BARGABINO,MICHAEL JOSH V.

FORTE, TIMOTHY

IV-ACSAD

CODE

POM.XML

pom.xml (springboot-restful-webservices)

```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>3.0.0-M4</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>net.javaguides</groupId>
    <artifactId>springboot-restful-webservices</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>springboot-restful-webservices</name>
    <description>Demo project for Spring Boot Restful Webservices</description>
    <properties>
        <java.version>17</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>
```

pom.xml (springboot-restful-webservices)

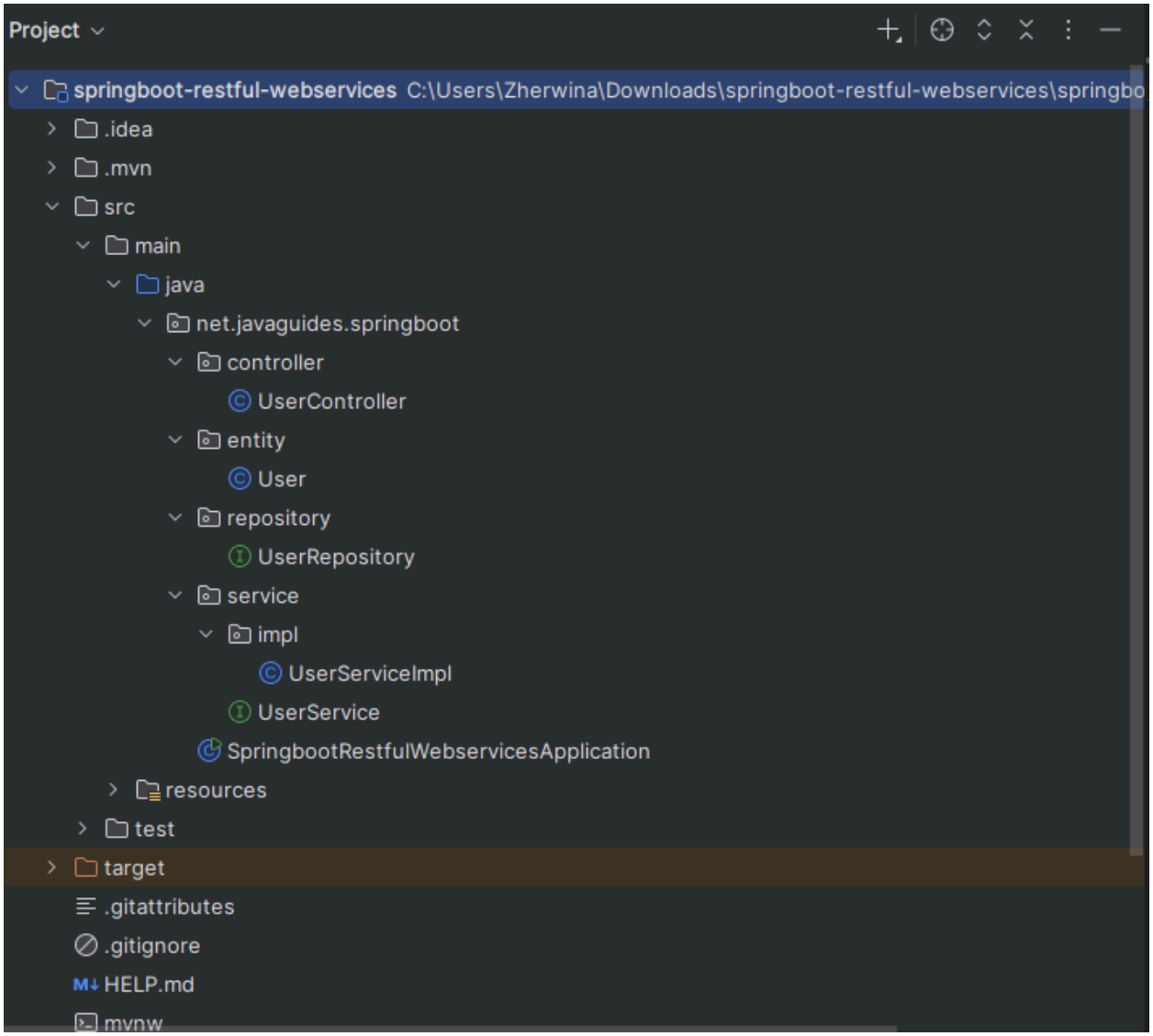
```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    <dependencies>
        <dependency>
            <artifactId>mysql-connector-java</artifactId>
