Retail Banking System

MFPE (POD - 1) ADM21JF010

Mentor : Karthikeyan Balakrishnan

Trainer : Saurabh Kumar Sharma

Coach : Madhumathi Malarvannan

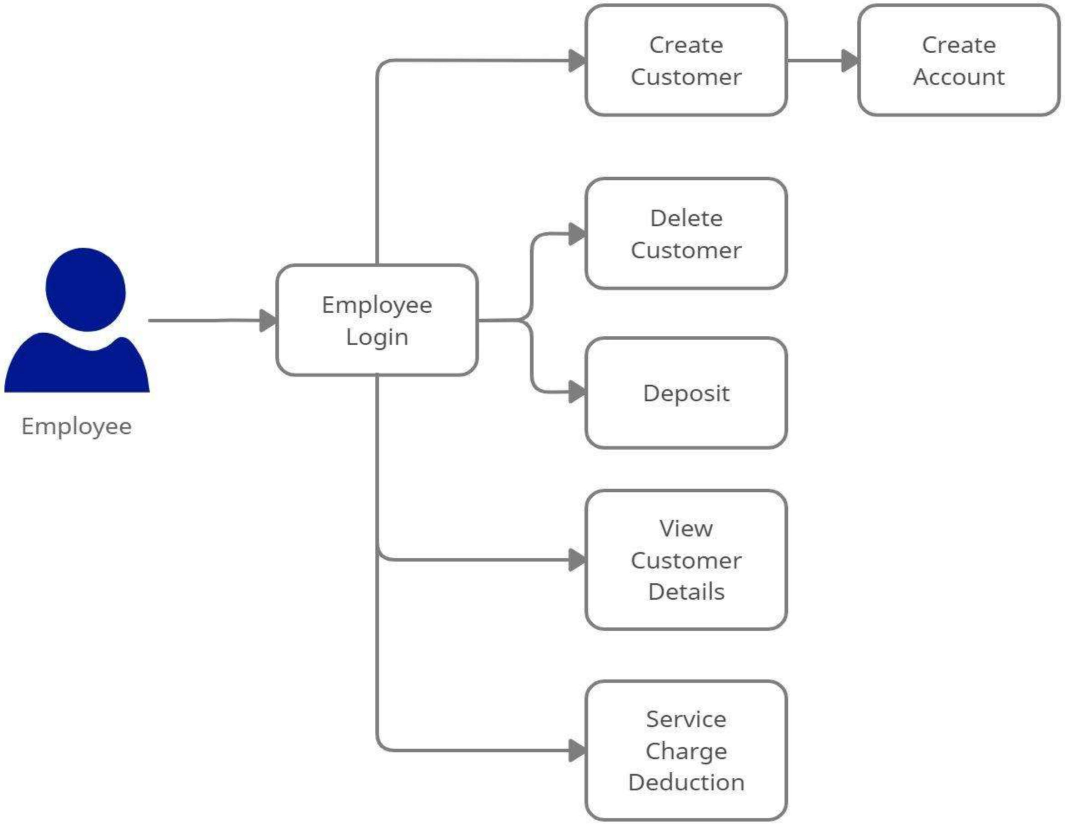
Team Members

- Akash Yadav(2064581)
- Sharath Nair(2063655)
- Brahmanaidu V(2063357)
- Sekhar Kuppili (2063130)
- Yashraj Sisode (2062867)

