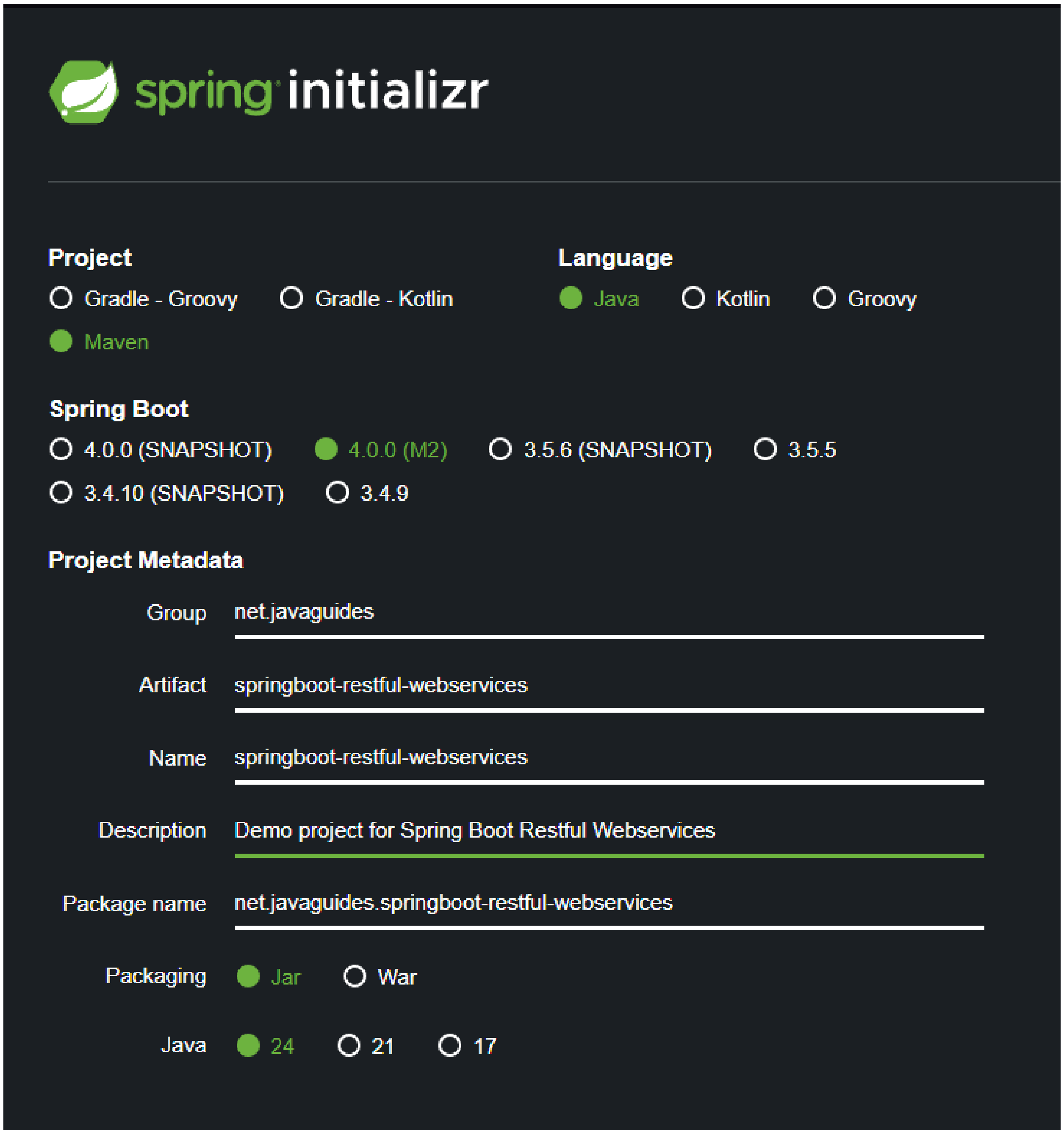
CABUS, CLEMENT HAROLD MIGUEL

IV-ACSAD

CODE:

Springboot Initializer:



Project

○ Gradle - Groovy ○ Gradle - Kotlin [✓ Maven]

Language

[✓ Java] ○ Kotlin ○ Groovy

Spring Boot

○ 4.0.0 (SNAPSHOT) [✓ 4.0.0 (M2)] ○ 3.5.6 (SNAPSHOT) ○ 3.5.5 ○ 3.4.10 (SNAPSHOT) ○ 3.4.9

Project Metadata

Group	net.javaguides
Artifact	springboot-restful-webservices
Name	springboot-restful-webservices
Description	Demo project for Spring Boot Restful Webservices
Package name	net.javaguides.springboot-restful-webservices
Packaging	[✓ Jar] ○ War
Java	[✓ 24] ○ 21 ○ 17

- Pom.xml

```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>4.0.0-M2</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>
    <url/>
    <licenses>
       <license/>
    </licenses>
    <developers>
       <developer/>
    </developers>
    <scm>
       <connection/>
       <developerConnection/>
       <tag/>
       <url/>
    </scm>
    <properties>
       <java.version>24</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>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.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <configuration>
                    <annotationProcessorPaths>
                        <path>
                            <groupId>org.projectlombok</groupId>
                            <artifactId>lombok</artifactId>
                        </path>
                    </annotationProcessorPaths>
                </configuration>
            </plugin>
            <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>
```

- UserController.java

```
package net.javaguides.springboot_restful_webservices.controller;

import lombok.AllArgsConstructor;
import net.javaguides.springboot_restful_webservices.entity.User;
import
net.javaguides.springboot_restful_webservices.service.UserService;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@AllArgsConstructor
@RequestMapping("api/users")
public class UserController {
    private UserService userService;

    // build create User REST API
    @PostMapping
    public ResponseEntity<User> createUser(@RequestBody User user){
        User savedUser = userService.createUser(user);
        return new ResponseEntity<>(savedUser, HttpStatus.CREATED);
    }

    // build get user by id REST API
    // http://localhost:8080/api/users/1
    @GetMapping("{id}")
    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
    public ResponseEntity<List<User>> getAllUsers(){
        List<User> users = userService.getAllUsers();
        return new ResponseEntity<>(users, HttpStatus.OK);
    }

    // Build Update User REST API
    @PutMapping("{id}")
    // 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}")
    public ResponseEntity<String> deleteUser(@PathVariable("id") Long
userId){
        userService.deleteUser(userId);
        return new ResponseEntity<>("User successfully deleted!",
HttpStatus.OK);
    }
}
```

- User.java

```
package net.javaguides.springboot_restful_webservices.entity;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
```

```java
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;

@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
@Entity
@Table(name = "users")
public class User {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    @Column(nullable = false)
    private String firstName;
    @Column(nullable = false)
    private String lastName;
    @Column(nullable = false, unique = true)
    private String email;
}
```

- UserRepository.java

```java
package net.javaguides.springboot_restful_webservices.repository;

import net.javaguides.springboot_restful_webservices.entity.User;
import org.springframework.data.jpa.repository.JpaRepository;

public interface UserRepository extends JpaRepository<User, Long> {
}
```

- UserServiceImpl.java

```java
package net.javaguides.springboot_restful_webservices.service.impl;

import lombok.AllArgsConstructor;
import net.javaguides.springboot_restful_webservices.entity.User;
import
net.javaguides.springboot_restful_webservices.repository.UserRepository;
import net.javaguides.springboot_restful_webservices.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
@AllArgsConstructor
public class UserServiceImpl implements UserService {
    private UserRepository userRepository;

    @Override
    public User createUser(User user) {
        return userRepository.save(user);
    }

    @Override
    public User getUserById(Long userId) {
        Optional<User> optionalUser = userRepository.findById(userId);
        return optionalUser.get();
    }

    @Override
    public List<User> getAllUsers() {
        return userRepository.findAll();
    }

    @Override
    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
    public void deleteUser(Long userId) {
        userRepository.deleteById(userId);
    }
}
```