            <scope>runtime</scope>
        </dependency>
        <dependency>
            <groupId>org.projectlombok</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>
```

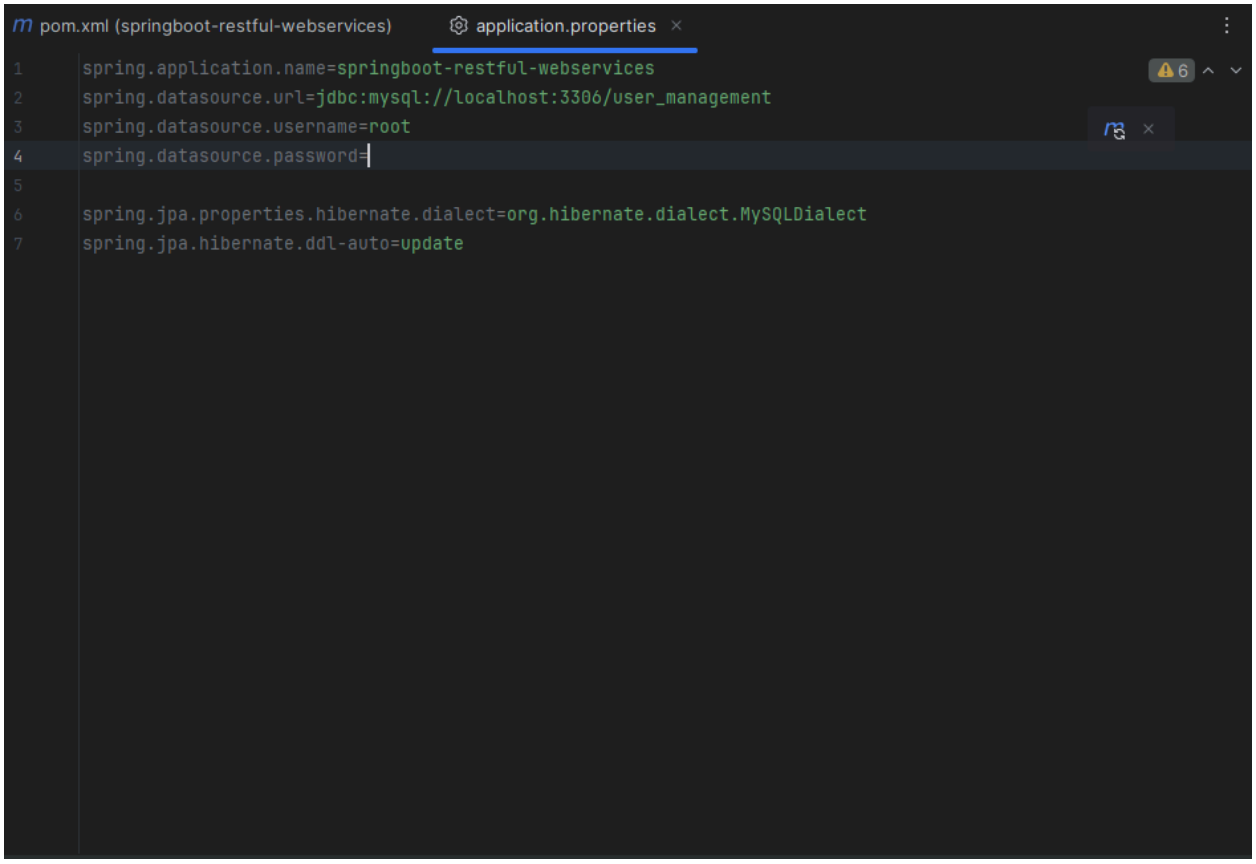
pom.xml (springboot-restful-webservices)

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    <build>
        <plugins>
            <plugin>
                <configuration>
                    </excludes>
                </configuration>
            </plugin>
        </plugins>
    </build>
    <repositories>
        <repository>
            <id>spring-milestones</id>
            <name>Spring Milestones</name>
            <url>https://repo.spring.io/milestone</url>
            <snapshots>
                <enabled>false</enabled>
