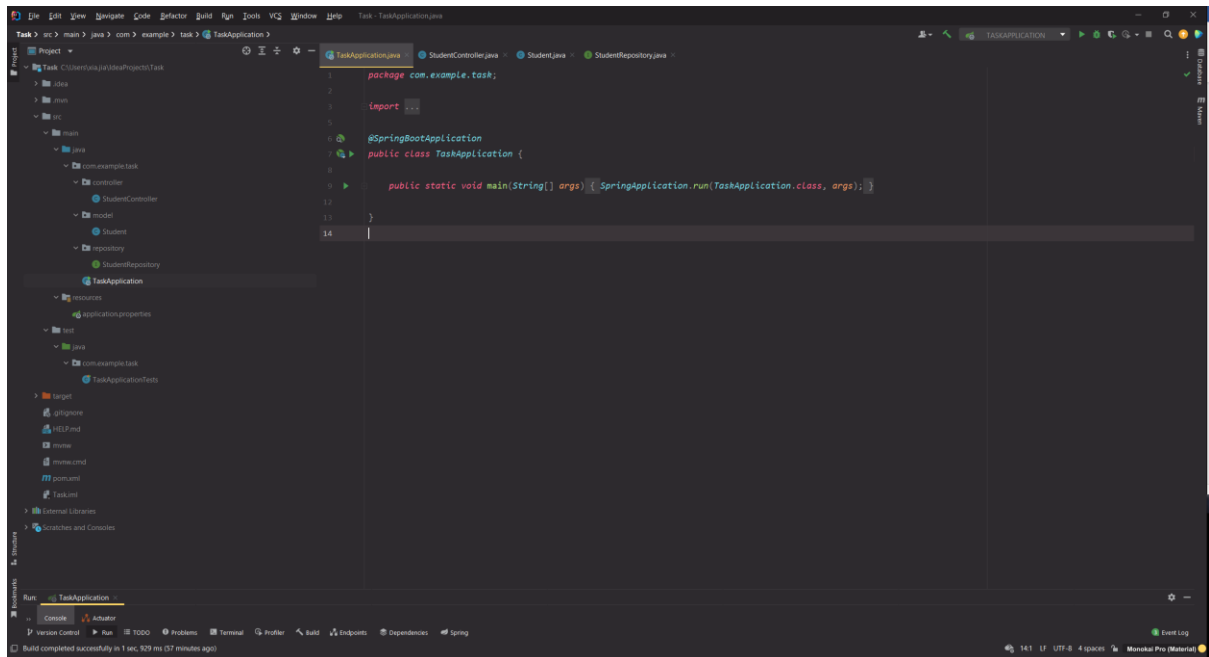


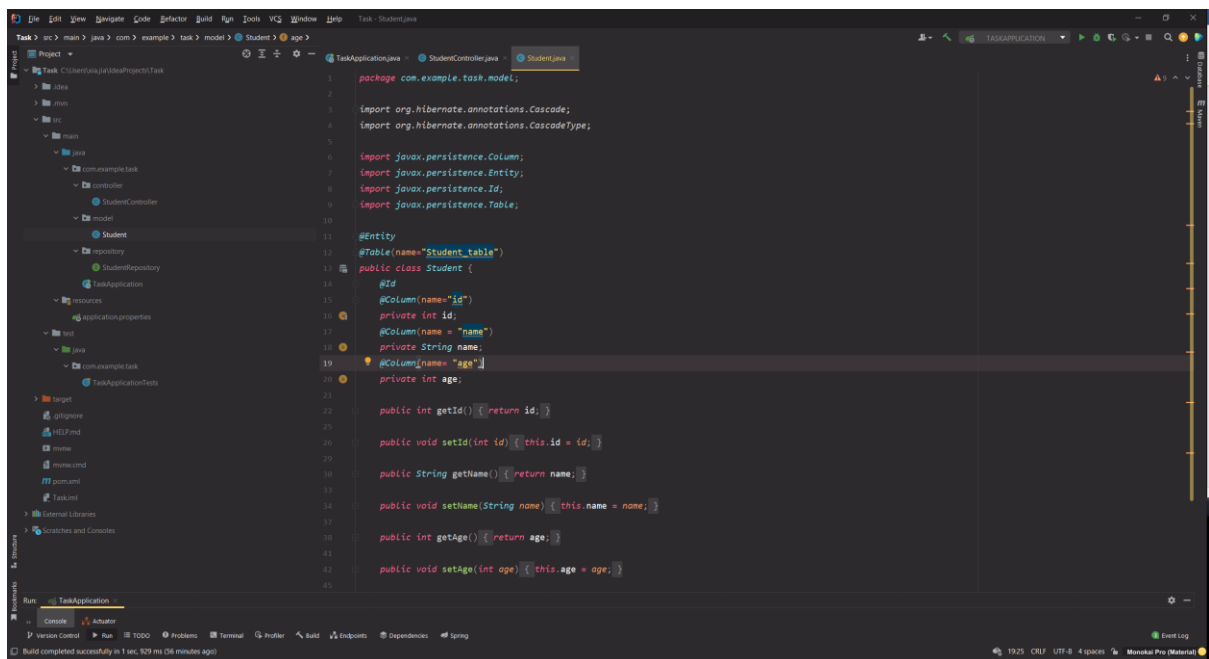
Part1 : Code

Main.java



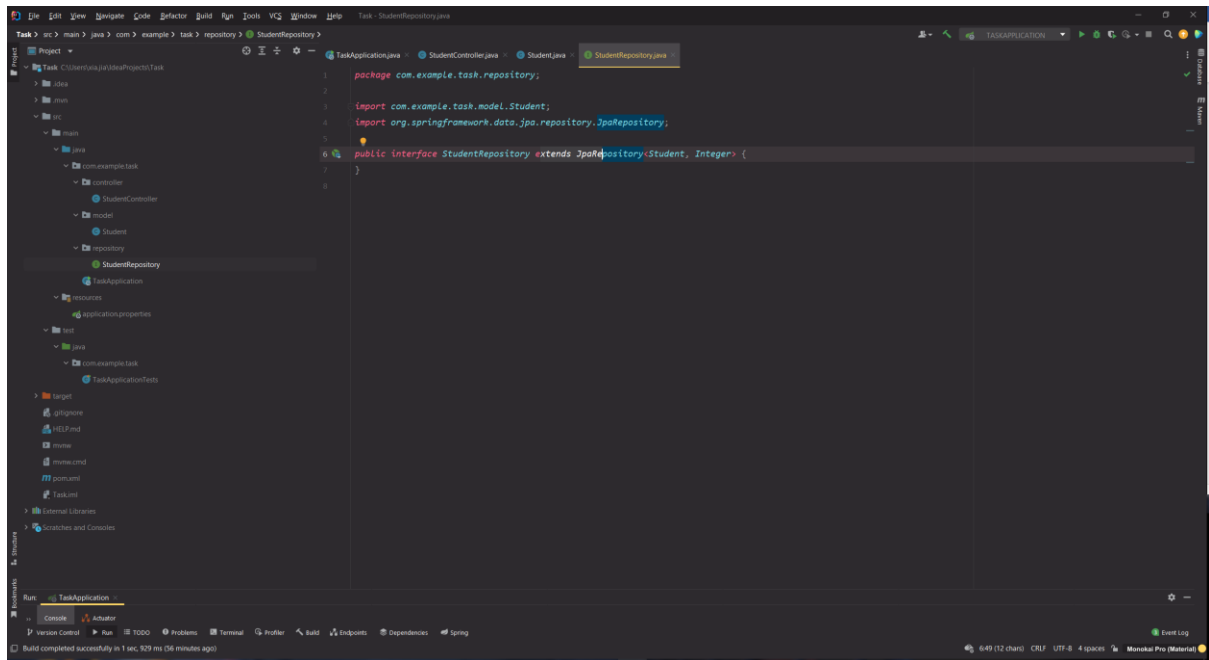
```
1 package com.example.task;
2
3 import ...
4
5 @SpringBootApplication
6 public class TaskApplication {
7
8     public static void main(String[] args) { SpringApplication.run(TaskApplication.class, args); }
9
10 }
11
12
13
14
```

