**1.User and order Microservices**

**Project Structure**

user-order-microservices/

├── user-service/

│ ├── pom.xml

│ └── src/

│ └── main/

│ ├── java/

│ │ └── com/example/userservice/

│ │ ├── UserServiceApplication.java

│ │ ├── controller/

│ │ │ └── UserController.java

│ │ ├── entity/

│ │ │ └── User.java

│ │ └── repository/

│ │ └── UserRepository.java

│ └── resources/

│ └── application.yml

│

├── order-service/

│ ├── pom.xml

│ └── src/

│ └── main/

│ ├── java/

│ │ └── com/example/orderservice/

│ │ ├── OrderServiceApplication.java

│ │ ├── controller/

│ │ │ └── OrderController.java

│ │ ├── entity/

│ │ │ └── Order.java

│ │ └── repository/

│ │ └── OrderRepository.java

│ └── resources/

│ └── application.yml

**User-service**

**UserController.java**

package com.example.userservice.controller;

import com.example.userservice.entity.User;

import com.example.userservice.repository.UserRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.\*;

import java.util.List;

@RestController

@RequestMapping("/users")

public class UserController {

    @Autowired

    private UserRepository userRepository;

    @PostMapping

    public User createUser(@RequestBody User user) {

        return userRepository.save(user);

    }

    @GetMapping

    public List<User> getAllUsers() {

        return userRepository.findAll();

    }

    @GetMapping("/{id}")

    public User getUserById(@PathVariable Long id) {

        return userRepository.findById(id).orElse(null);

    }

}

**User.java**

package com.example.userservice.entity;

import jakarta.persistence.\*;

@Entity

public class User {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private Long id;

    private String name;

    private String email;

    // Constructors

    public User() {

    }

    public User(Long id, String name, String email) {

        this.id = id;

        this.name = name;

        this.email = email;

    }

    // Getters & Setters

    public Long getId() {

        return id;

    }

    public void setId(Long id) {

        this.id = id;

    }

    public String getName() {

        return name;

    }

    public void setName(String name) {

        this.name = name;

    }

    public String getEmail() {

        return email;

    }

    public void setEmail(String email) {

        this.email = email;

    }

}

**UserRepository.java**

package com.example.userservice.repository;

import com.example.userservice.entity.User;

import org.springframework.data.jpa.repository.JpaRepository;

public interface UserRepository extends JpaRepository<User, Long> {

}

UserServiceApplication.java

package com.example.userservice.userservice;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class UserServiceApplication {

    public static void main(String[] args) {

        SpringApplication.run(UserServiceApplication.class, args);

    }

}

**Application.yml**

server:

  port: 8081

spring:

  datasource:

    url: jdbc:mysql://localhost:3306/userdb

    username: root

    password: Spoorthi@28

  jpa:

    hibernate:

      ddl-auto: update

    show-sql: true

**Pom.xml**

<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 http://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <groupId>com.example</groupId>

    <artifactId>user-service</artifactId>

    <version>0.0.1-SNAPSHOT</version>

    <packaging>jar</packaging>

    <name>user-service</name>

    <description>User Microservice</description>

    <parent>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-parent</artifactId>