            </snapshots>
        </repository>
    </repositories>
    <pluginRepositories>
        <pluginRepository>
            <id>spring-milestones</id>
            <name>Spring Milestones</name>
            <url>https://repo.spring.io/milestone</url>
            <snapshots>
                <enabled>false</enabled>
            </snapshots>
```

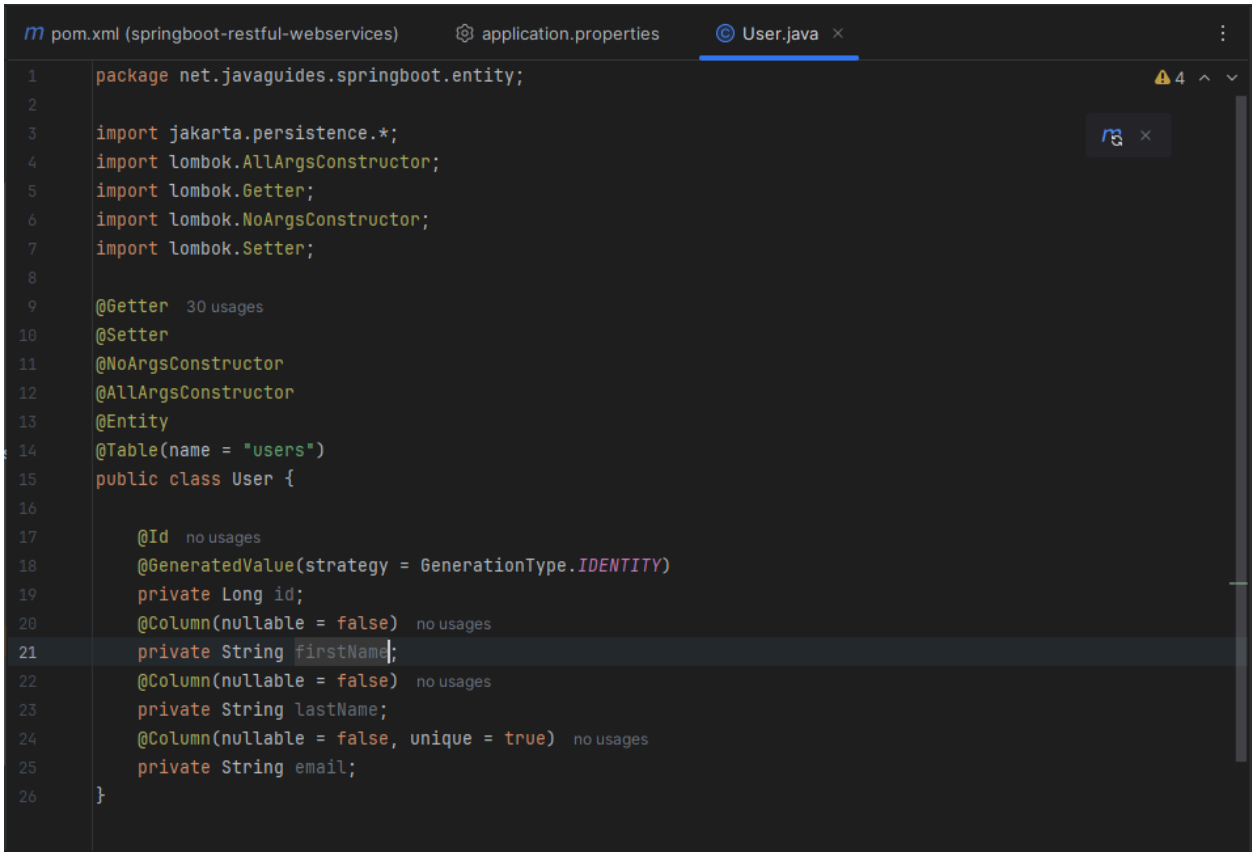
PROJECT STRUCTURE:



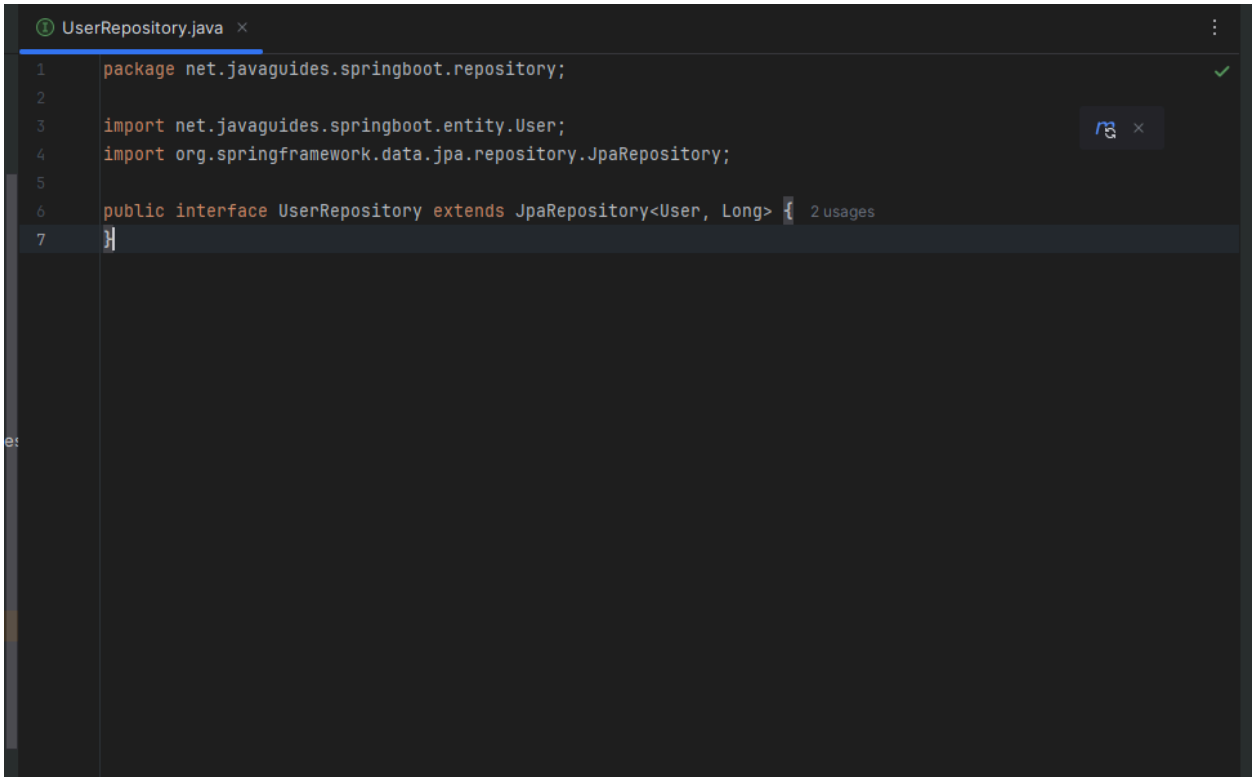
APPLICATION.PROPERTIES:



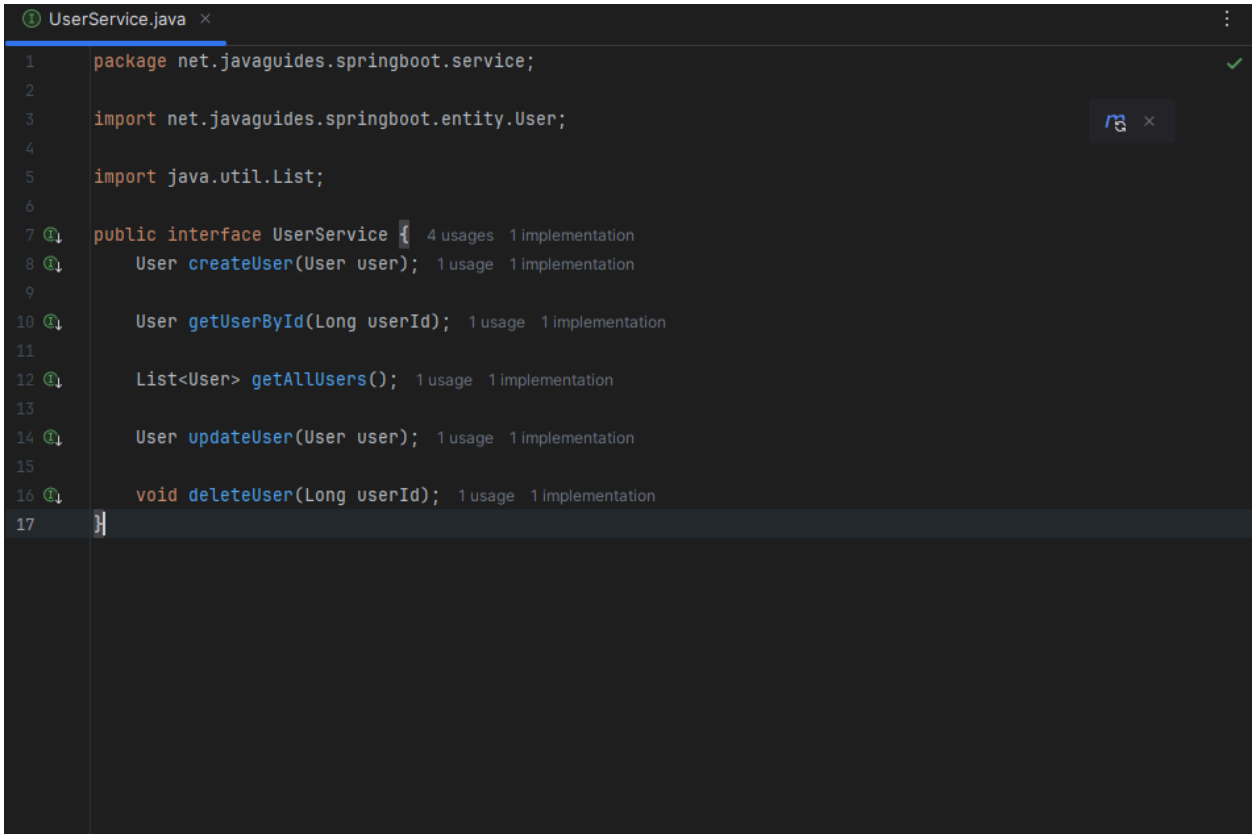
USER.JAVA


pom.xml (springboot-restful-webservices)
application.properties
User.java
4
package net.javaguides.springboot.entity;
import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
@Getter 30 usages
@Setter
@NoArgsConstructor
@AllArgsConstructor
@Entity
