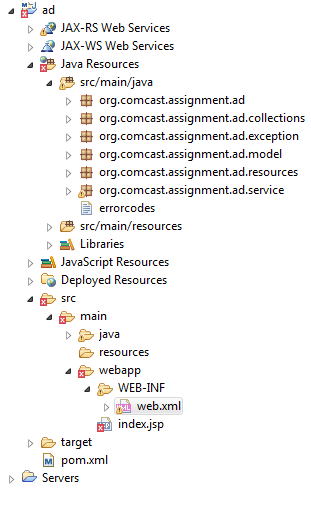
Instructions :

1 This is a maven jersey project, so configure the pom.xml and download the jar files.

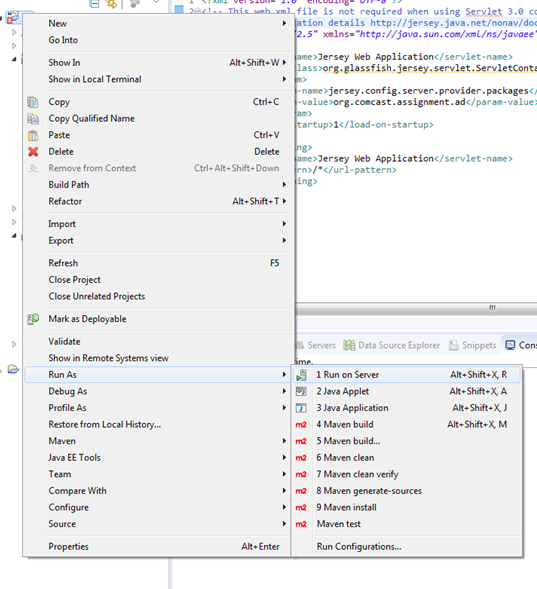


2. The whole project structure is like this:



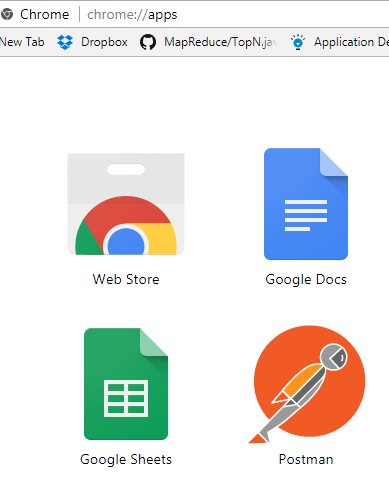
3. Add and configure a tomcat Server

4. Run this project on the tomcat Server.



Test:

1. Add Postman app in chrome browser.



1. Run the postman app.
2. Make function 1, create a new ad, test:

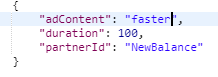
Test 1: regular test

Uri: <http://localhost:8089/ad> Post method

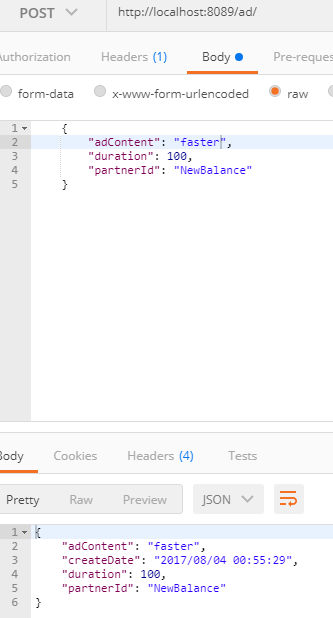
Header:



Body:

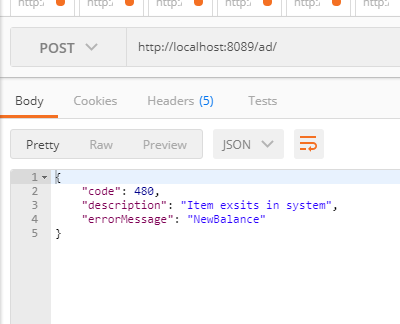


Return :



Test 2: Dulplication test: Do the same request again.

Return: error message: items exists in the system.



