include datatypes/ErrorMessage.raml
  Restaurant: !include datatypes/Restaurant.raml
```

Describe 200 status code of GET /restaurants as shown below



```
/restaurants:
  get:
    is: [Pageable,Acceptable]
    responses:
      200:
        body:
          type: array
          items: Restaurant

      500:
        body:
          type: ErrorMessage
```

Describe 200 and 500 status codes for   GET /restaurants/{restaurant_id}



```
  /{restaurant_id}:
    get:
      responses:
        200:
          body:
            type: Restaurant
        500:
          body:
            type: ErrorMessage
```

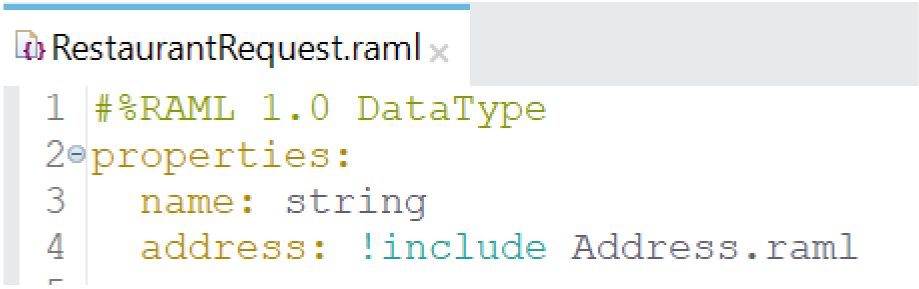
Now we want to describe the request body got POST /restaurants

Can we use Restaurant Datatype for describing body to POST /restaurants?

No. Because links doesnot make sense when adding a restaurant. Also, in request body, we don’t want to send restaurantId.

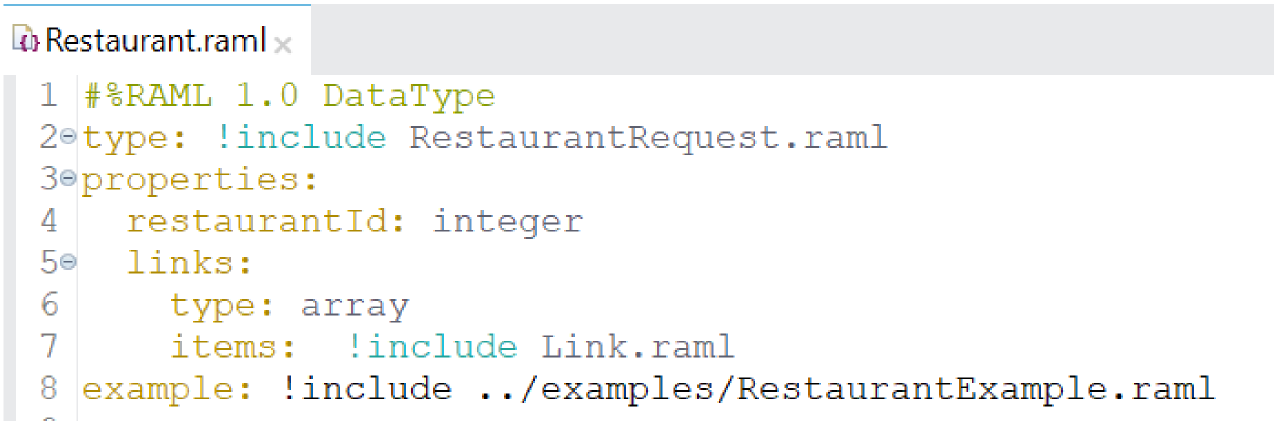
So, Better to Create a separate DataType for request body

Create a new DataType with name RestaurantRequest.raml as shown below:



```
#%RAML 1.0 DataType
properties:
  name: string
  address: !include Address.raml
```

We can update Restaurant.raml to inherit from RestaurantRequest.raml as shown below:

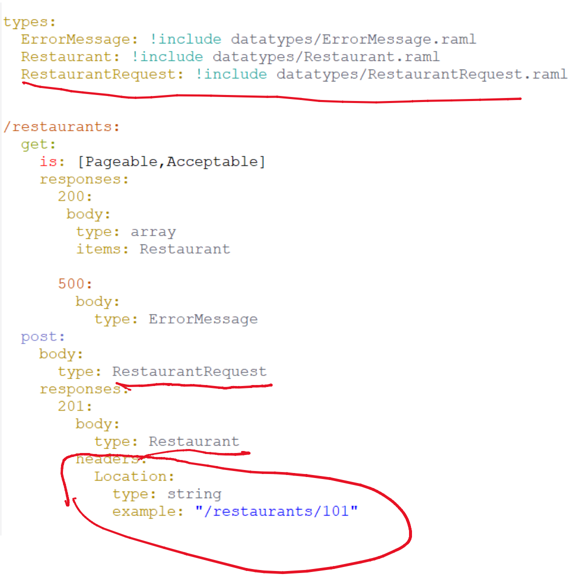


```
#%RAML 1.0 DataType
type: !include RestaurantRequest.raml
properties:
  restaurantId: integer
  links:
    type: array
    items:  !include Link.raml
example: !include ../examples/RestaurantExample.raml
```

Now declare RestaurantRequest Datatype in the main raml and use it for describing POST /restaurants as shown below:

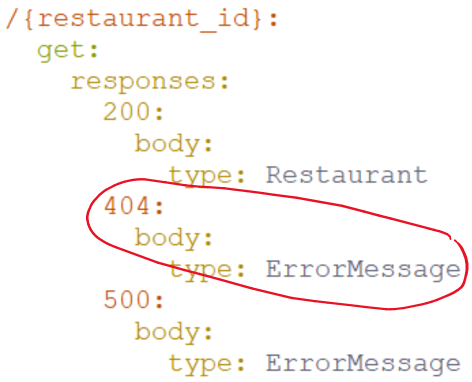
Also describe 201 status code for POST /restaurants as shown below:

Observe that we have used Restaurant as response body for 201 status code



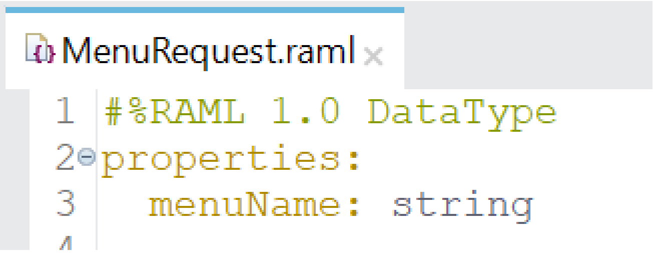
```
types:
  ErrorMessage: !include datatypes/ErrorMessage.raml
  Restaurant: !include datatypes/Restaurant.raml
  RestaurantRequest: !include datatypes/RestaurantRequest.raml

/restaurants:
  get:
    is: [Pageable,Acceptable]
    responses:
      200:
       body:
        type: array
        items: Restaurant

      500:
        body:
          type: ErrorMessage
  post:
    body:
      type: RestaurantRequest
    responses:
      201:
        body:
          type: Restaurant
        headers:
          Location:
            type: string
            example: "/restaurants/101"
```

Describe 404 status code for restaurants/{restaurant_id } as shown below

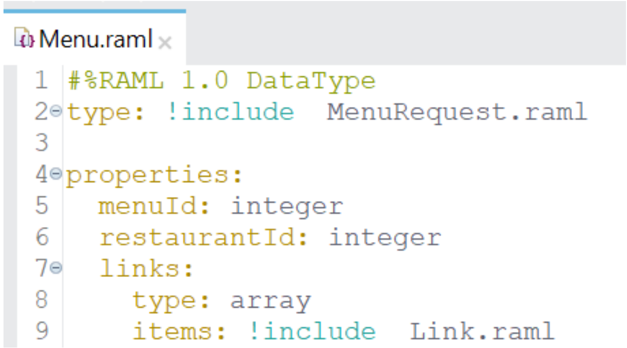


