Monolith_Microservice

Deliveryservice

DeliveryServiceApplication

```
package com.example.delivery;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class DeliveryServiceApplication {

    public static void main(String[] args) {

        SpringApplication.run(DeliveryServiceApplication.class, args);

    }

}
```

Delivery.java

```
package com.example.delivery.entity;


import jakarta.persistence.*;


@Entity

public class Delivery {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private Long id;


    private String address;

    private String status;

            public Long getId() {
```

```
		return id;
	}
	public void setId(Long id) {
		this.id = id;
	}
	public String getAddress() {
		return address;
	}
	public void setAddress(String address) {
		this.address = address;
	}
	public String getStatus() {
		return status;
	}
	public void setStatus(String status) {
		this.status = status;
	}

}
```

DeliveryController.java

```
package com.example.delivery.controller;

import com.example.delivery.entity.Delivery;
import com.example.delivery.service.DeliveryService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import java.util.List;

@RestController
@RequestMapping("/deliveries")
public class DeliveryController {

    @Autowired
    private DeliveryService deliveryService;

    @PostMapping
    public Delivery scheduleDelivery(@RequestBody Delivery delivery) {
        return deliveryService.scheduleDelivery(delivery);
    }

    @GetMapping
    public List<Delivery> getAllDeliveries() {
        return deliveryService.getAllDeliveries();
    }
}
```

DeliveryRepository.java

```
package com.example.delivery.repository;

import com.example.delivery.entity.Delivery;
import org.springframework.data.jpa.repository.JpaRepository;

public interface DeliveryRepository extends JpaRepository<Delivery, Long> {
}
```

DeliveryService.java

```
package com.example.delivery.service;


import com.example.delivery.entity.Delivery;

import java.util.List;


public interface DeliveryService {

    Delivery scheduleDelivery(Delivery delivery);

    List<Delivery> getAllDeliveries();

}
```

DeliveryServiceImp1.java

```
package com.example.delivery.service;


import com.example.delivery.entity.Delivery;

import com.example.delivery.repository.DeliveryRepository;

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

import org.springframework.stereotype.Service;


import java.util.List;


@Service

public class DeliveryServiceImpl implements DeliveryService {


    @Autowired

    private DeliveryRepository deliveryRepository;


    @Override
    public Delivery scheduleDelivery(Delivery delivery) {
        return deliveryRepository.save(delivery);
    }

    @Override
    public List<Delivery> getAllDeliveries() {
        return deliveryRepository.findAll();
    }
}
```

Application.properties

```
spring.application.name=DeliveryService
server:
  port: 8083

spring:
  application:
    name: delivery-service
  datasource:
    url: jdbc:h2:mem:deliverydb
    driver-class-name: org.h2.Driver
    username: sa
    password:
  h2:
    console:
      enabled: true
  jpa:
    hibernate:
      ddl-auto: update
    show-sql: true
```

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>3.5.4</version>
		<relativePath/> <!-- lookup parent from repository -->
	</parent>
	<groupId>com.example</groupId>
	<artifactId>deliveryservice</artifactId>
	<version>0.0.1-SNAPSHOT</version>
	<name>DeliveryService</name>
	<description>Demo project for Spring Boot</description>
	<url/>
	<licenses>
		<license/>
	</licenses>
	<developers>
		<developer/>
	</developers>
	<scm>
```

```xml
    <connection/>
    <developerConnection/>
    <tag/>
    <url/>
</scm>
<properties>
    <java.version>21</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.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>
```

Orderservice

OrderServiceApplication.java

```
package com.example.order;

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);
    }
}
```

Order.java

```
package com.example.order.entity;

import jakarta.persistence.*;
import java.time.LocalDateTime;
import java.util.List;

@Entity
@Table(name = "orders")
public class Order {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private Long restaurantId;
    private String customerName;
    private LocalDateTime orderTime;

    @OneToMany(mappedBy = "order", cascade = CascadeType.ALL)
```

```
private List<OrderItem> items;

    public Long getId() {
          return id;
    }

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

    public Long getRestaurantId() {
          return restaurantId;
    }

    public void setRestaurantId(Long restaurantId) {
          this.restaurantId = restaurantId;
    }

    public String getCustomerName() {
          return customerName;
    }

    public void setCustomerName(String customerName) {
          this.customerName = customerName;
    }

    public LocalDateTime getOrderTime() {
```

```
                return orderTime;
        }

        public void setOrderTime(LocalDateTime orderTime) {
                this.orderTime = orderTime;
        }

        public List<OrderItem> getItems() {
                return items;
        }

        public void setItems(List<OrderItem> items) {
                this.items = items;
        }