- UserService.java

```java
package net.javaguides.springboot_restful_webservices.service;

import net.javaguides.springboot_restful_webservices.entity.User;

import java.util.List;

public interface UserService {
    User createUser(User user);

    User getUserById(Long userId);

    List<User> getAllUsers();

    User updateUser(User user);

    void deleteUser(Long userId);
}
```

- SpringbootRestfulWebservicesApplication.java

```java
package net.javaguides.springboot_restful_webservices;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class SpringbootRestfulWebservicesApplication {

    public static void main(String[] args) {
       SpringApplication.run(SpringbootRestfulWebservicesApplication.class,
args);
    }

}
```

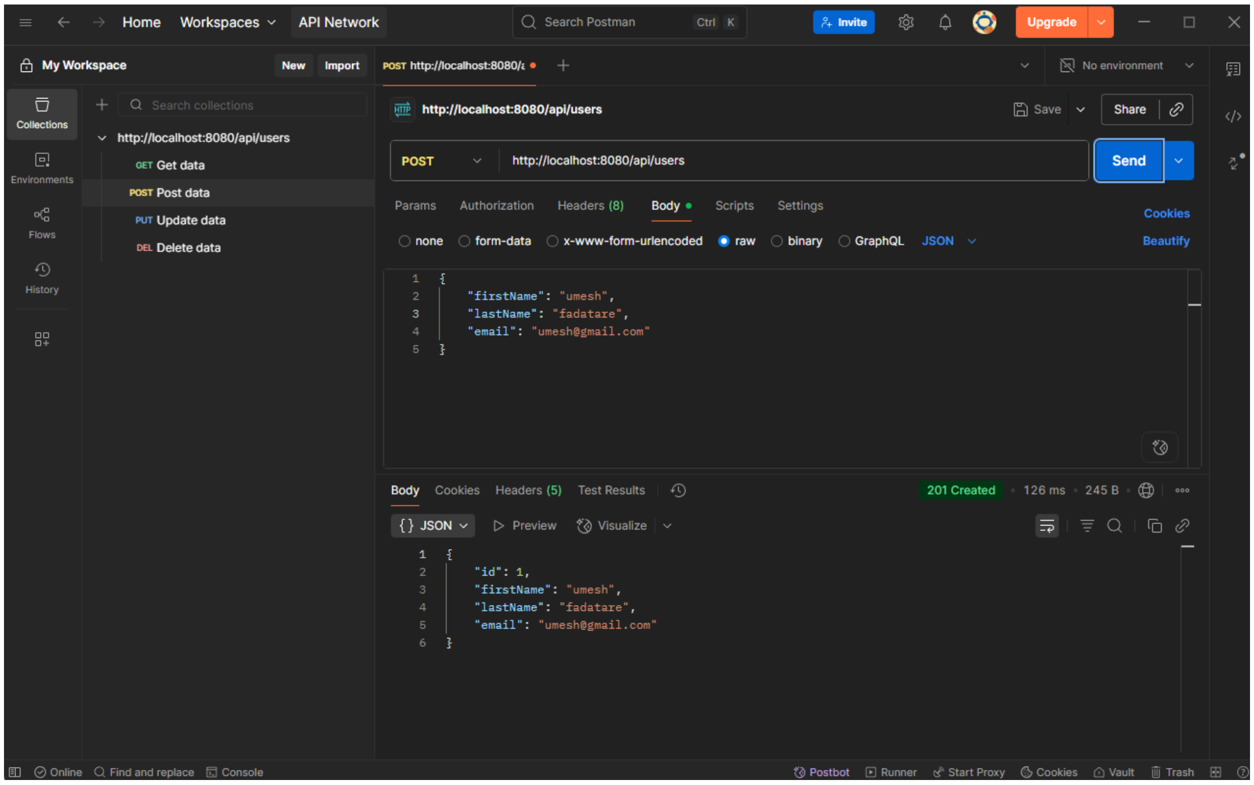
- Application.properties

```properties
spring.application.name=springboot-restful-webservices
spring.datasource.url=jdbc:mysql://localhost:3306/user_management
spring.datasource.username=root
spring.datasource.password=

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect
spring.jpa.hibernate.ddl-auto=update
```

