**Building Rest API for Spring CT technical test:**

Functionalities and their respective request URL mapping:

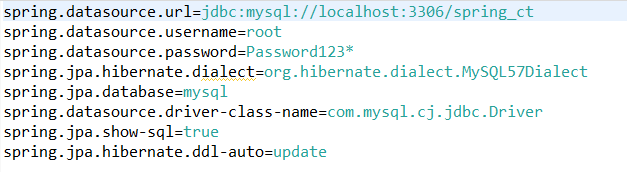
* Add a Course [Name, Professor Name, Description]
  + -> <http://localhost:8080/addCourse> [POST]
* Add a Student [Name, Email, Phone]
  + -> [http://localhost:8080/addStudent](http://localhost:8080/addCourse) [POST]
* Allocate a Student to one or more Course
  + -> [http://localhost:8080/addCoursesToStudent](http://localhost:8080/addCourse) [PUT]
* List Students [Name, Email, Phone, Course Enrolled (comma separated string)]
  + -> [http://localhost:8080/students](http://localhost:8080/addCourse) [GET]
* Delete a given Course
  + -> [http://localhost:8080/deletCourse/{id}](http://localhost:8080/addCourse) [DELETE]
* Delete a given Student
  + -> [http://localhost:8080/deletStudent/{id}](http://localhost:8080/deletStudent/%7bid%7d) [DELETE]
* Should be Get a Students data for a given Course
  + -> [http://localhost:8080/getStudentsByCourse/{id}](http://localhost:8080/getStudentsByCourse/%7bid%7d) [GET]

**SETUP and USAGE INSTRUCTIONS:**

**Step 1:**

Setting Up the Database: run the schema queries give with this document to create a schema and its corresponding tables.

Enter your datasource (Use MySql database, as driver used is for MySql) details in Application.properties file.



**Database Structure and Data:**



**Step 2:**

Method 1: Create a War file of the Application and deploy it in your Servlet Container and run the application. These APIs can be consumed by above mentioned URLs.

Method 2: Import the application in your IDE and run the application. It contains Integrated Tomcat Web server, hence it will get deployed immediately and ready to use at localhost:8080.

**Server used:** Tomcat Integrated Server

**POM.xml:**

**Parent**

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.4.5</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