@Table(name = "users")
public class User {
@Id no usages
@GeneratedValue(strategy = GenerationType.IDENTITY)
private Long id;
@Column(nullable = false) no usages
private String firstName;
@Column(nullable = false) no usages
private String lastName;
@Column(nullable = false, unique = true) no usages
private String email;
}

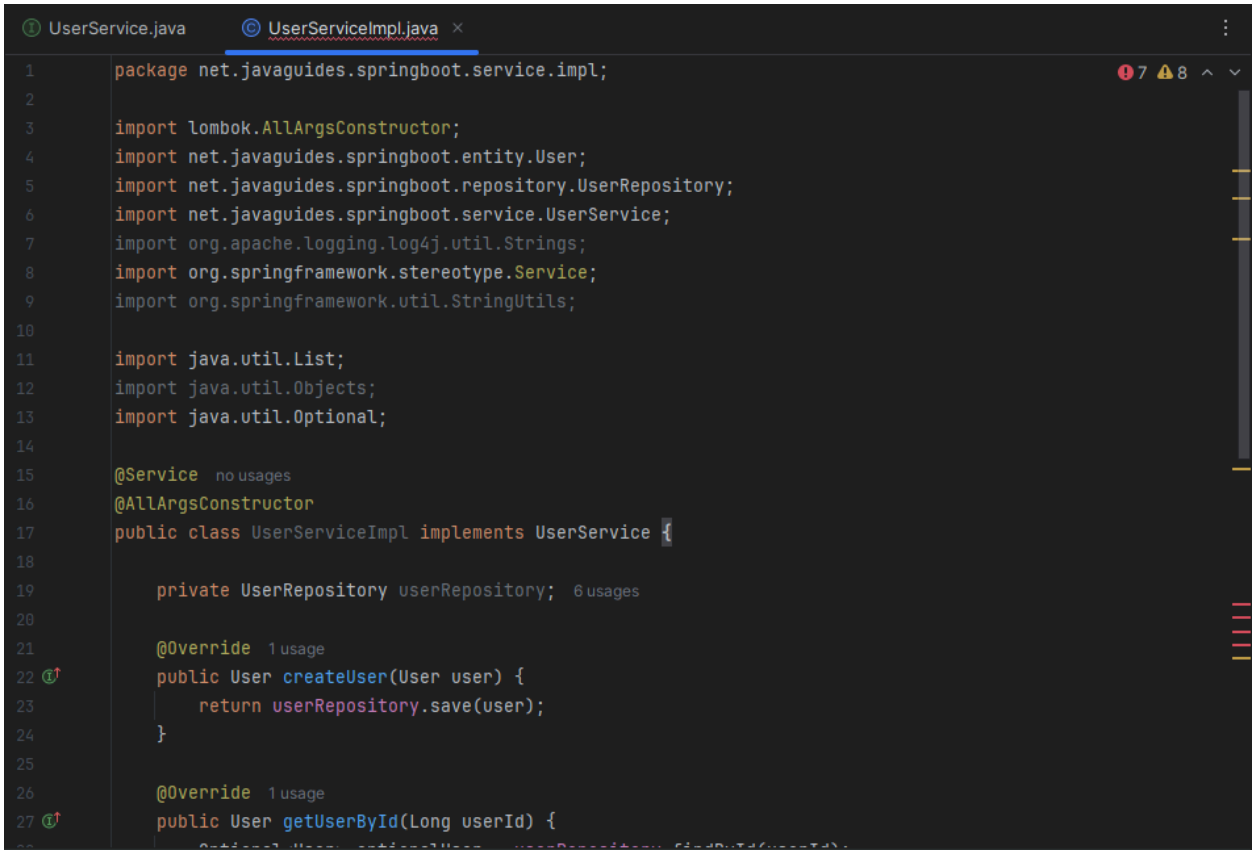
USER REPOSITORY:

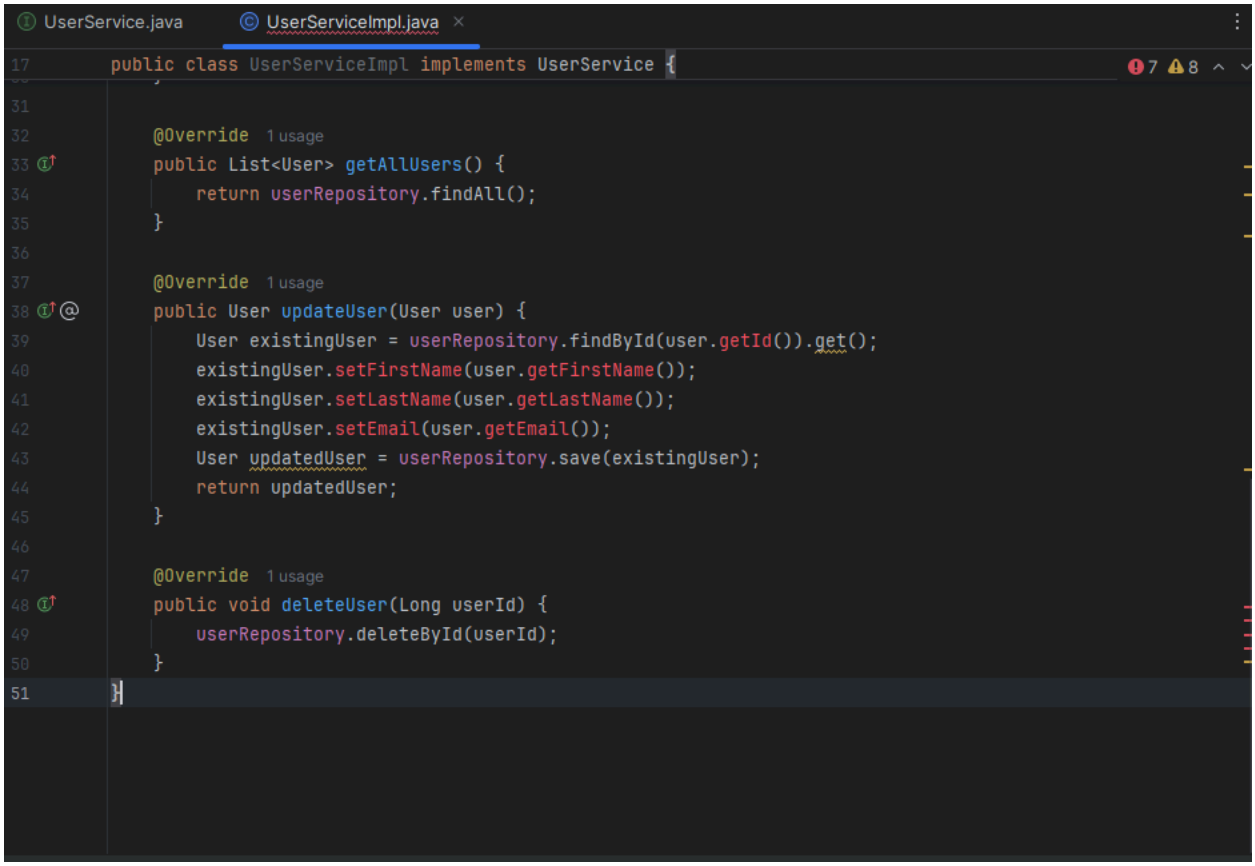


USERSERVICE:

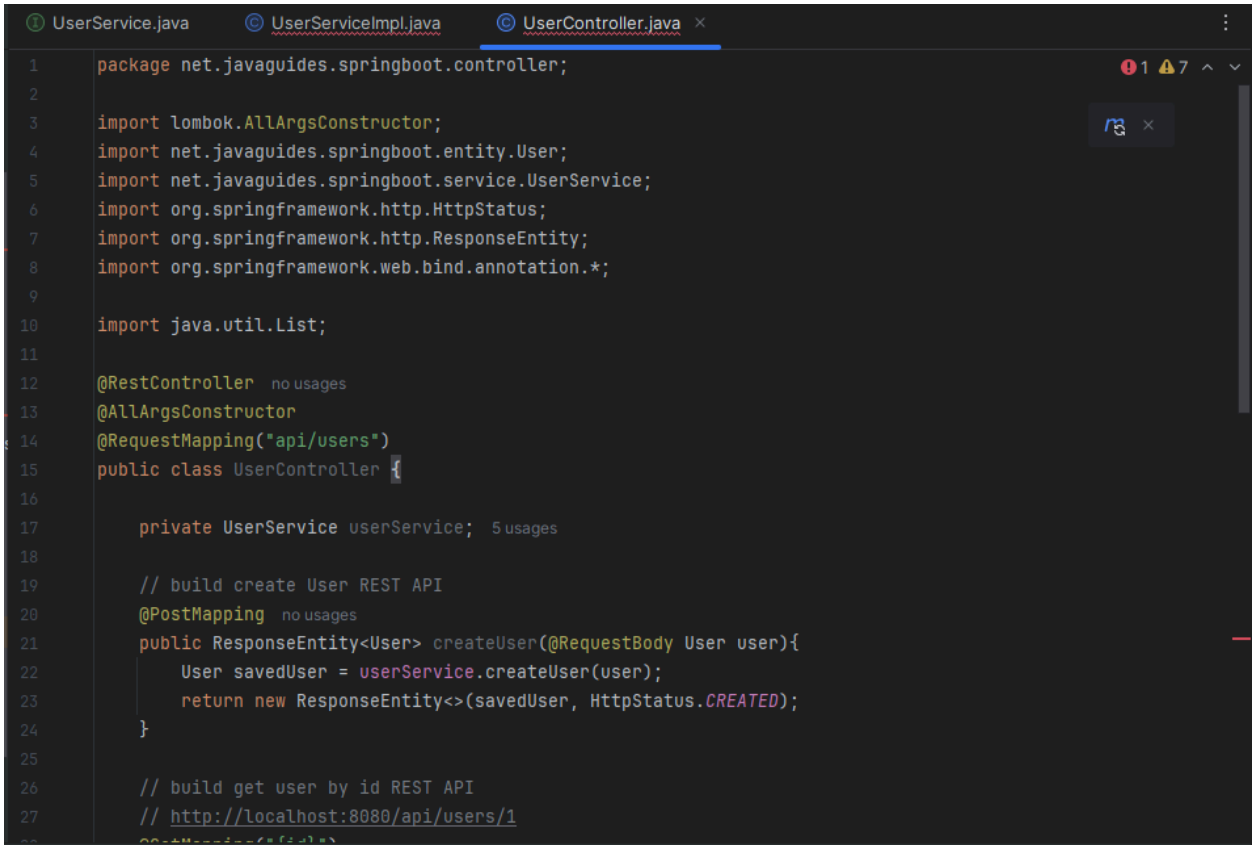


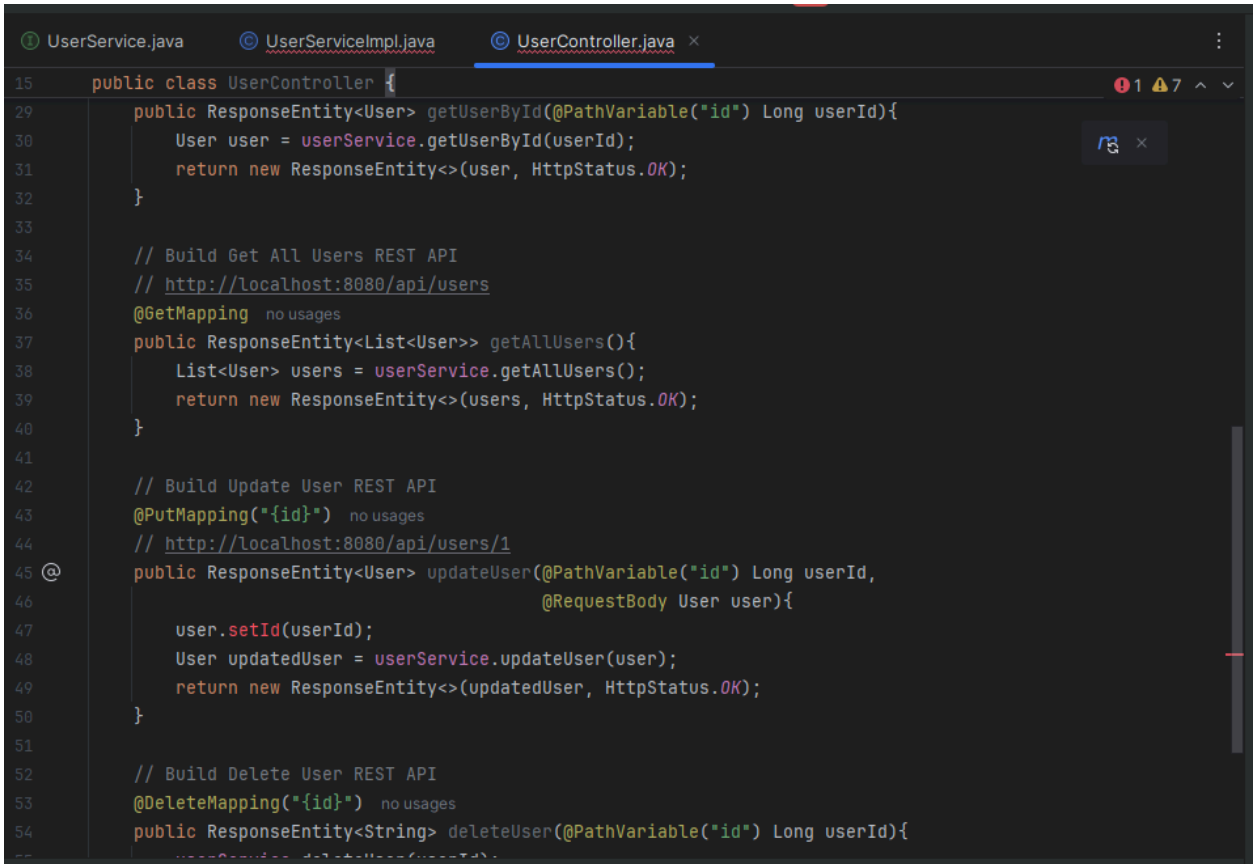
USERSERVICEIMPL:


UserService.java
UserServiceImpl.java
7 8
package net.javaguides.springboot.service.impl;
import lombok.AllArgsConstructor;
import net.javaguides.springboot.entity.User;
import net.javaguides.springboot.repository.UserRepository;
