

BARGABINO,MICHAEL JOSH V.

FORTE, TIMOTHY

IV-ACSAD

CODE

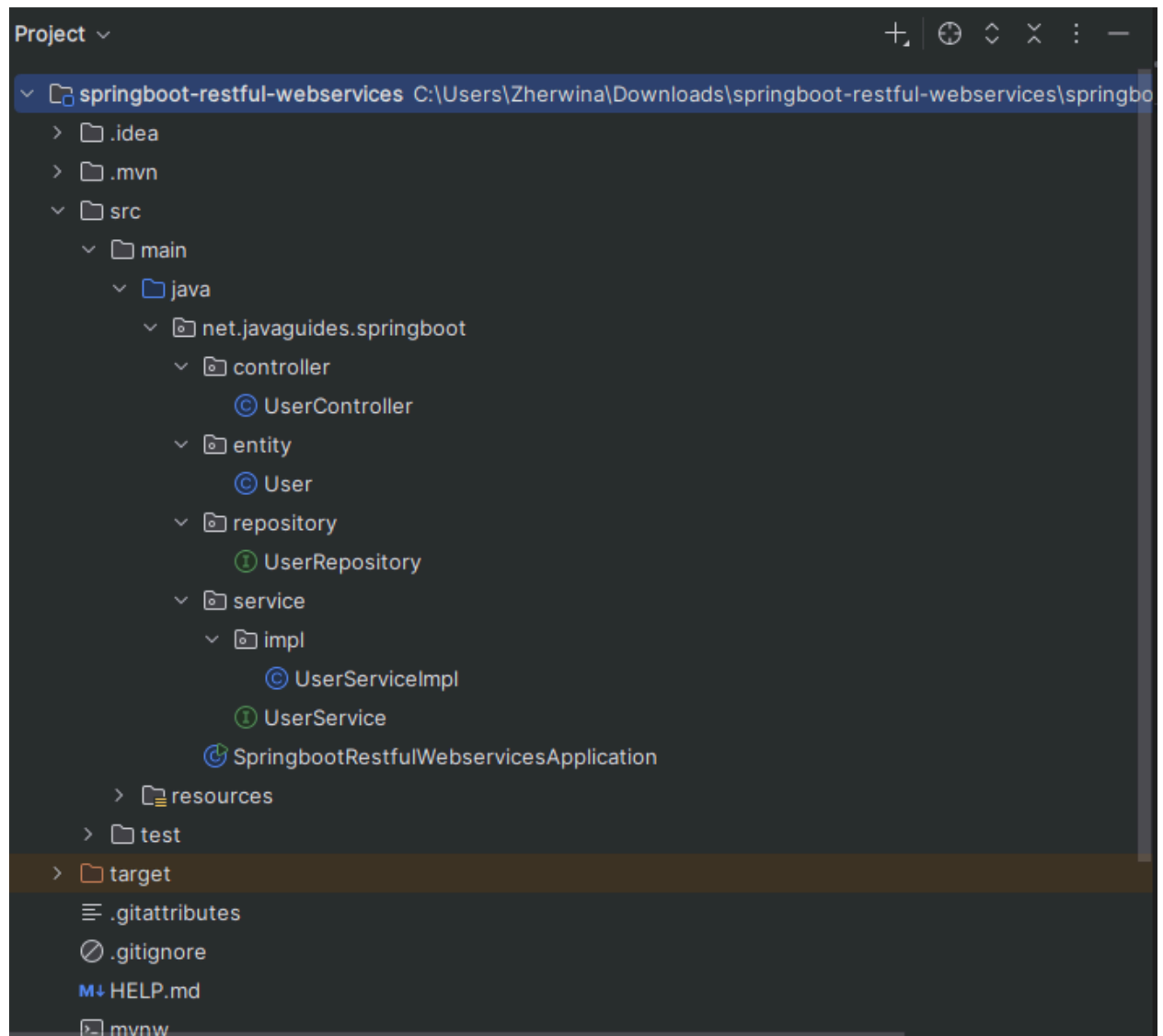
POM.XML

```
m pom.xml (springboot-restful-webservices) x
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
4     <modelVersion>4.0.0</modelVersion>
5     <parent>
6         <groupId>org.springframework.boot</groupId>
7         <artifactId>spring-boot-starter-parent</artifactId>
8         <version>3.0.0-M4</version>
9         <relativePath/> <!-- lookup parent from repository -->
10    </parent>
11    <groupId>net.javaguides</groupId>
12    <artifactId>springboot-restful-webservices</artifactId>
13    <version>0.0.1-SNAPSHOT</version>
14    <name>springboot-restful-webservices</name>
15    <description>Demo project for Spring Boot Restful Webservices</description>
16    <properties>
17        <java.version>17</java.version>
18    </properties>
19    <dependencies>
20        <dependency>
21            <groupId>org.springframework.boot</groupId>
22            <artifactId>spring-boot-starter-data-jpa</artifactId>
23        </dependency>
24        <dependency>
25            <groupId>org.springframework.boot</groupId>
26            <artifactId>spring-boot-starter-web</artifactId>
27        </dependency>
28    </dependencies>
29 </project>
```

