# **Spring boot**

**Step 1: Project Setup**

1. Open your preferred web browser and navigate to Spring Initializr. [ <https://start.spring.io/> ]
2. Set the following project details:

* Project: Select "Grade-Groovy."
* Language: Choose "Java."
* Spring Boot: Choose the latest stable version.
* Project Metadata:

Group: com.example (or your preferred group name).

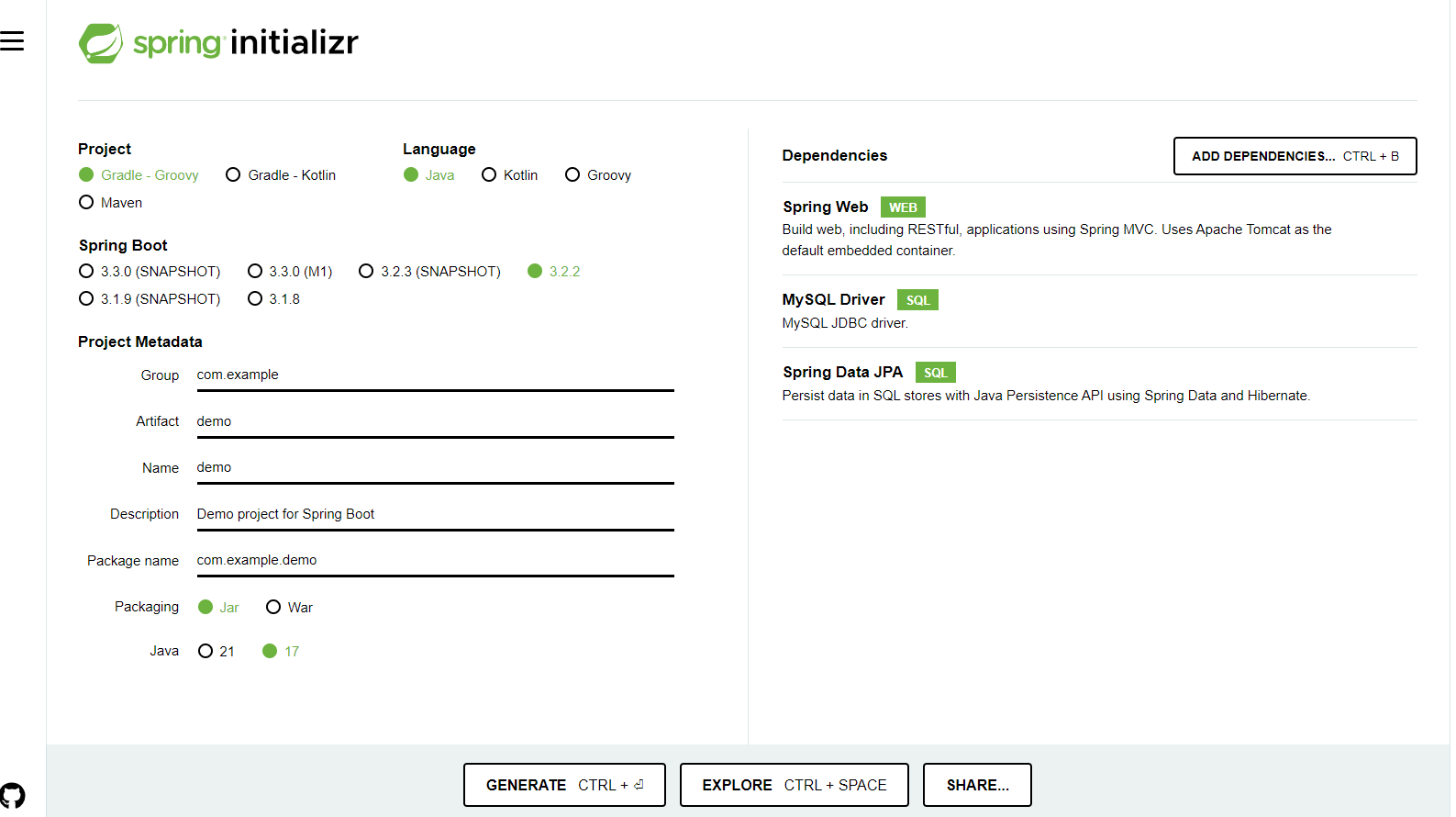
Artifact: crud-demo (or your preferred project name).

Name: CrudDemoApplication (or your preferred application name).

Description: Spring Boot CRUD Demo.

* Packaging: Select "Jar."
* Java: Choose your preferred Java version.

1. Click on “ADD DEPENDENCIES” to add dependencies.
   * Spring Web
   * MySQL Driver
   * Spring Data JPA
2. Click on the "Generate" button to download the project as a ZIP file.

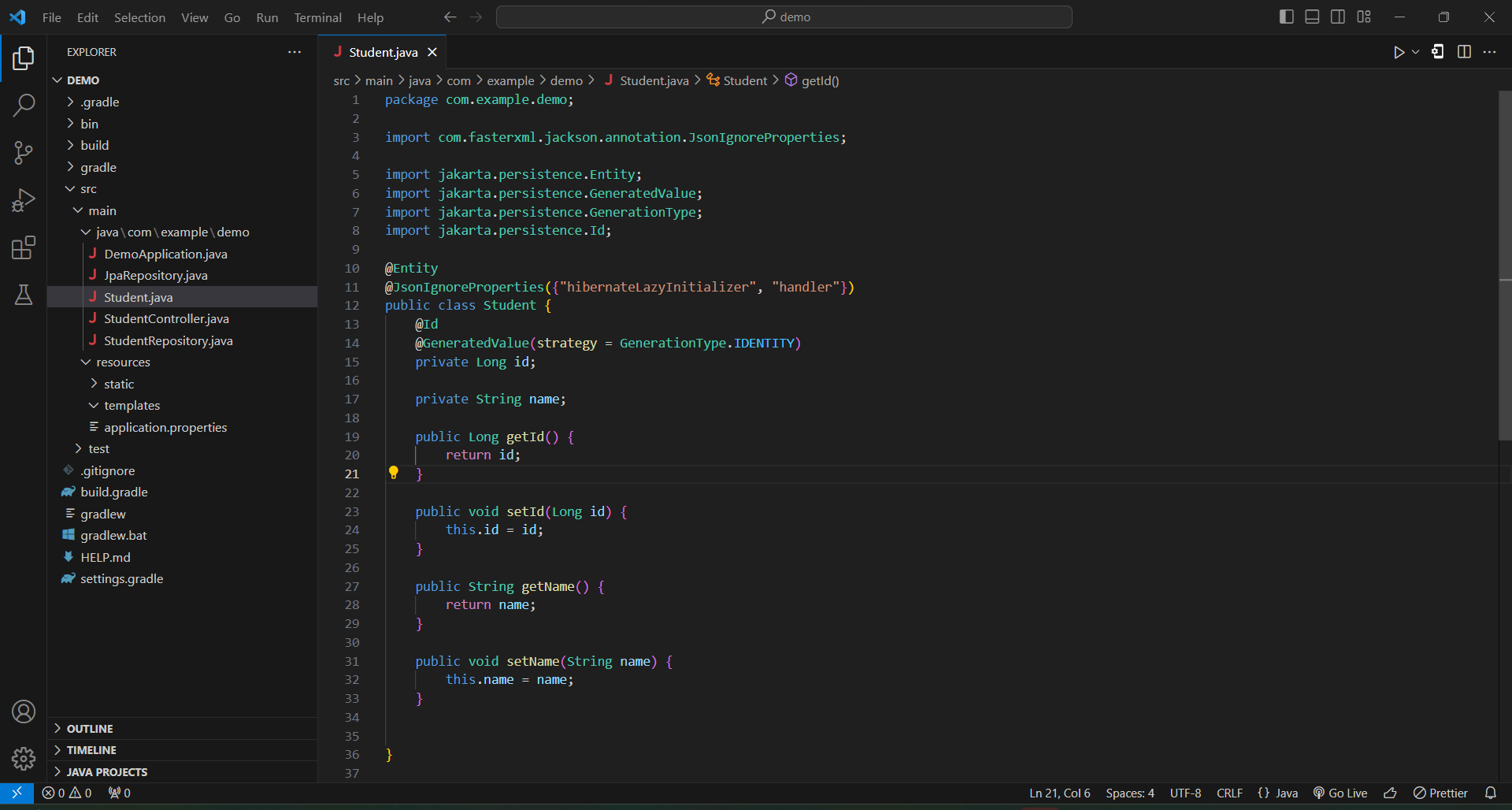


**Step 2: Open the Project**

* Extract the downloaded ZIP file to your preferred workspace.
* Open your favorite IDE (Eclipse, IntelliJ, etc.).

