

Docker Assignment On Microservices in Jersey Web Application for CRUD Operations on MySQL Database In RESTful Manner

Name : Manoj Sadashiv Popalghat

Roll No : 18223

Department : Mathematics

Class : M.Sc. Industrial Mathematics with Computer Applications

Subject : DevOps and Services

GitHub : https://github.com/manojpopalghat/crudapi_docker.git

How Implements microservice for CRUD Operations on Mysql Database ?

For implement microservice , I used Jersey Maven web application for CRUD API's in RESTful manner. Web App contains 'UserResource', 'UserRepository', 'User'(POJO) classes which deals with CRUD Operations and MySQL Database, which supports support PUT, POST, READ, DELETE Operations.

In Database there is 'userDB' database having 'users' table fulfills given requirements of User Schema.

User Object

```
{
Id : Integer
Name : String
Age : Integer
Department : String
Subjects : String
}
```

Web App files Explanations :

all classes in com.manoj.crud package

'User' class :

It is simple POJO having all Above User properties and methods

'UserResource' class:

this class contains all CRUD API's, which supports support PUT, POST, READ, DELETE Operations.

'UserRepository' class :

this class deals with database (DAO).

Database Explainatons:

in DB I created new User 'Admin' having 'userDB' database contains 'users' table
see description of userDB.users

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(100)	YES		NULL	
age	int(11)	YES		NULL	
department	varchar(50)	YES		NULL	
subject	varchar(200)	YES		NULL	

API Testing :

index file(not parsed in Postman)

<http://localhost:8080/CRUD/>

This shows simply table of Users in Database

Microservice in Jersey RESTful Web Application

ID	Name	Age	Department	Subjects
111	manoj	24	Mathematics	devOps
112	harshal	22	Maths	ML,Data-Science
113	ankita	23	Maths	ML
114	Lalit	23	Maths	Data_Science,ML
115	aniket	23	Computer Science	IoT
201	anil koyal	23	Statistics	python,Data-Science,Regression-Analysis,Data-Mining,Multi Variate-Analysis

To Check CRUD Operations in RESTful manner use following links(Supports in POSTMAN)
GET Method for all users:<http://localhost:8080/CRUD/webapi/users>
GET Method for particular user with id=101:<http://localhost:8080/CRUD/webapi/users/read/101>
POST Method for a user:<http://localhost:8080/CRUD/webapi/users/create>
PUT Method for a user:<http://localhost:8080/CRUD/webapi/users/update/>
DELETE Method for a user with id=101:<http://localhost:8080/CRUD/webapi/users/delete/101>

1)READ API:

To test the read api, open POSTMAN and enter the following as the request URL.

<http://localhost:8080/CRUD/webapi/users>

Select Method 'GET', and enter above request to read all users
this will return Array of JSON Objects .

http://localhost:8080/

GET http://localhost:8080/CRUD/webapi/users/

Headers (2)