Student.java

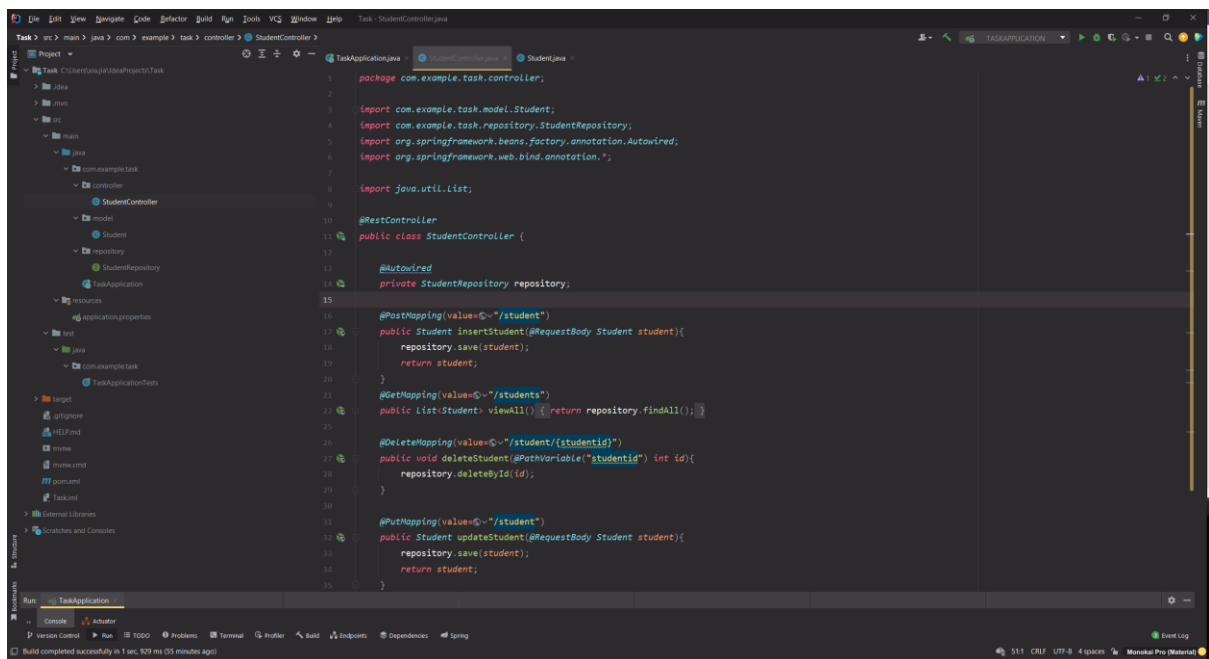


```
1 package com.example.task.model;
2
3 import org.hibernate.annotations.Cascade;
4 import org.hibernate.annotations.CascadeType;
5
6 import javax.persistence.Column;
7 import javax.persistence.Entity;
8 import javax.persistence.Id;
9 import javax.persistence.Table;
10
11 @Entity
12 @Table(name="Student_table")
13 public class Student {
14     @Id
15     @Column(name="id")
16     private int id;
17     @Column(name = "name")
18     private String name;
19     @Column(name = "age")
20     private int age;
21
22     public int getId() { return id; }
23
24     public void setId(int id) { this.id = id; }
25
26     public String getName() { return name; }
27
28     public void setName(String name) { this.name = name; }
29
30     public int getAge() { return age; }
31
32     public void setAge(int age) { this.age = age; }
33
34 }
35
```