POSTMAN RESULTS:

Create User REST API:



POST http://localhost:8080/api/users

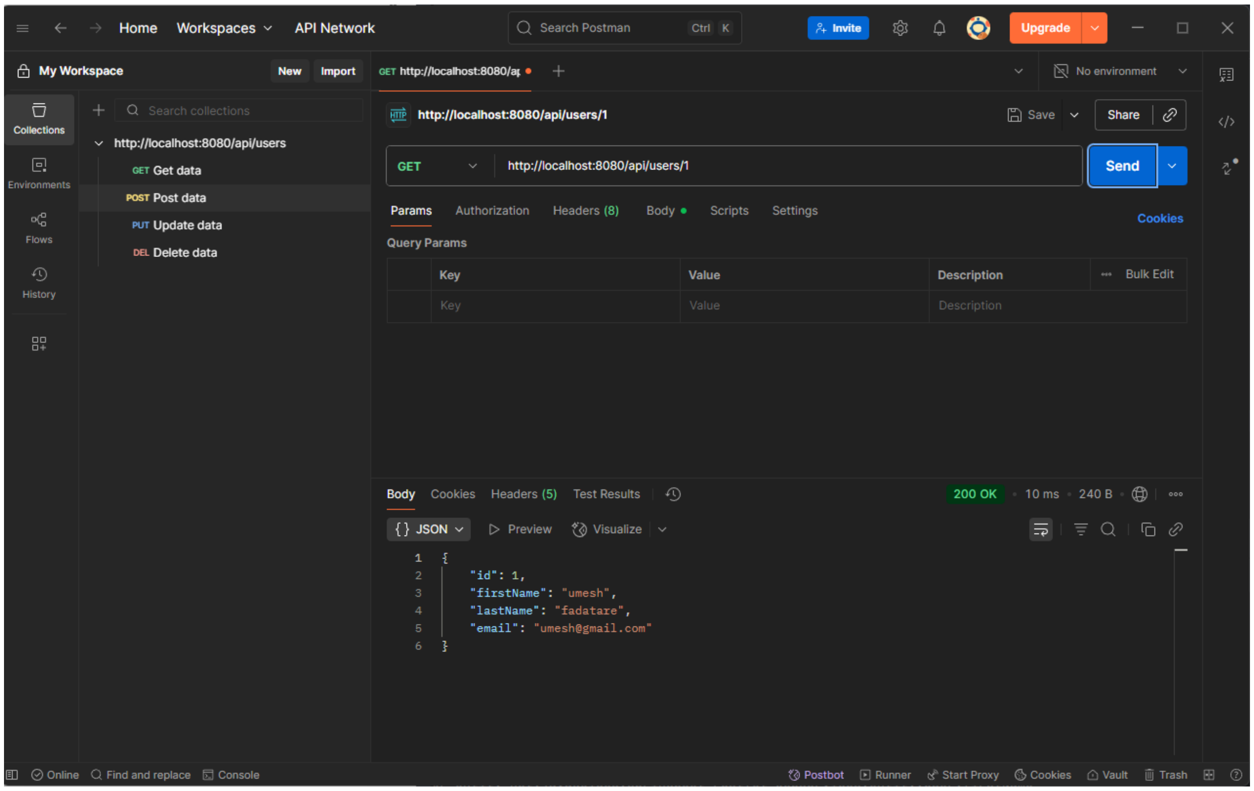
Body: none, form-data, x-www-form-urlencoded, [✓ raw], binary, GraphQL — JSON

```
{
    "firstName": "umesh",
    "lastName": "fadatare",
    "email": "umesh@gmail.com"
}
```