Key	Value	Description
Content-Type	application/json	
Accept	application/json	

Body

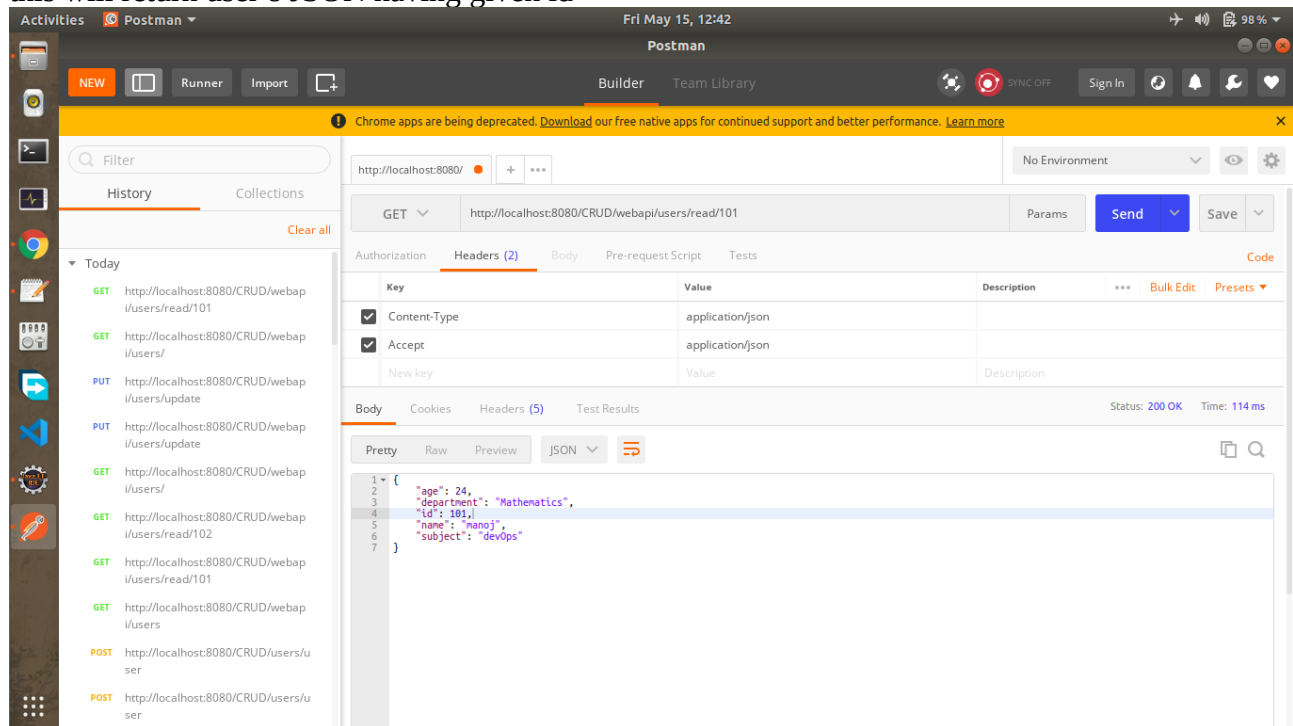
JSON

```
1 {
2   {
3     "age": 24,
4     "department": "Mathematics",
5     "id": 101,
6     "name": "manoj",
7     "subject": "devOps"
8   },
9   {
10    "age": 22,
11    "department": "Maths",
12    "id": 102,
13    "name": "harshal",
14    "subject": "ML,Data-Science"
15  },
16  {
17    "age": 23,
18    "department": "Maths",
19    "id": 103,
20    "name": "ankita",
21    "subject": "ML"
22  }
23 }
```

To read user with some id enter below request

<http://localhost:8080/CRUD/webapi/users/read/101>

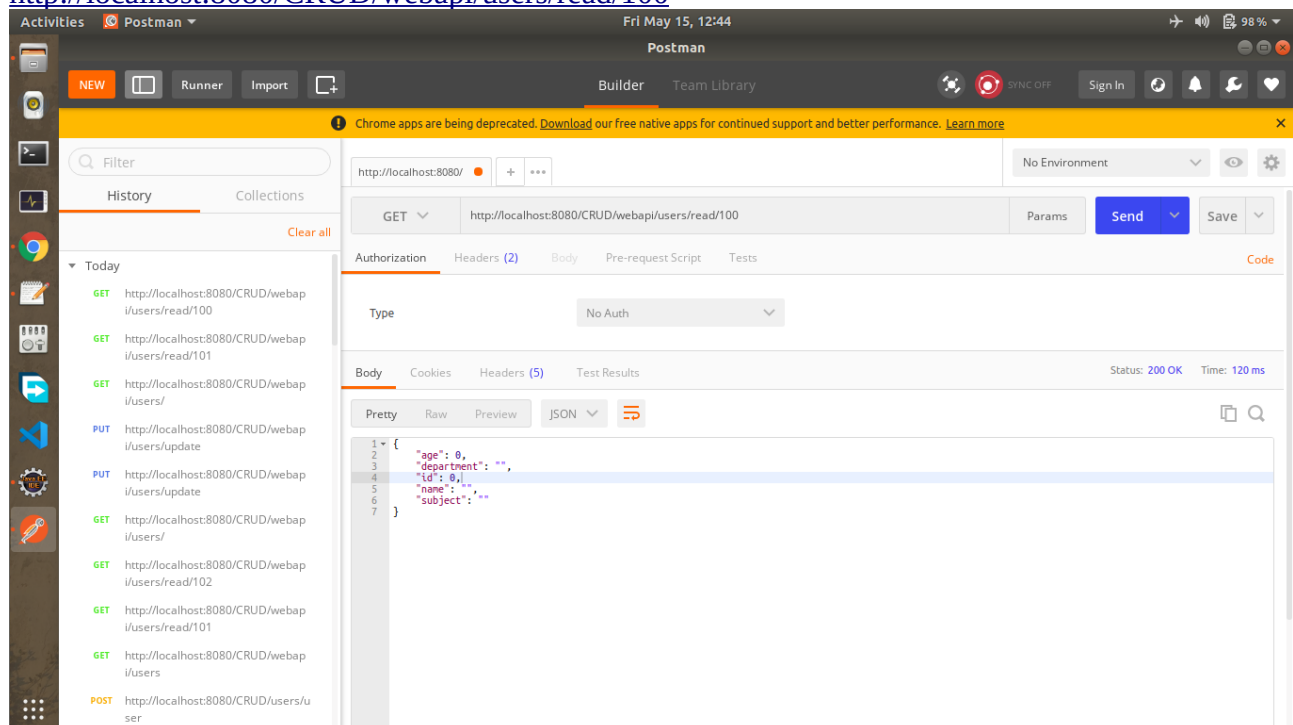
this will return user's JSON having given id