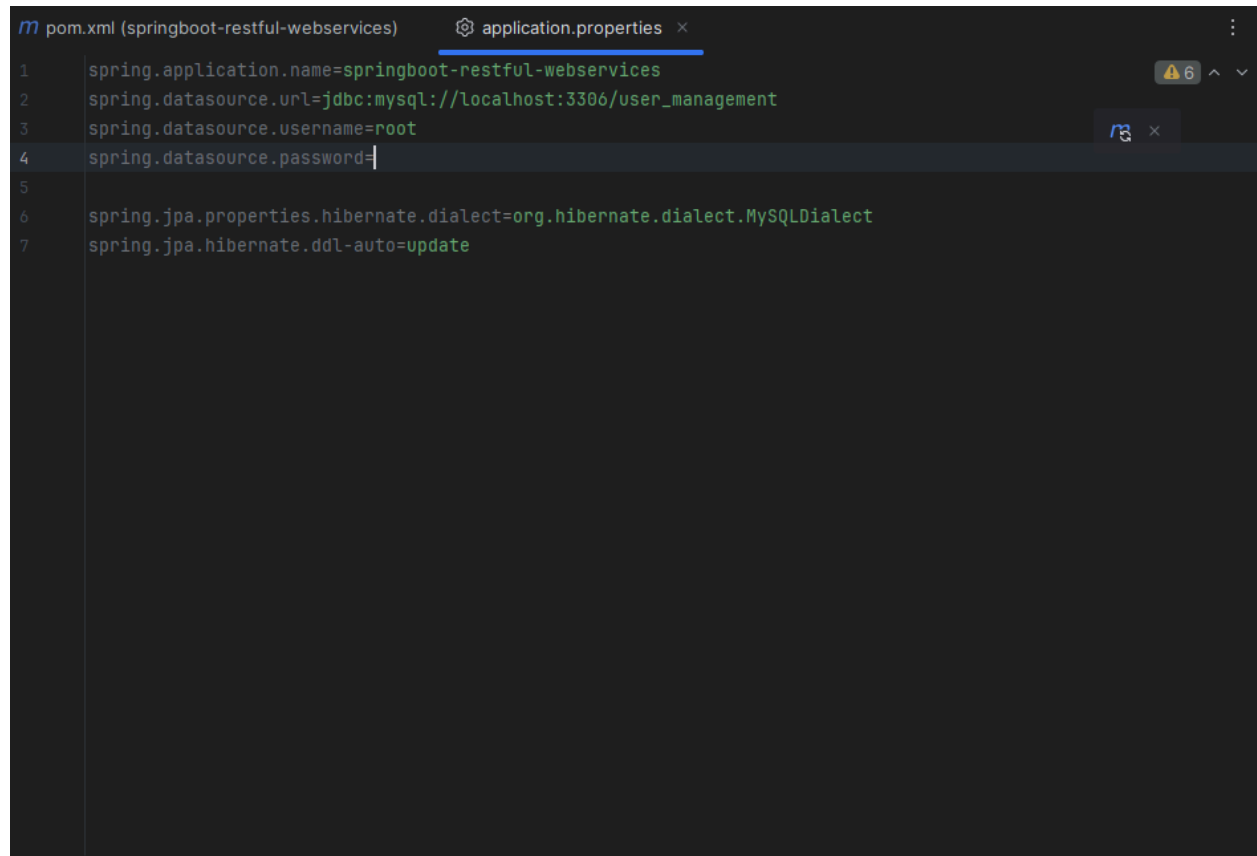
```
m pom.xml (springboot-restful-webservices) x
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ✓
19 <dependencies>
29 <dependency>
31 <artifactId>mysql-connector-java</artifactId>
32 <scope>runtime</scope>
33 </dependency>
34 Ⓢ
35 <dependency>
36 <groupId>org.projectlombok</groupId>
37 <artifactId>lombok</artifactId>
38 <optional>true</optional>
39 Ⓢ
40 <dependency>
41 <groupId>org.springframework.boot</groupId>
42 <artifactId>spring-boot-starter-test</artifactId>
43 <scope>test</scope>
44 </dependency>
45 </dependencies>
46
47 <build>
48 <plugins>
49 <plugin>
50 <groupId>org.springframework.boot</groupId>
51 Ⓢ <artifactId>spring-boot-maven-plugin</artifactId>
52 <configuration>
53 <excludes>
54 <exclude>
55 <groupId>org.projectlombok</groupId>
<artifactId>lombok</artifactId>
```

```
m pom.xml (springboot-restful-webservices) x
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ✓
46 <build>
47 <plugins>
48 <plugin>
51 <configuration>
57 </excludes>
58 </configuration>
59 </plugin>
60 </plugins>
61 </build>
62 <repositories>
63 <repository>
64 <id>spring-milestones</id>
65 <name>Spring Milestones</name>
66 <url>https://repo.spring.io/milestone</url>
67 <snapshots>
68 <enabled>false</enabled>
69 </snapshots>
70 </repository>
71 </repositories>
72 <pluginRepositories>
73 <pluginRepository>
74 <id>spring-milestones</id>
75 <name>Spring Milestones</name>
76 <url>https://repo.spring.io/milestone</url>
77 <snapshots>
78 <enabled>false</enabled>
79 </snapshots>
```