import net.javaguides.springboot.service.UserService;
import org.apache.logging.log4j.util.Strings;
import org.springframework.stereotype.Service;
import org.springframework.util.StringUtils;
import java.util.List;
import java.util.Objects;
import java.util.Optional;
@Service no usages
@AllArgsConstructor
public class UserServiceImpl implements UserService {
private UserRepository userRepository; 6 usages
@Override 1 usage
public User createUser(User user) {
return userRepository.save(user);
}
@Override 1 usage
public User getUserById(Long userId) {


UserService.java
UserServiceImpl.java
public class UserServiceImpl implements UserService {
7 8
@Override 1 usage
public List<User> getAllUsers() {
return userRepository.findAll();
}
@Override 1 usage
public User updateUser(User user) {
User existingUser = userRepository.findById(user.getId()).get();
existingUser.setFirstName(user.getFirstName());
existingUser.setLastName(user.getLastName());
existingUser.setEmail(user.getEmail());
User updatedUser = userRepository.save(existingUser);
return updatedUser;
}
@Override 1 usage
public void deleteUser(Long userId) {
userRepository.deleteById(userId);
}
}

USERCONTROLLER:

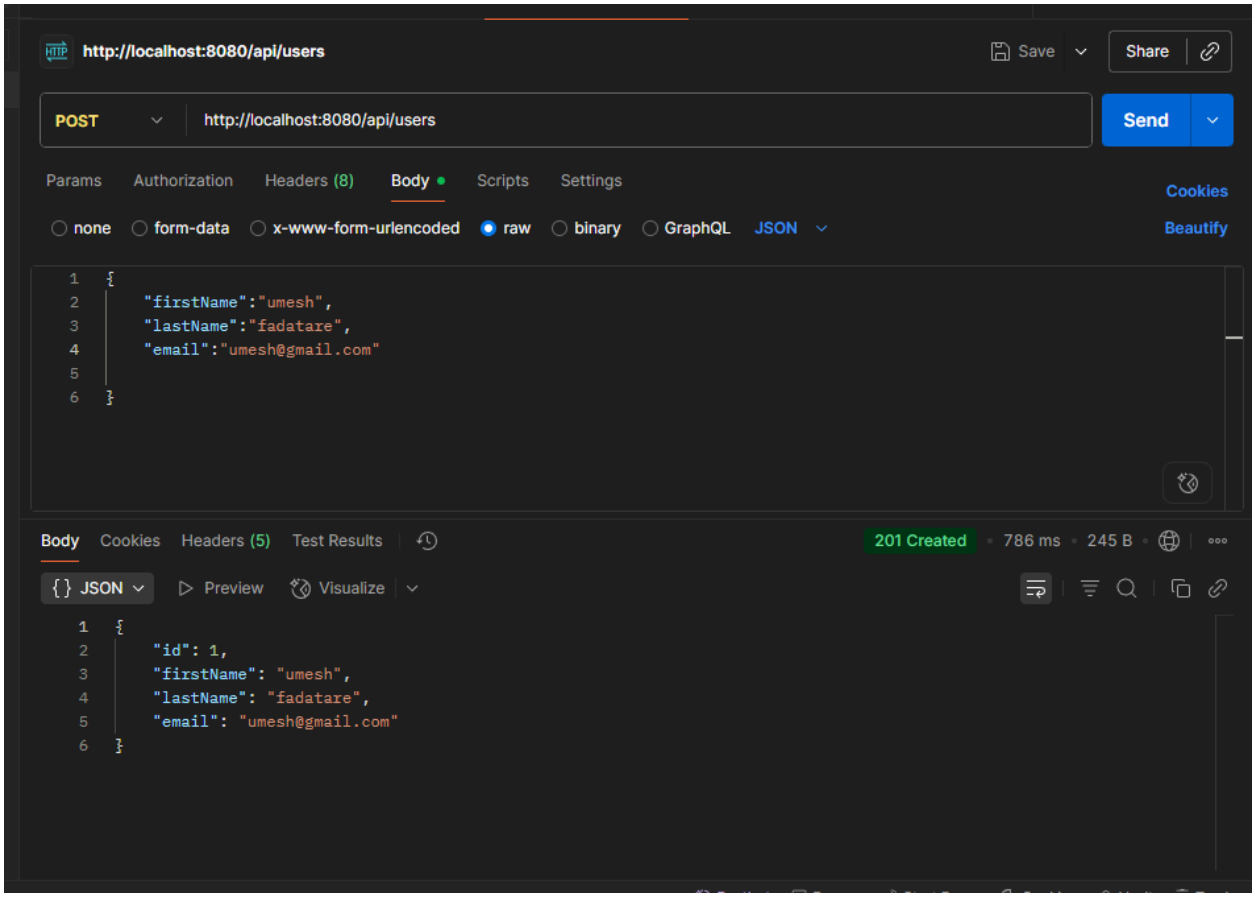



UserService.java
UserServiceImpl.java
UserController.java
public class UserController {
public ResponseEntity<User> getUserById(@PathVariable("id") Long userId){
User user = userService.getUserById(userId);
return new ResponseEntity<>(user, HttpStatus.OK);
}
// Build Get All Users REST API
// http://localhost:8080/api/users
@GetMapping no usages
