

Spring Cloud Assignment Solutions

Assignment 1 : Microservices

@Entity

public class Ticket {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "id", updatable = false, nullable = false)

private Long id;

private Long numberoftickets;

private BigDecimal totalprice ;

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

```
public Long getNumeroftickets() {  
  
    return numeroftickets;  
  
}  
  
public void setNumeroftickets(Long numeroftickets) {  
  
    this.numeroftickets = numeroftickets;  
  
}  
  
public BigDecimal getTotalprice() {  
  
    return totalprice;  
  
}  
  
public void setTotalprice(BigDecimal totalprice) {  
  
    this.totalprice = totalprice;  
  
}  
  
}  
  
public interface TicketRepo extends JpaRepository<Ticket, Long> {  
  
}  
  
@RestController  
  
@RequestMapping("/ticketapi")
```

```

public class TicketResrtController {

    @Autowired

    TicketRepo repo;

    @RequestMapping(value="/tickets/{quantity}",
method=RequestMethod.GET)

    public Ticket resrveTickets(@PathVariable("quantity") Long quantity) {

        Ticket ticket = new Ticket();

        ticket.setNumberoftickets(quantity);

        //new BigDecimal(quantity * 50.00);

        ticket.setTotalprice(new BigDecimal(quantity * 50.00));

        return repo.save(ticket);

    }

}

```

```

@RestController

```

```

@RequestMapping("/priceapi")

```

```

public class PriceRestController {

```

```

    @Autowired

```

```

    private PriceRepo repo;

    @RequestMapping(value="/prices", method =
RequestMethod.POST)

    public Price create(@RequestBody Price price) {

        return repo.save(price);

    }

    @RequestMapping(value="/prices/{id}",
method=RequestMethod.GET)

    public Price findPriceById(@PathVariable("id") Long id) {

        return repo.findById(id).get();

    }

    @RequestMapping(value="/prices/event/{target}",
method=RequestMethod.GET)

    public Price findPriceByTarget(@PathVariable("target") String target)
{

        return repo.findPriceByTarget(target);

    }

}

```

```
public interface PriceRepo extends JpaRepository<Price, Long> {  
  
    Price findPriceByTarget(String target);  
  
}
```

@Entity

```
public class Price {
```

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "id", updatable = false, nullable = false)

```
    private Long id ;
```

```
    private String target;
```

```
    private BigDecimal price;
```

```
    public Long getId() {
```

```
        return id;
```

```
}
```

```
    public void setId(Long id) {
```

```
        this.id = id;
```

```
}
```

```
    public BigDecimal getPrice() {  
  
        return price;  
    }  
  
    public void setPrice(BigDecimal price) {  
  
        this.price = price;  
    }  
  
    public String getTarget() {  
  
        return target;  
    }  
  
    public void setTarget(String target) {  
  
        this.target = target;  
    }  
}
```

Assignment 2 : Service Discovery using Eureka

Ticket service:

pom.xml

add version in properties:

```
<properties>
```

```
<java.version>1.8</java.version>
```

```
<spring-cloud.version>Hoxton.SR8</spring-cloud.version>
```

```
</properties>
```

Add dependency:

```
<dependency>
```

```
<groupId>org.springframework.cloud</groupId>
```

```
<artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
```

```
</dependency>
```

Add dependency management

```
<dependencyManagement>
```

```
<dependencies>
```

```
<dependency>
```

```
<groupId>org.springframework.cloud</groupId>
```

```
<artifactId>spring-cloud-dependencies</artifactId>
```

```
<version>${spring-cloud.version}</version>
```

```
<type>pom</type>
```

```
<scope>import</scope>
```

```
</dependency>
```

```
</dependencies>
```

```
</dependencyManagement>
```

In TicketServiceApplication.java

```
@SpringBootApplication
```

```
@EnableEurekaClient
```

```
public class TicketServiceApplication {
```

```
    public static void main(String[] args) {
```

```
        SpringApplication.run(TicketServiceApplication.class, args);
```

```
    }
```

```
}
```

In application.properties

```
spring.application.name=ticket-service
```

```
eureka.client.service-url.defaultZone=http://localhost:8761/eureka/
```


Price service:

pom.xml

add version in properties:

```
<properties>
```

```
<java.version>1.8</java.version>
```

```
<spring-cloud.version>Hoxton.SR8</spring-cloud.version>
```

```
</properties>
```

Add dependency:

```
<dependency>
```

```
<groupId>org.springframework.cloud</groupId>
```

```
<artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
```

```
</dependency>
```

Add dependency management

```
<dependencyManagement>
```

```
<dependencies>
```

```
<dependency>
```

```
<groupId>org.springframework.cloud</groupId>
```

<artifactId>spring-cloud-dependencies</artifactId>

<version>\${spring-cloud.version}</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

In PriceserviceApplication.java

@SpringBootApplication

@EnableEurekaClient

public class PriceserviceApplication {

public static void main(String[] args) {

SpringApplication.run(PriceserviceApplication.class, args);

}

}

In application.properties

server.port:9090

spring.application.name=price-service

eureka.client.service-url.defaultZone=http://localhost:8761/eureka/

Assignment 3 : FeignClient

Ticket service:

pom.xml

Add dependency:

```
<dependency>
```

```
<groupId>org.springframework.cloud</groupId>
```

```
<artifactId>spring-cloud-starter-openfeign</artifactId>
```

```
</dependency>
```

In model

add Price.java

```
public class Price {
```

```
    private Long id;
```

```
    private BigDecimal price;
```

```
    public Long getId() {  
  
        return id;  
    }  
  
    public void setId(Long id) {  
  
        this.id = id;  
    }  
  
    public BigDecimal getPrice() {  
  
        return price  
    }  
  
    public void setPrice(BigDecimal price) {  
  
        this.price = price  
    }  
}
```