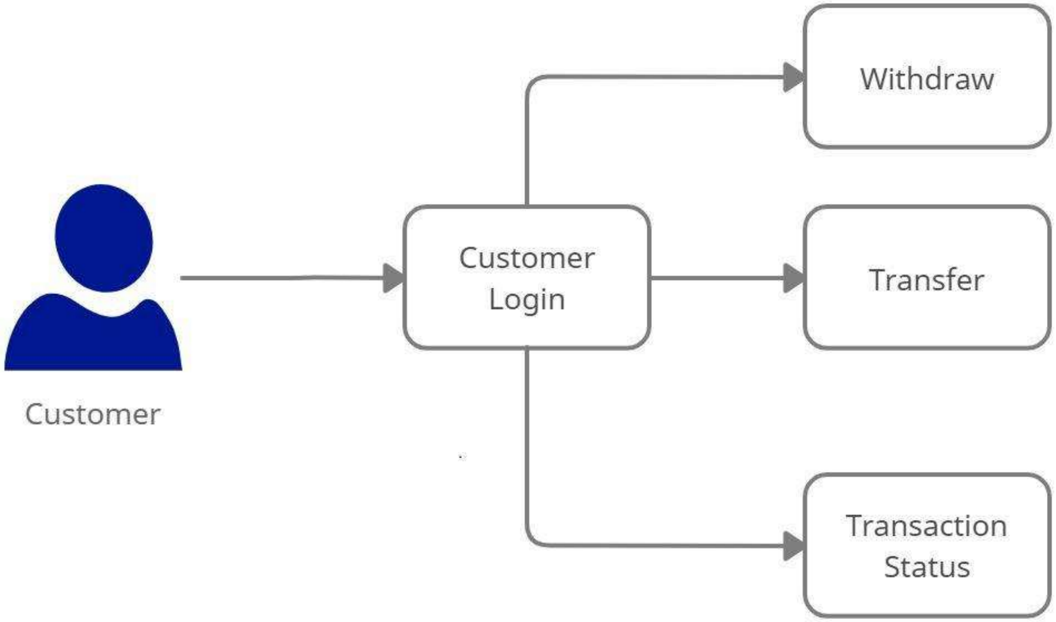
Introduction

- The Retail Banking System provides **Microservice Architecture based software solution.**
- It provides **high Availability operations** for their **client & bank employees** without disruption.
- **Different Microservices combines together** to perform together as a **Unified application.**