public ResponseEntity<List<User>> getAllUsers(){
List<User> users = userService.getAllUsers();
return new ResponseEntity<>(users, HttpStatus.OK);
}
// Build Update User REST API
@PutMapping("{id}") no usages
// http://localhost:8080/api/users/1
public ResponseEntity<User> updateUser(@PathVariable("id") Long userId,
@RequestBody User user){
user.setId(userId);
User updatedUser = userService.updateUser(user);
return new ResponseEntity<>(updatedUser, HttpStatus.OK);
}
// Build Delete User REST API
@DeleteMapping("{id}") no usages
public ResponseEntity<String> deleteUser(@PathVariable("id") Long userId){

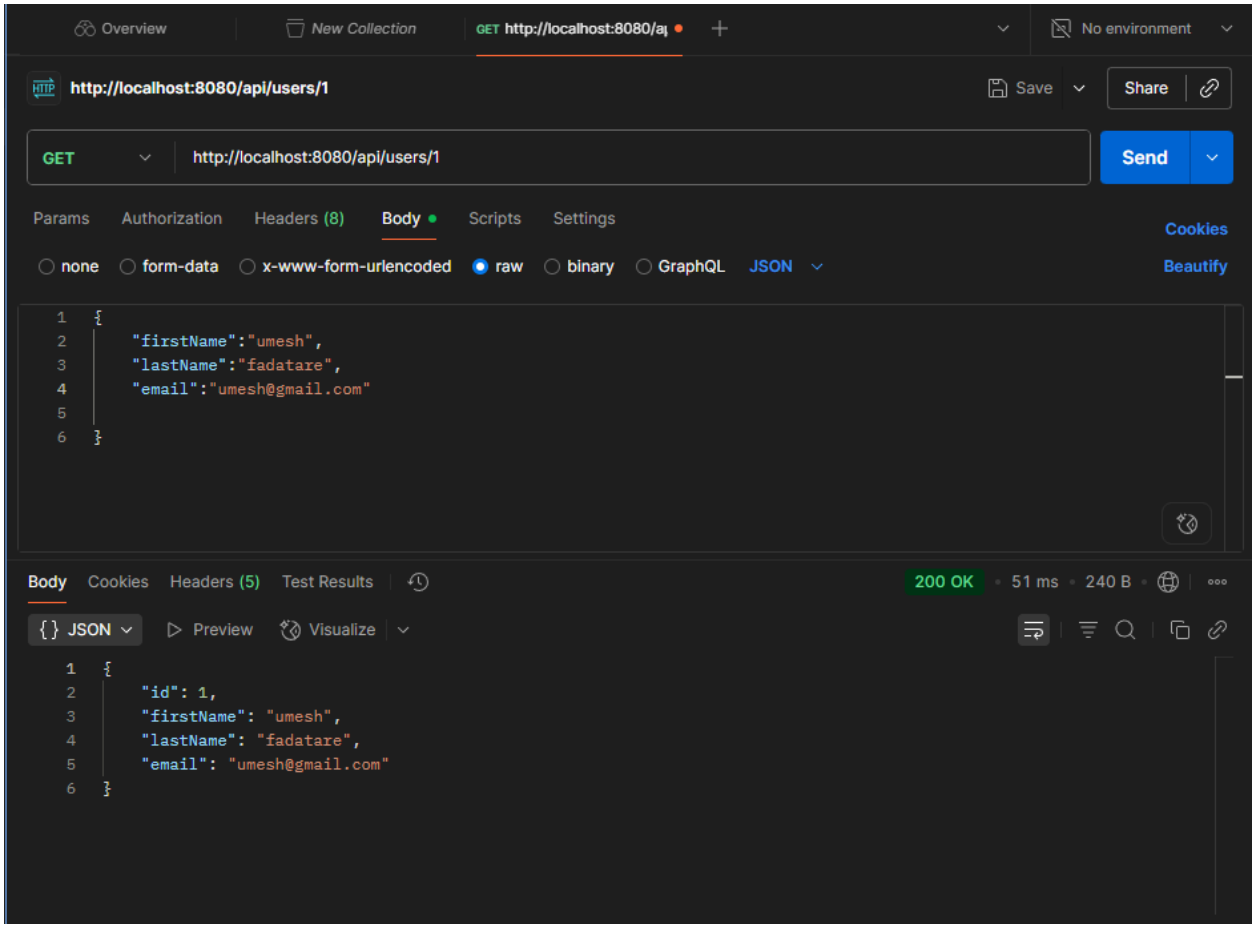

// Build Delete User REST API
@DeleteMapping("{id}") no usages
public ResponseEntity<String> deleteUser(@PathVariable("id") Long userId){