if you entered wrong id then this will return an empty JSON Values

e.g. id 100 is not present

<http://localhost:8080/CRUD/webapi/users/read/100>



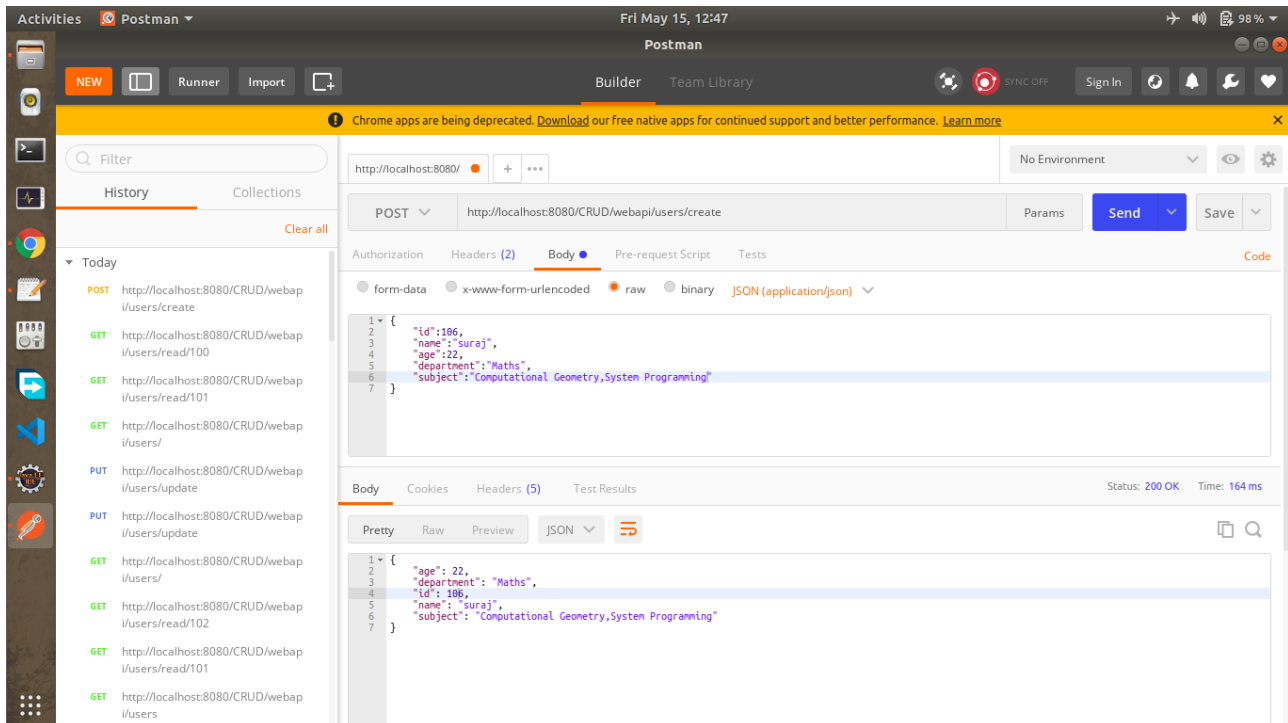
2)CREATE API:

To test the create api, open POSTMAN and enter the following as the request URL.