**Other major dependencies**

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-rest</artifactId>

</dependency>

<dependency>

<groupId>mysql</groupId>

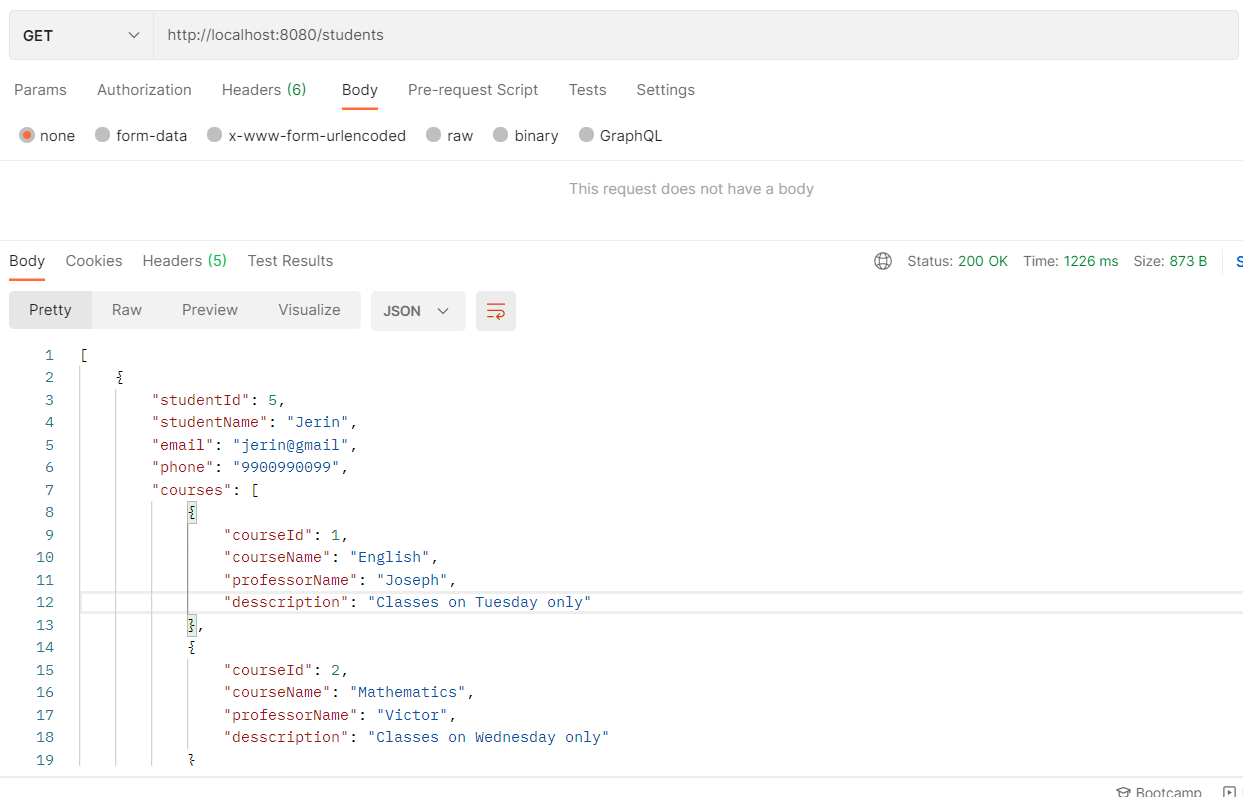
<artifactId>mysql-connector-java</artifactId>