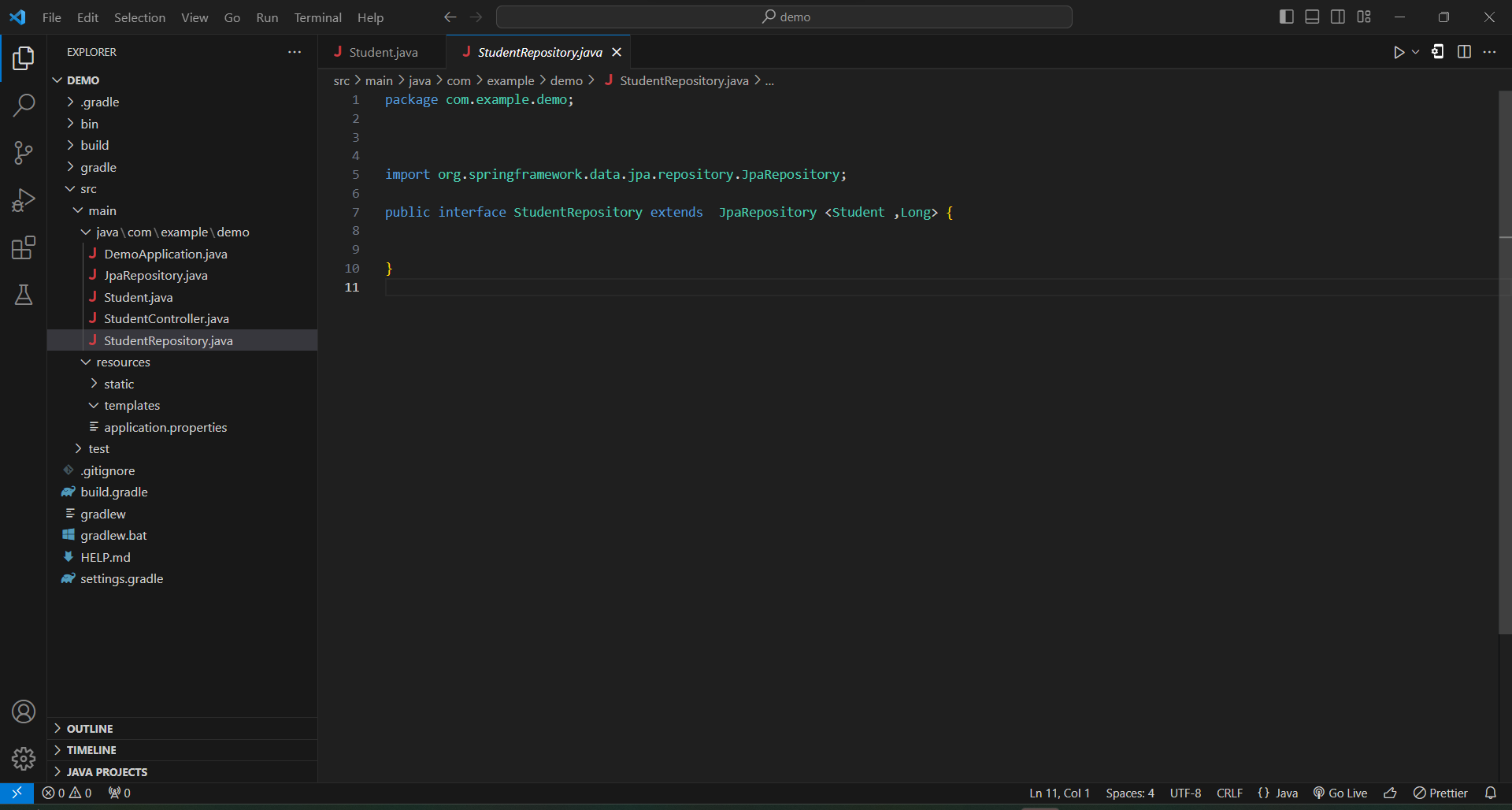
**Step 3: Create Entity Class**

* In the existing com.example.demo package, create a class named **Student** with fields like **id** and **name**.
* Add appropriate annotations such as **@Entity**, **@Id**, and others for JPA.



