**Exercise 1: Create Microservices for Account and Loan**

**Objective:**

Develop two Spring Boot-based microservices: one for handling account details and another for loan details. Each service is to be run independently with its own port and endpoint.

**1. Account Microservice**

**Step-by-Step:**

1. Use [Spring Initializr](https://start.spring.io/).
2. Group: com.cognizant
3. Artifact: account
4. Dependencies:
   * Spring Boot DevTools
   * Spring Web
5. Generate and download the project ZIP.
6. Extract and place in D:\<employee\_id>\microservices\account
7. Open terminal, navigate to the account directory, run:
8. mvn clean package
9. Import the project into Eclipse.

**Controller Implementation:**

@RestController

@RequestMapping("/accounts")

public class AccountController {

@GetMapping("/{number}")

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

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

response.put("number", number);

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

response.put("balance", 234343);

return response;

}

}

**application.properties:**

server.port=8080

Run the application and access:

http://localhost:8080/accounts/00987987973432

**2. Loan Microservice**

**Steps:**

Repeat the same steps as above with:

* Artifact: loan
* Change the server port to avoid conflict.

**Controller Implementation:**

@RestController

@RequestMapping("/loans")

public class LoanController {

@GetMapping("/{number}")

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

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

response.put("number", number);

response.put("type", "car");

response.put("loan", 400000);

response.put("emi", 3258);

response.put("tenure", 18);

return response;

}

}

**application.properties:**

server.port=8081

Run and test:

http://localhost:8081/loans/H00987987972342

**Exercise 2: Create Eureka Discovery Server and Register Microservices**

**Objective:**

Setup Eureka Discovery Server and register both account and loan microservices with it.

**1. Eureka Discovery Server**

**Steps:**

1. Use Spring Initializr:
   * Group: com.cognizant
   * Artifact: eureka-discovery-server
   * Dependency: Eureka Server
2. Generate and extract the project.
3. Add the following annotation to the main class:

@EnableEurekaServer

@SpringBootApplication

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

1. Run and verify at:

http://localhost:8761

**2. Register Account and Loan Services**

**Dependencies (pom.xml):**

Add to both microservices:

<dependency>

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

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

</dependency>

**Main Class Annotation:**

@EnableDiscoveryClient

@SpringBootApplication

**application.properties (for account):**

spring.application.name=account-service

server.port=8080

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

**application.properties (for loan):**

spring.application.name=loan-service

server.port=8081

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

Run Eureka first, then start both services. Confirm registration at:

http://localhost:8761

**Exercise 3: Create a Spring Cloud API Gateway with Global Logging Filter**

**Objective:**

Create a Spring Cloud API Gateway that forwards requests to a registered greet-service and logs each request via a global filter.

**1. Greet Service**

**Steps:**

1. Use Spring Initializr:
   * Group: com.cognizant
   * Artifact: greet-service
   * Dependencies:
     + Spring Web
     + Eureka Discovery Client
2. Controller:

@RestController

@RequestMapping("/greet")

public class GreetController {

@GetMapping

public String greet() {

return "Hello World";

}

}

**application.properties:**

server.port=8082

spring.application.name=greet-service

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

Ensure it registers at Eureka server.

**2. API Gateway**

**Steps:**

1. Use Spring Initializr:
   * Artifact: api-gateway
   * Dependencies:
     + Spring Cloud Gateway
     + Eureka Discovery Client
2. 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

Test API Gateway forwarding:

http://localhost:9090/greet-service/greet

**3. Implement Global Logging Filter**

**LogFilter.java:**

@Component

public class LogFilter implements GlobalFilter {

private static final Logger logger = LoggerFactory.getLogger(LogFilter.class);

@Override

public Mono<Void> filter(ServerWebExchange exchange, GatewayFilterChain chain) {

logger.info("Incoming request: {}", exchange.getRequest().getURI());

return chain.filter(exchange);

}

}

Check the console logs in API Gateway when the greet endpoint is accessed.