**BANKING APPLICATION**

Service Registry

Customer service

Authentication Service

Debit Service

Transfer Service

Client

API

Gateway

***Architecture***

The Banking project is aimed to give the simulation of how the microservices work and how we can achieve microservice architecture in spring boot.

The project contains following microservices working together.

1. Customer Microservice
2. Debit Credit Microservice
3. Transfer Microservice
4. Service Registry
5. Authentication Microservice
6. Gateway Microservice

Each of these microservices are described below in details.

1. **Customer Service**

This microservice is used to create, update, delete and view the customers of the Banking application.  
It contains the following endpoints.

1. Create

<http://localhost:8085/customer/create>

This endpoint is used to create a new user. The endpoint is not secured because anyone without the token should be able to access this endpoint for creating new user.

1. Update

<http://localhost:8085/customer/update>

This endpoint is used to update the existing user’s details. Here we must pass the customer object in request body and bearer token in authentication.

1. Delete

<http://localhost:8085/customer/delete>/{username}

This endpoint is used to delete the customer. We need to pass the username here as path variable. Also, we need the bearer token for authorization.

1. View

<http://localhost:8085/customer/view>/{username}

This endpoint is used to view a particular Customer. We must pass the username as a path variable while calling this endpoint. Also, we need the bearer token for authorization.

1. View All

<http://localhost:8085/customer/viewAll>

This endpoint is used to view all the customers that are present in the database. It also needs bearer token for authorization.

1. View By Account Number

[http://localhost:8085/customer/viewByAccount/{accountNumber}](http://localhost:8085/customer/viewByAccount/%7baccountNumber%7d)

This endpoint is used to get the customer data by using their account number. This also needs authorization.

1. **Debit Credit Microservice**

This debit credit service is used to debit and credit money into the customers account. We have to pass the account number along with the amount and the amount will be credited or debited to the user’s account.

The service has following endpoints.

1. Debit

<http://localhost:8085/debitcredit/debit>

This endpoint is used to debit the money into customer’s account. It needs the authorization token i.e., the bearer token.

1. Credit

<http://localhost:8085/debitcredit/credit>

This endpoint is used to credit the money into customer’s account. It needs the authorization token i.e., the bearer token.

1. **Transfer Money Service**

This service is used to transfer money from one account to another account. This service takes the from Account and to Account and amount as parameter. Then it transfers the money to the respective account.

The following endpoint is used to access this service - <http://localhost:8085/transfermoney/transfer>

1. **Service Registry**

This service acts as a service registry. All the microservices are registered here and it stores the information about each microservice.

The service is accessible at the following endpoint - <http://localhost:8080/eureka>

1. **Authentication service**

This service is used to authenticate each request that is going through the Gateway service. This service generates the JWT Token for validation. This token is passed in the authorization token as bearer token in each request.

The service has following endpoints.

1. Token

<http://localhost:8085/token>

This endpoint is used to create a new JWT token. It takes the customer object in which the username and password is defined. If the username and password are correct then only token is generated otherwise we get 403 Forbidden error.

1. Validate

<http://localhost:8085/validate>

This endpoint is used to validate the tokens. It needs the token as a query parameter.

1. **Gateway Service**

The gateway service is used to create an API gateway for our application. All requests will go through this gateway. This will be endpoint that is accessible to the Client or browser.

The service is hosted at – <http://localhost:8085>