**WEEK 05**

1. **Creating Microservices for Account and Loan**

**Project Structure**

microservices-root/

│

├── api-gateway/

├── discovery-server/

├── config-server/ (optional)

├── account-service/

└── loan-service/

**Eureka Discovery Server**

**discovery-server**

**pom.xml**

<dependency>

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

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

</dependency>

**Main Class**

public class DiscoveryServerApplication {

public static void main(String[] args) {

SpringApplication.run(DiscoveryServerApplication.class, args);

}

}

**application.yml**

server:

port: 8761

eureka:

client:

register-with-eureka: false

fetch-registry: false

**API Gateway**

**api-gateway**

**pom.xml**

<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>

**application.yml**

server:

port: 8080

spring:

application:

name: api-gateway

cloud:

gateway:

routes:

- id: account-service

uri: lb://account-service

predicates:

- Path=/accounts/\*\*

- id: loan-service

uri: lb://loan-service

predicates:

- Path=/loans/\*\*

eureka:

client:

service-url:

defaultZone: http://localhost:8761/eureka/

**Account Service**

**Entity**

public class Account {

private Long id;

private String customerName;

private String accountNumber;

private String accountType;

}

**Repository**

public interface AccountRepository extends JpaRepository<Account, Long> {}

**Controller**

public class AccountController {

private AccountRepository repository;

public List<Account> getAll() {

return repository.findAll();

}

public Account create(@RequestBody Account account) {

return repository.save(account);

}

}

**application.yml**

server:

port: 8081

spring:

application:

name: account-service

datasource:

url: jdbc:h2:mem:testdb

driver-class-name: org.h2.Driver

username: sa

password:

jpa:

hibernate:

ddl-auto: update

eureka:

client:

service-url:

defaultZone: http://localhost:8761/eureka/

**Loan Service**

**Entity**

public class Loan {

private Long id;

private String loanNumber;

private String customerName;

private Double amount;

}

**Repository & Controller**

public interface LoanRepository extends JpaRepository<Loan, Long> {}

public class LoanController {

private LoanRepository repository;

public List<Loan> getAll() {

return repository.findAll();

}

public Loan create(@RequestBody Loan loan) {

return repository.save(loan);

}

}

**application.yml**

server:

port: 8082

spring:

application:

name: loan-service

datasource:

url: jdbc:h2:mem:testdb

driver-class-name: org.h2.Driver

username: sa

password:

jpa:

hibernate:

ddl-auto: update

eureka:

client:

service-url:

defaultZone: http://localhost:8761/eureka/

**Gateway Routing in Action**

* Access accounts: http://localhost:8080/accounts
* Access loans: http://localhost:8080/loans