

## Spring\_GET\_POST\_DELETE\_UPDATE

<https://springjava.com/spring-boot/spring-boot-crud-operation-using-rest>

controller  
model  
repository  
service  
    impl  
        serviceImpl  
        service  
MainMethod

---

Start project

application.properties

```
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/Employee
spring.datasource.username=root
spring.datasource.password=Data#123456!
spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect
t.MySQL8Dialect
server.port=12345
```

Pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.7.8</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.example</groupId>
    <artifactId>Spring_GET_POST_DELETE_UPDATE</artifactId>
```

```
<version>0.0.1-SNAPSHOT</version>
<name>Spring_GET_POST_DELETE_UPDATE</name>
<description>Demo project for Spring Boot</description>
<properties>
  <java.version>11</java.version>
</properties>
<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-web</artifactId>
    </dependency>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-devtools</artifactId>
      <scope>runtime</scope>
      <optional>true</optional>
    </dependency>
    <dependency>
      <groupId>com.h2database</groupId>
      <artifactId>h2</artifactId>
      <scope>runtime</scope>
    </dependency>
    <dependency>
      <groupId>com.mysql</groupId>
      <artifactId>mysql-connector-j</artifactId>
      <scope>runtime</scope>
    </dependency>
    <dependency>
      <groupId>org.projectlombok</groupId>
      <artifactId>lombok</artifactId>
      <optional>true</optional>
    </dependency>
```

```

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
        <configuration>
          <excludes>
            <exclude>
              <groupId>org.projectlombok</groupId>
              <artifactId>lombok</artifactId>
            </exclude>
          </excludes>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>

```

1).model.

```

@NoArgsConstructor
@AllArgsConstructor
@Getter
@Setter
@Entity
@Table(name = "students")
public class Student {
    @Id
    private Integer UserId;
    private String name;
}

```

## 2).repository

```
public interface StudentRepository extends
JpaRepository<Student, Integer> {
    Student save(Student student);
    void delete(Student UserId);
}
```

## 3).service

```
@Service
public interface StudentService {
    public List<Student> getStudent();
    public void save(Student student);
    public Student findById(Integer Id);
    public void delete(Student student);
}
```

## 4).serviceimpl

```
@Service
public class StudentServiceImpl implements StudentService {
    @Autowired
    private StudentRepository studentRepository;
    @Override
    public List<Student> getStudent() {
        return studentRepository.findAll();
    }
    @Override
    public void save(Student student) {
        studentRepository.save(student);
    }
    @Override
    public Student findById(Integer Id) {
        Optional<Student> studentResult=
studentRepository.findById(Id);
        Student student=null;
        if (studentResult.isPresent()){
```

```

        student=studentResult.get();
    }
    return student;
}
@Override
public void delete(Student student) {
    studentRepository.delete(student);
}
}

```

## 5).controller

```

@RestController
public class StudentController {
    @Autowired
    private StudentService studentService;
    @Autowired
    private StudentRepository studentRepository;

    @GetMapping("/student")
    public List<Student> getStudent() {
        return studentService.getStudent();
    }

    @PostMapping("/student")
    public Student save(@RequestBody Student student) {
        studentService.save(student);
        return student;
    }

    @DeleteMapping("/student/{UserId}")
    public void delete(@PathVariable Student UserId) {
        studentService.delete(UserId);
    }

    @PutMapping("/student/{UserId}")
    public Student updateStudent(@RequestBody Student
student, @PathVariable Student UserId ){
        Student student1 =
studentService.findById(UserId.getUserId());

```

```
        student1.setName(student.getName());  
        studentService.save(student1);  
        return student1;  
    }  
}
```

#### 6).main method

```
@SpringBootApplication  
public class SpringGetPostDeleteUpdateApplication {  
    public static void main(String[] args) {  
  
        SpringApplication.run(SpringGetPostDeleteUpdateApplication.class, args);  
    }  
}
```

## Final project

---