PROJECT STRUCTURE:



APPLICATION.PROPERTIES:

A screenshot of an IDE window titled 'pom.xml (springboot-restful-webservices)' with a sub-tab 'application.properties'. The code is as follows:

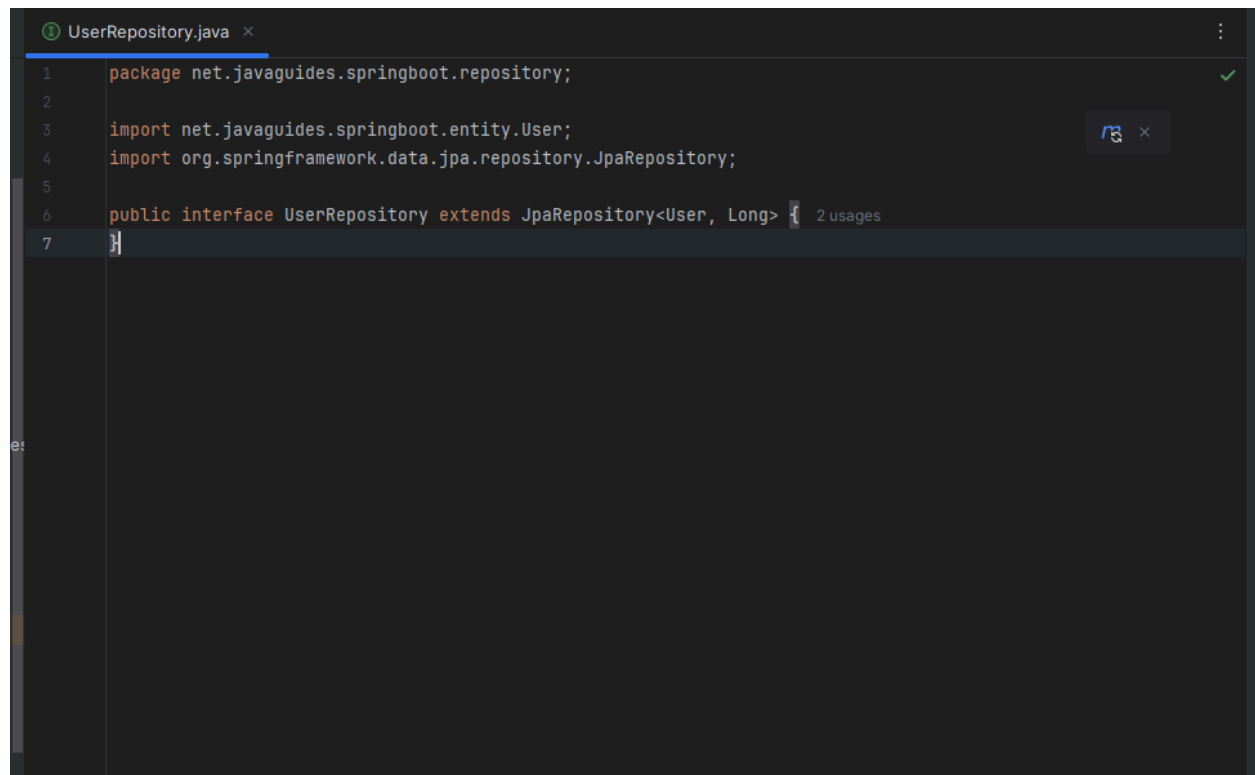
```
1 spring.application.name=springboot-restful-webservices
2 spring.datasource.url=jdbc:mysql://localhost:3306/user_management
3 spring.datasource.username=root
4 spring.datasource.password=
5
6 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect
7 spring.jpa.hibernate.ddl-auto=update
```

The editor has a dark theme. Line 4 is currently selected. There are some UI elements on the right side, including a warning icon with the number 6 and a small icon with an 'x'.

