**Week-5: Microservices with Spring Boot 3 and Spring Cloud**

| **Creating Microservices for account and loan**  **Project Structure:**          **Code:**  AccountApplication.java  package com.cognizant.account;  import org.springframework.boot.SpringApplication;  import org.springframework.boot.autoconfigure.SpringBootApplication;  import org.springframework.cloud.client.discovery.EnableDiscoveryClient;  @SpringBootApplication  @EnableDiscoveryClient  public class AccountApplication {  public static void main(String[] args) {  SpringApplication.run(AccountApplication.class, args);  }  }  AccountController.java  package com.cognizant.account;  import org.springframework.web.bind.annotation.\*;  @RestController  @RequestMapping("/accounts")  public class AccountController {  @GetMapping("/{number}")  public Account getAccount(@PathVariable String number) {  return new Account(number, "savings", 234343);  }  }  class Account {  private String number;  private String type;  private int balance;  public Account(String number, String type, int balance) {  this.number = number;  this.type = type;  this.balance = balance;  }  public String getNumber() { return number; }  public String getType() { return type; }  public int getBalance() { return balance; }  }  application.properties  server.port=8080  spring.application.name=account-service  eureka.client.service-url.defaultZone=<http://localhost:8761/eureka>  LoanApplication.java  package com.cognizant.loan;  import org.springframework.boot.SpringApplication;  import org.springframework.boot.autoconfigure.SpringBootApplication;  import org.springframework.cloud.client.discovery.EnableDiscoveryClient;  @SpringBootApplication  @EnableDiscoveryClient  public class LoanApplication {  public static void main(String[] args) {  SpringApplication.run(LoanApplication.class, args);  }  }  LoanController.java  package com.cognizant.loan;  import org.springframework.web.bind.annotation.\*;  @RestController  @RequestMapping("/loans")  public class LoanController {  @GetMapping("/{number}")  public Loan getLoan(@PathVariable String number) {  return new Loan(number, "car", 400000, 3258, 18);  }  }  class Loan {  private String number;  private String type;  private int loan;  private int emi;  private int tenure;  public Loan(String number, String type, int loan, int emi, int tenure) {  this.number = number;  this.type = type;  this.loan = loan;  this.emi = emi;  this.tenure = tenure;  }  public String getNumber() { return number; }  public String getType() { return type; }  public int getLoan() { return loan; }  public int getEmi() { return emi; }  public int getTenure() { return tenure; }  }  Application.properties  server.port=8081  spring.application.name=loan-service  eureka.client.service-url.defaultZone=<http://localhost:8761/eureka>  EurekaDiscoveryServerApplication  package com.cognizant.eureka\_discovery\_server;  import org.springframework.boot.SpringApplication;  import org.springframework.boot.autoconfigure.SpringBootApplication;  import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;  @SpringBootApplication  @EnableEurekaServer  public class EurekaDiscoveryServerApplication {  public static void main(String[] args) {  SpringApplication.run(EurekaDiscoveryServerApplication.class, args);  }  }  Application.properties  spring.application.name=eureka-discovery-server  server.port=8761  eureka.client.register-with-eureka=false  eureka.client.fetch-registry=false  ApiGatewayApplication  package com.cognizant.api\_gateway;  import org.springframework.boot.SpringApplication;  import org.springframework.boot.autoconfigure.SpringBootApplication;  @SpringBootApplication  public class ApiGatewayApplication {  public static void main(String[] args) {  SpringApplication.run(ApiGatewayApplication.class, args);  }  }  LoggingFilter.java  package com.cognizant.apigateway.filter;  import org.slf4j.Logger;  import org.slf4j.LoggerFactory;  import org.springframework.cloud.gateway.filter.GatewayFilterChain;  import org.springframework.cloud.gateway.filter.GlobalFilter;  import org.springframework.core.Ordered;  import org.springframework.http.server.reactive.ServerHttpRequest;  import org.springframework.http.server.reactive.ServerHttpResponse;  import org.springframework.stereotype.Component;  import org.springframework.web.server.ServerWebExchange;  import reactor.core.publisher.Mono;  @Component  public class LoggingFilter implements GlobalFilter, Ordered {  private static final Logger logger = LoggerFactory.getLogger(LoggingFilter.class);  @Override  public Mono<Void> filter(ServerWebExchange exchange, GatewayFilterChain chain) {  ServerHttpRequest request = exchange.getRequest();  logger.info("Incoming Request: {} {}", request.getMethod(), request.getURI());  return chain.filter(exchange).then(Mono.fromRunnable(() -> {  ServerHttpResponse response = exchange.getResponse();  logger.info("Outgoing Response Status: {}", response.getStatusCode());  }));  }  @Override  public int getOrder() {  return -1; // Priority (lower means higher priority)  }  }  Application.properties  server.port=9090  spring.application.name=api-gateway  eureka.client.service-url.defaultZone=http://localhost:8761/eureka  spring.cloud.gateway.discovery.locator.enabled=true  spring.cloud.gateway.discovery.locator.lower-case-service-id=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>3.5.3</version>  <relativePath/> <!-- lookup parent from repository -->  </parent>  <groupId>com.cognizant</groupId>  <artifactId>eureka-discovery-server</artifactId>  <version>0.0.1-SNAPSHOT</version>  <name>eureka-discovery-server</name>  <description>Demo project for Spring Boot</description>  <url/>  <licenses>  <license/>  </licenses>  <developers>  <developer/>  </developers>  <scm>  <connection/>  <developerConnection/>  <tag/>  <url/>  </scm>  <properties>  <java.version>17</java.version>  <spring-cloud.version>2025.0.0</spring-cloud.version>  </properties>  <dependencies>  <dependency>  <groupId>org.springframework.cloud</groupId>  <artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-devtools</artifactId>  <scope>runtime</scope>  <optional>true</optional>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-test</artifactId>  <scope>test</scope>  </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>  **Output** |
| --- |