StudentRepository



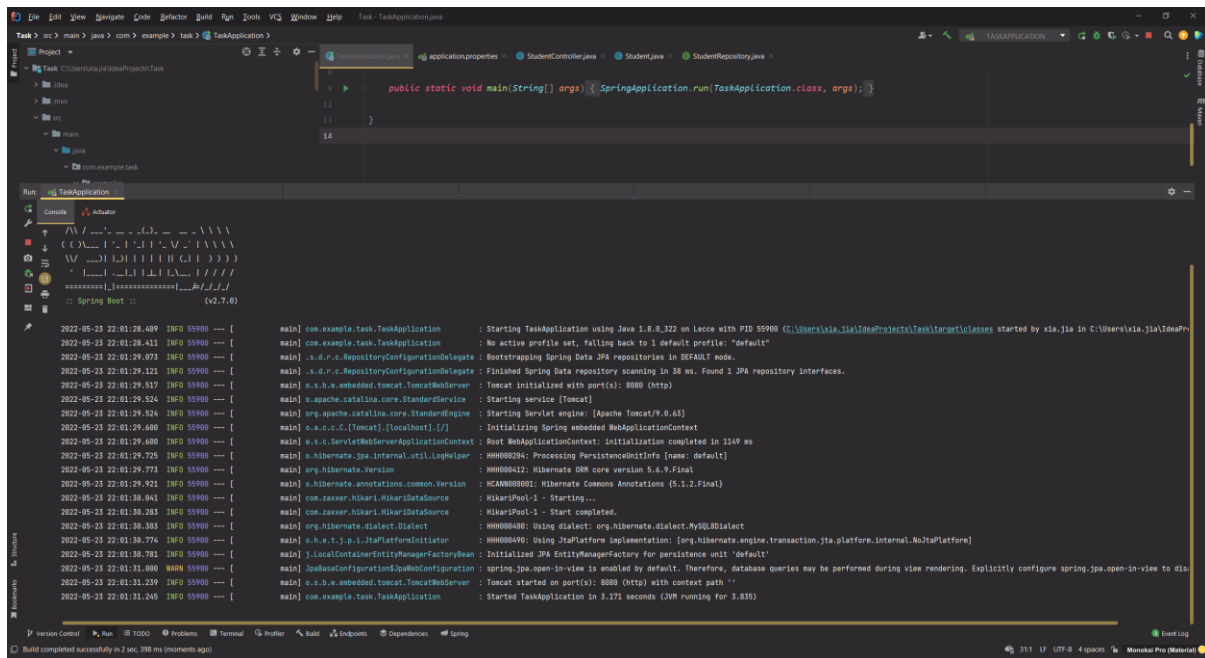
StudentController.java



```
spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto=update

spring.datasource.url=jdbc:mysql://localhost:3306/jdbc
spring.datasource.username=root
spring.datasource.password=root1234
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

It can work

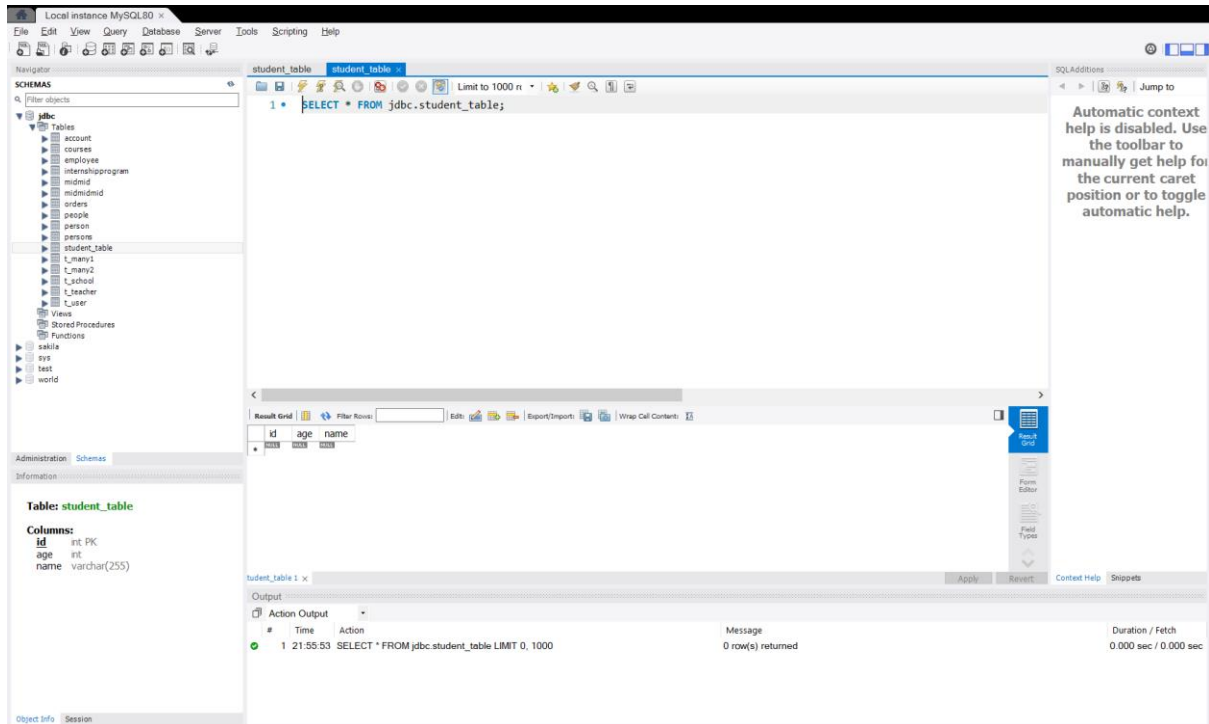