Response: 201 Created · 126 ms · 245 B

```
{
    "id": 1,
    "firstName": "umesh",
    "lastName": "fadatare",
    "email": "umesh@gmail.com"
}
```

Get Single User REST API:



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

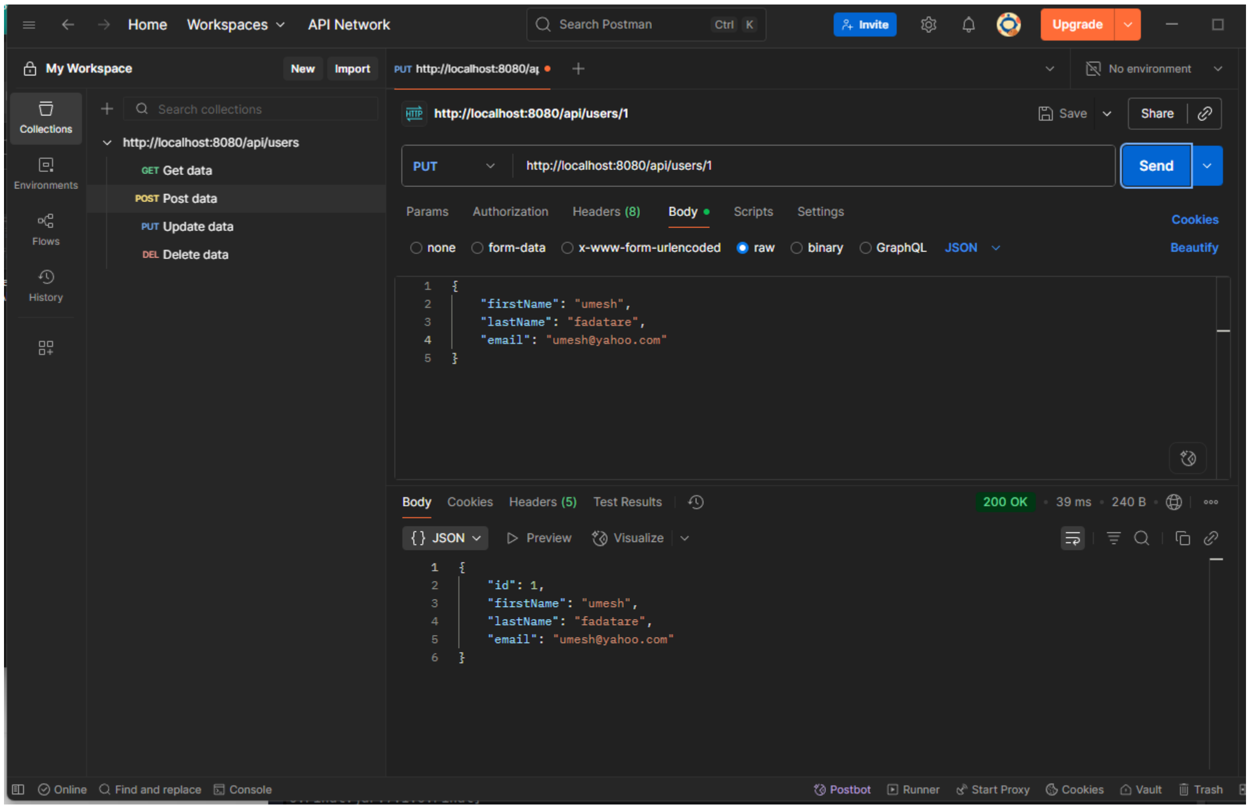
GET http://localhost:8080/api/users/1

Body Cookies Headers (5) Test Results — 200 OK · 10 ms · 240 B

JSON

```
{
    "id": 1,
    "firstName": "umesh",
    "lastName": "fadatare",
    "email": "umesh@gmail.com"
}
```

Update User REST API:



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

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

Body: none form-data x-www-form-urlencoded [✓ raw] binary GraphQL JSON

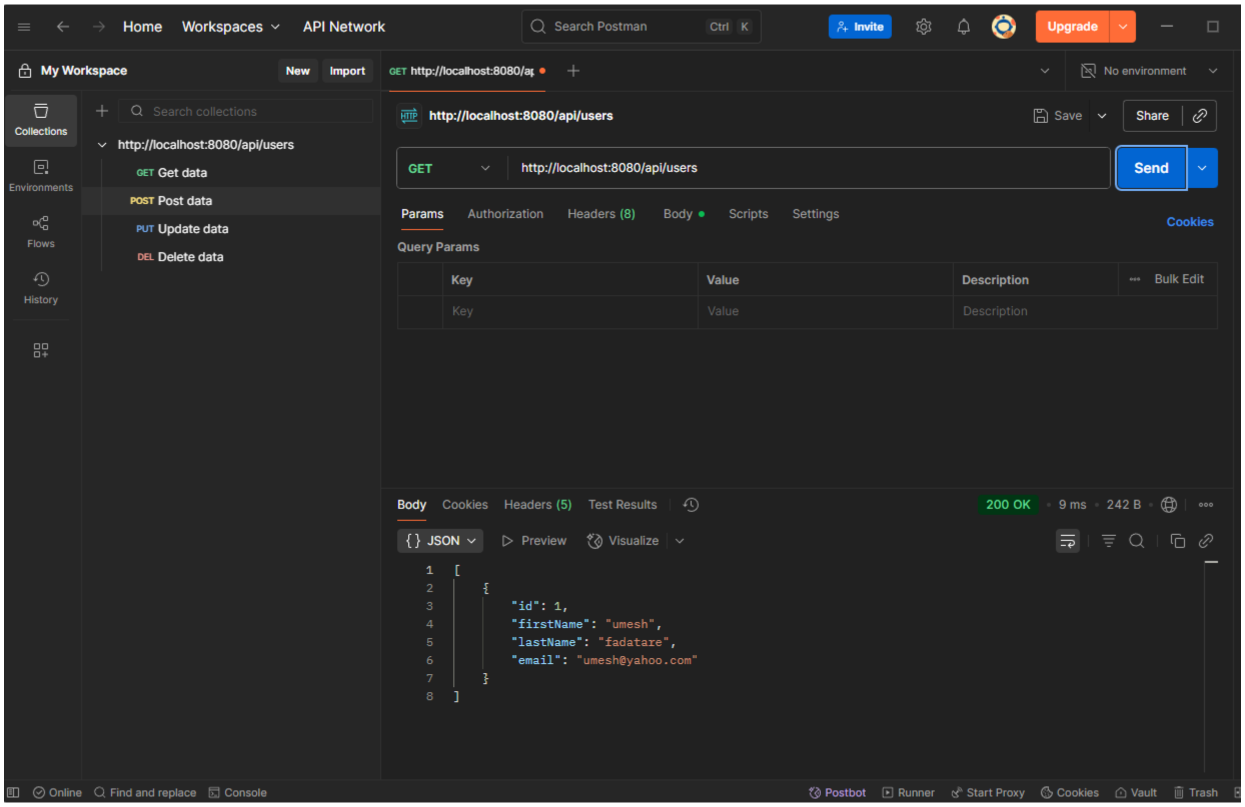
```
{
    "firstName": "umesh",
    "lastName": "fadatare",
    "email": "umesh@yahoo.com"
}
```

Body Cookies Headers (5) Test Results — 200 OK · 39 ms · 240 B

JSON

```
{
    "id": 1,
    "firstName": "umesh",
    "lastName": "fadatare",
    "email": "umesh@yahoo.com"
}
```

Get All Users REST API:



http://localhost:8080/api/users
- GET Get data
- POST Post data
- PUT Update data
- DEL Delete data