        <version>3.2.4</version>

        <relativePath/> <!-- lookup parent from repository -->

    </parent>

    <dependencies>

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-web</artifactId>

        </dependency>

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-data-jpa</artifactId>

        </dependency>

        <dependency>

            <groupId>com.mysql</groupId>

            <artifactId>mysql-connector-j</artifactId>

            <scope>runtime</scope>

        </dependency>

    </dependencies>

    <build>

        <plugins>

            <plugin>

                <groupId>org.springframework.boot</groupId>

                <artifactId>spring-boot-maven-plugin</artifactId>

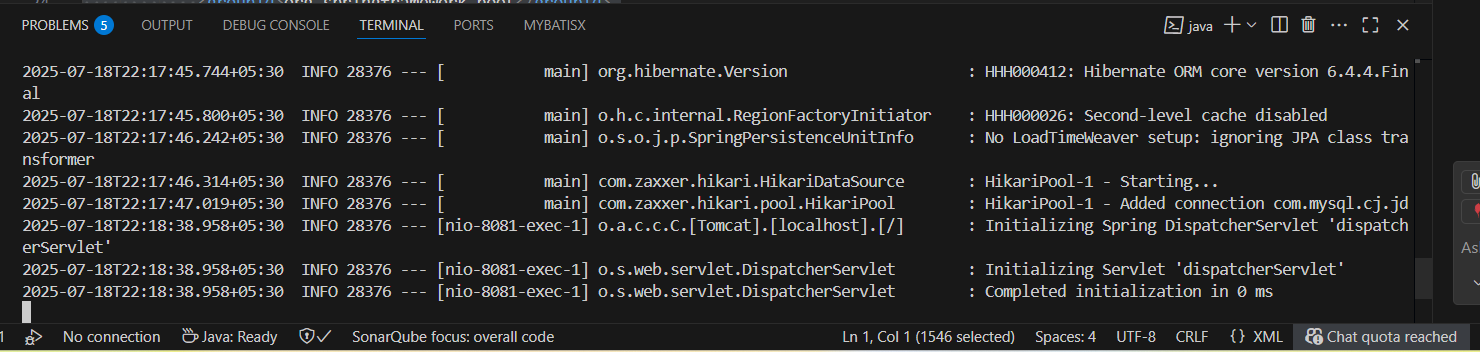
            </plugin>

        </plugins>

    </build>

</project>

**Output:**



**Order-service**

**OrderController.java**

package com.example.orderservice.controller;

import com.example.orderservice.entity.Order;

import com.example.orderservice.repository.OrderRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.\*;

import java.util.List;

@RestController

@RequestMapping("/orders")

public class OrderController {

    @Autowired

    private OrderRepository repo;

    @PostMapping

    public Order create(@RequestBody Order order) {

        return repo.save(order);

    }

    @GetMapping

    public List<Order> list() {

        return repo.findAll();

    }

    @GetMapping("/user/{userId}")

    public List<Order> getByUser(@PathVariable Long userId) {

        return repo.findByUserId(userId);

    }

}

**Order.java**

package com.example.orderservice.entity;

import jakarta.persistence.\*;

@Entity

public class Order {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private Long id;

    private Long userId;

    private String product;

    private Double amount;

    public Long getId() { return id; }

    public void setId(Long id) { this.id = id; }

    public Long getUserId() { return userId; }

    public void setUserId(Long userId) { this.userId = userId; }

    public String getProduct() { return product; }

    public void setProduct(String product) { this.product = product; }

    public Double getAmount() { return amount; }

    public void setAmount(Double amount) { this.amount = amount; }

}

**OrderRepository.java**

package com.example.orderservice.repository;

import com.example.orderservice.entity.Order;

import org.springframework.data.jpa.repository.JpaRepository;

import java.util.List;

public interface OrderRepository extends JpaRepository<Order, Long> {

    List<Order> findByUserId(Long userId);

}

OrderServiceApplication.java

package com.example.orderservice;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class OrderServiceApplication {

    public static void main(String[] args) {

        SpringApplication.run(OrderServiceApplication.class, args);

    }

}

**Application.yml**

server:

  port: 8082

spring:

  datasource:

    url: jdbc:mysql://localhost:3306/orderdb

    username: root

    password: Spoorthi@28

  jpa:

    hibernate:

      ddl-auto: update

    show-sql: true

**pom.xml**

<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 http://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <groupId>com.example</groupId>

    <artifactId>order-service</artifactId>

    <version>0.0.1-SNAPSHOT</version>

    <packaging>jar</packaging>

    <name>order-service</name>

    <description>Order Microservice</description>

    <parent>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-parent</artifactId>

        <version>3.2.4</version>

    </parent>

    <dependencies>

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-web</artifactId>

        </dependency>

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-data-jpa</artifactId>

        </dependency>

        <dependency>

            <groupId>com.mysql</groupId>

            <artifactId>mysql-connector-j</artifactId>

            <scope>runtime</scope>

        </dependency>

    </dependencies>

    <build>

  <plugins>

    <plugin>

      <groupId>org.springframework.boot</groupId>

      <artifactId>spring-boot-maven-plugin</artifactId>

      <version>3.2.4</version>

      <configuration>

        <mainClass>com.example.orderservice.OrderServiceApplication</mainClass>

      </configuration>

    </plugin>

  </plugins>

</build>

</project>

**Output:**