```
Task > src > main > java > com > example > task > TaskApplication >
Project: Task
Run: TaskApplication
TaskApplication.java
public static void main(String[] args) {
    SpringApplication.run(TaskApplication.class, args);
}

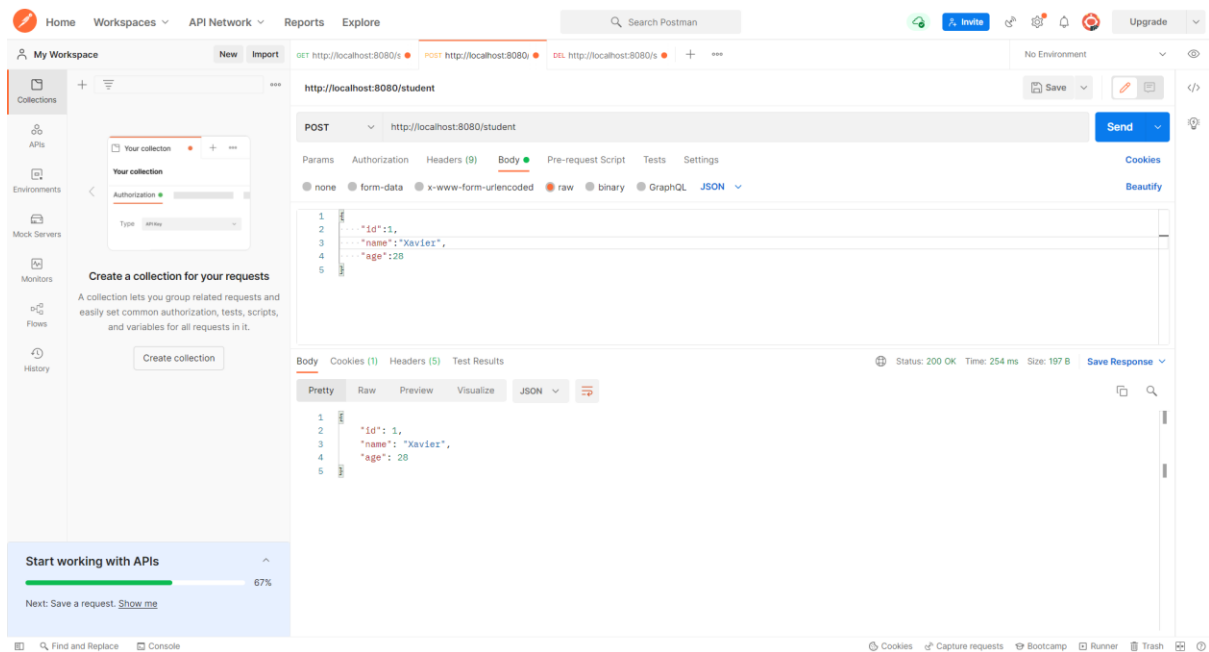
Run: TaskApplication
Console:
2022-05-23 22:01:28.409 INFO 55900 [main] com.example.task.TaskApplication : Starting TaskApplication using Java 1.8.0_322 on Lecce with PID 55900 (C:\Users\xia.jia\IdeaProjects\Task\target\classes started by xia.jia in C:\Users\xia.jia\IdeaProjects\Task)
2022-05-23 22:01:28.411 INFO 55900 [main] com.example.task.TaskApplication : No active profile set, falling back to 1 default profile: 'default'
2022-05-23 22:01:29.073 INFO 55900 [main] org.springframework.data.jpa.repository.config.JpaRepositoriesConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2022-05-23 22:01:29.121 INFO 55900 [main] org.springframework.data.jpa.repository.config.JpaRepositoriesConfigurationDelegate : Finished Spring Data repository scanning in 38 ms. Found 1 JPA repository interfaces.
2022-05-23 22:01:29.517 INFO 55900 [main] org.springframework.boot.web.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2022-05-23 22:01:29.524 INFO 55900 [main] org.apache.catalina.core.StandardService : Starting service [Tomcat]
2022-05-23 22:01:29.524 INFO 55900 [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.63]
2022-05-23 22:01:29.680 INFO 55900 [main] org.springframework.boot.web.embedded.tomcat.TomcatWebServer : Initializing Spring embedded WebApplicationContext
2022-05-23 22:01:29.680 INFO 55900 [main] org.springframework.boot.web.embedded.tomcat.TomcatWebServer : Root WebApplicationContext: initialization completed in 1149 ms
2022-05-23 22:01:29.725 INFO 55900 [main] org.hibernate.jpa.internal.util.LogHelper : HHH0000294: Processing PersistenceUnitInfo [name: default]