}
```

OrderItem.java

```
package com.example.order.entity;

import jakarta.persistence.*;

@Entity
public class OrderItem {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
private String itemName;

private int quantity;


@ManyToOne

@JoinColumn(name = "order_id")

private Order order;


    public Long getId() {

        return id;

    }


    public void setId(Long id) {

        this.id = id;

    }


    public String getItemName() {

        return itemName;

    }


    public void setItemName(String itemName) {

        this.itemName = itemName;

    }


    public int getQuantity() {

        return quantity;

    }
```

```
        public void setQuantity(int quantity) {
                this.quantity = quantity;
        }

        public Order getOrder() {
                return order;
        }

        public void setOrder(Order order) {
                this.order = order;
 }
}
```

OrderController.java

```
package com.example.order.controller;

import com.example.order.entity.Order;
import com.example.order.service.OrderService;
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 OrderService orderService;

    @PostMapping
    public Order placeOrder(@RequestBody Order order) {
        return orderService.placeOrder(order);
    }

    @GetMapping
    public List<Order> getAllOrders() {
        return orderService.getAllOrders();
    }
}
```

OrderItemRepository.java

```
package com.example.order.repository;

import com.example.order.entity.OrderItem;
import org.springframework.data.jpa.repository.JpaRepository;

public interface OrderItemRepository extends JpaRepository<OrderItem, Long> {
}
```

OrderRepository.java

```
package com.example.order.repository;

import com.example.order.entity.Order;
import org.springframework.data.jpa.repository.JpaRepository;

public interface OrderRepository extends JpaRepository<Order, Long> {
}
```

OrderService.java

```
package com.example.order.service;


import com.example.order.entity.Order;
import java.util.List;


public interface OrderService {
    Order placeOrder(Order order);
    List<Order> getAllOrders();
}
```

OrderServiceImp1.java

```
package com.example.order.service;


import com.example.order.entity.Order;
import com.example.order.repository.OrderRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;


import java.time.LocalDateTime;
import java.util.List;


@Service
public class OrderServiceImpl implements OrderService {


    @Autowired
    private OrderRepository orderRepository;
```

```
  @Override
  public Order placeOrder(Order order) {
    order.setOrderTime(LocalDateTime.now());
    return orderRepository.save(order);
  }

  @Override
  public List<Order> getAllOrders() {
    return orderRepository.findAll();
  }
}
```

Application.properties

```
spring.application.name=OrderService
server:
 port: 8082

spring:
 application:
  name: order-service
 datasource:
  url: jdbc:h2:mem:orderdb
  driver-class-name: org.h2.Driver
  username: sa
  password:
 h2:
  console:
    enabled: true
 jpa:
  hibernate:
    ddl-auto: update
  show-sql: true
```

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>3.5.4</version>
		<relativePath/> <!-- lookup parent from repository -->
	</parent>
	<groupId>com.example</groupId>
	<artifactId>orderservice</artifactId>
	<version>0.0.1-SNAPSHOT</version>
	<name>OrderService</name>
	<description>Demo project for Spring Boot</description>
	<url/>
	<licenses>
		<license/>
	</licenses>
	<developers>
```

```
    <developer/>
</developers>
<scm>
    <connection/>
    <developerConnection/>
    <tag/>
    <url/>
</scm>
<properties>
    <java.version>21</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.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>
```

Restaurantservice

RestaurantServieApplication.java

```
package com.example.restaurant;


import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;


@SpringBootApplication

public class RestaurantServiceApplication {

    public static void main(String[] args) {

        SpringApplication.run(RestaurantServiceApplication.class, args);

    }

}
```

MenuItem.java

```
package com.example.restaurant.entity;


import jakarta.persistence.*;


@Entity

public class MenuItem {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private Long id;

    private String name;

    private double price;


    @ManyToOne
```

```
@JoinColumn(name = "restaurant_id")
private Restaurant restaurant;

    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 double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }
```

```
        public Restaurant getRestaurant() {
                return restaurant;
        }

        public void setRestaurant(Restaurant restaurant) {
                this.restaurant = restaurant;
        }
}
```

