Project Overview

You have provided a distributed system composed of four main components built with Spring Boot and Spring Cloud:

1. **eureka-discovery-server**: Acts as the service registry where other microservices register themselves.
2. **account-service**: A microservice for handling account-related operations.
3. **loan-service**: A microservice for handling loan-related operations.
4. **api-gateway**: A single entry point that routes external requests to the appropriate microservice.

This setup allows for a scalable and resilient architecture.

How to Run the System

To run this application, you must start the services in a specific order. Each command should be run from the root directory of its respective module (e.g., eureka-discovery-server/, account/, etc.).

1. Start the Eureka Discovery Server

This is the central registry and must be started first.

- **Directory:** eureka-discovery-server/
- **Command:** mvn spring-boot:run
- **Port:** 8761

You can verify it's running by visiting http://localhost:8761 in your browser. You should see the Eureka dashboard.

2. Start the Microservices

These services can be started in any order after the Eureka server is running.

- **Account Service:**
  - **Directory:** account/
  - **Command:** mvn spring-boot:run
  - **Port:** 8081
- **Loan Service:**
  - **Directory:** loan/
  - **Command:** mvn spring-boot:run
  - **Port:** 8082

After starting, you should see ACCOUNT-SERVICE and LOAN-SERVICE listed as registered instances on the Eureka dashboard.

3. Start the API Gateway

This is the final component to start. It will handle and route all incoming requests.

- **Directory:** api-gateway/
- **Command:** mvn spring-boot:run
- **Port:** 8080

API Endpoints (via API Gateway)

All requests should now go through the API Gateway on port 8080.

1. Account Service Endpoint

- **URL:** http://localhost:8080/account
- **Method:** GET
- **Description:** Routes the request to the account-service.

Example Request:

```
curl http://localhost:8080/account
```

Expected Output:

The gateway will forward the request, and you will receive the response from the account-service:

```
Account Service is Up!
```

2. Loan Service Endpoint

- **URL:** http://localhost:8080/loan
- **Method:** GET
- **Description:** Routes the request to the loan-service.

Example Request:

```
curl http://localhost:8080/loan
```

Expected Output:

The gateway will forward the request, and you will receive the response from the loan-service:

```
Loan Service is Up!
```

Additionally, for every request made to the API Gateway, the RequestLoggingFilter will print the request path to the console where the API Gateway is running. For example:

```
Incoming request: /account
Incoming request: /loan
```