<scope>runtime</scope>

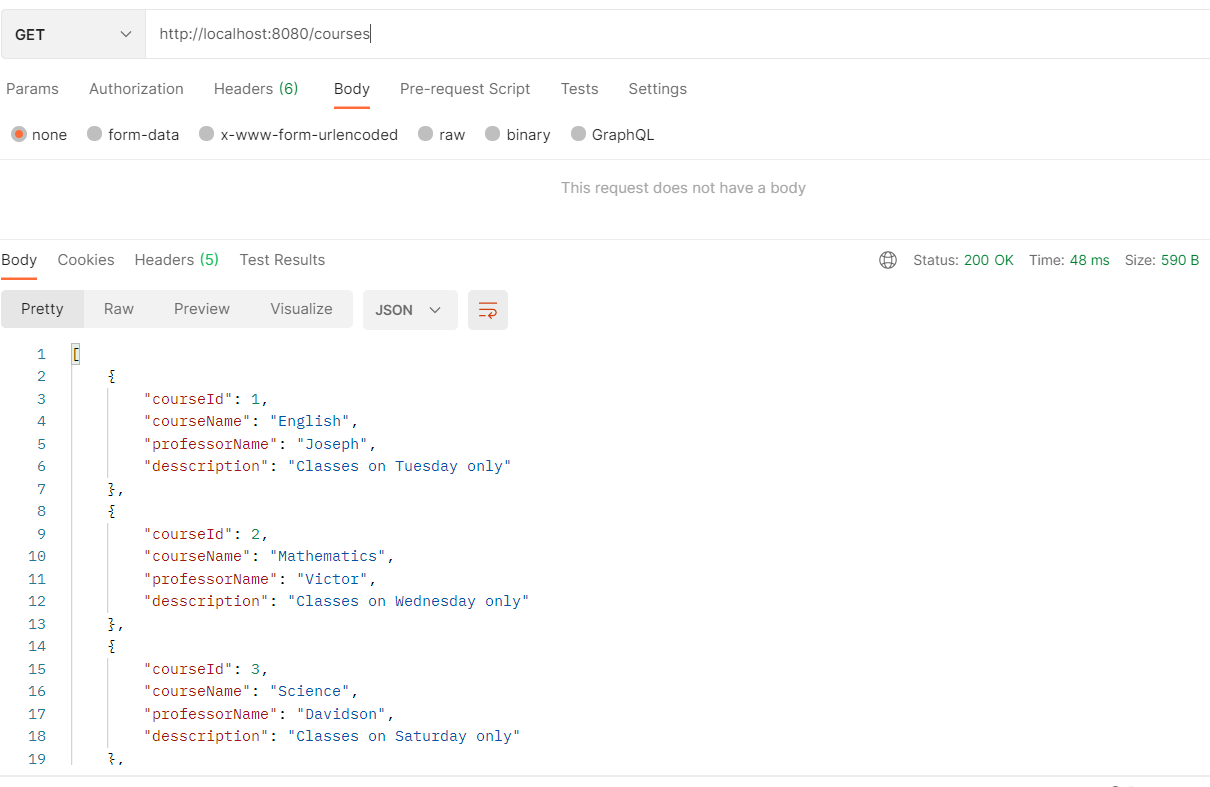
</dependency>

**Sample Outputs:**

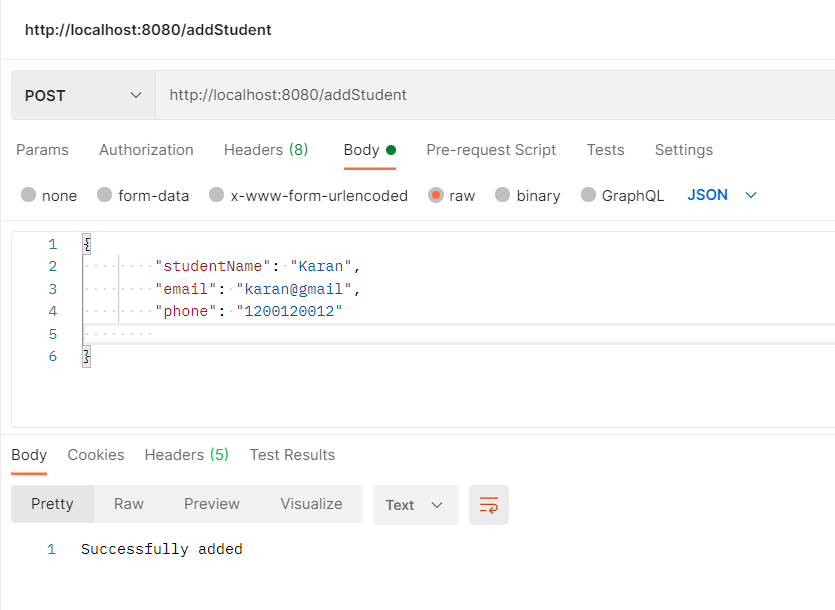
1. <http://localhost:8080/students> [Fetch all the Students]



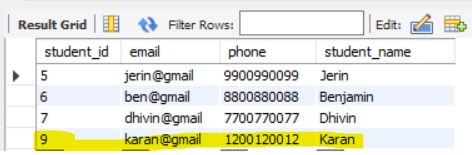
1. <http://localhost:8080/courses> [Fetch all Courses]

****

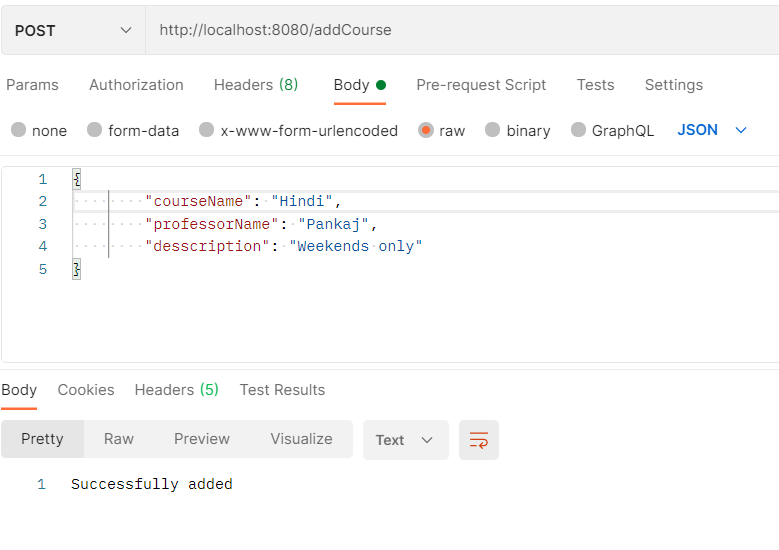
1. <http://localhost:8080/addStudent> [Adding a Student]