In Ticket.java add property price

```
public class Ticket{  
  
    @Id  
  
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
private Long id;

private BigDecimal totalticket ;

private Integer ticketnumber;

@Transient

private Long priceId;

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public BigDecimal getTotalticket() {

return totalticket

}

public void setTotalticket(BigDecimal totalticket) {

this.totalticket = totalticket
```

```
}
```

```
    public Integer getTicketnumber() {
```

```
        return ticketnumber
```

```
    }
```

```
    public void setTicketNumber(Integer ticketnumber) {
```

```
        this.ticketnumber = ticketnumber
```

```
    }
```

```
    public Long getpriceld() {
```

```
        return priceld;
```

```
    }
```

```
    public void setPriceld(Long priceld) {
```

```
        this.priceld = priceld;
```

```
    }
```

```
}
```

In TicketserviceApplication.java

```
@SpringBootApplication
```

```
@EnableEurekaClient
```

@EnableFeignClients

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

In restClients PriceClient.java

```
@FeignClient("PRICE-SERVICE")  
  
public interface PriceClient {  
  
    @GetMapping("/priceapi/price/{id}")  
  
    Price getPrice(@PathVariable("id") Long id);  
  
}
```

In controller TicketRestController.java

```
@RestController  
  
@RequestMapping("/ticketapi")  
  
public class TicketRestController {
```

@Autowired

PriceClient priceClient;

@Autowired

private TicketRepo repo;

@RequestMapping(value = "/tickets", method=
RequestMethod.POST)

public Ticket create(@RequestBody Ticket ticket) {

Price price = priceClient.getPrice(ticket.getPriceId());

ticket.setTotalticket(price.getPrice()*ticket.getTicketnumber());

return repo.save(ticket);

}

}

Assignment 4 : Sleuth and Zipkin

server:

port: 8060

eureka:

instance:

preferIpAddress: true

instance-id:

client:

serviceUrl:

defaultZone: http://192.168.133.1:8761/eureka/

spring:

application:

name: billingservice

main:

allow-bean-definition-overriding: true

datasource:

password: root

url:

jdbc:mysql://localhost:3306/mydb?useSSL=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC

username: root

jpa:

show-sql: true

zipkin:

base-url: http://localhost:9411/

sleuth:

sampler:

probability: 1

feign:

enabled: true

```
<?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>2.3.3.RELEASE</version>
```

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

</parent>

<groupId>com.example</groupId>

<artifactId>couponservice</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>couponservice</name>

<description>Coupon Sercie</description>

<properties>

<java.version>1.8</java.version>

<spring-cloud.version>Greenwich.RELEASE</spring-cloud.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.cloud</groupId>

<artifactId>spring-cloud-starter-openfeign</artifactId>

<version>2.2.5.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-netflix-ribbon</artifactId>

<version>2.2.5.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-hystrix</artifactId>

<version>2.2.5.RELEASE</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-sleuth</artifactId>

</dependency>

<dependency>

```
<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-zipkin</artifactId>

</dependency>

<dependency>

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

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

<scope>test</scope>

<exclusions>

<exclusion>

<groupId>org.junit.vintage</groupId>

<artifactId>junit-vintage-engine</artifactId>

</exclusion>

</exclusions>

</dependency>

</dependencies>

<dependencyManagement>

<dependencies>
```

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-dependencies</artifactId>

<version>\${spring-cloud.version}</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

```
</build>
```

```
</project>
```

Assignment 5 : Ribbon

IPriceClient.java

```
@FeignClient("PRICE-SERVICE")
```

```
@RibbonClient("PRICE-SERVICE")
```

```
public interface IPriceClient {
```

```
    @GetMapping("/pricesapi/prices/{ticketname}")
```

```
    Price getPrices(@PathVariable("ticketname") String ticketname);
```

```
}
```

PriceRestController.java

```
@RestController
```

```
@RequestMapping("/pricesapi")
```

```
public class PriceRestController {
```

```
    @Autowired
```

```
    public IPriceDao pricedao;
```



```
@RequestMapping(value="/prices/{ticketname}", method =  
RequestMethod.GET)
```

```
public Price findPriceByName(@PathVariable("ticketname") String  
ticketname) {
```

```
    System.out.println("Server 2");
```

```
    return pricedao.findPriceByticketname(ticketname);
```

```
}
```

```
}
```

```
application.properties:
```

```
spring.datasource.url=jdbc:mysql://localhost:3306/mydb
```

```
spring.datasource.username=root
```

```
spring.datasource.password=admin
```

```
spring.application.name=price-service
```

```
eureka.client.service-url.defaultZone=http://localhost:8761/eureka/
```

```
server.port=8081
```

Assignment 6 : Zuul API Proxy Gateway

IPriceClient.java

@FeignClient("PRICE-SERVICE")

@RibbonClient("PRICE-SERVICE")

public interface IPriceClient {

@GetMapping("/pricesapi/prices/{ticketname}")

Price getPrices(@PathVariable("ticketname") String ticketname);

}

PriceRestController.java

@RestController

@RequestMapping("/pricesapi")

public class PriceRestController {

@Autowired

public IPriceDao pricedao;

@RequestMapping(value="/prices/{ticketname}", method =
RequestMethod.GET)

public Price findPriceByName(@PathVariable("ticketname") String
ticketname) {

```
System.out.println("Server 2");

return pricedao.findPriceByticketname(ticketname);

}

}
```

application.properties:

spring.datasource.url=jdbc:mysql://localhost:3306/mydb

spring.datasource.username=root

spring.datasource.password=admin

spring.application.name=price-service

eureka.client.service-url.defaultZone=http://localhost:8761/eureka/

server.port=8081