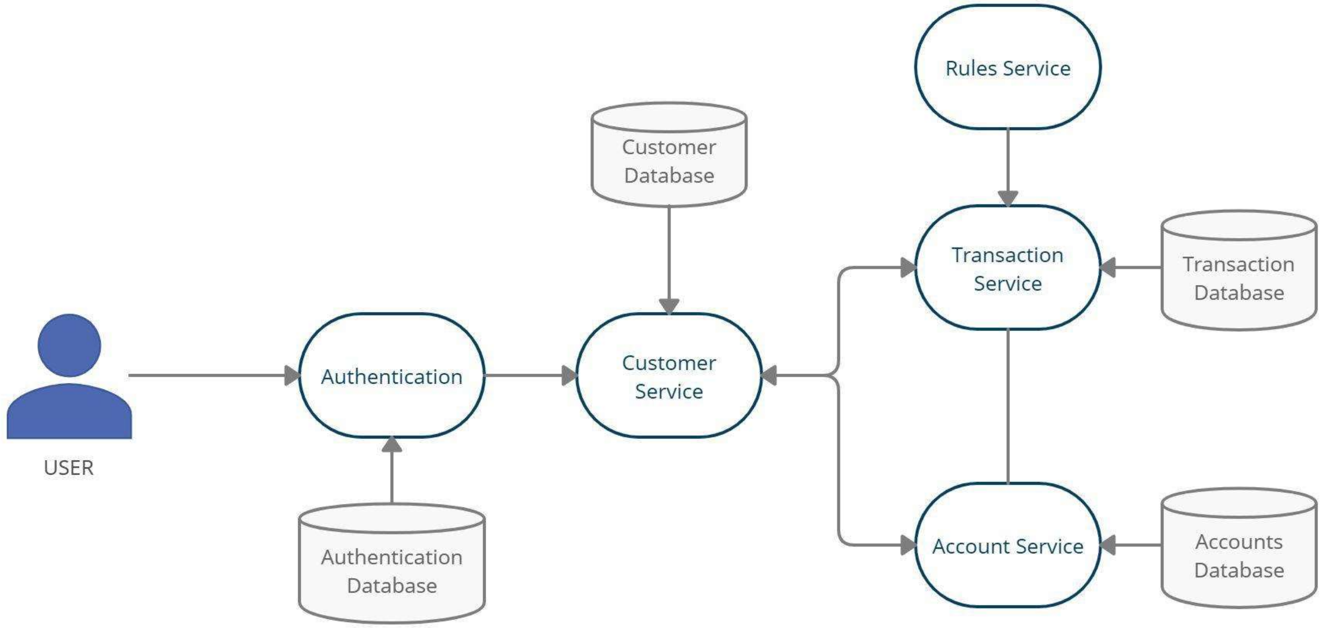
Architecture Diagram for Employee



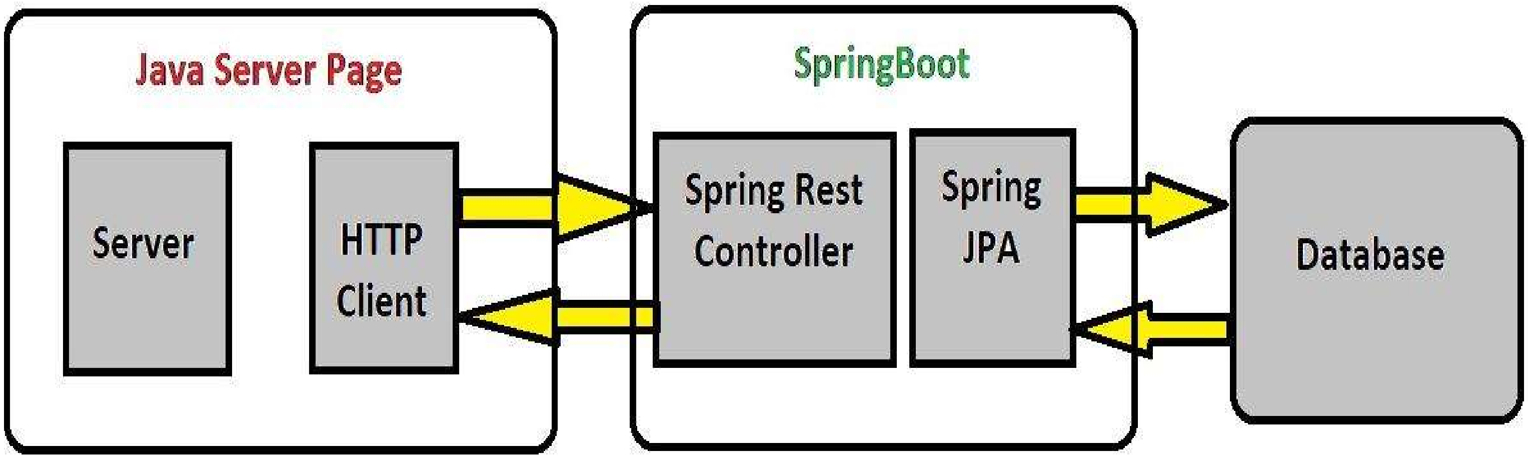
Architecture Diagram for Customer



WorkFlow:


Rules Service
Customer Database
Transaction Service
Transaction Database
USER
Authentication
Customer Service
Account Service
Accounts Database
Authentication Database

Client - Server Architecture



Technologies Used

- Front End : **Java Server Page**
- Backend : **Java, Spring Boot, Restful-Services**
- Database : **H2 - database**
- Tools : **Spring Tool Suite, Swagger, Postman, AWS**

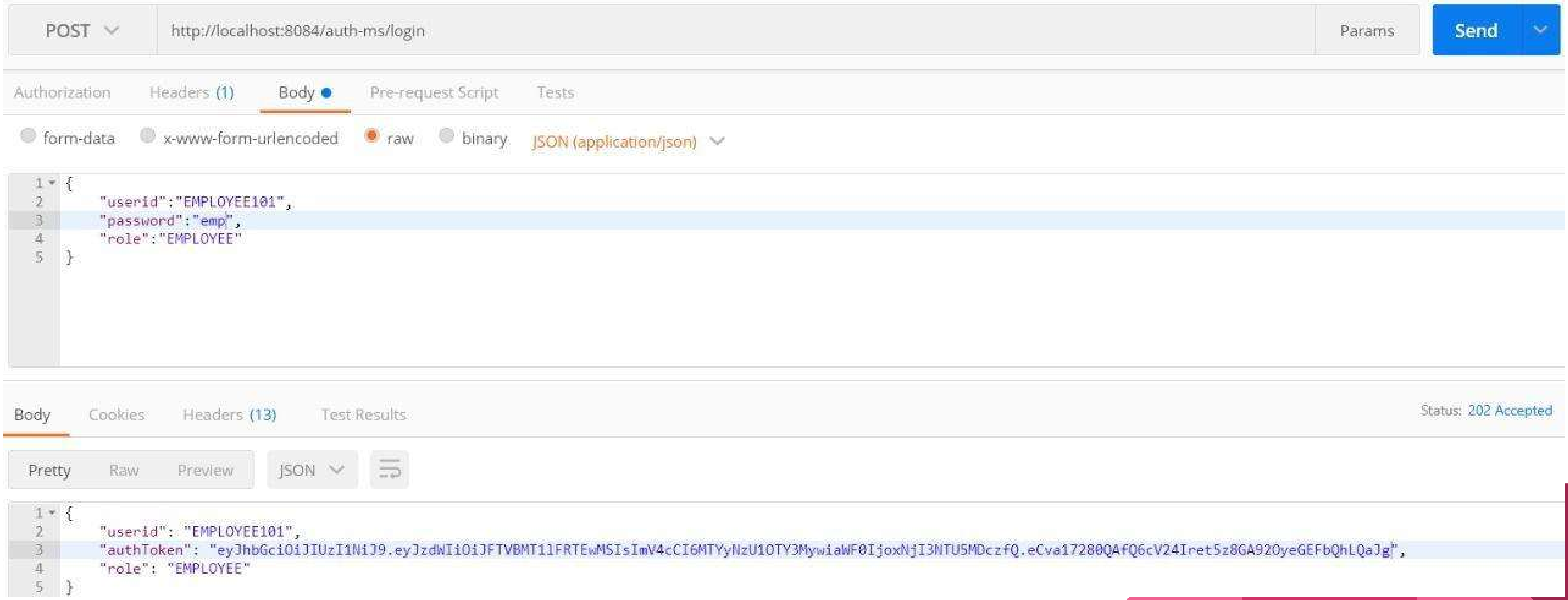
Working Microservices:

- Authorization microservice
- Customer microservice
- Account microservice
- Transaction microservice
- Rules microservice

Authentication Microservice

- Authentication Microservice will perform **operations like:**
  - **Login &**
  - **Logout**
- Authentication will provide the **JWT** token after user’s validation.
- Providing the **authorization** based on **user's login**

Authorization Microservice - Sample request



```
{
    "userid":"EMPLOYEE101",
    "password":"emp",
    "role":"EMPLOYEE"
}
```

Status: 202 Accepted

```
{
    "userid": "EMPLOYEE101",
    "authToken": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJFTVBMT1lFRTEwMSIsImV4cCI6MTYyNzU1OTY3MywiaWF0IjoxNjI3NTU5MDczfQ.eCva17280QAfQ6cV24Iret5z8GA92OyeGEFbQhLQaJg",
    "role": "EMPLOYEE"
}
```

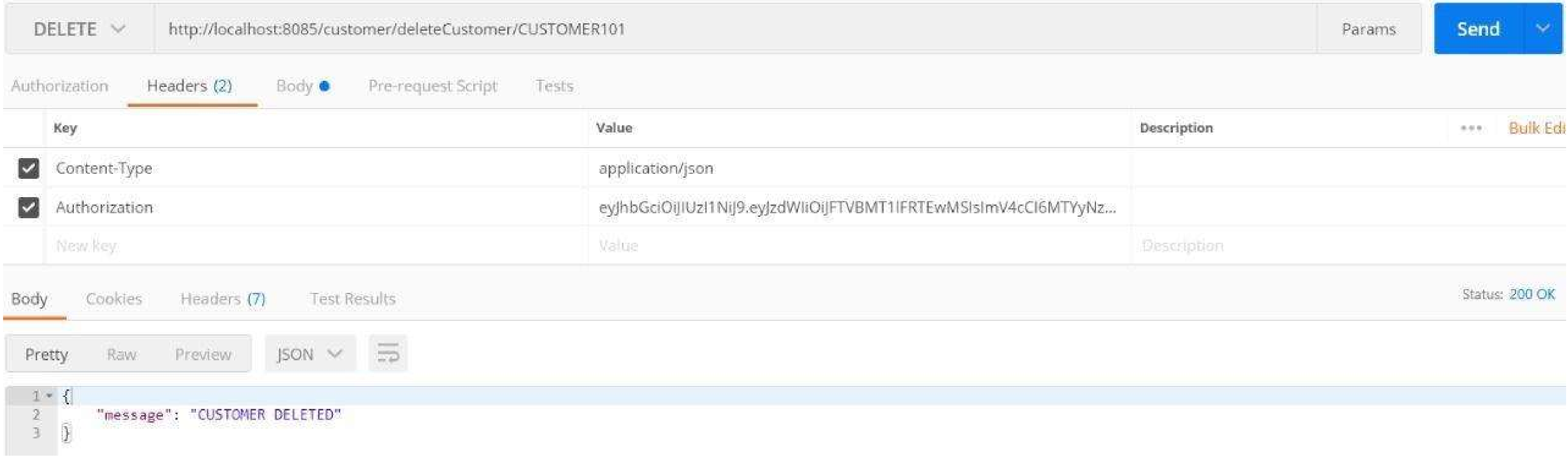
Customer Microservice

❖ Customer microservice will performs **operations like :**

- Creating Customer Details
- Fetching Customer Details
- Deletion of Customer

❖ It even interacts with authentication and account microservices for the purpose of sharing details.