USER.JAVA

```
m pom.xml (springboot-restful-webservices) application.properties User.java x
1 package net.javaguides.springboot.entity;
2
3 import jakarta.persistence.*;
4 import lombok.AllArgsConstructor;
5 import lombok.Getter;
6 import lombok.NoArgsConstructor;
7 import lombok.Setter;
8
9 @Getter 30 usages
10 @Setter
11 @NoArgsConstructor
12 @AllArgsConstructor
13 @Entity
14 @Table(name = "users")
15 public class User {
16
17     @Id no usages
18     @GeneratedValue(strategy = GenerationType.IDENTITY)
19     private Long id;
20     @Column(nullable = false) no usages
21     private String firstName;
22     @Column(nullable = false) no usages
23     private String lastName;
24     @Column(nullable = false, unique = true) no usages
25     private String email;
26 }
```

USER REPOSITORY:



```
1 package net.javaguides.springboot.repository;
2
3 import net.javaguides.springboot.entity.User;
4 import org.springframework.data.jpa.repository.JpaRepository;
5
6 public interface UserRepository extends JpaRepository<User, Long> {
7 }
```

USERSERVICE:

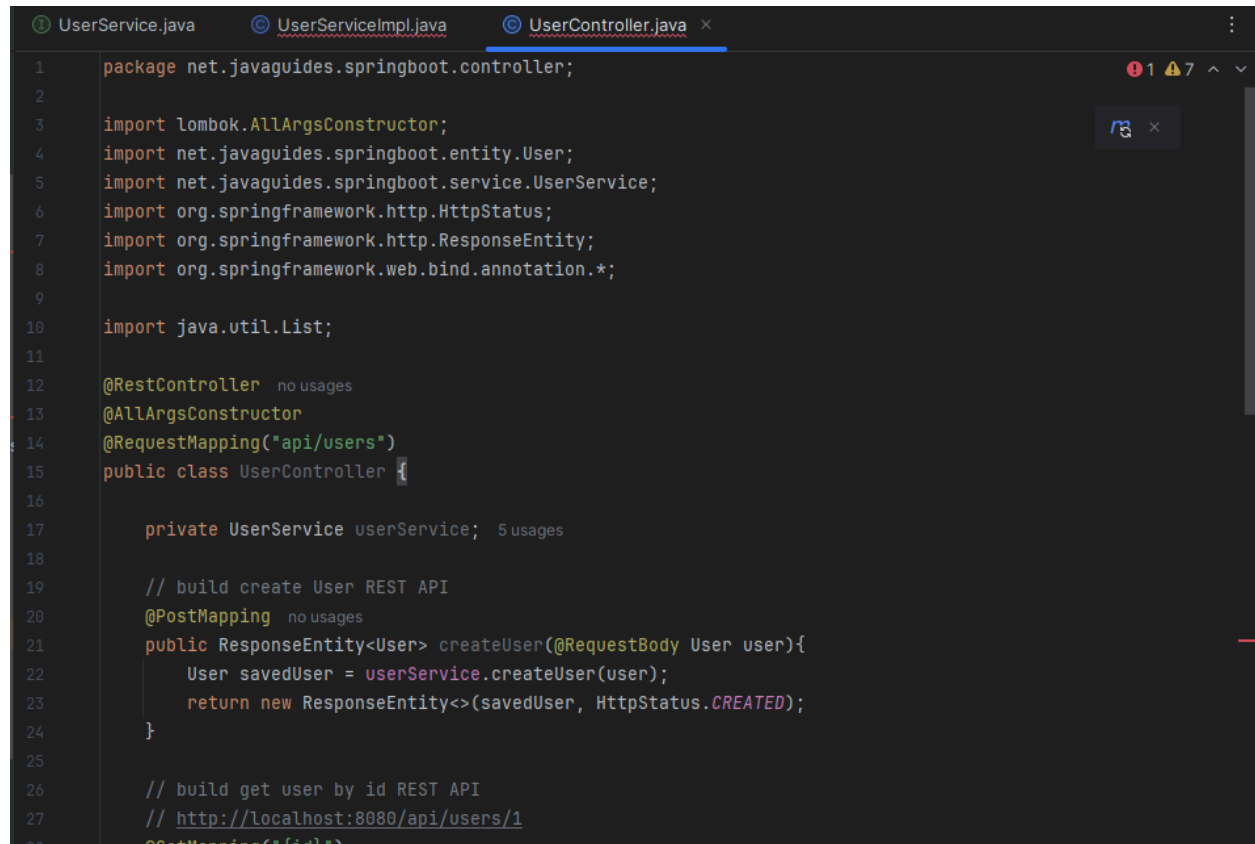
```
UserService.java x
1 package net.javaguides.springboot.service;
2
3 import net.javaguides.springboot.entity.User;
4
5 import java.util.List;
6
7 public interface UserService { 4 usages 1 implementation
8     User createUser(User user); 1 usage 1 implementation
9
10    User getUserById(Long userId); 1 usage 1 implementation
11
12    List<User> getAllUsers(); 1 usage 1 implementation
13
14    User updateUser(User user); 1 usage 1 implementation
15
16    void deleteUser(Long userId); 1 usage 1 implementation
17 }
```

