**Create microservices for account and loan**

**TECHNOLOGIES USED**

* Spring Boot (for microservices)
* Spring Cloud Gateway (API Gateway)
* Eureka Server (Service Registry)
* Spring Web
* Spring Boot DevTools
* Spring Boot Actuator (optional for monitoring)
* VSCode + Java Extension Pack

**STEP 1: CREATE SPRING BOOT PROJECTS**

Use <https://start.spring.io/> or VSCode Spring Initializr extension.

Generate **4 projects** separately:

**API Gateway**

**Dependencies**:

* Spring Cloud Gateway
* Eureka Discovery Client
* Spring Boot DevTools
* Spring Web

**Service Registry (Eureka Server)**

**Dependencies**:

* Eureka Server
* Spring Boot DevTools
* Spring Web

**Account Service**

**Dependencies**:

* Spring Web
* Eureka Discovery Client
* Spring Boot DevTools
* Spring Boot Actuator (optional)

**Loan Service**

**Dependencies**:

* Spring Web
* Eureka Discovery Client
* Spring Boot DevTools
* Spring Boot Actuator (optional)

**CONFIGURE EUREKA SERVER**

**service-registry/src/main/resources/application.yml**

server:

port: 8761

eureka:

client:

register-with-eureka: false

fetch-registry: false

**ServiceRegistryApplication.java**

@SpringBootApplication

@EnableEurekaServer

public class ServiceRegistryApplication {

public static void main(String[] args) {

SpringApplication.run(ServiceRegistryApplication.class, args);

}

}

**CONFIGURE ACCOUNT & LOAN SERVICE**

**Both application.yml (Account and Loan):**

server:

port: 8081 # Account or 8082 # Loan

spring:

application:

name: account-service # or loan-service

eureka:

client:

service-url:

defaultZone: http://localhost:8761/eureka

**Controller Example for Account**

@RestController

@RequestMapping("/account")

public class AccountController {

@GetMapping("/info")

public String getInfo() {

return "Account Service";

}

}

**Controller Example for Loan**

@RestController

@RequestMapping("/loan")

public class LoanController {

@GetMapping("/info")

public String getInfo() {

return "Loan Service";

}

}

**CONFIGURE API GATEWAY**

**application.yml**

server:

port: 8080

spring:

application:

name: api-gateway

cloud:

gateway:

routes:

- id: account-service

uri: lb://account-service

predicates:

- Path=/account/\*\*

- id: loan-service

uri: lb://loan-service

predicates:

- Path=/loan/\*\*

eureka:

client:

service-url:

defaultZone: http://localhost:8761/eureka

**RUN THE SERVICES**

**Order to run the services in VSCode Terminal (or using Spring Boot Dashboard):**

1. service-registry
2. account-service
3. loan-service
4. api-gateway

**TEST THE ENDPOINTS**

Use browser or Postman:

* http://localhost:8080/account/info → Account Service
* http://localhost:8080/loan/info → Loan Service

**: REGISTER ALL SERVICES**

Each microservice automatically registers to Eureka and is discoverable via Gateway.

You can open Eureka Dashboard at:

http://localhost:8761