<http://localhost:8080/CRUD/webapi/users/create>

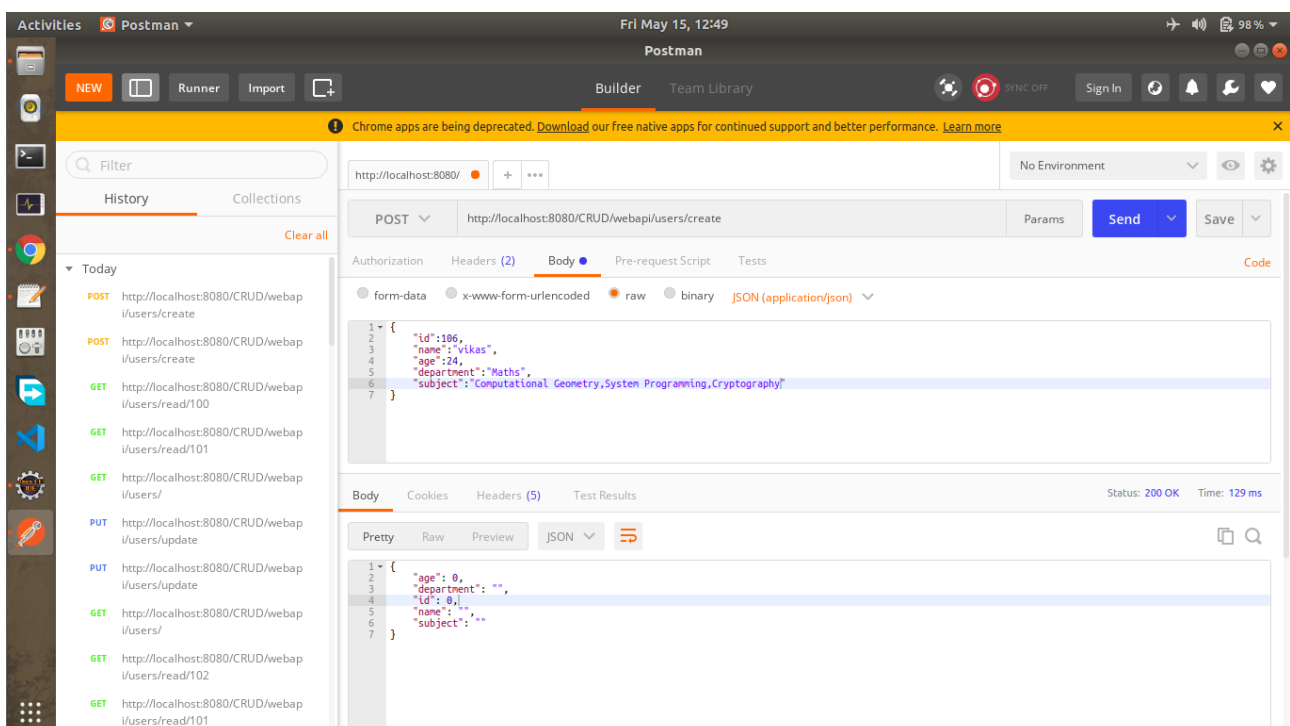
Select Method 'POST' and enter raw JSON data and it will returns entered JSON

e.g.



and if the user is already present with given id then this will not stored data and returns empty JSON Values

e.g.



