Create Eureka Discovery Server and register Microservices Project - Hands-on Documentation

# 1. Overview

This project demonstrates the implementation of a Spring Boot Microservices architecture with service discovery, centralized routing via Spring Cloud Gateway, and basic inter-service communication. The following microservices are developed and integrated: Account Service, Loan Service, Greet Service, API Gateway with Global Filter, and Eureka Discovery Server.

# 2. Microservices Setup

Each microservice is configured to run on a different port:

* • Account - Port 8080
* • Loan - Port 8081
* • Greet Service - Port 8082
* • API Gateway - Port 9090
* • Eureka Discovery Server - Port 8761

# 3. API Gateway and Global Filter

The API Gateway is set up using Spring Cloud Routing Gateway. A global filter is also implemented to log request details and enhance visibility across microservices.

Key classes:

* • GlobalFilter: Implements GlobalFilter interface
* • GatewayFilterChain: Used for request filtering

# 4. Routes Available

The following routes are accessible through the API Gateway:

* • /account/\*\* → Routes to Account Service
* • /loan/\*\* → Routes to Loan Service
* • /greet-service/\*\* → Routes to Greet Service

# 5. Discovery Client Usage

The @EnableDiscoveryClient annotation is used in the main class of each service (including API Gateway) to enable service registration with Eureka.

# 6. Sample Controllers

Each service includes a basic REST controller to verify route configuration.

* • AccountController - returns account-related response
* • LoanController - returns loan-related response
* • GreetController - returns greeting message

# 7. Testing the Routes

Test the routes using curl, Postman, or any REST client:

* • http://localhost:9090/account/accounts/{Acc-no}
* • http://localhost:9090/loan/loans/{Loan-no}
* • http://localhost:9090/greet-service/greet