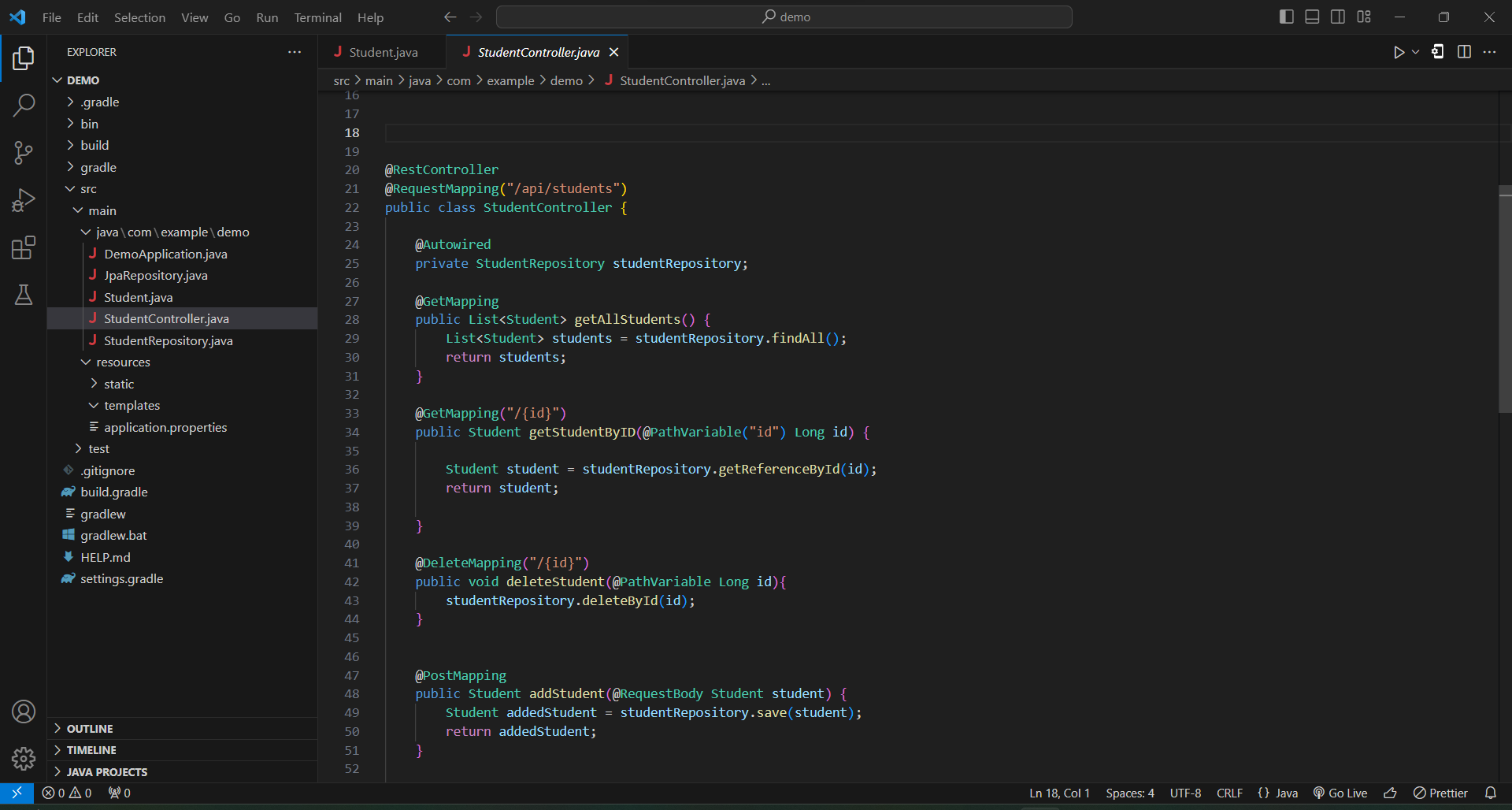
**Step 4: Create Repository Interface**

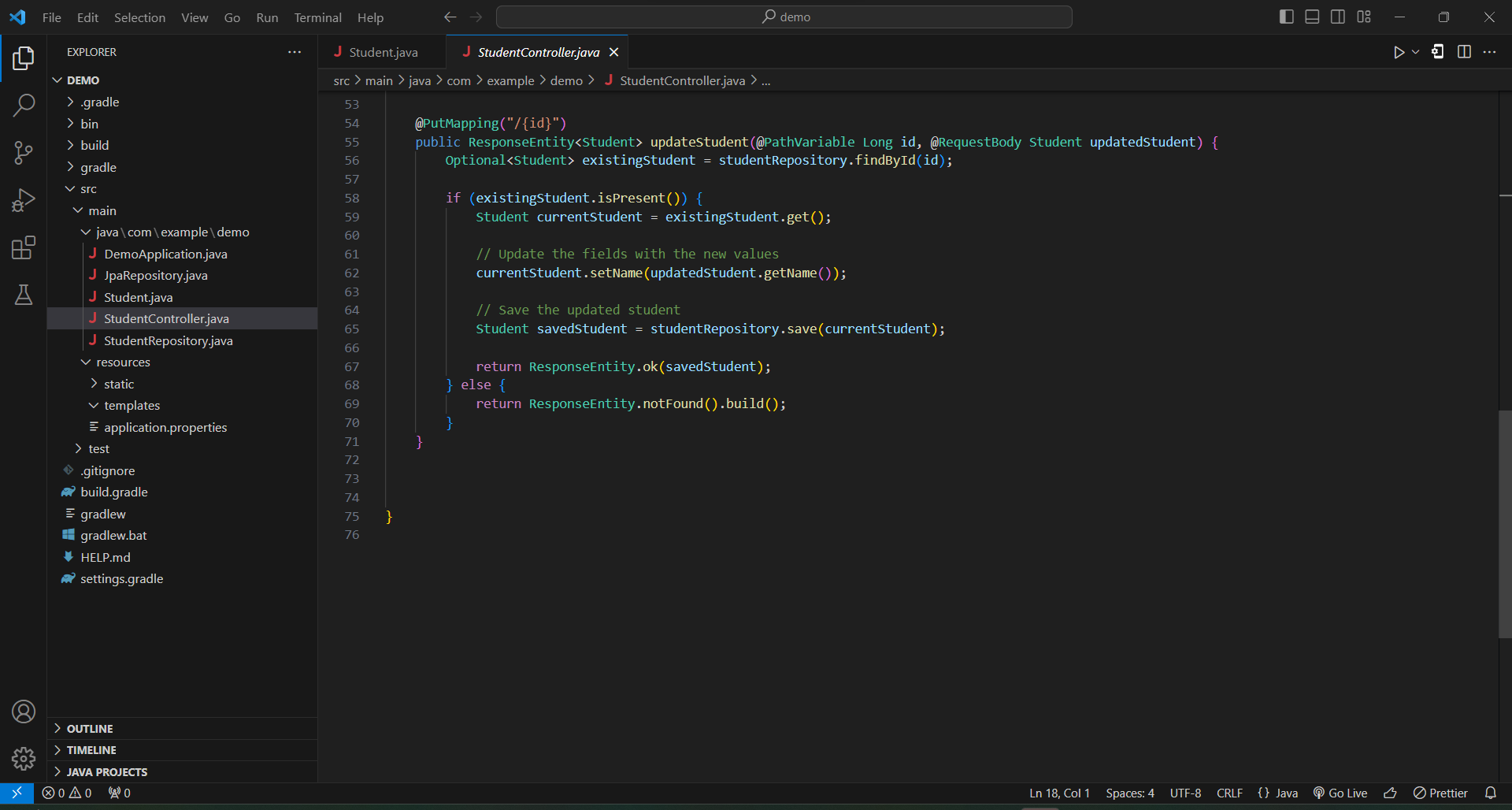
* Inside the StidentRepository.java, create an interface named StudentRepository that extends JpaRepository<Student, Long>.

****

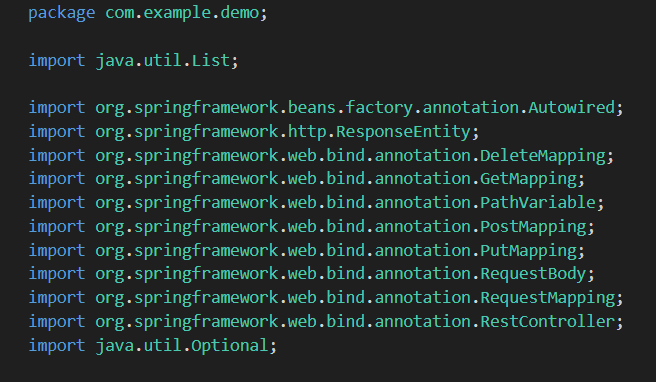
**Step 5: Create Controller**

* Create a file named StudentController.
* Inside the StudentController, modify the existing class StudentController.
* Annotate the class with @RestController.
* Implement methods for handling CRUD operations using StudentService.