USERSERVICEIMPL:

```
UserService.java  UserServiceImpl.java x
1 package net.javaguides.springboot.service.impl;
2
3 import lombok.AllArgsConstructor;
4 import net.javaguides.springboot.entity.User;
5 import net.javaguides.springboot.repository.UserRepository;
6 import net.javaguides.springboot.service.UserService;
7 import org.apache.logging.log4j.util.Strings;
8 import org.springframework.stereotype.Service;
9 import org.springframework.util.StringUtils;
10
11 import java.util.List;
12 import java.util.Objects;
13 import java.util.Optional;
14
15 @Service no usages
16 @AllArgsConstructor
17 public class UserServiceImpl implements UserService {
18
19     private UserRepository userRepository; 6 usages
20
21     @Override 1 usage
22     public User createUser(User user) {
23         return userRepository.save(user);
24     }
25
26     @Override 1 usage
27     public User getUserById(Long userId) {
28         Optional<User> optionalUser = userRepository.findById(userId);
29     }
30 }
```

```
UserService.java  UserServiceImpl.java x
17 public class UserServiceImpl implements UserService {
31
32     @Override 1 usage
33     public List<User> getAllUsers() {
34         return userRepository.findAll();
35     }
36
37     @Override 1 usage
38     public User updateUser(User user) {
39         User existingUser = userRepository.findById(user.getId()).get();
40         existingUser.setFirstName(user.getFirstName());
41         existingUser.setLastName(user.getLastName());
42         existingUser.setEmail(user.getEmail());
43         User updatedUser = userRepository.save(existingUser);
44         return updatedUser;
45     }
46
47     @Override 1 usage
48     public void deleteUser(Long userId) {
49         userRepository.deleteById(userId);
50     }
51 }
```


USERCONTROLLER:



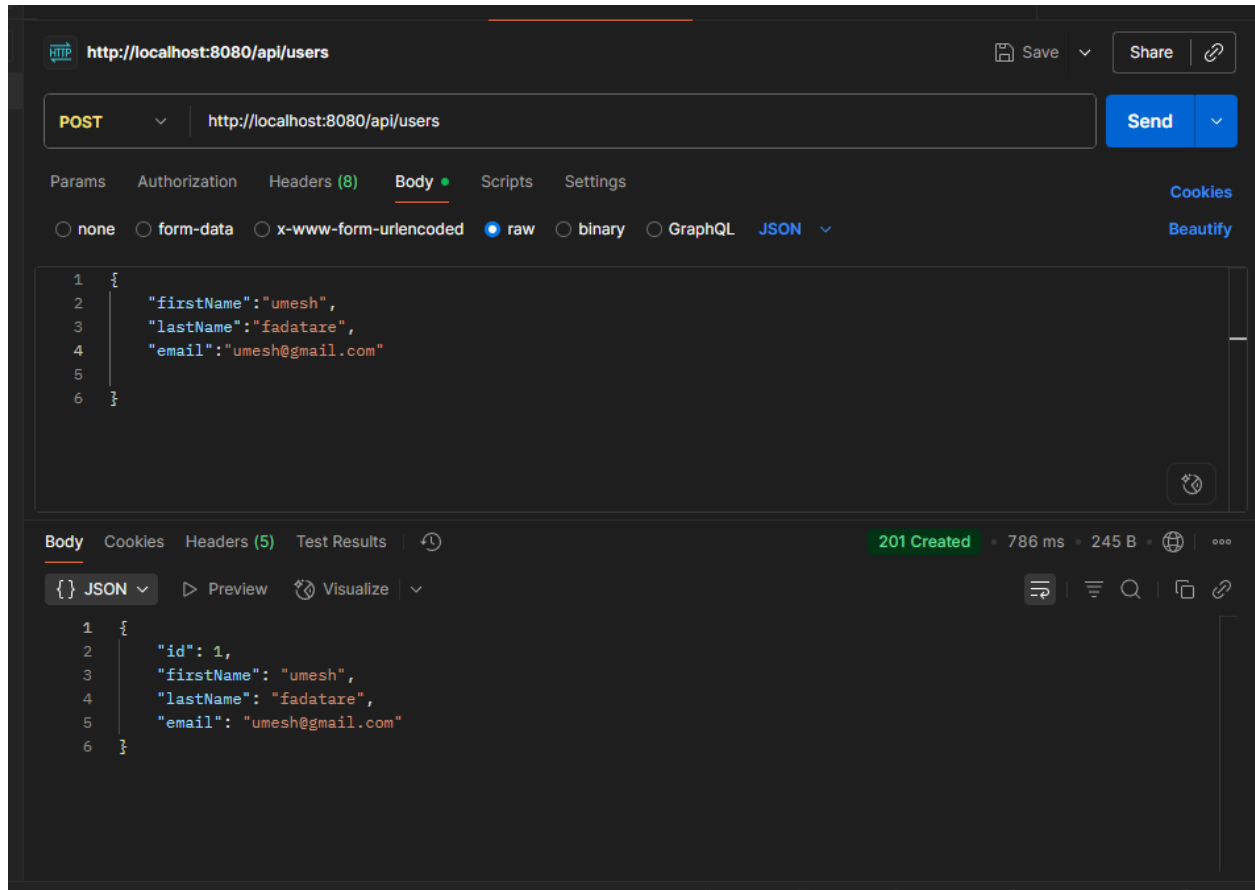
```
1 package net.javaguides.springboot.controller;
2
3 import lombok.AllArgsConstructor;
4 import net.javaguides.springboot.entity.User;
5 import net.javaguides.springboot.service.UserService;
6 import org.springframework.http.HttpStatus;
7 import org.springframework.http.ResponseEntity;
8 import org.springframework.web.bind.annotation.*;
9
10 import java.util.List;
11
12 @RestController no usages
13 @AllArgsConstructor
14 @RequestMapping("api/users")
15 public class UserController {
16
17     private UserService userService; 5 usages
18
19     // build create User REST API
20     @PostMapping no usages
21     public ResponseEntity<User> createUser(@RequestBody User user){
22         User savedUser = userService.createUser(user);
23         return new ResponseEntity<>(savedUser, HttpStatus.CREATED);
24     }
25
26     // build get user by id REST API
27     // http://localhost:8080/api/users/1
28     @GetMapping("/{id}")
```

```
UserService.java  UserServiceImpl.java  UserController.java x
15  public class UserController {
29      public ResponseEntity<User> getUserById(@PathVariable("id") Long userId){
30          User user = userService.getUserById(userId);
31          return new ResponseEntity<>(user, HttpStatus.OK);
32      }
33
34      // Build Get All Users REST API
35      // http://localhost:8080/api/users
36      @GetMapping no usages
37      public ResponseEntity<List<User>> getAllUsers(){
38          List<User> users = userService.getAllUsers();
39          return new ResponseEntity<>(users, HttpStatus.OK);
40      }
41
42      // Build Update User REST API
43      @PutMapping("/{id}") no usages
44      // http://localhost:8080/api/users/1
45      @ public ResponseEntity<User> updateUser(@PathVariable("id") Long userId,
46                                              @RequestBody User user){
47          user.setId(userId);
48          User updatedUser = userService.updateUser(user);
49          return new ResponseEntity<>(updatedUser, HttpStatus.OK);
50      }
51
52      // Build Delete User REST API
53      @DeleteMapping("/{id}") no usages
54      public ResponseEntity<String> deleteUser(@PathVariable("id") Long userId){
55          userService.deleteUser(userId);
56      }
57  }
```

