Customer Management Service:

Running the Service:

Clone the repository containing the Customer Management Service code.

Navigate to the project directory and run the service using a command like ‘ ./mvnw spring-boot:run ‘(assuming Maven wrapper is included).

Ensure that the service successfully registers itself with the Eureka server.

Sample URLs:

Add Customer: POST

http://localhost:9091/customers

Get All Customers: GET
http://localhost:9091/customers/all

Get Single Customer Details: GET
http://localhost:9091/customers/{customerId}

Update Customer Details: PUT
http://localhost:9091/customers/{customerId}

Delete Customer: DELETE
http://localhost:9091/customers/{customerId}

Account Management Service:

Running the Service:

Clone the repository containing the Account Management Service code.

Navigate to the project directory and run the service using a command like ‘./mvnw spring-boot:run’ (assuming Maven wrapper is included).

Ensure that the service successfully registers itself with the Eureka server.

Sample URLs:

Add Account: POST

http://localhost:9092/account/{customerId}

Add Money to Account: PUT

http://localhost:9092/account/add-money/{accountId}?customerId={customerId}&amount={accountId}

Withdraw Money from Account: PUT

http://localhost:9092/account/withdraw-money/{accountId}?customerId={customerId}&amount={accountId}

Get Account Details: GET

http://localhost:9092/account/{accountId}

Delete Account: DELETE

http://localhost:9092/account/{accountId}

API Gateway:

Running the Gateway:

Clone the repository containing the API Gateway code.

Navigate to the project directory and run the service using a command like' ./mvnw spring-boot:run' (assuming Maven wrapper is included).

Ensure that the service successfully registers itself with the Eureka server.

Sample URLs:

Access Customer Management Service: http://localhost:9093/customer/{endpoint}

Access Account Management Service: http://localhost:9093/account/{endpoint}

Eureka Server:

Running the Server:

Clone the repository containing the Eureka Server code.

Navigate to the project directory and run the service using a command like ‘ ./mvnw spring-boot:run ‘(assuming Maven wrapper is included).

The Eureka server should start and be accessible at http://localhost:8761.

Sample URL:

Eureka Dashboard: http://localhost:8761

Centralized Configuration Management:

Setting up Configuration Server:

Clone the repository containing the Configuration Server code.

Configure the server to fetch configuration files from a Git repository or any other source.

Start the Configuration Server using a command like ‘./mvnw spring-boot:run’.

The server should register itself with Eureka.

Sample URL :

Configuration Server Endpoint: http://localhost:9095

To ensure proper execution and functionality of the microservices architecture, follow this order:

1. **Eureka Server**: Start the Eureka server first to allow service registration and discovery.

2. **Centralized Configuration Management Server**: Start the configuration server next to provide configurations to other microservices.

3. **Customer Management Service**: Start the customer management service so that it can register itself with Eureka and fetch necessary configurations.

4. **Account Management Service**: Start the account management service after the customer service, ensuring it registers with Eureka and fetches configurations.