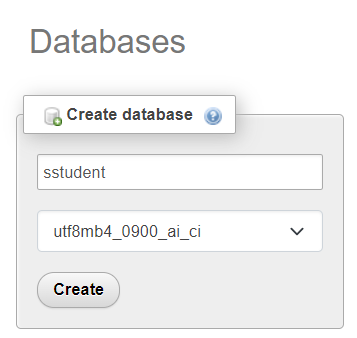




* Add following imports

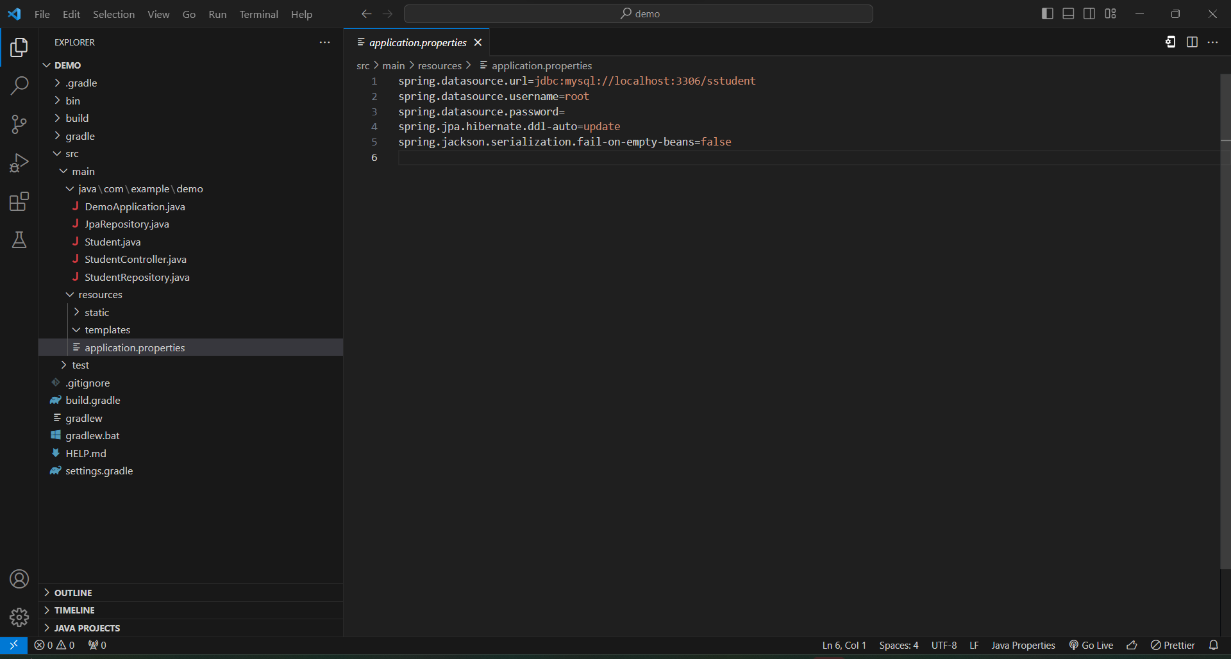


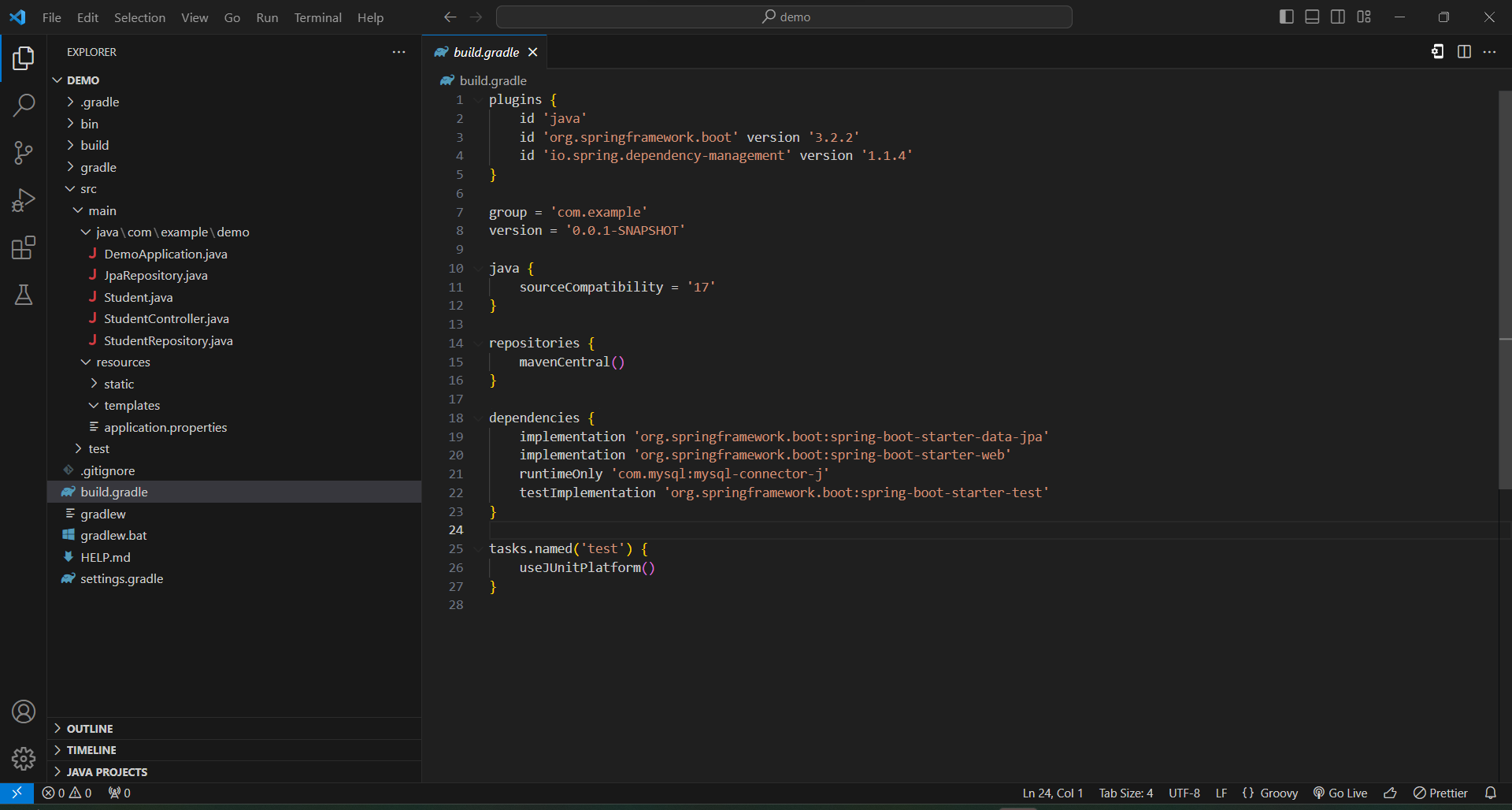
**Step 6: Create Database**

****

**Step 7: configuration properties**

* Set the application.properties.



* Check build.gradle file.

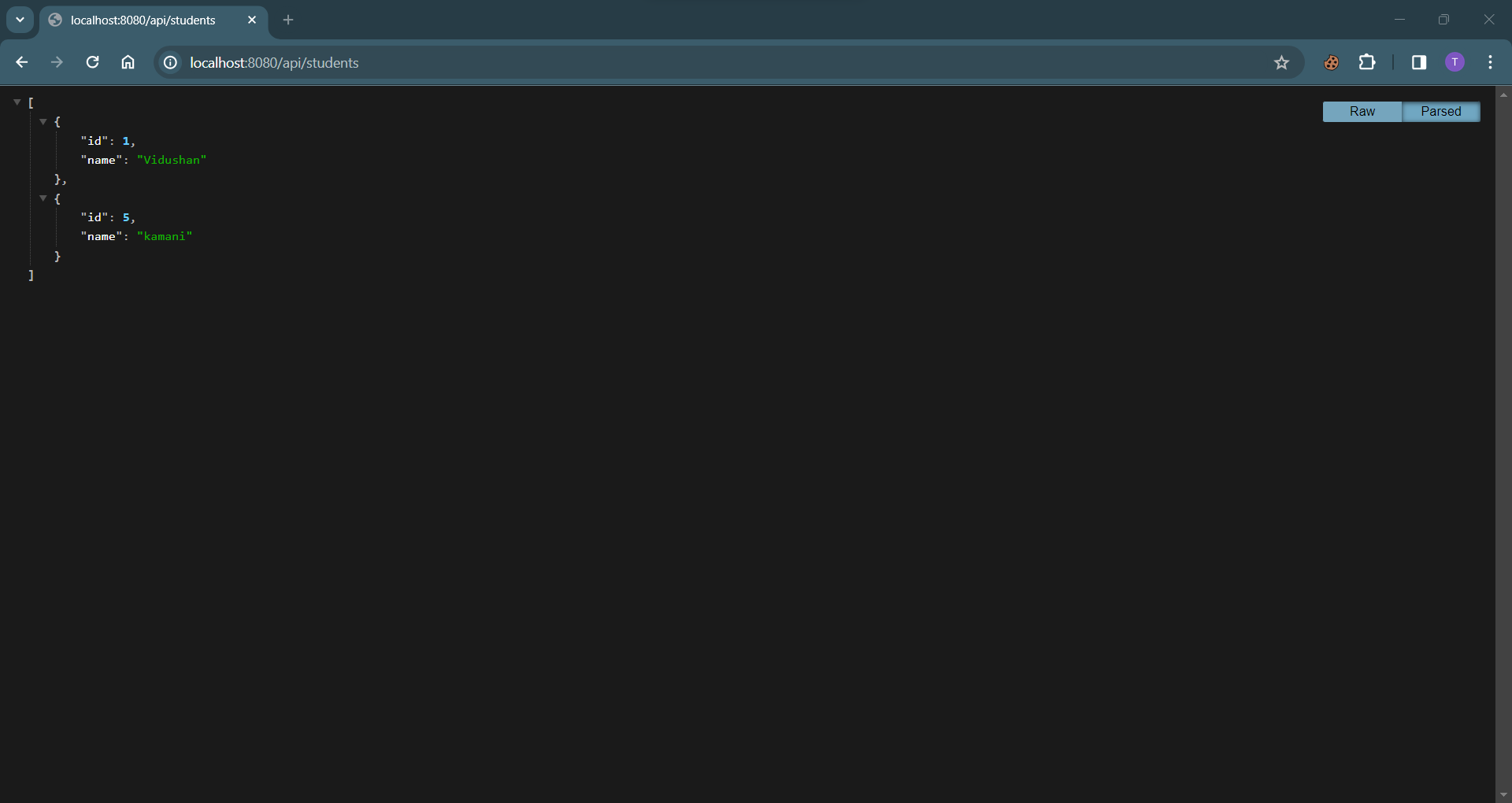
**Step 8: Run the Application**

* To run the application, run the following command in a terminal window (in the complete) directory.
* Ensure that the application starts without errors.

