**Sample Spring Boot project API document**

# **Introduction**

This is a sample spring boot project those who are seeking to learn about how to create a simple spring boot crud application. This project has implemented all the crud operations in a simple way. You can learn how to do crud operations in spring boot.

# **Technologies Used**

Spring Boot

MySQL

Hibernate

# **IDE used**

IntelliJ IDEA

# **API Signatures**

Here I have run this application in localhost and 8883 port. You can change the port as you wish.

|  |  |  |
| --- | --- | --- |
| API signature | Method | Description |
| localhost:8883/employee/save | POST | This API will save an employee in employee database. (a record will insert to employee table). |
| localhost:8883/employee/getall | GET | This API will give details of all the employees as a List. |
| localhost:8883/employee/getone/{employeeId} | GET | This API will give details of the employee, that we send the employee id in the url. |
| localhost:8883/employee/delete/{employeeId} | DELETE | This API will give delete the employee, that we send the employee id in the url.(delete the record of that employee from employee table) |
| localhost:8883/employee/update | PUT | This API will update the details of an employee. (update the employee table) |

# **Let’s test the API**

## **Import project to IDE**

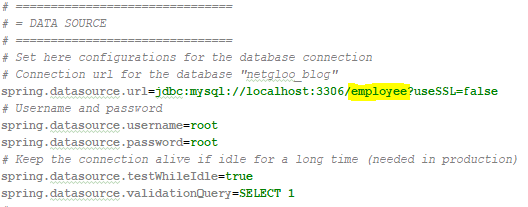
First you must clone the project from the GitHub repository and then open the project using IntelliJ as below.

File -> Open -> *select your project*

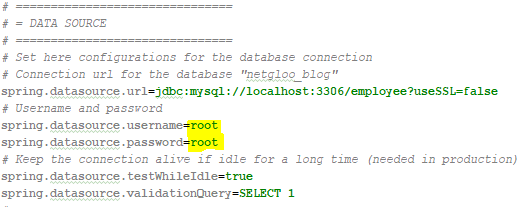
## **Create the Database**

You must create your database.

Create a database as ***employee*** in your MySQL workbench. You can create your own database. it doesn’t have to be the name as ***employee*** in my case I used the database as ***employee****.* You can create your oen database. If you are creating your own database, then you must change the database name in property file to the name of your changed database. You can find it in below image, the database name is highlighted.



And, database ***username*** and ***password*** must change according to your MySQL configuration.



**NOTE: -**

You only need to create the database , you must not create the tables. when you run the application Hibernate will create the tables in the ***employee*** database for you.

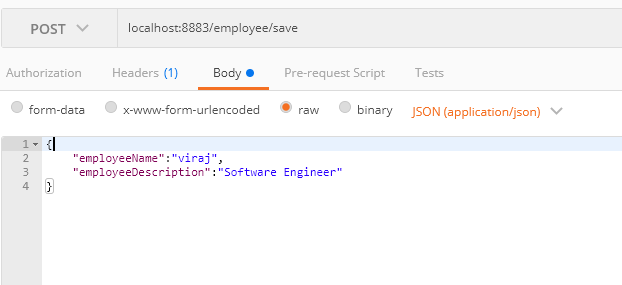
Then Build and Run your project.

You can test all the routes of the API using postman. Here I have include all the screen shots of API calls using postman.