Restaurant.java

```
package com.example.restaurant.entity;

import jakarta.persistence.*;
import java.util.List;

@Entity
public class Restaurant {
   @Id
   @GeneratedValue(strategy = GenerationType.IDENTITY)
   private Long id;
   private String name;
   private String location;

   @OneToMany(mappedBy = "restaurant", cascade = CascadeType.ALL)
   private List<MenuItem> menuItems;

        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 getLocation() {
        return location;
}

public void setLocation(String location) {
        this.location = location;
}

public List<MenuItem> getMenuItems() {
        return menuItems;
}

public void setMenuItems(List<MenuItem> menuItems) {
        this.menuItems = menuItems;
    }
}
```

MenuItemController.java

```
package com.example.restaurant.controller;

import com.example.restaurant.entity.MenuItem;
import com.example.restaurant.repository.MenuItemRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/menu-items")
public class MenuItemController {

    @Autowired
    private MenuItemRepository menuItemRepository;

    @GetMapping
    public List<MenuItem> getAllItems() {
        return menuItemRepository.findAll();
    }

    @PostMapping
    public MenuItem createItem(@RequestBody MenuItem item) {
        return menuItemRepository.save(item);
    }
}
```

RestaurantController.java

```
package com.example.restaurant.controller;

import com.example.restaurant.entity.Restaurant;
import com.example.restaurant.service.RestaurantService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/restaurants")
public class RestaurantController {

    @Autowired
    private RestaurantService restaurantService;

    @GetMapping
    public List<Restaurant> getAllRestaurants() {
        return restaurantService.getAllRestaurants();
    }

    @GetMapping("/{id}")
    public Restaurant getRestaurantById(@PathVariable Long id) {
        return restaurantService.getRestaurantById(id);
    }

    @PostMapping
    public Restaurant createRestaurant(@RequestBody Restaurant restaurant) {
        return restaurantService.saveRestaurant(restaurant);
    }

    @DeleteMapping("/{id}")
    public void deleteRestaurant(@PathVariable Long id) {
        restaurantService.deleteRestaurant(id);
    }
}
```

MenuItemRepository.java

```
package com.example.restaurant.repository;

import com.example.restaurant.entity.MenuItem;
import org.springframework.data.jpa.repository.JpaRepository;

public interface MenuItemRepository extends JpaRepository<MenuItem, Long> {}
```

RestaurantRepository.java

```
package com.example.restaurant.repository;

import com.example.restaurant.entity.Restaurant;
import org.springframework.data.jpa.repository.JpaRepository;

public interface RestaurantRepository extends JpaRepository<Restaurant, Long> {}
```

RestaurantService.java

```
package com.example.restaurant.service;

import com.example.restaurant.entity.Restaurant;
import java.util.List;

public interface RestaurantService {
    List<Restaurant> getAllRestaurants();
    Restaurant getRestaurantById(Long id);
    Restaurant saveRestaurant(Restaurant restaurant);
    void deleteRestaurant(Long id);
}
```

RestaurantServiceImp1.java

```
package com.example.restaurant.service;

import com.example.restaurant.entity.Restaurant;
import com.example.restaurant.repository.RestaurantRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class RestaurantServiceImpl implements RestaurantService {

    @Autowired
    private RestaurantRepository restaurantRepository;
    public List<Restaurant> getAllRestaurants() {
        return restaurantRepository.findAll();
    }

    public Restaurant getRestaurantById(Long id) {
        return restaurantRepository.findById(id).orElse(null);
    }

    public Restaurant saveRestaurant(Restaurant restaurant) {
        return restaurantRepository.save(restaurant);
    }

    public void deleteRestaurant(Long id) {
        restaurantRepository.deleteById(id);
    }
}
```

Application.properties

```
spring.application.name=RestaurantService
server:
  port: 8083

spring:
  datasource:
    url: jdbc:h2:mem:restaurantdb
    driver-class-name: org.h2.Driver
    username: sa
    password:
  jpa:
    hibernate:
      ddl-auto: update
    show-sql: true
  h2:
    console:
      enabled: true
      path: /h2-console
```

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>3.5.4</version>
		<relativePath/> <!-- lookup parent from repository -->
	</parent>
	<groupId>com.example</groupId>
	<artifactId>restaurantservice</artifactId>
	<version>0.0.1-SNAPSHOT</version>
	<name>RestaurantService</name>
	<description>Demo project for Spring Boot</description>
```

```
<url/>

<licenses>

    <license/>

</licenses>

<developers>

    <developer/>

</developers>

<scm>

    <connection/>

    <developerConnection/>

    <tag/>

    <url/>

</scm>

<properties>

    <java.version>21</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.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>
```