**How to run only the backend assignment**

**Technologies used-** Java, Spring Boot, H2 Database, Maven, Angular JS

a) Clone the project from the git repository

b) Make sure that you have maven and Java installed in your local machine.

c) Import maven project from DemoBE folder

d) Run "mvn clean install" command inside the DemoBE directory to download maven dependencies required by spring.

e) Then go to file "DemoApplication.java" and run it as SpringBoot application.

f) The application will be up at port 1122

g) To connect to the database, go to url "http://localhost:1122/h2-console/", after the application is up.

h) Other configurable properties related to database, port, logging are mentioned in "application.properties" file

------------------------------------------------------------------------------------------------------------------------------------------

Things covered as part of the assignment

1) Added CRUD operation for Student class object.

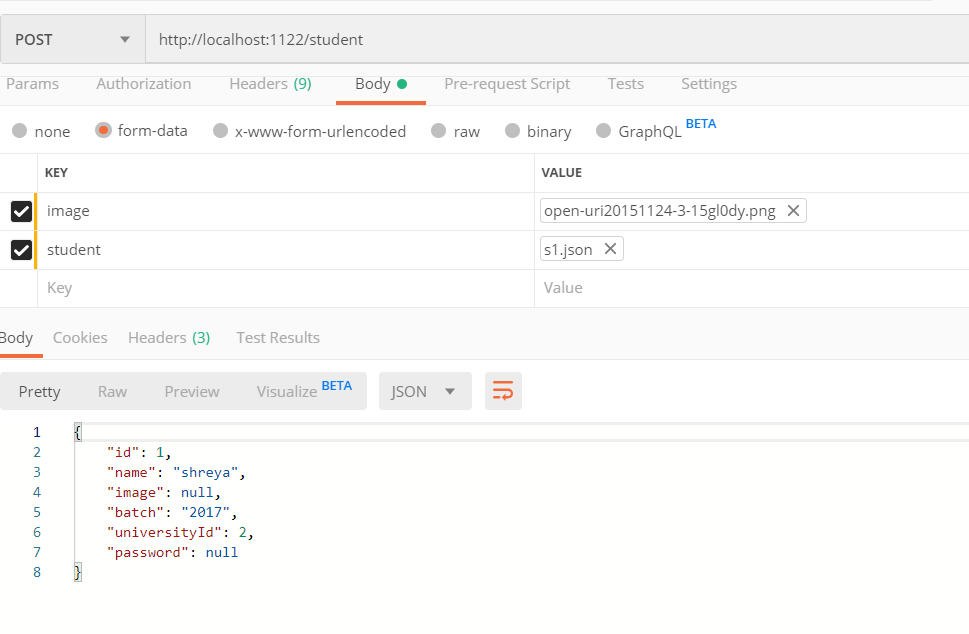
2) Student image is also stored as a multipart file.

Usecases

--------------------------------------------------------------------------------------------------------------------

POST - http://localhost:1122/student/

We have to pass the request as form-data in body where we are passing key value pair of image and student as files



.png contains any image

s1.json contains below json object

{

"id":"1",

"batch":"2017",

"name":"shreya",

"universityId":"2",

"password":"cccccc"

}

Response-

{

"id": 1,

"name": "shreya",

"image": null,

"batch": "2013",

"universityId": 2,

"password": null

}

We are making passowrd as null in response body.

Response is stored in H2 database in STUDENT\_TABLE as below-

ID BATCH IMAGE NAME PASSWORD UNIVERSITY

2 2017 89504e... (667116 bytes) shreya ccccc 2

Image is stored in bytes.

-------------------------------------------------------------------------------------------------------

GET BY ID- id=1

GET - http://localhost:1122/student/1

Response-

{

"id": 1,

"name": "shreya",

"image": null,

"batch": "2017",

"universityId": 2,

"password": null

}

------------------------------------------------------------------------------------------------------------

GET all students

GET - http://localhost:1122/student/

Response- List of all students stored in database

------------------------------------------------------------------------------------------------------------

DELETE BY ID-

DELETE - http://localhost:1122/student/2

Response- FORBIDDEN

Student with id will be deleted from H2 database.

-----------------------------------------------------------------------------------------------------------------

UPDATE-

POST - http://localhost:1122/student/

Request -

student.json

{

"id":"1",

"batch":"2017",

"name":"polll",

"universityId":"2",

"password":"sss"

}

Same as POST request with some field change, it will update the existing student. In above example, name and password is changed.

In H2 database stored as below-

ID BATCH IMAGE NAME PASSWORD UNIVERSITY

1 2017 89504e... (667116 bytes)polll sss 2

--------------------------------------------------------------------------------------------------------------------------------------

2) Validation in student request- Added @NotNull and @NotEmpty in Student object, which will automatically not allow to persist in database, if mandatory fields are missing.

POST - http://localhost:1122/student/

Request with some fields(universityId) missing as below-

{

"id":"1",

"batch":"2017",

"name":"polll",

"universityId":"",

"password":"sss"

}

Response-

{

"timestamp": "2019-11-11T18:48:49.793+0000",

"status": 500,

"error": "Internal Server Error",

"message": "Could not commit JPA transaction; nested exception is javax.persistence.RollbackException: Error while committing the transaction",

"path": "/student/"

}

----------------------------------------------------------------------------------------------------------------

3) Error handling if in GET request , student does not exist

REQUEST -

GET POST - http://localhost:1122/student/200

RESPONSE- NoStudentException custom exception is thrown and below response in postman

{

"timestamp": "2019-11-11T18:49:50.496+0000",

"status": 500,

"error": "Internal Server Error",

"message": "No such student exist",

"path": "/student/200"

}

----------------------------------------------------------------------------------------------------------

4)

a) Added CRUD operation for University class object.

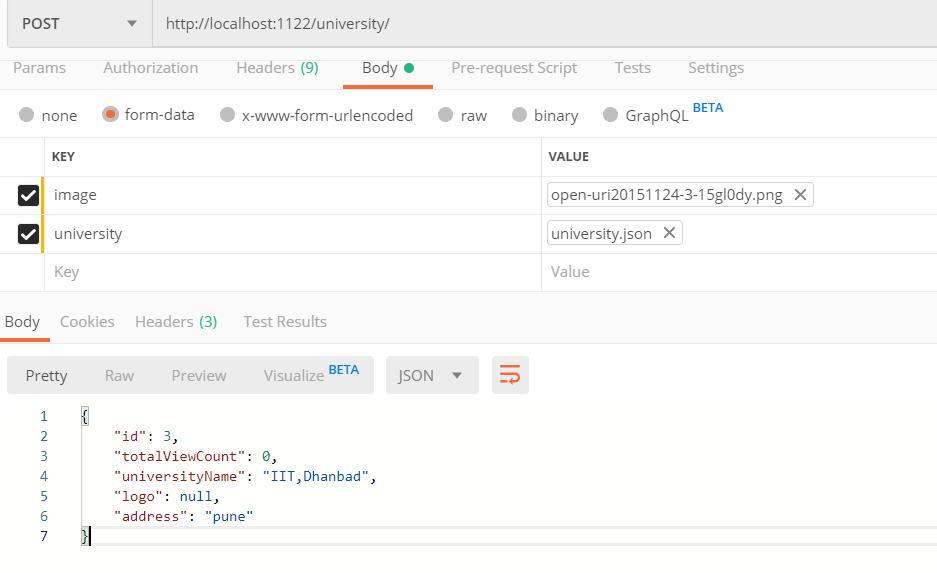
b) University logo is uploaded as multipart image file

Use cases

---------------------------------------------------------------------------

Same as above

POST http://localhost:1122/university/



university.json

{

"id":"1",

"totalViewCount":"0",

"universityName": "IIT,Dhanbad",

"address":"Pune"

}

SELECT \* FROM UNIVERSITY\_TABLE ;

ID ADDRESS LOGO TOTAL\_VIEW\_COUNT UNIVERSITY\_NAME

3 pune 89504e... (667116 bytes) 0 IIT,Dhanbad

University data is stored in DB

Remaining get, update and delete operations on university object is same as student object.

---------------------------------------------------------------------------------------------------------

5) a) Helper class Credential which contains id and password is used to perform the login operation

b) Login Service is used for authentication whether correct username and password is used.

That's all I could cover in two days.