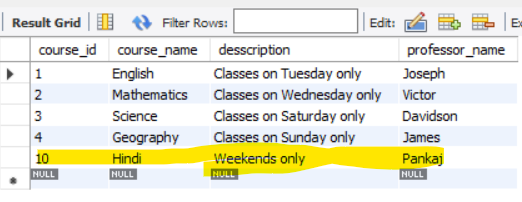
****

**Data persisted:**

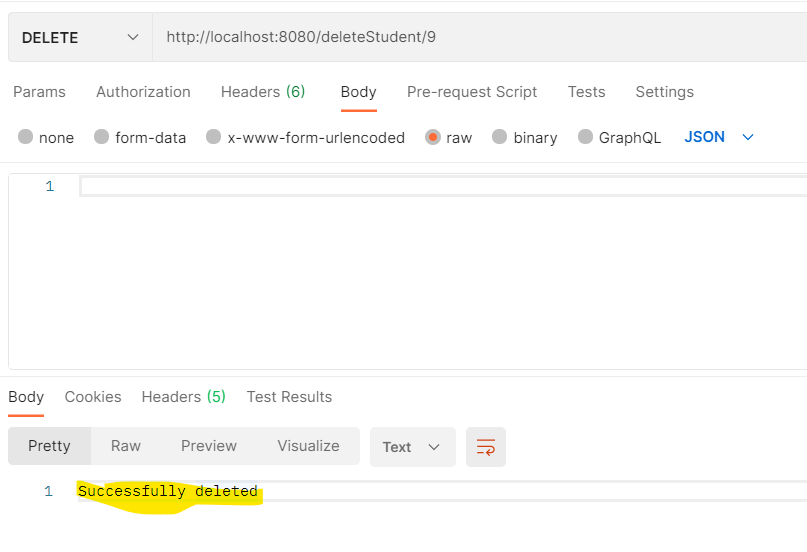


1. <http://localhost:8080/addCourse> [Adding a Course]

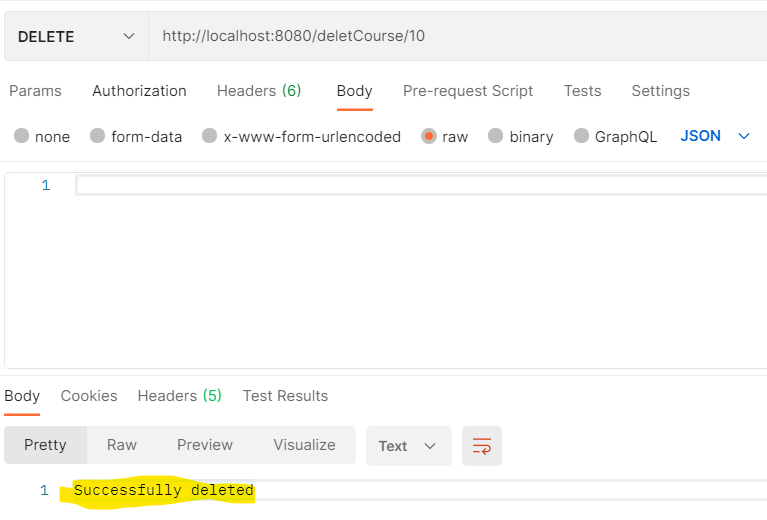
**Data persisted:**



1. <http://localhost:8080/deleteStudent/9> [Deleting a Student]



1. <http://localhost:8080/deletCourse/10> [Deleting a particular Course]



1. <http://localhost:8080/getStudentsByCourse/3> [Retrieving List of Students for one particular course]