```
/{restaurant_id}:
  get:
    responses:
      200:
        body:
          type: Restaurant
      404:
        body:
          type: ErrorMessage
      500:
        body:
          type: ErrorMessage
```

Similar to Restaurant Datatypes, create data types for Menu and MenuRequest as show below:

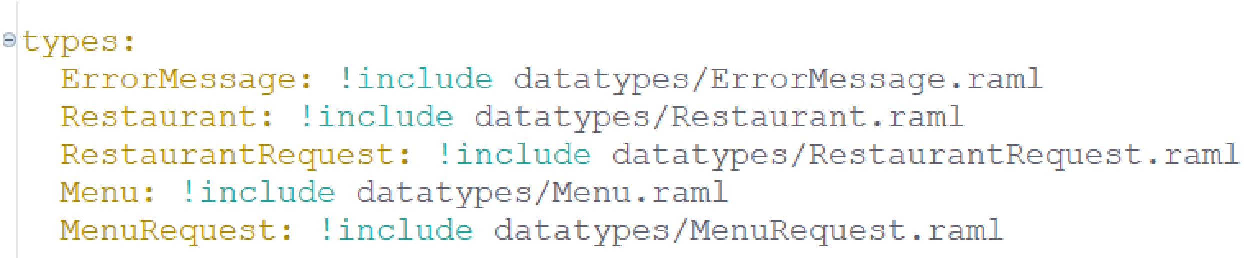


```
#%RAML 1.0 DataType
properties:
  menuName: string
```



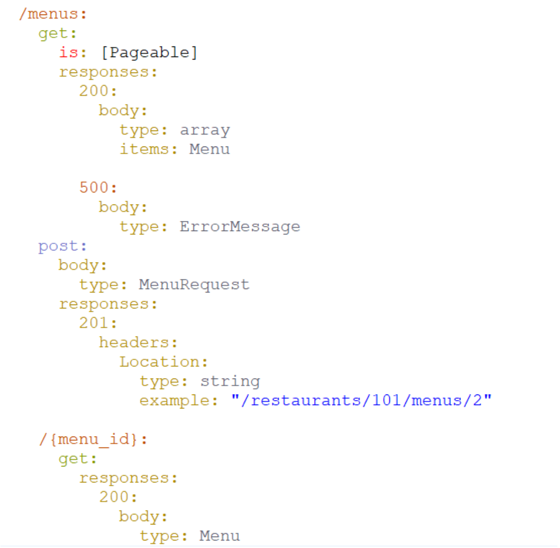
```
#%RAML 1.0 DataType
type: !include  MenuRequest.raml

properties:
  menuId: integer
  restaurantId: integer
  links:
    type: array
    items: !include  Link.raml
```

Declare these datatypes in main raml as shown below:



```
types:
  ErrorMessage: !include datatypes/ErrorMessage.raml
  Restaurant: !include datatypes/Restaurant.raml
  RestaurantRequest: !include datatypes/RestaurantRequest.raml
  Menu: !include datatypes/Menu.raml
  MenuRequest: !include datatypes/MenuRequest.raml
```

use above datatypes   for describing /restaurants/{restaurant_id}/menus and /restaurants/{restaurant_id}/menus/{menu_id} as shown below



```
/menus:
  get:
    is: [Pageable]
    responses:
      200:
        body:
          type: array
          items: Menu

      500:
        body:
          type: ErrorMessage
  post:
    body:
      type: MenuRequest
    responses:
      201:
        headers:
          Location:
            type: string
            example: "/restaurants/101/menus/2"

  /{menu_id}:
    get:
      responses:
        200:
          body:
            type: Menu
```

This is the end of the Exercise