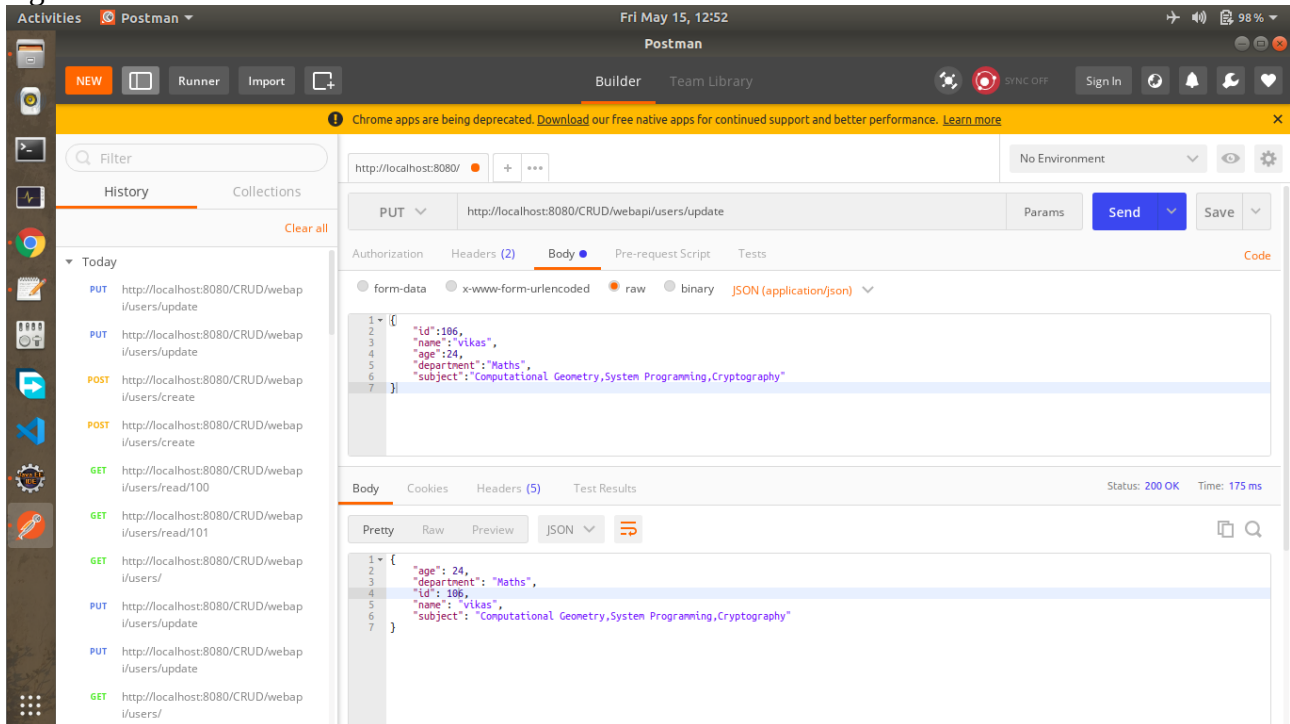
3)UPDATE API:

To test the update api, open POSTMAN and enter the following as the request URL.

<http://localhost:8080/CRUD/webapi/users/update>

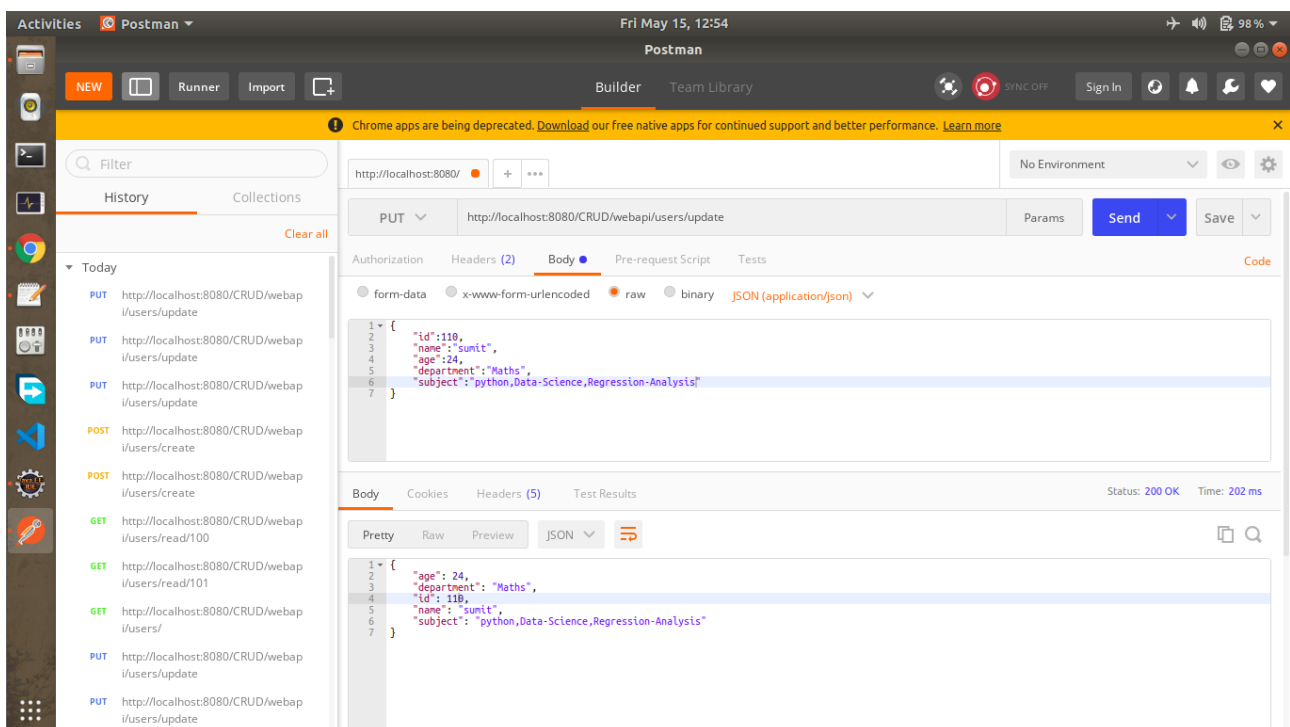
Select Method 'PUT' and enter raw JSON data to update data (data updated with respect to id) and return JSON Object of updated data