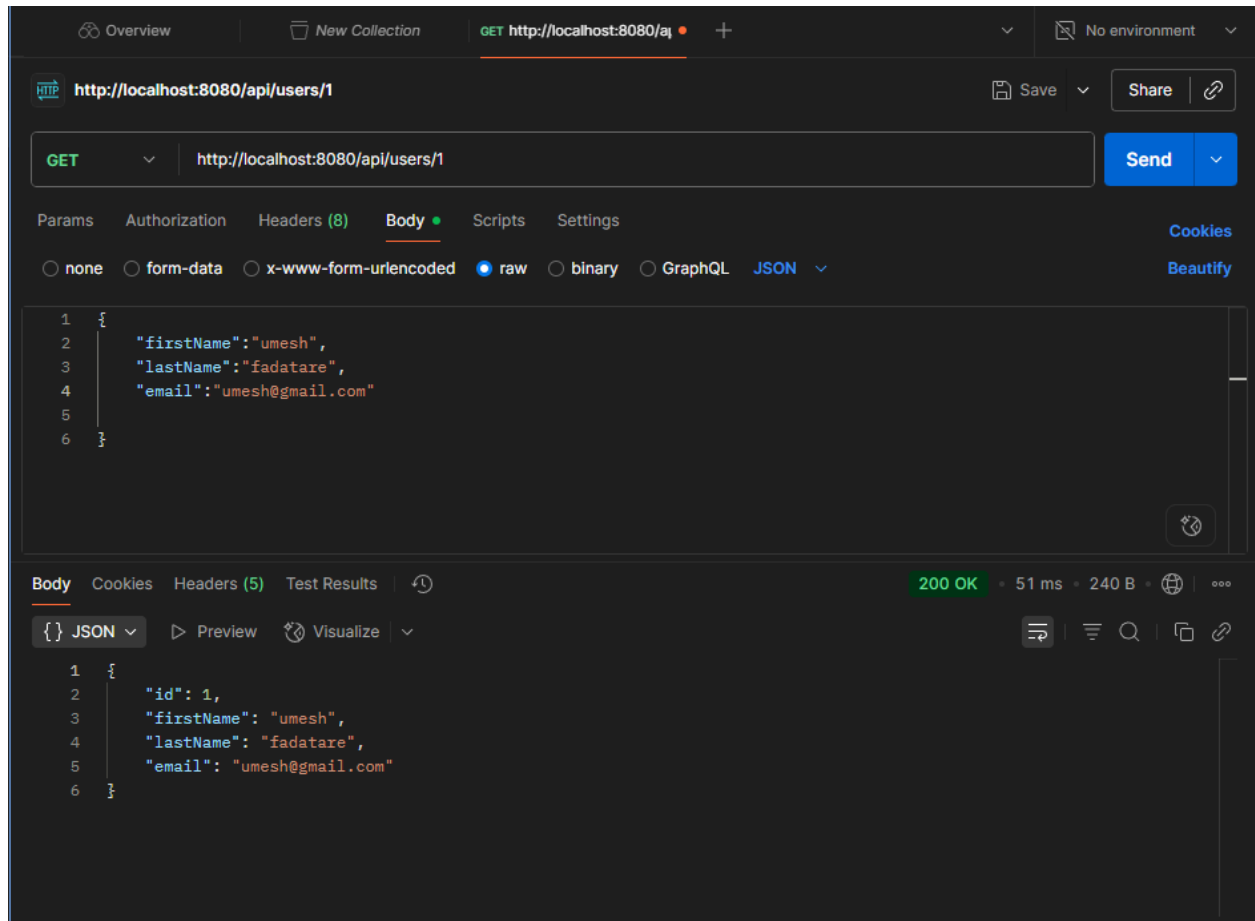
```
// Build Delete User REST API
@DeleteMapping("/{id}") no usages
public ResponseEntity<String> deleteUser(@PathVariable("id") Long userId){
    userService.deleteUser(userId);
    return new ResponseEntity<>({ body: "User successfully deleted!", HttpStatus.OK);
}
}
```

TESTING

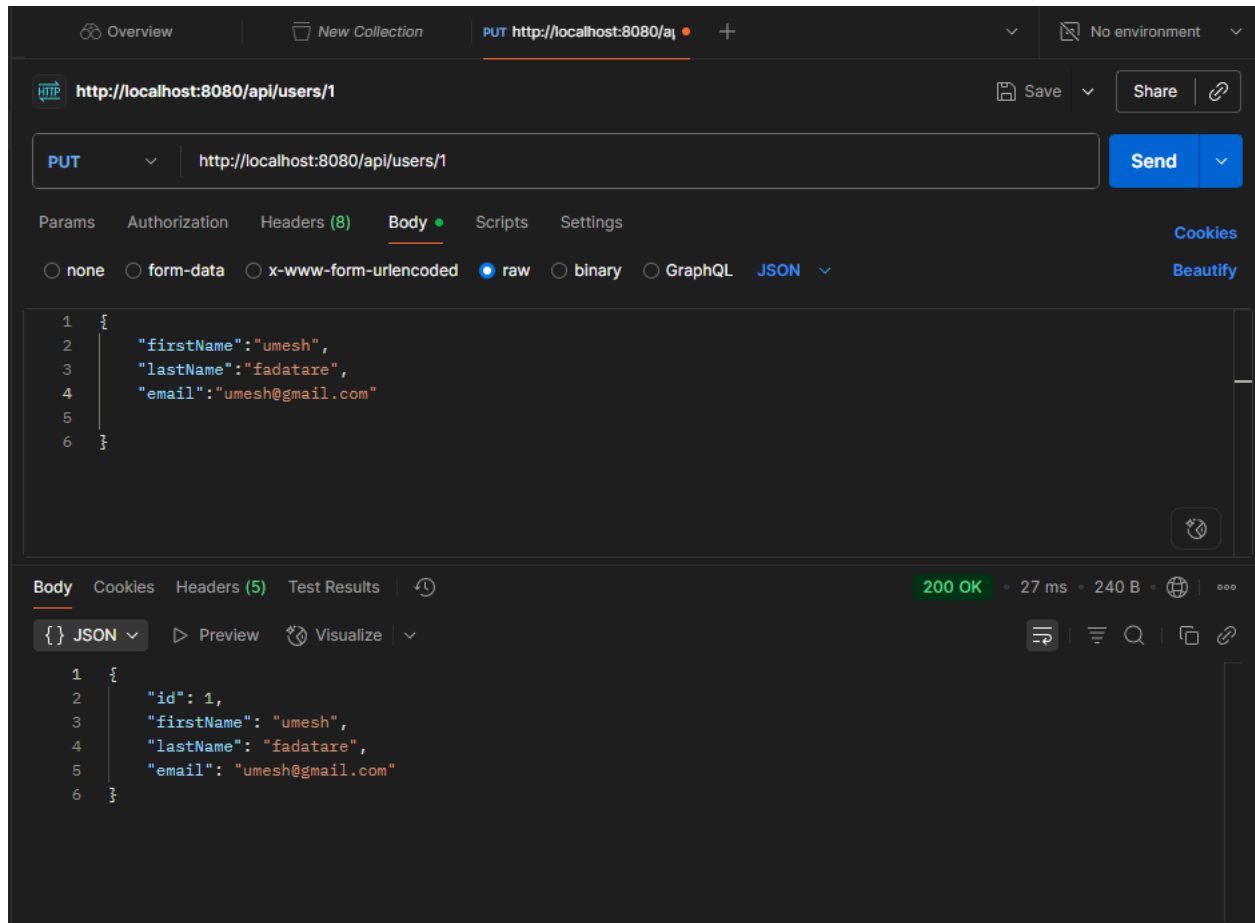
Create User REST API:



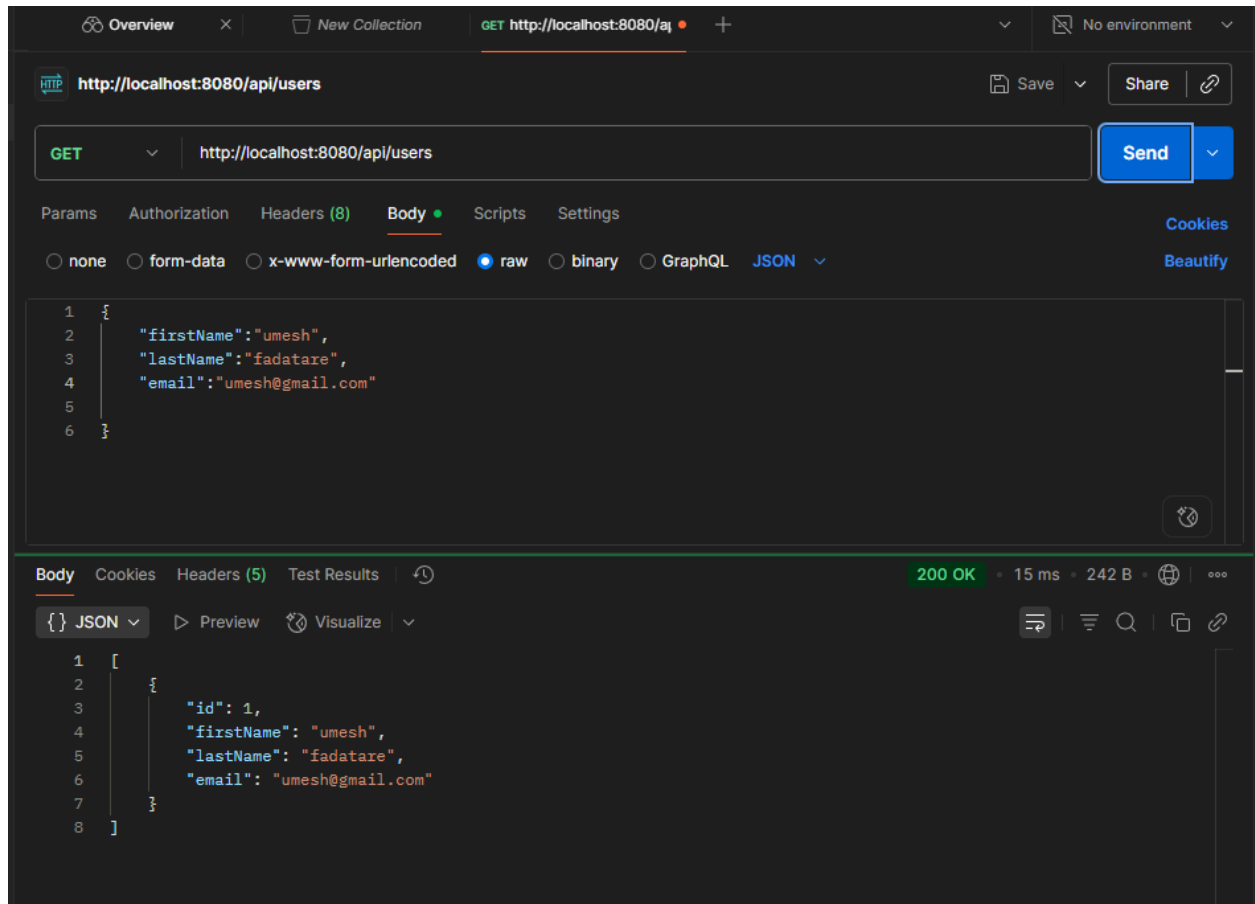
Get Single User REST API:



Update User REST API:



Get All Users REST API:



Delete User REST API:

Overview

New Collection

DEL http://localhost:8080/api/users/1

No environment

HTTP

http://localhost:8080/api/users/1

Save

Share

DELETE

http://localhost:8080/api/users/1

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

Cookies

Beautify

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

☒ JSON

```
1 {
2   "firstName": "umesh",
3   "lastName": "fadatare",
4   "email": "umesh@gmail.com"
5 }
6 }
```

Body

Cookies

Headers (5)

Test Results

200 OK

79 ms

190 B

Raw

Preview

Visualize

1 User successfully deleted!