GET http://localhost:8080/api/users

Query Params

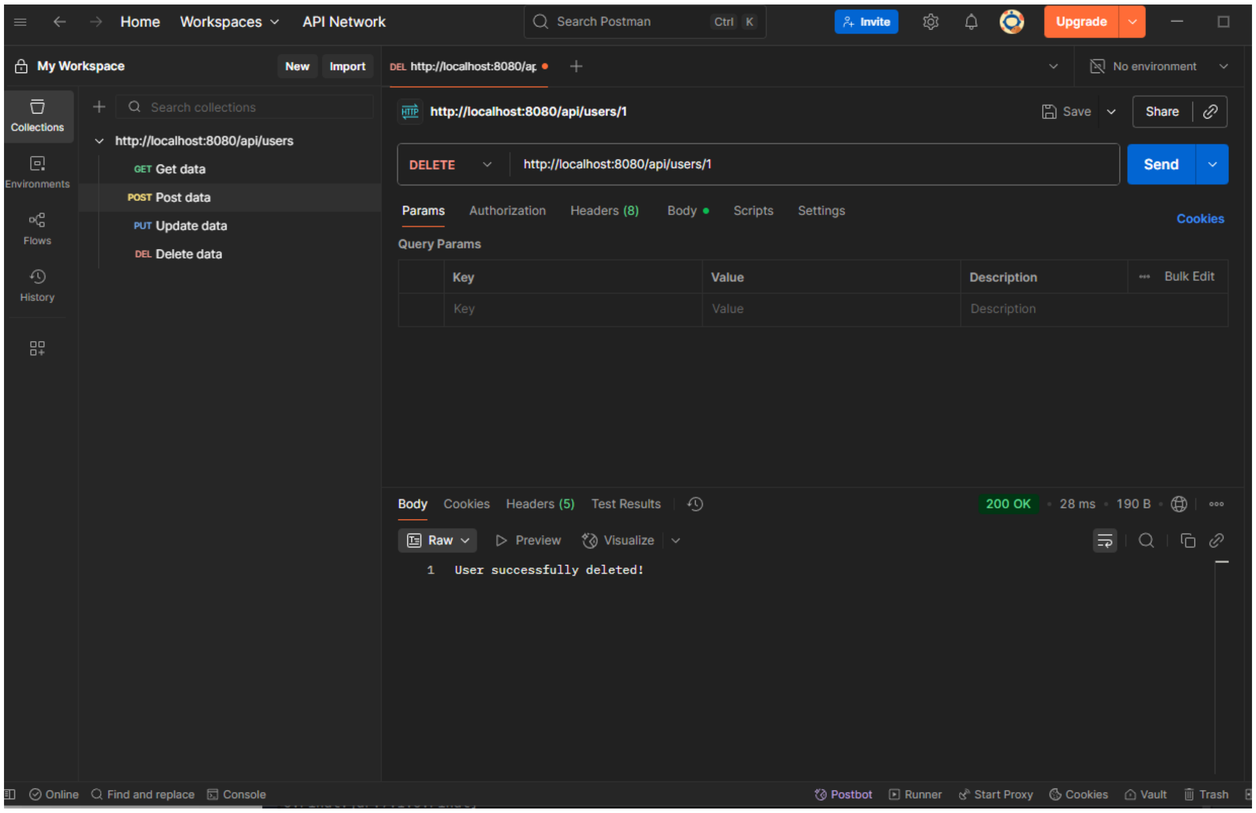
Key	Value	Description
Key	Value	Description

Body · Cookies · Headers (5) · Test Results — 200 OK · 9 ms · 242 B

JSON

```
[
    {
        "id": 1,
        "firstName": "umesh",
        "lastName": "fadatare",
        "email": "umesh@yahoo.com"
    }
]
```

Delete User REST API:



http://localhost:8080/api/users
- GET Get data
- POST Post data
- PUT Update data
- DEL Delete data

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

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (5) Test Results — 200 OK · 28 ms · 190 B

```
User successfully deleted!
```