e.g.



if user is not present then it will create a new user

e.g.



4)DELETE API:

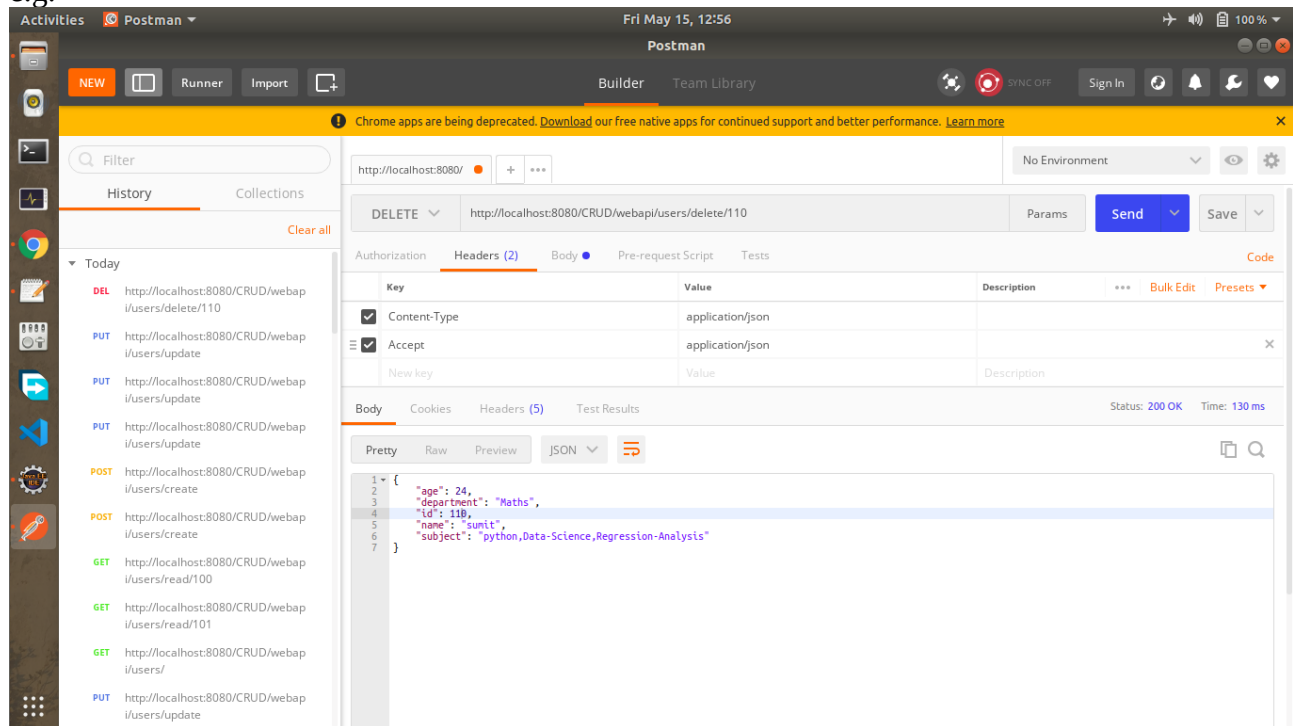
To test the delete api, open POSTMAN and enter the following as the request URL.

