

LAB- Api Design

Defining resourceTypes


Create a folder with name “resourceTypes” under your project

Copy the files collection.raml and member.raml given to you

Open collection.raml and observe how we have described the get and post as optional methods. Observe how we defined place holders

Similarly, open member.raml and understand it

Now, defined these resource types in main raml as shown below:



```
1 #%RAML 1.0
2 title: restaurants-sapi
3 baseUrl: http://localhost:8081/api
4 mediaType:
5   - application/json
6
7
8 resourceTypes:
9   collection: !include resourceTypes/collection.raml
10  member: !include resourceTypes/member.raml
11
12 traits:
13   Pageable: !include traits/pageable.raml
14   Acceptable: !include traits/Acceptable.raml
15
```

Apply these resource types as shown below:

```
» /restaurants:
  type: collection
  get:
    is: [Pageable,Acceptable]
  post:
    /{restaurant_id}:
      type: member
      get:
      put:
      delete:
    /menus:
      type: collection
      get:
        is: [Pageable]
      post:
        /{menu_Id}:
          type: member
          get:
          delete:
```

Publishing to Exchange

We want to define a reusable fragment for datatypes.

Create a new Api specification Project

Select “New API Fragment” tab and give the name as “restaurantdatatypes”

Select fragment type as “Data Type” and click Finish.

New API Specification

Create a new API Specification Project

Create an API Specification or fragment from scratch

Username Business Group

New API Specification **New API Fragment**

Project Name

Select your fragment type ✓

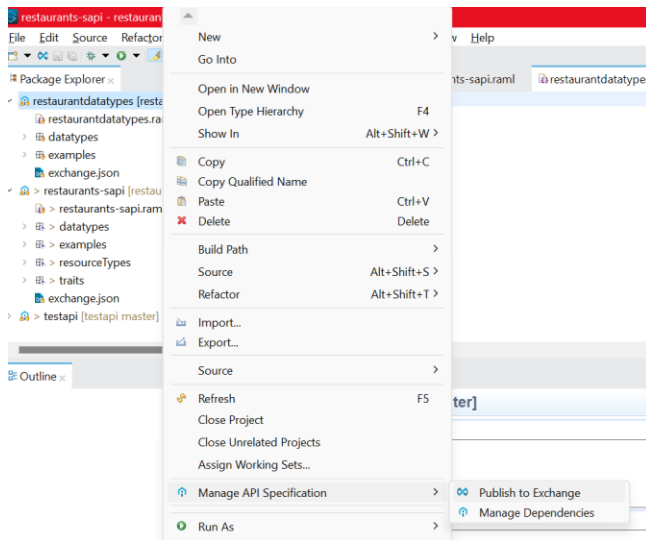
① An API Specification will be cloned in Anypoint Design Center after being created.

Copy the `datatypes` and `examples` folders from your API specification project into the API fragment project.

You can delete both these folders from the main API specification project.

Commit the changes in the fragment project and push the changes.

Now, right-click on the fragment project and select “manage API specification”
click publish to exchange as shown below:

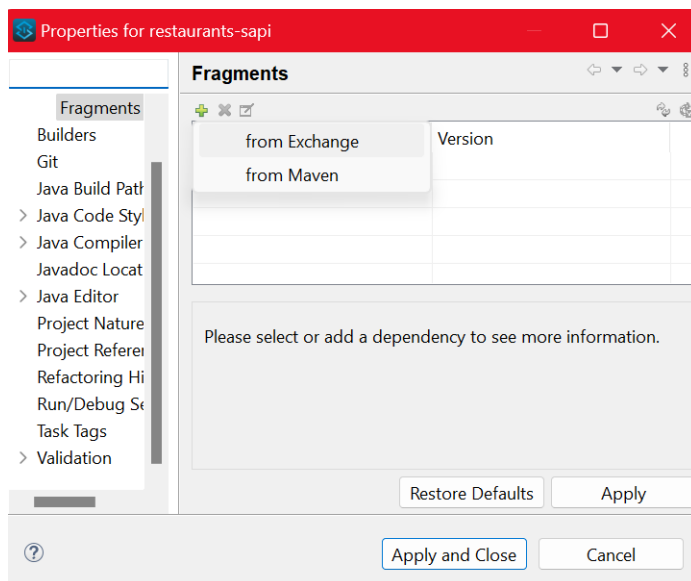


Give asset version as 1.0.0 and click finish

Your asset will be published to exchange

Now Right click on the restaurant-sapi project and select “manage API Specification” and click “Manage Dependencies”

Click on “+” and select “from Exchange”



Search for restaurantdatatypes and select it, add it and finish.

This fragment will be added as a dependency to the restaurants-sapi project

Now open the main raml file of the restaurants-sapi.raml

Update the !include in all datatypes as

```
!include /exchange_modules/{yourorgid}/restaurantdatatypes/1.0.0/datatypes/<<filename>>.raml
```

Update collection.raml and member.raml accordingly

All errors should be gone

Finally, publish your api specification project also in to exchange.

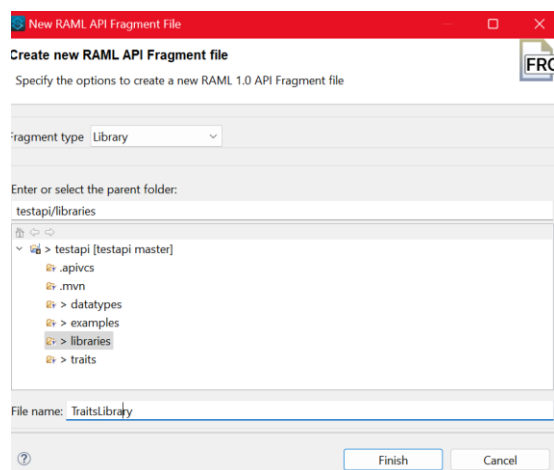
Create Libraries

Now we want to create a single library of traits ,types and resource types

Create a folder with name libraries inside your api specification project

Right click on the folder and select new->Raml Api Fragment file

Select fragment type as Library and give the file name as TraitsLibrary.raml as shown below:



Cut the traits definition in the main raml file and write the following in the library file

```
#%RAML 1.0 Library

traits:
  Pageable: !include ../traits/Pageable.raml
  Acceptable: !include ../traits/MyAcceptable.raml
```

In the main raml, write like below:

```
uses:
  traitslibrary: libraries/TraitsLibrary.raml
```

Now where ever we were using traits, replace it like below:

```
» /restaurants:
»   get:
      is: [traitslibrary.Pageable, traitslibrary.Acceptable]
```

Congratulations!! You know how to create a RAML.

This is the end of the Exercise