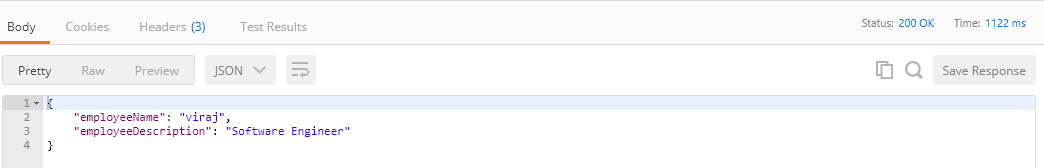
## **POST**

## **localhost:8883/employee/save**

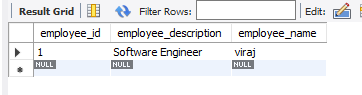
**Request**



**Response**



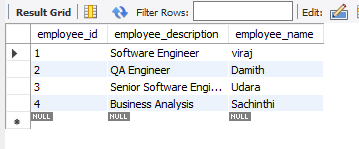
**Database**



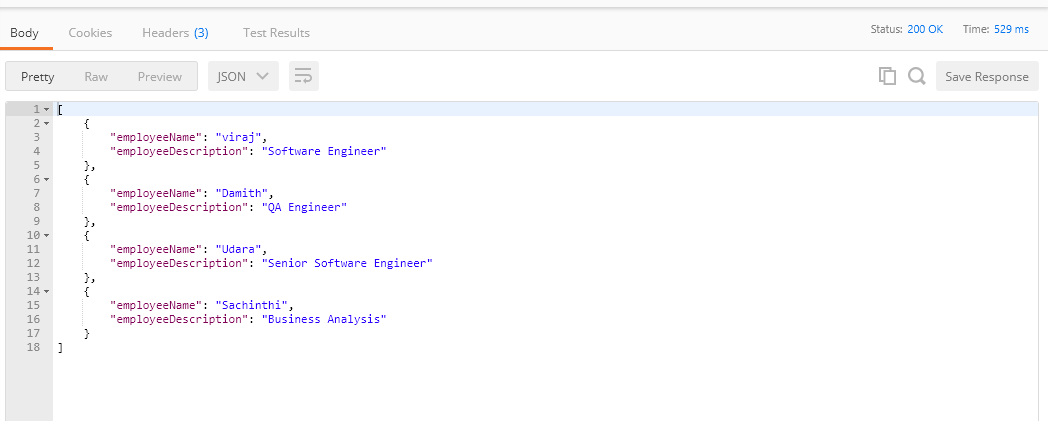
## **GET**

## **localhost:8883/employee/getall**

**Database**



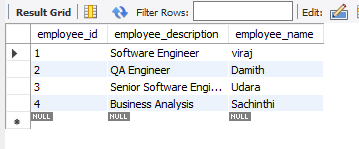
**Response**



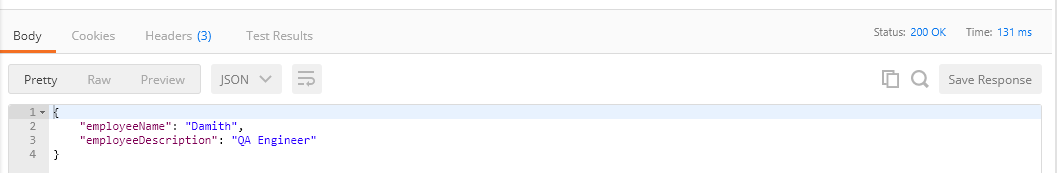
## **GET**

## **localhost:8883/employee/getone/2**

**Database**



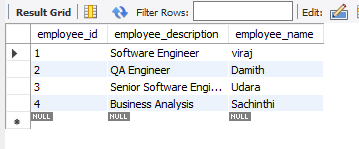
**Response**



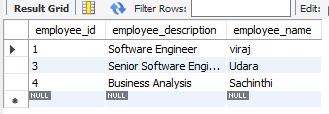
## **DELETE**

## **localhost:8883/employee/delete/2**

**Database** (Before delete the record which has the employee Id is 2)



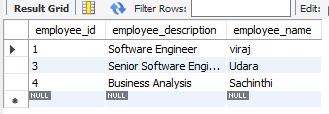
**Database** (After delete the record which has the employee Id as 2)



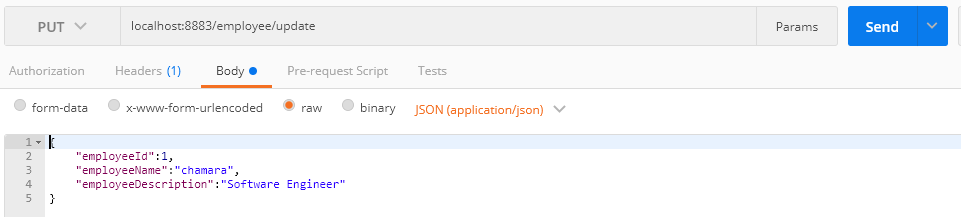
## **PUT**

## **localhost:8883/employee/update**

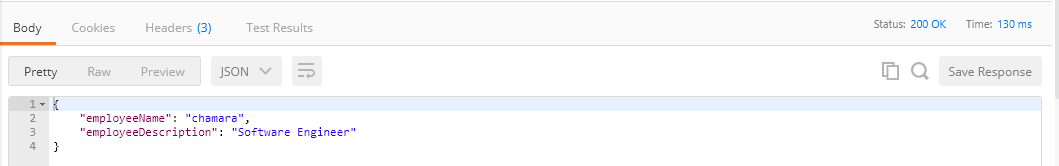
**Database** (Before update the record which has the id as 1 )



**Request**



**Response**



**Database** (After update the record which has the id as 1)