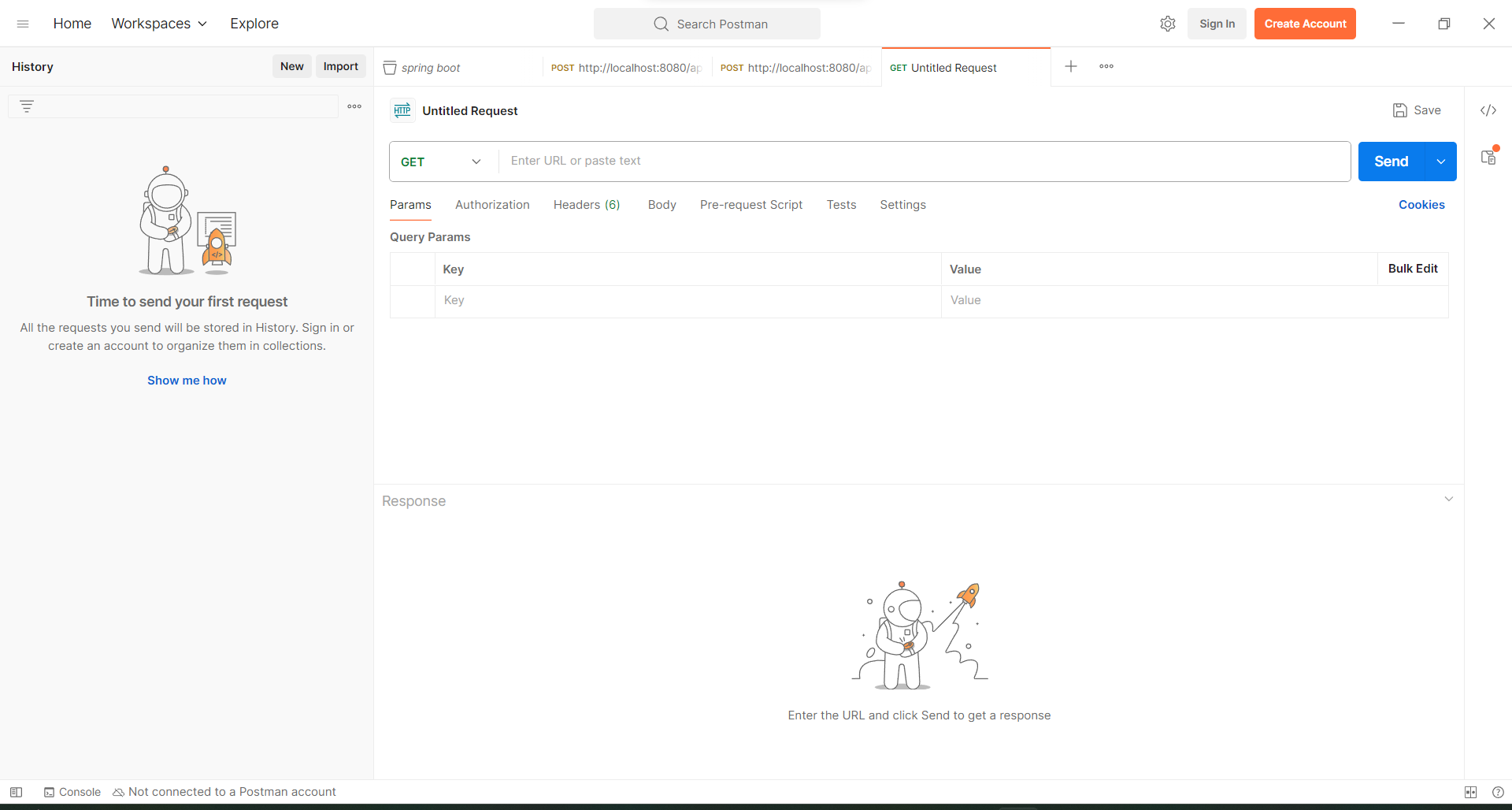
‘gradlew bootRun’

**Step 9: Test CRUD Operations**

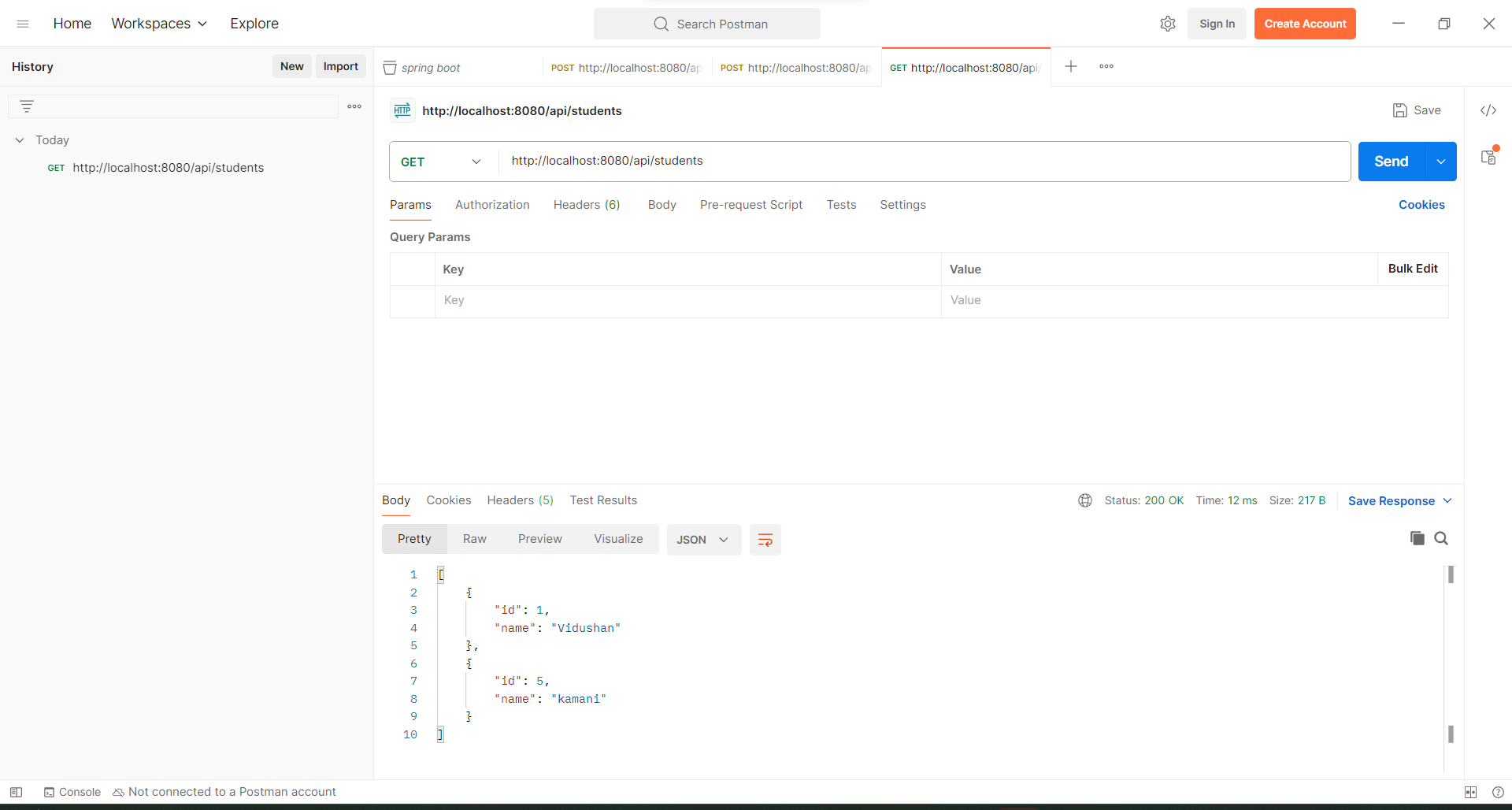
* Use tools like Postman or Swagger to test the CRUD operations (Create, Read, Update, Delete) for the Student entity.
* First, check the **Read** operation.
* Type this URL in the browser. ‘localhost:8080/api/students’



* Open postman.