1. Make function 2, retrieve an ad, test:

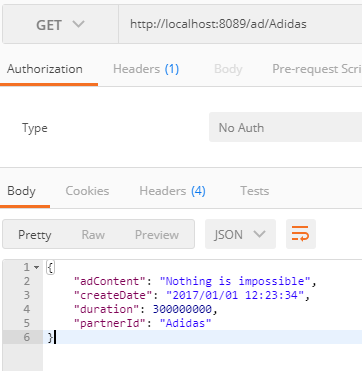
1 Regular test:

Uri : <http://localhost:8089/ad/Adidas> method: get

Adidas is hard code in the system:



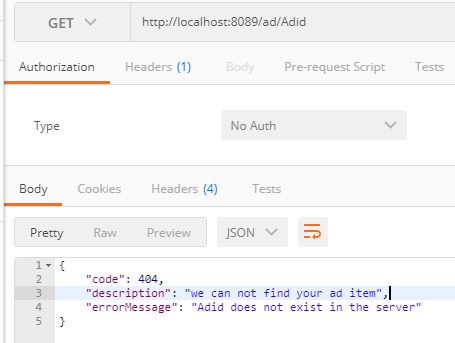
Result :



2 Not found data test:

Uri : http://localhost:8089/ad/Adid method: get

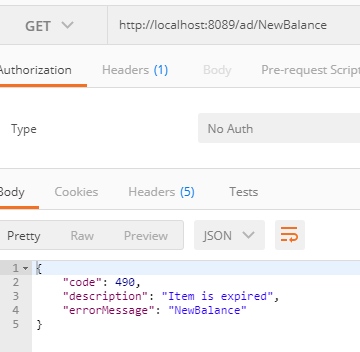
Return : error message: adid does not exist in the server



3 Expired data test:

Uri : <http://localhost:8089/ad/NewBalance> method:get

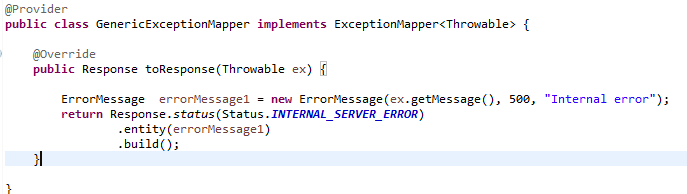
Return error message : item is expired.



Additional questions:

1

In the code, we need hand all the failure requests in the following cods. We could add an globe exception handler, named GenericExceptionMapper.



2

Advantage:

Jersey is an intuitive choice to set up the rest system, since it implement the Jax-rs stander. And it has the good performance in the work. For the light project it is good choice, since it could easily to set up.

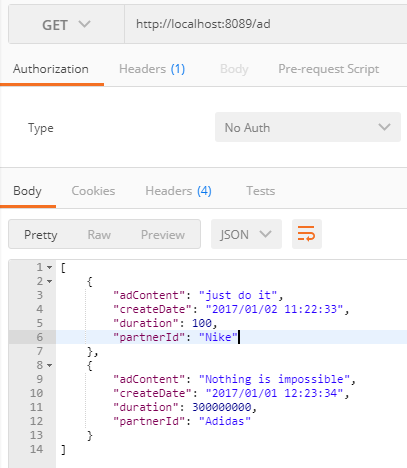
Compare with Spring MVC

Using Spring MVC to implement rest ful is another good choice. Since Spring MVC has its own stander, and it is very integrate with other spring product like spring security. For the a large system I prefer to use the Spring mvc to set the rest ful api. Since Jersey only works on the servlet layer to implement restful and Spring MVC is a completion system.

3.

Uri : [http://localhost:8089/ad/](http://localhost:8089/ad/NewBalance) method:get

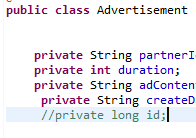
Return :



1. If allow a partner has multiple ad campaigns. We need to change the hashMap key value pair and advertisement class.

Since it will change previous codes a lot, I just describe my codes for this function.

1．add （private long id） to advertisement class.



2． Change the hashMap structure to



3. We generate a unique id for each successful user request.



4. We change the logic of the retrieve and add ad.

For adding: we can not only judge whether the partner exist in the system. We need add an determination of the expiring time. We could do like this: judge the id first, if not find, add directly, if find, add another if clause: If the ad expired, add new, if not expired, reject. Codes maybe like this:

