**Displaying User Feedback**

**Developer Details:**

N Prasad

BTECH (Department of Computer science and Engineering)

BS Abdur Rahman Crescent Institute of science and technology.

**Contact**: [prasadnithi15@gmail.com](mailto:prasadnithi15@gmail.com)

**Source code:**

Git repository link:🡪 <https://github.com/prasad-ops/phase-2-project>

**PROBLEM STATEMENT:**

Create a Spring Boot project that will capture user feedback using a REST endpoint. The REST resource will take in parameters using HTTP POST. The feedback data will be then added to a database table.

**REQUIREMENTS:**

* Eclipse/IntelliJ: An IDE to code for the application
* Java 8: A programming language to develop the prototype
* Git: To connect and push files from the local system to GitHub
* GitHub: To store the application code and track its versions
* Scrum: An efficient agile framework to deliver the product incrementally
* Specification document: Any open-source document or Google Docs
* Apache Tomcat web server.
* Spring boot with hibernate.

**Git Repository links:**

Git repository link:🡪 <https://github.com/prasad-ops/phase-2-project>

**ALGORITHM:**

* Start the program.
* Create a MySQL table named feedback for storing feedback data
* An entity class Feedback should be made with annotations to link it with the feedback table
* A repository class should then map the entity class to the CrudRepository interface
* Create a REST controller class to create the REST endpoint. It should take in parameters using the POST protocol
* Data received in the REST controller will be then saved into the database
* Create a springfox swagger-ui for performing crud operation.
* The step-by-step process involved in completing this task should be documented
* Stop the program

**HOW TO EXECUTE:**

STEP 1:

Download the code from the git repository and run the code as **Spring application** default port used is 9200. For changing the port

Server.port=\*\*\*\*

This code is written in application.properties

After executing the code the table is created in h2 database.

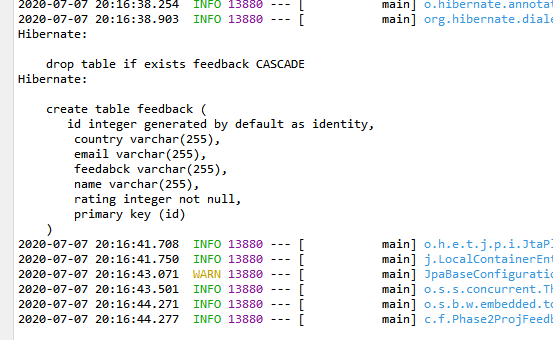


Fig 1. Table creation query

STEP 2:

In-order to view the fields in the database.

Open this URL in the browser - l<http://localhost:9200/h2>

Set JDBC URL as - jdbc:h2:mem:testdb

Then click connect.

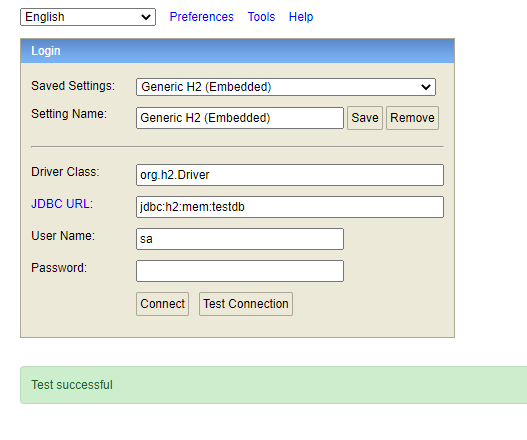


Fig 2. H2 database connection.

STEP 3:

The crud operation can be performed using postman tool or by using swagger-ui.

POST operation – it will create the fields in the form of Jason data and store it into the h2 database.

PUT operation – it will update the data based on primary key (id),

GET operation – it will fetch the data from the database.

DELETE operation – it will delete the data based on primary key (id).

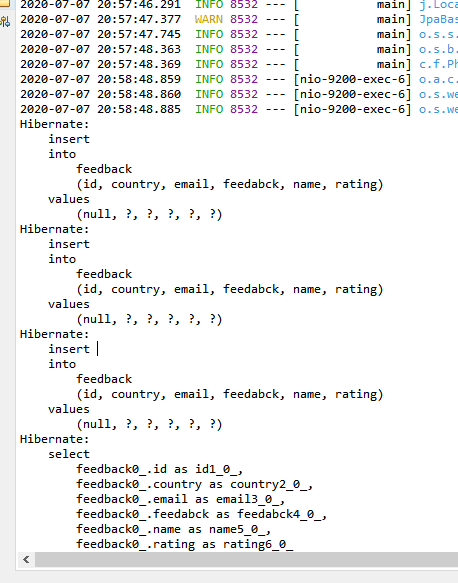


Fig 3. Log for each operations.

STEP 4:

For capturing user feedback using a REST endpoint

Open browser and open this link - <http://localhost:9200/swagger-ui.html>

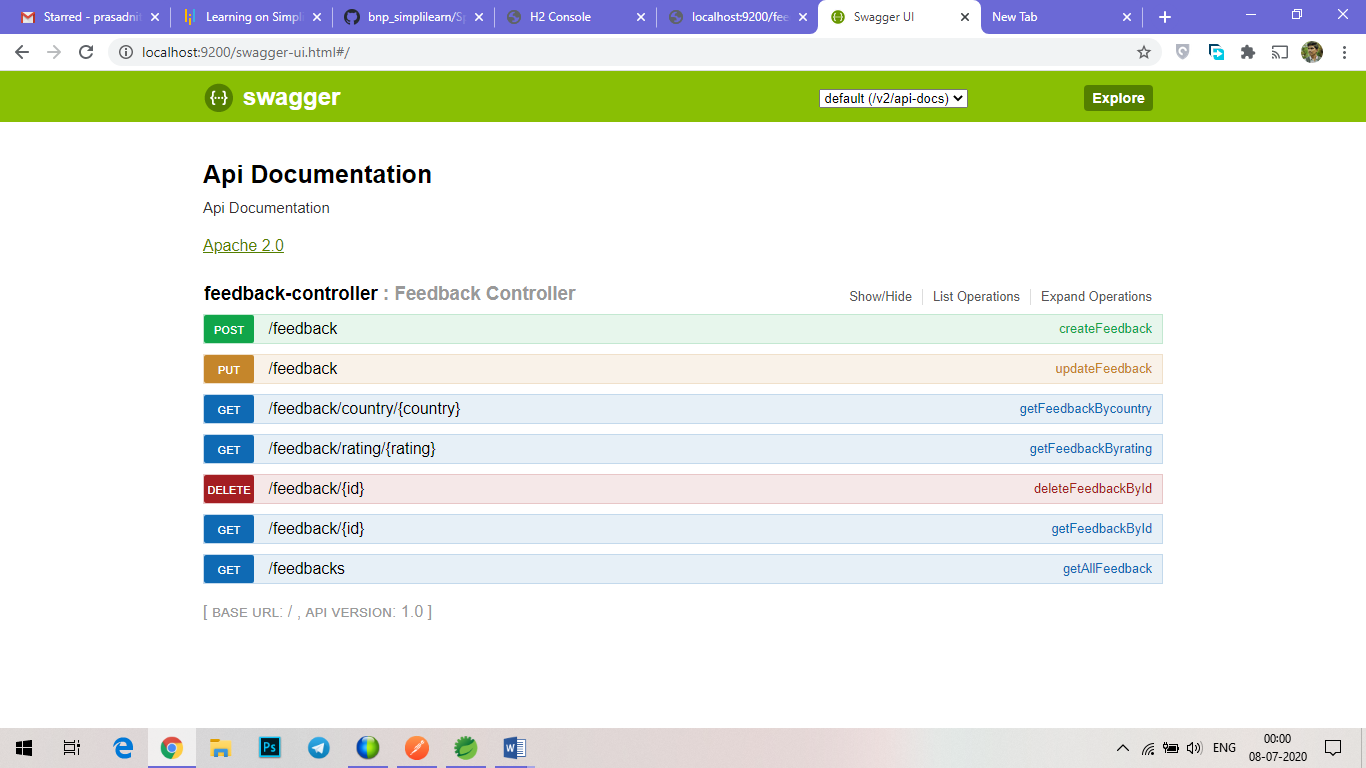


Fig 4. Swagger-ui.

STEP 5:

For posting the feedback, click POST

{

    "name": "suse",

    "country": "brazil",

    "email": "suseabi@gmail.com",

    "rating": 7,

    "feedback": "ok"

}

This statement will add the feedback into database.

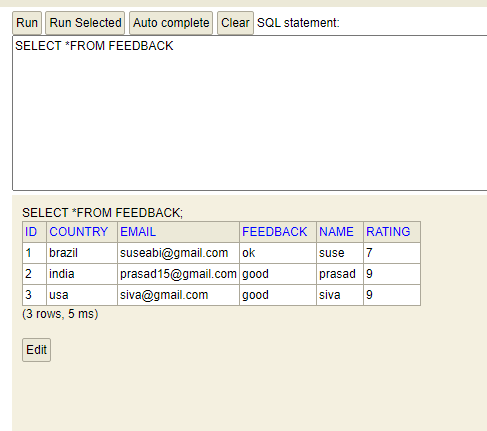


Fig 5. Creating feedback and storing it in database.

STEP 6:

For updating, click PUT.

{

    "id": 1,

    "name": "suse",

    "country": "brazil",

    "email": "suseabi@gmail.com",

    "rating": 7,

    "feedback": "ok"

}

This statement will update the database based on primary key (id).

STEP 7:

For deletion, click DELETE.

It will delete the field based on primary key

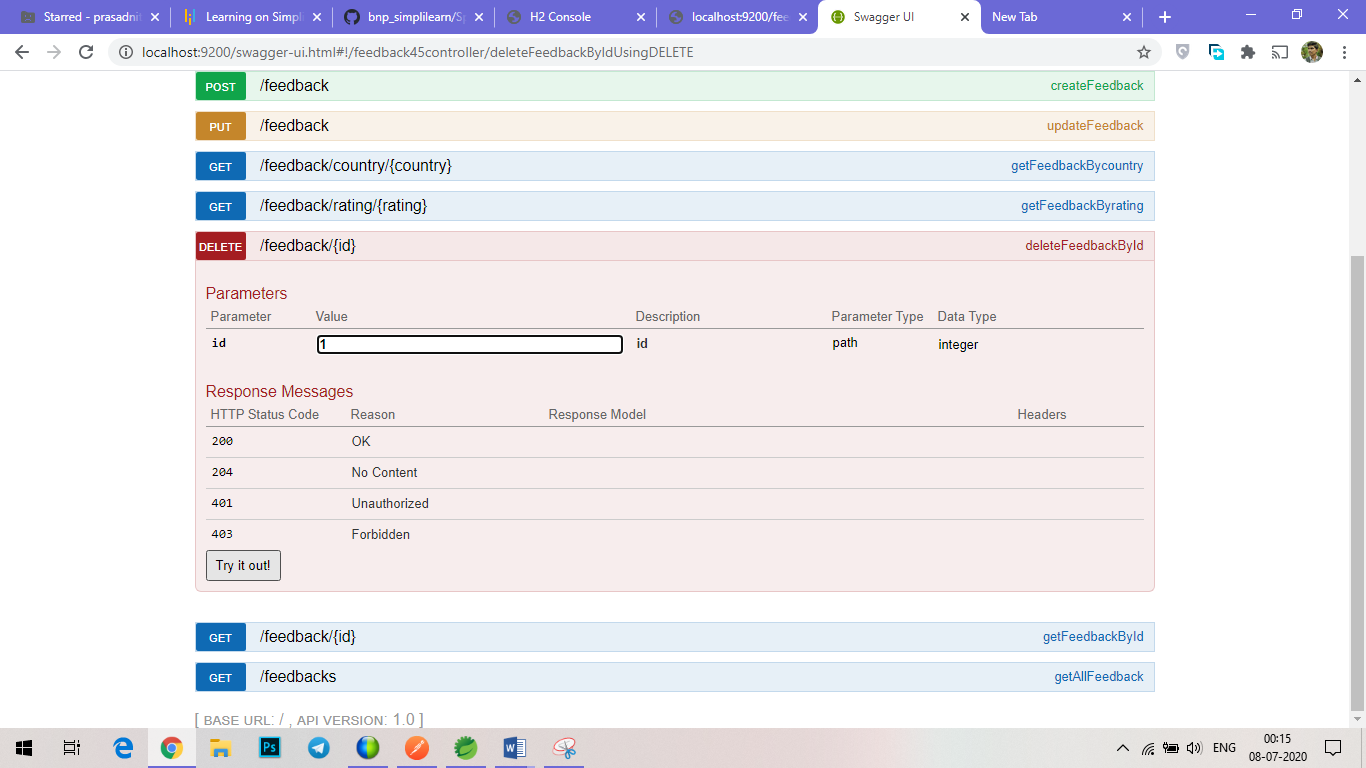


Fig 6. Delete by id.

STEP 8:

GET operation is done by three method ( getByID, getByCountry, getByRating).

Enter the valid request in the values for fetching the feedback

URL for GET operation

getByID - [http://localhost:9200/feedback/{id}](http://localhost:9200/feedback/%7bid%7d)

getByCountry - [http://localhost:9200/feedback/country/{country}](http://localhost:9200/feedback/country/%7bcountry%7d)

getByRating - [http://localhost:9200/feedback/rating/{rating}](http://localhost:9200/feedback/rating/%7brating%7d)

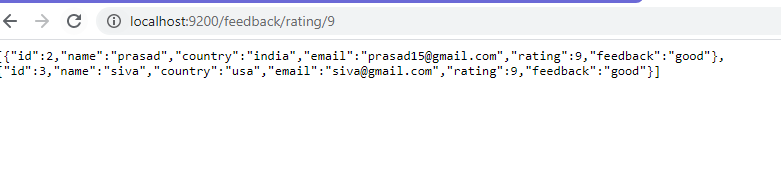


Fig 7. Get by rating {9}

Since h2 database is an embedded database. It will drop the table once the connection is closed. For permanent storage of data, use MSQL database. The same program with change in configuration is required in application.properties.

**CONCLUSION:**

Thus, the mentioned task in the problem statement are executed. The source code is pushed into the git repository. The git link for downloading the source code is given in document. An entity class Feedback is made with annotations to link it with the feedback table. A repository class is map the entity class to the Crud Repository interface. REST controller class is created to create the REST endpoint. It should take in parameters using the POST protocol. Data received in the REST controller was saved into the database.