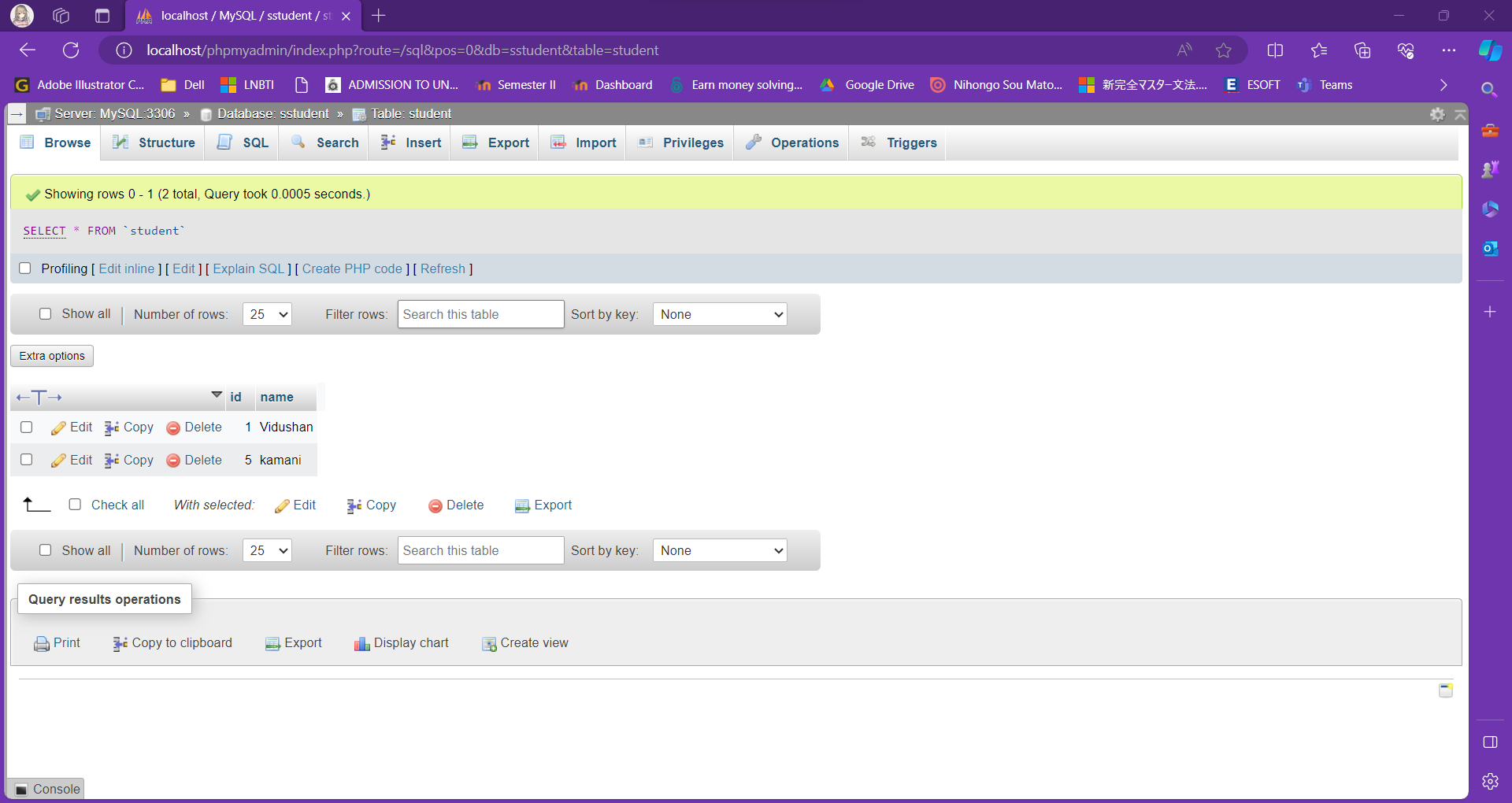
* Type URL ‘http://localhost:8080/api/students’.



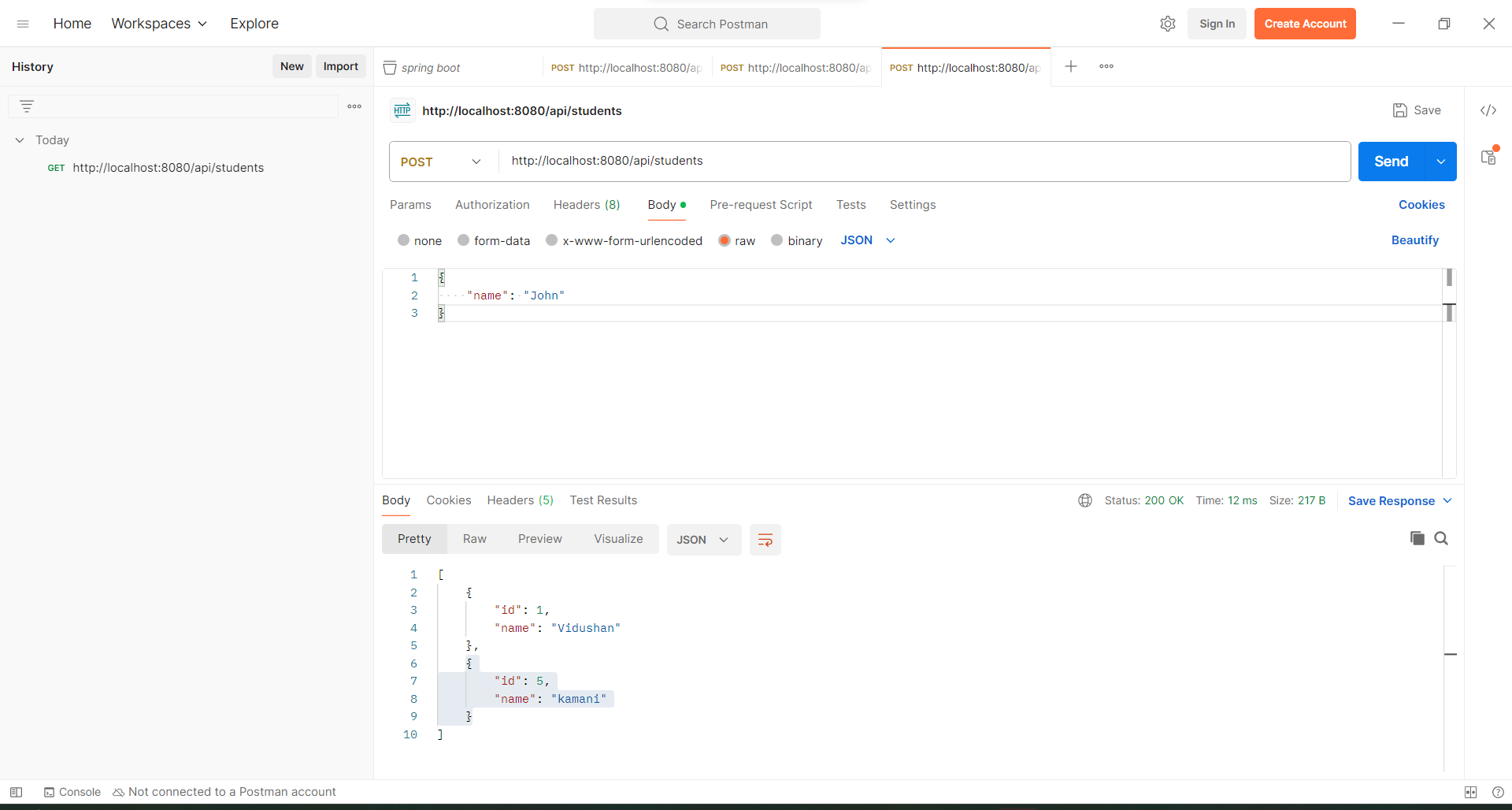
3.Click on Send

2.Insert ULR here

1.select method GET



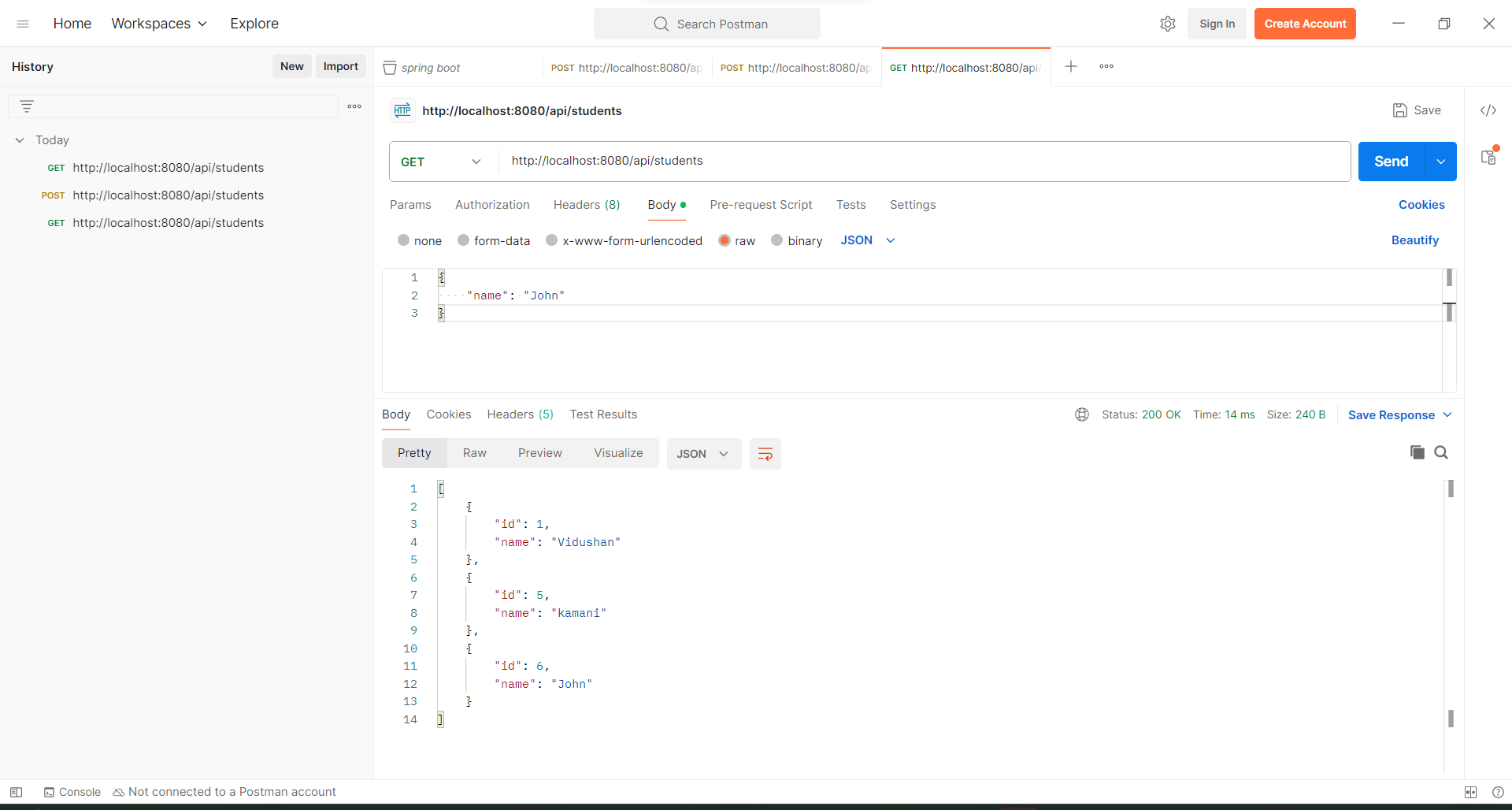
* Next check **Create** operation.

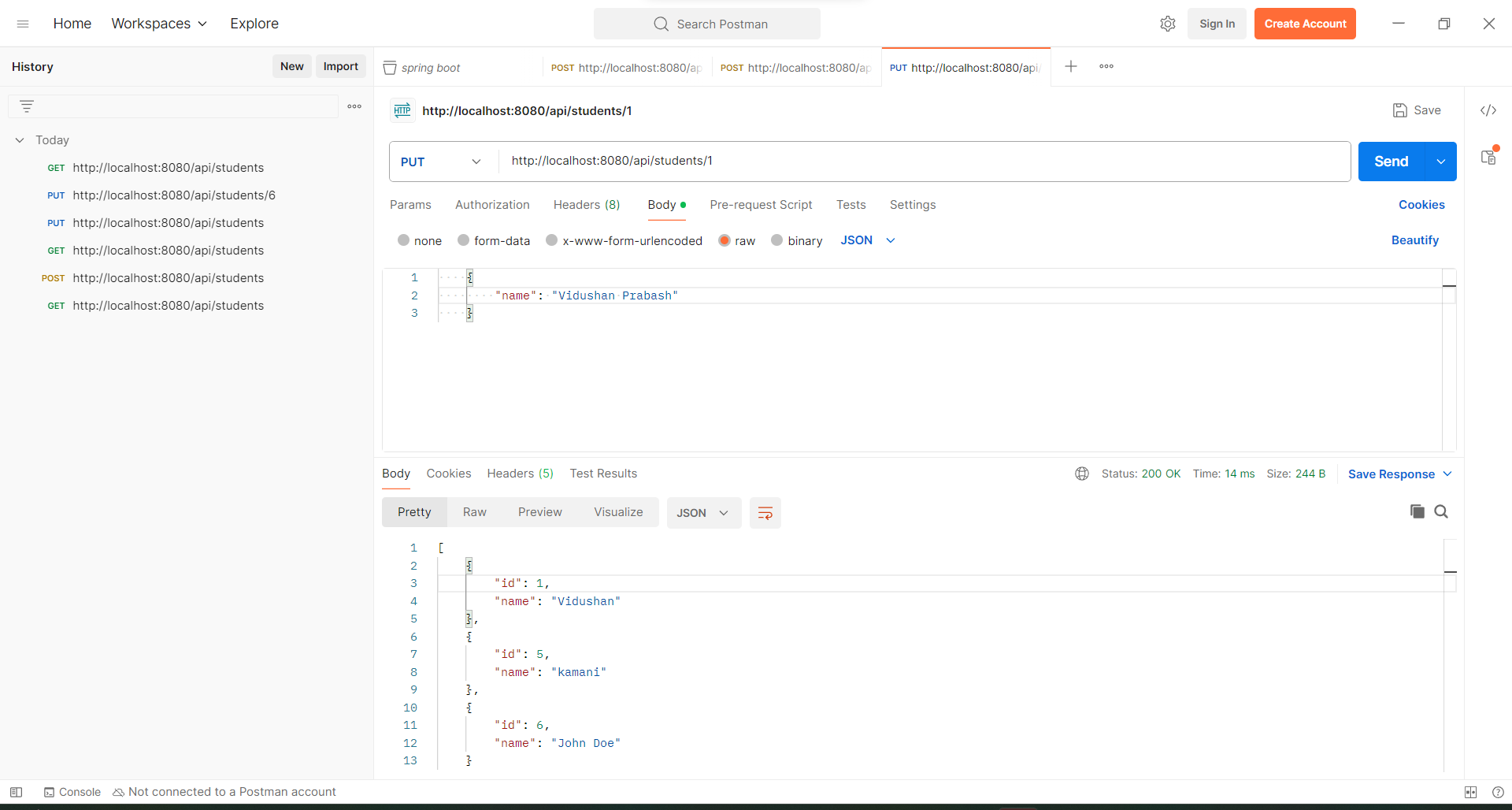


2.Type data you want to insert.

3.Select raw and JSON

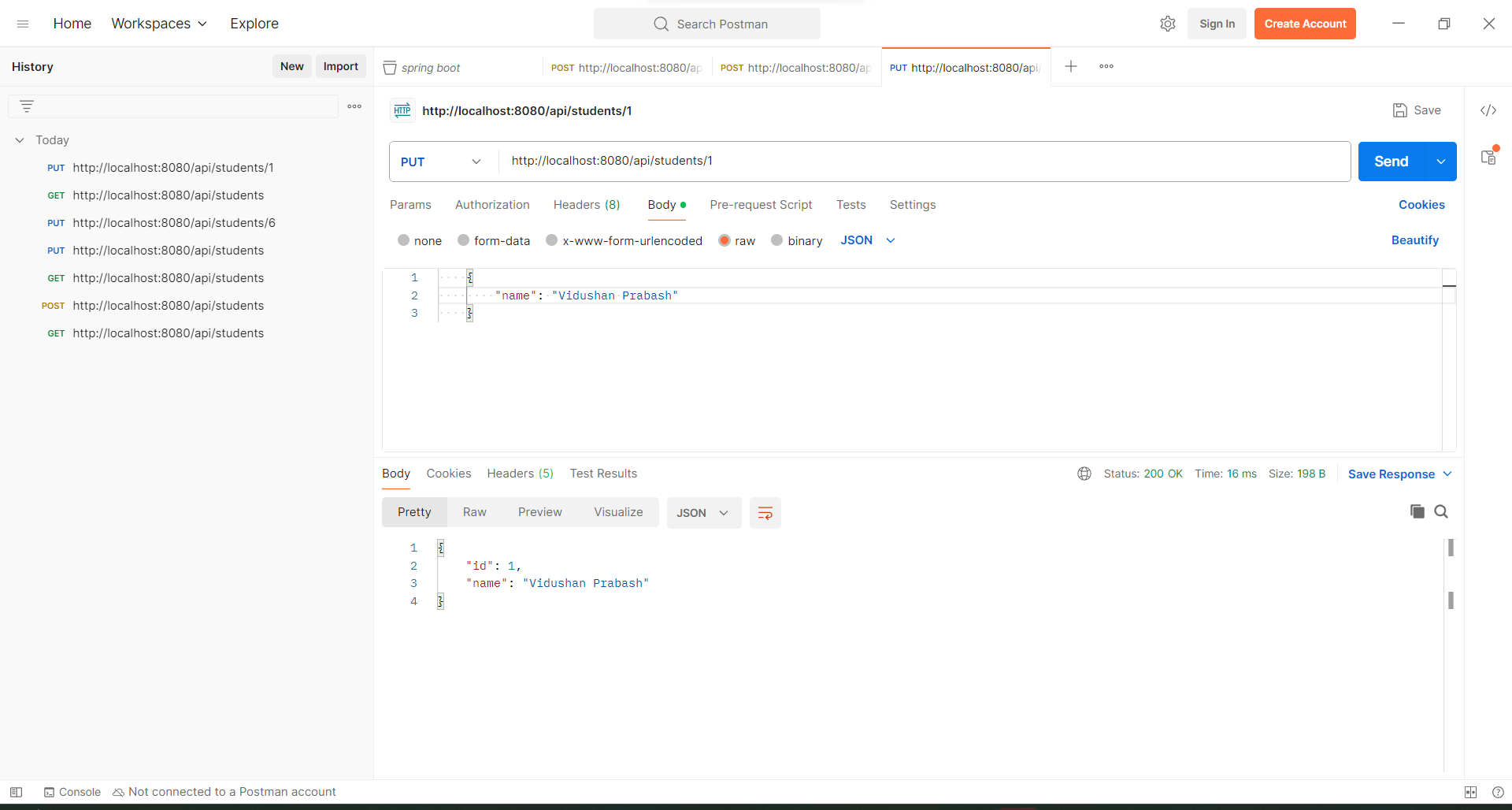
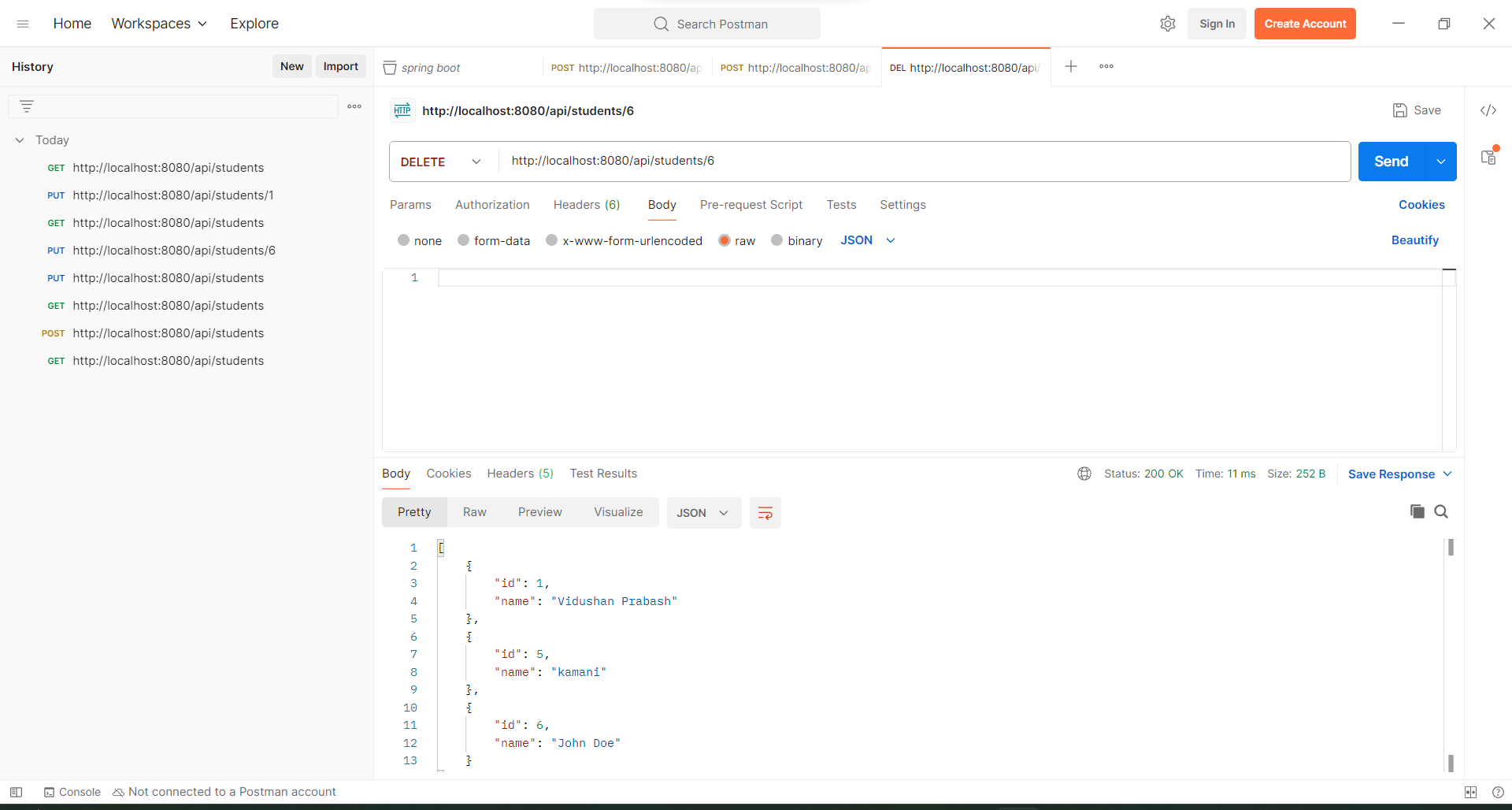
1.select method POST

* Data inserted successfully.
* Next check **Update** operation.

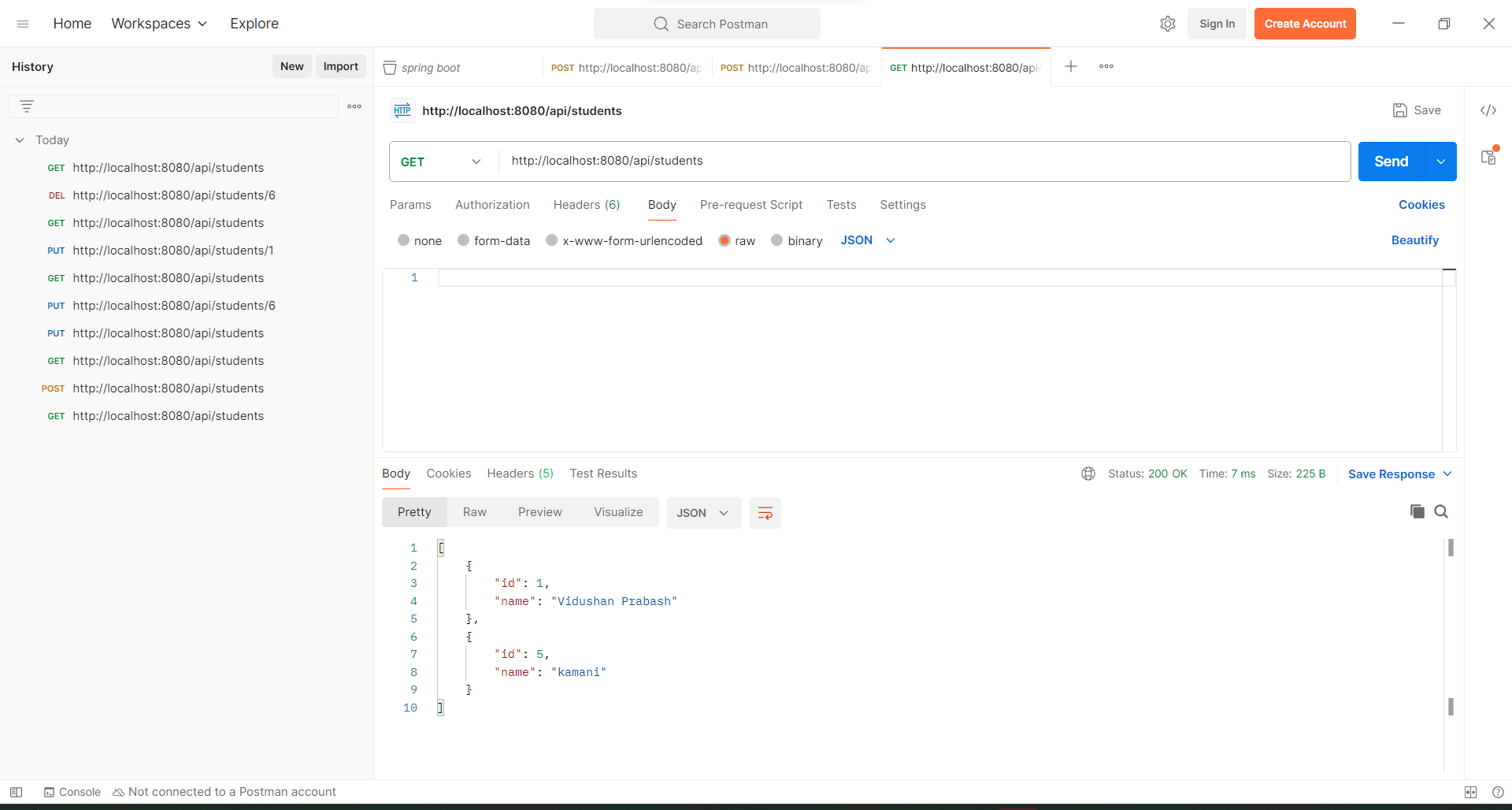


1.Type data you want to update

1.Select method PUT

* Successfully updated.
* Next, check the **Delete** operation.

1.Select method DELETE

* Check whether data has been successfully deleted.