userService.deleteUser(userId);
return new ResponseEntity<>(body: "User successfully deleted!", HttpStatus.OK);
}
}

TESTING

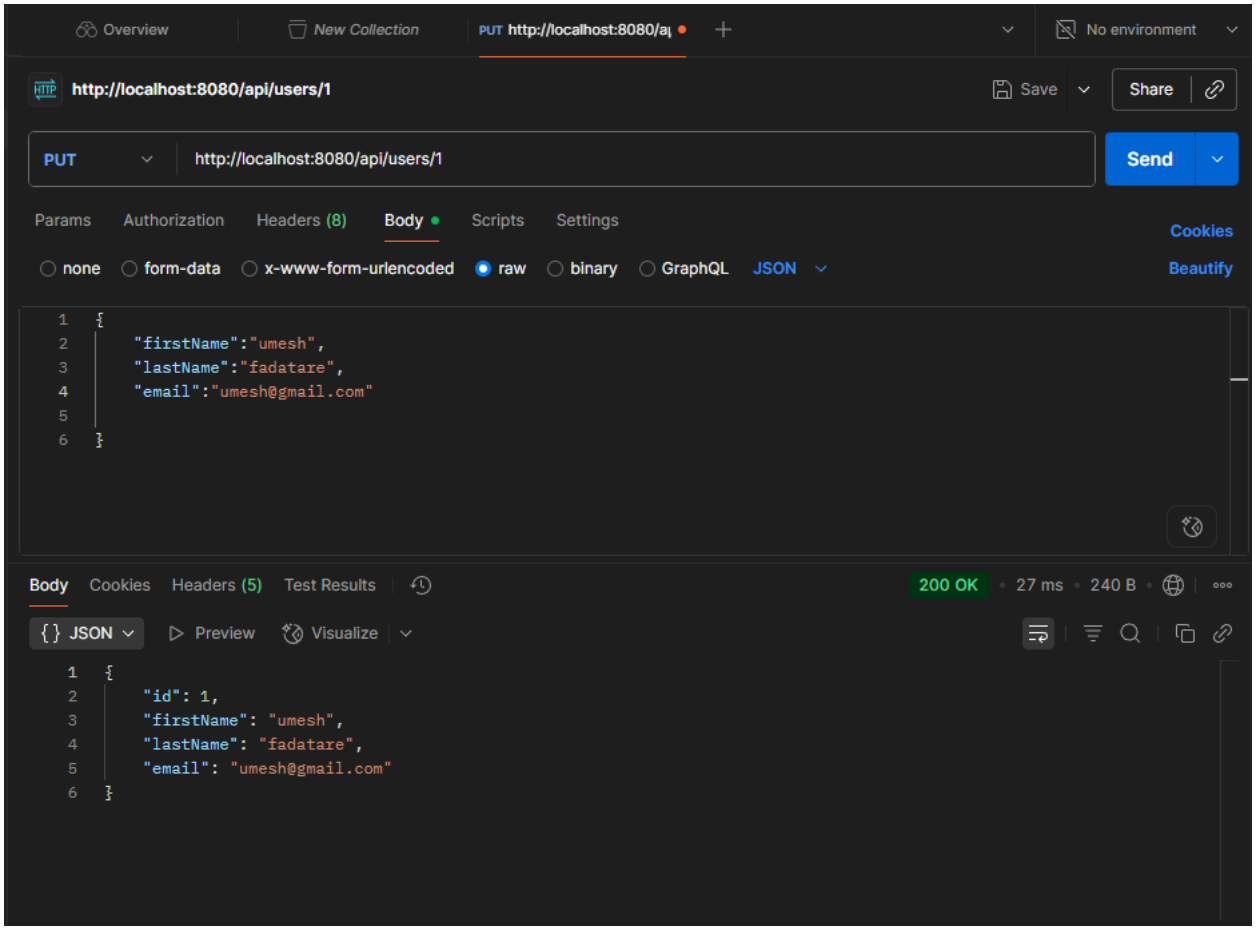
Create User REST API:



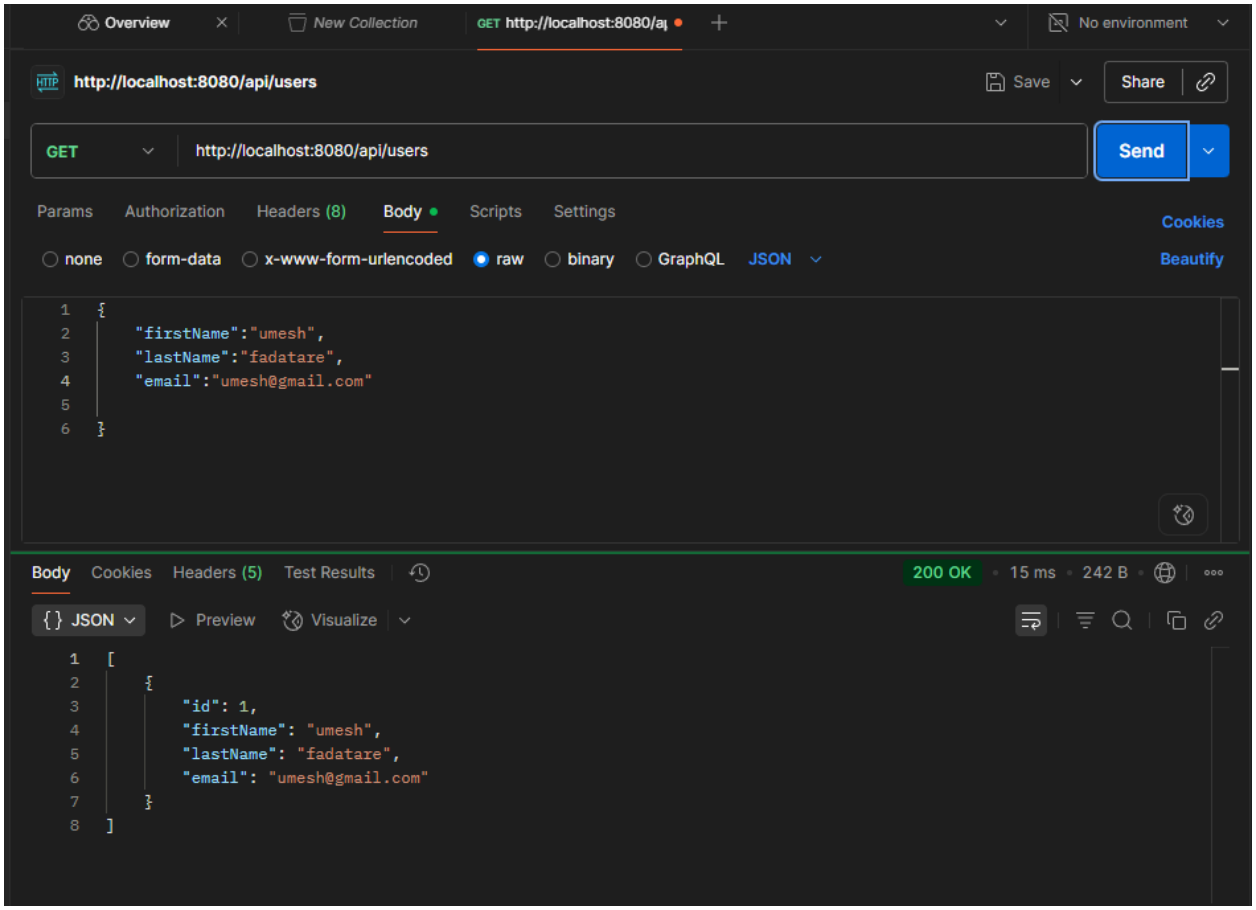
Get Single User REST API:



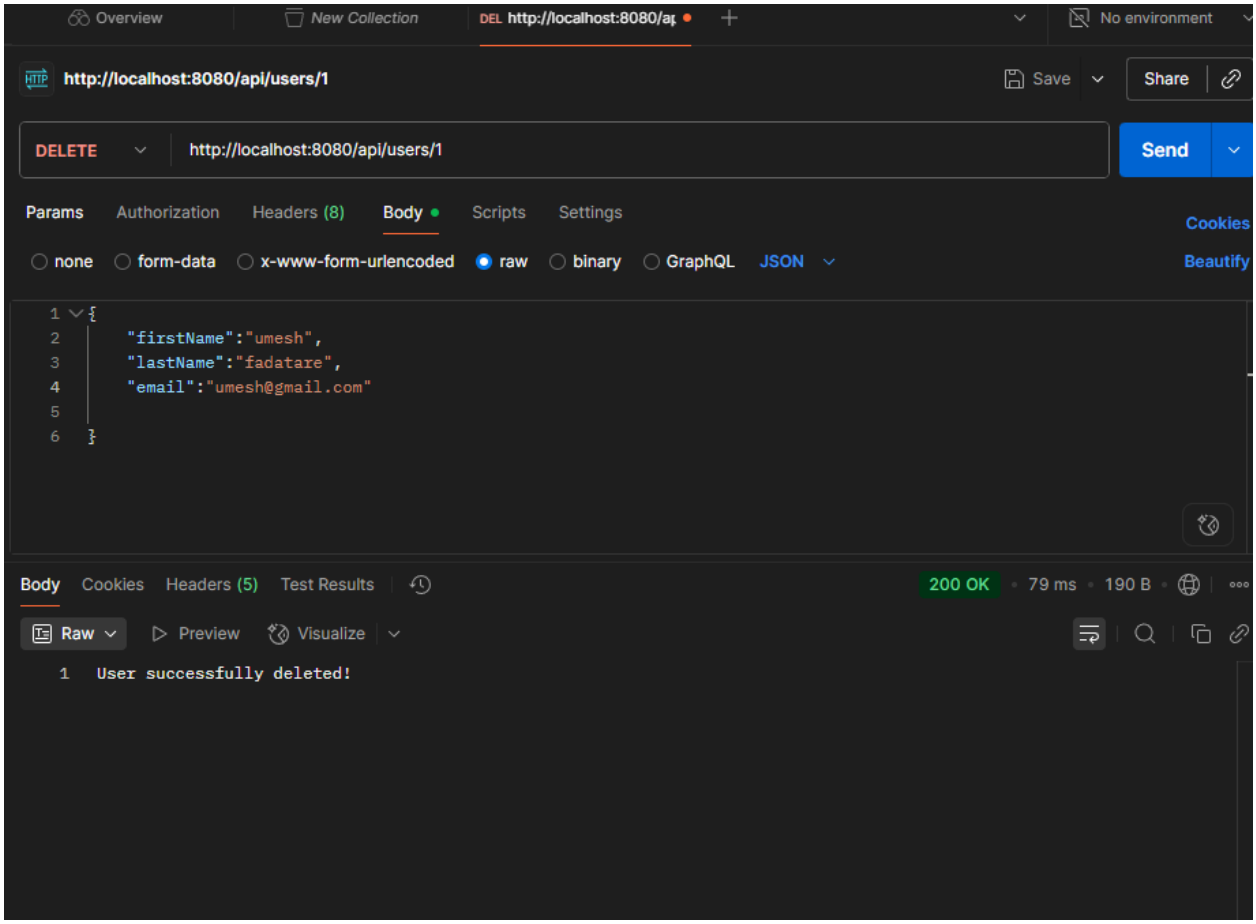
Update User REST API:



Get All Users REST API:



Delete User REST API:


Overview
New Collection
DEL http://localhost:8080/ap
No environment
http://localhost:8080/api/users/1
Save
Share
DELETE
http://localhost:8080/api/users/1
Send
Params
Authorization
Headers (8)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"firstName":"umesh",
"lastName":"fadatare",
"email":"umesh@gmail.com"
}
Body
Cookies
Headers (5)
Test Results
200 OK
79 ms
190 B
Raw
Preview
Visualize
User successfully deleted!