**WEEK 5 – ASSIGNMENT**

**EXERCISE 1: Creating Microservices for account and loan**

**AccountController.java**

@RestController

@RequestMapping("/accounts")

public class AccountController {

@GetMapping("/{number}")

public Map<String, Object> getAccount(@PathVariable String number) {

Map<String, Object> account = new HashMap<>();

account.put("number", number);

account.put("type", "savings");

account.put("balance", 234343);

return account;

}

}

**LoanController.java**

@RestController

@RequestMapping(“/loans”)

Public class LoanController {

@GetMapping(“/{number}”)

Public Map<String, Object> getLoan(@PathVariable String number) {

Map<String, Object> loan = new HashMap<>();

Loan.put(“number”, number);

Loan.put(“type”, “car”);

Loan.put(“loan”, 400000);

Loan.put(“emi”, 3258);

Loan.put(“tenure”, 18);

Return loan;

}

}

**Output:**

**Json**

**AccountController**

{

“number”: “00987987973432”,

“type”: “savings”,

“balance”: 234343

}

**LoanController**

{

“number”: “H00987987972342”,

“type”: “car”,

“loan”: 400000,

“emi”: 3258,

“tenure”: 18

}

**EXERCISE 2: Create Eureka Discovery Server and register microservice**

**1. Eureka Discovery Server**

**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.cognizant</groupId>

<artifactId>eureka-discovery-server</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>eureka-discovery-server</name>

<description>Eureka Discovery Server</description>

<packaging>jar</packaging>

<parent>

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

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

<version>3.2.4</version>

<relativePath/>

</parent>

<properties>

<java.version>17</java.version>

<spring-cloud.version>2023.0.1</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-starter</artifactId>

</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>

**EurekaDiscoveryServerApplication.java**

Package com.cognizant;

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**

Server.port=8761

Eureka.client.register-with-eureka=false

Eureka.client.fetch-registry=false

Logging.level.com.netflix.eureka=OFF

Logging.level.com.netflix.discovery=OFF

**Output:**

Application AMIs Availability Zones Status

ACCOUNT-SERVICE (1) (localhost) UP (1) – <http://localhost:8080>

LOAN-SERVICE (1) (localhost) UP (1) – <http://localhost:8081>

API-GATEWAY (1) (localhost) UP (1) – <http://localhost:9000>

**2. Account Microservice**

**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.cognizant</groupId>

<artifactId>account</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>account</name>

<description>Account Microservice</description>

<packaging>jar</packaging>

<parent>

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

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

<version>3.2.4</version>

<relativePath/>

</parent>

<properties>

<java.version>17</java.version>

<spring-cloud.version>2023.0.1</spring-cloud.version>

</properties>

<dependencies>

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

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

</dependency>

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

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

</dependency>

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

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</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>

**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.controller;

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

Import java.util.\*;

@RestController

@RequestMapping(“/accounts”)

Public class AccountController {

@GetMapping(“/{number}”)

Public Map<String, Object> getAccountDetails(@PathVariable String number) {

Map<String, Object> account = new HashMap<>();

Account.put(“number”, number);

Account.put(“type”, “savings”);

Account.put(“balance”, 234343);

Return account;

}

}

**Application.properties**

Server.port=8080

Spring.application.name=account-service

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

**Output:**

**Json**

{

“number”: “00987987973432”,

“type”: “savings”,

“balance”: 234343

}

**3. Loan Microservice**

**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.cognizant</groupId>

<artifactId>loan</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>loan</name>

<description>Loan Microservice</description>

<packaging>jar</packaging>

<parent>

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

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

<version>3.2.4</version>

<relativePath/>

</parent>

<properties>

<java.version>17</java.version>

<spring-cloud.version>2023.0.1</spring-cloud.version>

</properties>

<dependencies>

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

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

</dependency>

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

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

</dependency>

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

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</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>

**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.controller;

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

Import java.util.\*;

@RestController

@RequestMapping(“/loans”)

Public class LoanController {

@GetMapping(“/{number}”)

Public Map<String, Object> getLoanDetails(@PathVariable String number) {

Map<String, Object> loan = new HashMap<>();

Loan.put(“number”, number);

Loan.put(“type”, “car”);

Loan.put(“loan”, 400000);

Loan.put(“emi”, 3258);

Loan.put(“tenure”, 18);

Return loan;

}

}

**Application.properties**

server.port=8081

spring.application.name=loan-service

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

**Output:**

**Json**

{

“number”: “H00987987972342”,

“type”: “car”,

“loan”: 400000,

“emi”: 3258,

“tenure”: 18

}

**4. API Gateway**

**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.cognizant</groupId>

<artifactId>api-gateway</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>api-gateway</name>

<description>API Gateway</description>

<packaging>jar</packaging>

<parent>

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

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

<version>3.2.4</version>

<relativePath/>

</parent>

<properties>

<java.version>17</java.version>

<spring-cloud.version>2023.0.1</spring-cloud.version>

</properties>

<dependencies>

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

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

</dependency>

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

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

</dependency>

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

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

</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>

**ApiGatewayApplication.java**

Package com.cognizant.gateway;

Import org.springframework.boot.SpringApplication;

Import org.springframework.boot.autoconfigure.SpringBootApplication;

Import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

@SpringBootApplication

@EnableDiscoveryClient

Public class ApiGatewayApplication {

Public static void main(String[] args) { SpringApplication.run(ApiGatewayApplication.class, args);

}

}

**GlobalRequestLogger.java**

Package com.cognizant.gateway.filter;

Import org.springframework.cloud.gateway.filter.GlobalFilter;

Import org.springframework.core.Ordered;

Import org.springframework.stereotype.Component;

Import org.springframework.web.server.ServerWebExchange;

Import reactor.core.publisher.Mono;

@Component

Public class GlobalRequestLogger implements GlobalFilter, Ordered {

@Override

Public Mono<Void> filter(ServerWebExchange exchange, org.springframework.cloud.gateway.filter.GatewayFilterChain chain) {

System.out.println(“Request URI: “ + exchange.getRequest().getURI());

Return chain.filter(exchange);

}

@Override

Public int getOrder() {

Return -1;

}

}

**Application.properties**

Server.port=9000

Spring.application.name=api-gateway

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

Spring.cloud.gateway.routes[0].id=account-service

Spring.cloud.gateway.routes[0].uri=lb://account-service

Spring.cloud.gateway.routes[0].predicates[0]=Path=/accounts/\*\*

Spring.cloud.gateway.routes[1].id=loan-service

Spring.cloud.gateway.routes[1].uri=lb://loan-service

Spring.cloud.gateway.routes[1].predicates[0]=Path=/loans/\*\*

**Output:**

**Account Microservice via API Gateway**

**Json**

{

“number”: “00987987973432”,

“type”: “savings”,

“balance”: 234343

}

**Loan Microservice via API Gateway**

**Json**

{

“number”: “H00987987972342”,

“type”: “car”,

“loan”: 400000,

“emi”: 3258,

“tenure”: 18

}