<http://localhost:8080/CRUD/webapi/users/delete/110>

Select Method 'DELETE'

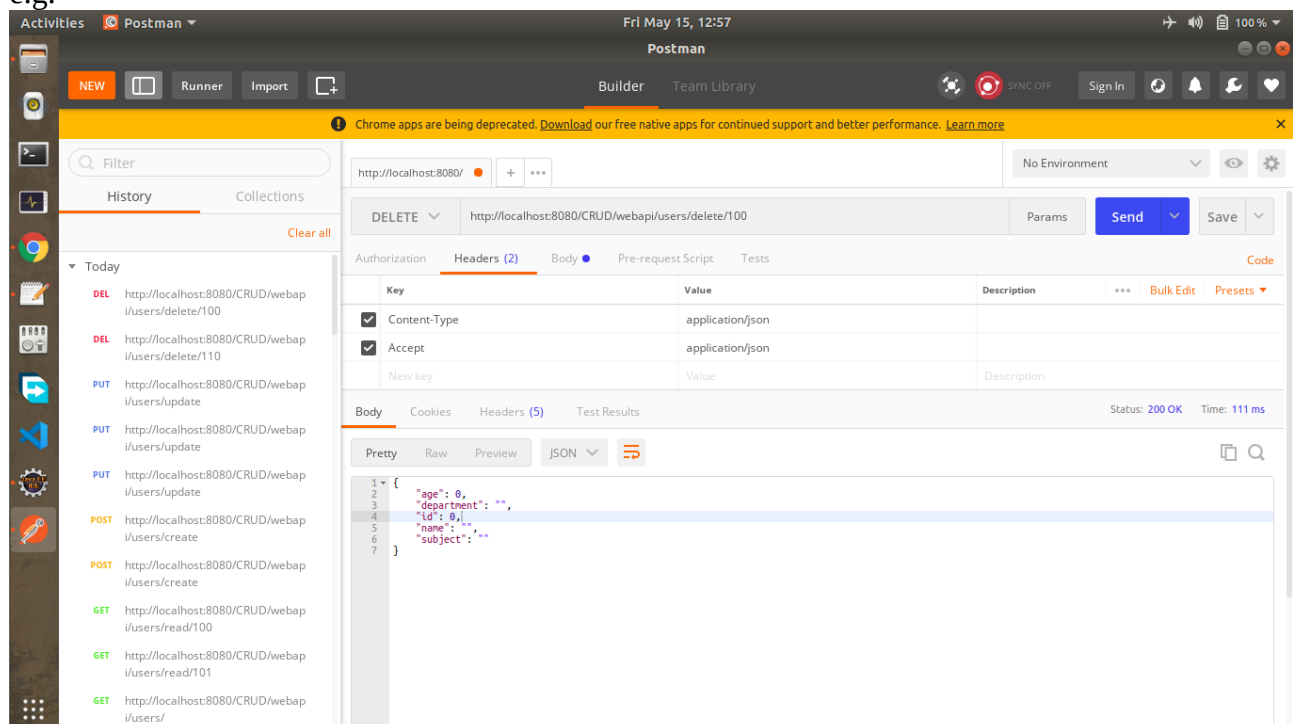
i.e. this method deletes user having id 110 and returns user's JSON having id=110

e.g.



And if user is not present with corresponding id then it return empty JSON Values

e.g.



How Containerised This App ?

For this Project We need :

- jdk 8 Runtime Environment
- Apache Tomcat 8.5 Server
- MySQL-Server

I created one container having these services and deployed this 'CRUD' Web-app in apache-tomcat-8.5.54/webapps and for usage from host machine, port-mapping is 8080:8080 and 3306:3306

How to Build Docker Image ?

Go to https://github.com/manojpopalghat/crudapi_docker.git clone or download zip and unzip it , go to this directory and run following commands

To Build image of Dockerfile

```
docker build -t manojpopalghat/crudapi_docker .
```

To Run Docker image

```
docker run -d -p 8080:8080 <image-id>  
docker run -d -p 8080:8080 --name=crudapi <container-id>
```

To Explore container

```
docker exec -it <container-id> bash
```

Postman Collection:

https://github.com/manojpopalghat/crudapi_docker/tree/master/Postman%20Collection