2022-05-23 22:01:29.773 INFO 55900 [main] org.hibernate.Version : HHH0000412: Hibernate ORM core version 5.4.9.Final
2022-05-23 22:01:29.921 INFO 55900 [main] org.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations (5.1.2.Final)
2022-05-23 22:01:30.041 INFO 55900 [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2022-05-23 22:01:30.283 INFO 55900 [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2022-05-23 22:01:30.303 INFO 55900 [main] org.hibernate.dialect.Dialect : HHH000490: using dialect: org.hibernate.dialect.MySQL8Dialect
2022-05-23 22:01:30.374 INFO 55900 [main] org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform : HHH000490: using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
2022-05-23 22:01:30.781 INFO 55900 [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2022-05-23 22:01:31.080 WARN 55900 [main] org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this behavior
2022-05-23 22:01:31.239 INFO 55900 [main] org.springframework.boot.web.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2022-05-23 22:01:31.245 INFO 55900 [main] com.example.task.TaskApplication : Started TaskApplication in 3.171 seconds (JVM running for 3.835s)
```

Part2 : Run

At first the database is empty



INSERT A STUDENT



```
Hibernate: select student0_.id as id1_0_0_, student0_.age as age2_0_0_, student0_.name as name3_0_0_ from student_table student0_ where student0_.id=?
Hibernate: insert into student_table (age, name, id) values (?, ?, ?)
```

The database changed

The screenshot shows the MySQL Workbench interface. On the left, the 'Schemas' pane shows a database named 'jdbc' with various tables. The 'student_table' is selected. The main query editor shows the following SQL query:

```
1 * SELECT * FROM jdbc.student_table;
```

The 'Output' pane at the bottom shows the results of the query execution:

#	Time	Action	Message	Duration / Fetch
1	21:55:53	SELECT * FROM jdbc.student_table LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
2	22:02:40	SELECT * FROM jdbc.student_table LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

The 'Table: student_table' information pane shows the following columns:

- id: int PK
- age: int
- name: varchar(255)

The 'Result Grid' shows the following data:

id	age	name
1	28	Xavier

UPDATE A STUDENT

The screenshot shows the Postman interface. A PUT request is being made to the endpoint `http://localhost:8080/student`. The request body is a JSON object:

```
{  "id": 1,  "name": "Vincent",  "age": 29}
```

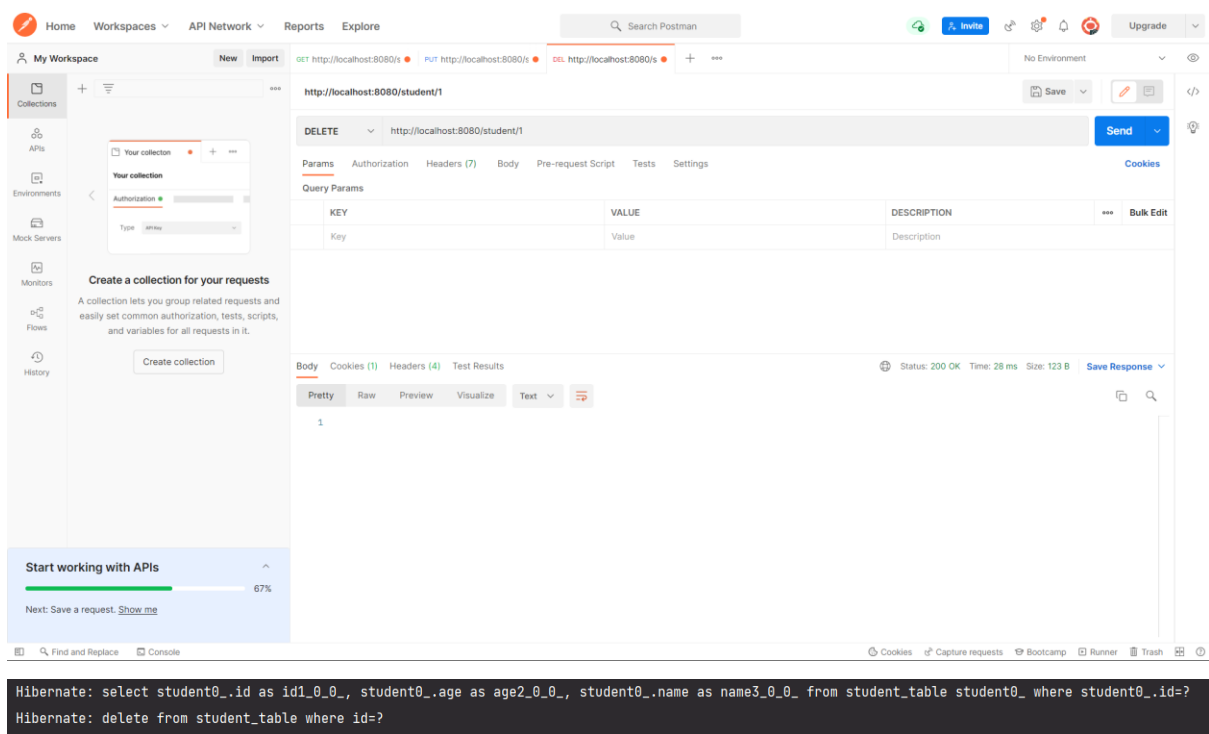
The response status is 200 OK. The response body is a JSON object:

```
{  "id": 1,  "name": "Vincent",  "age": 29}
```

```
Hibernate: select student0_.id as id1_0_0_, student0_.age as age2_0_0_, student0_.name as name3_0_0_ from student_table student0_ where student0_.id=?  
Hibernate: update student_table set age=?, name=? where id=?
```

	id	age	name
▶	1	29	Vincent
✱	NULL	NULL	NULL

DELETE A STUDENT

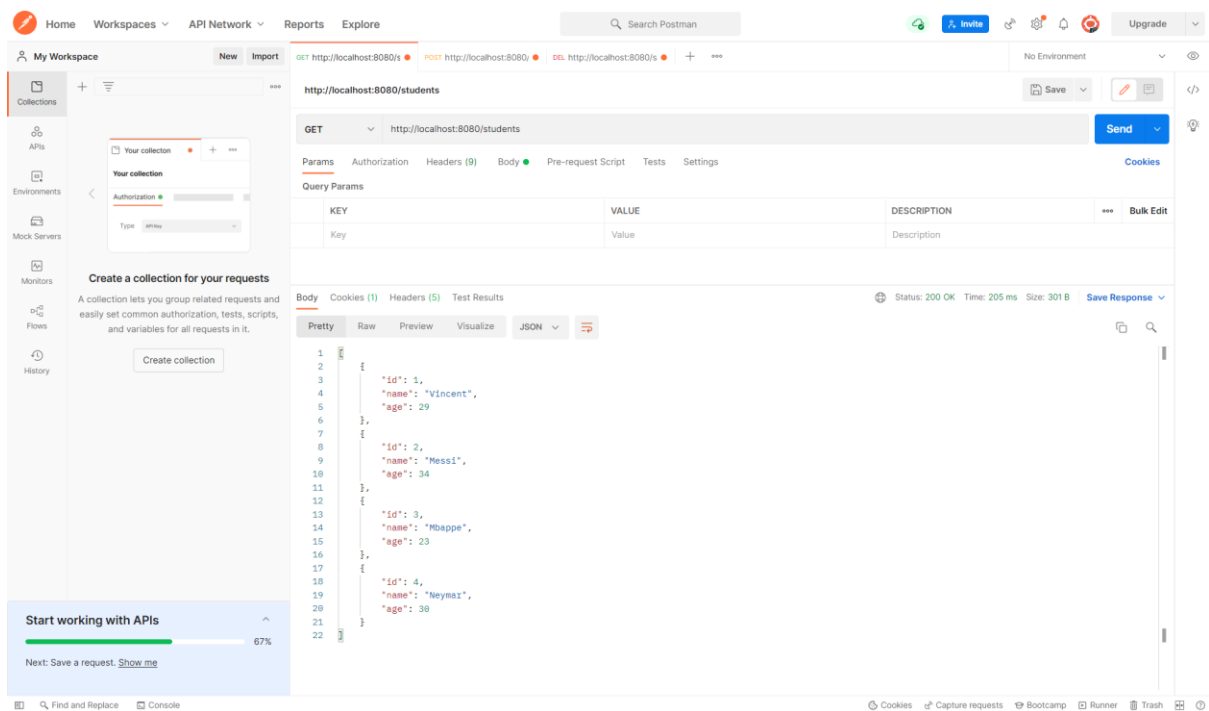


The Postman interface shows a DELETE request to `http://localhost:8080/student/1`. The response status is 200 OK, with a time of 28 ms and a size of 123 B. The response body is empty.

Below the interface, a terminal window shows the following Hibernate SQL queries:

```
Hibernate: select student0_.id as id1_0_0_, student0_.age as age2_0_0_, student0_.name as name3_0_0_ from student_table student0_ where student0_.id=?
Hibernate: delete from student_table where id=?
```

SHOW STUDENTS



The Postman interface shows a GET request to `http://localhost:8080/students`. The response status is 200 OK, with a time of 205 ms and a size of 301 B. The response body is a JSON array of student objects.

```
[{"id": 1, "name": "Vincent", "age": 29}, {"id": 2, "name": "Messi", "age": 34}, {"id": 3, "name": "Mbappe", "age": 23}, {"id": 4, "name": "Neymar", "age": 38}]
```

	id	age	name
▶	1	29	Vincent
	2	34	Messi
	3	23	Mbappe
	4	30	Neymar
•	NULL	NULL	NULL