Customer Microservice - Sample request



```
DELETE http://localhost:8085/customer/deleteCustomer/CUSTOMER101
```

	Key	Value	Description
☑	Content-Type	application/json	
☑	Authorization	eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJFTVBMT1lFRTEwMSIsImV4cCI6MTYyNz...	
	New key	Value	Description

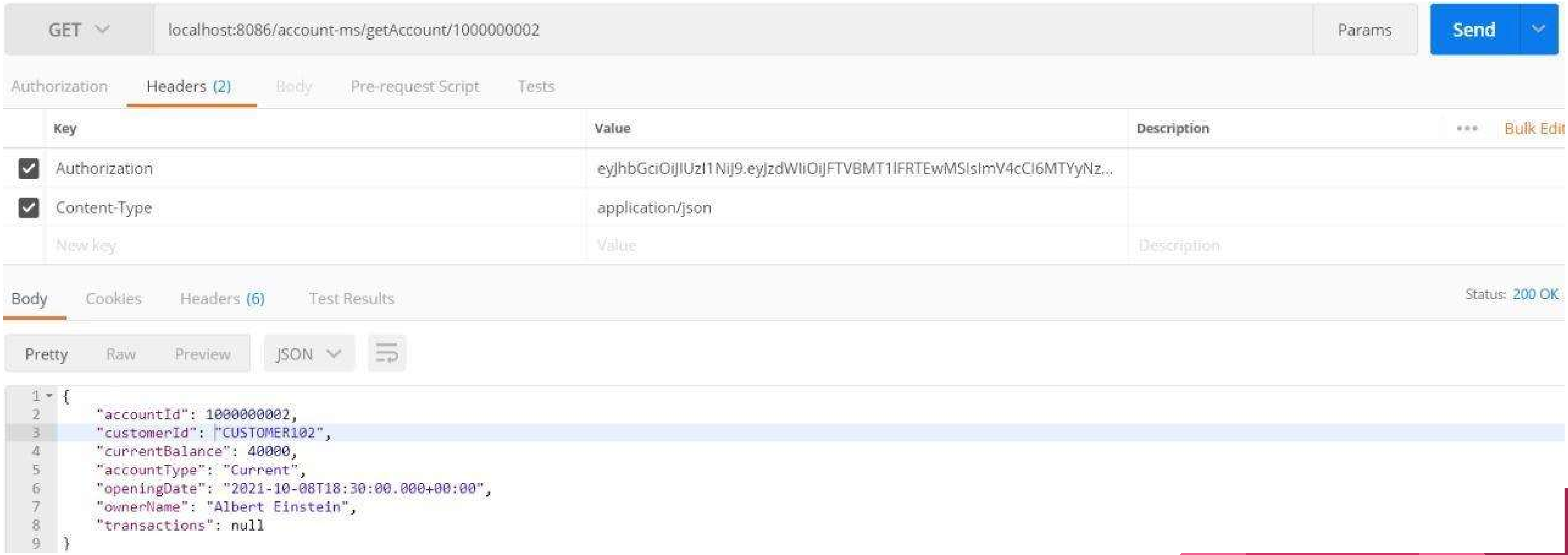
Status: 200 OK

```
{
    "message": "CUSTOMER DELETED"
}
```

Account Microservice

- Account Microservice will perform **following operations:**
  - **Customer Account Creation**
  - **Fetching Account Details** for Each Customer
  - **Deposit and Withdraw from customer's account.**
- **Interacts** with **transaction microservice** for **validating the account details and transferring the amount.**

Account Microservice - Sample request



GET localhost:8086/account-ms/getAccount/1000000002

	Key	Value	Description
☑	Authorization	eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJFTVBMT1lFRTEwMSIsImV4cCI6MTYyNz...	
☑	Content-Type	application/json	
	New key	Value	Description

Status: 200 OK

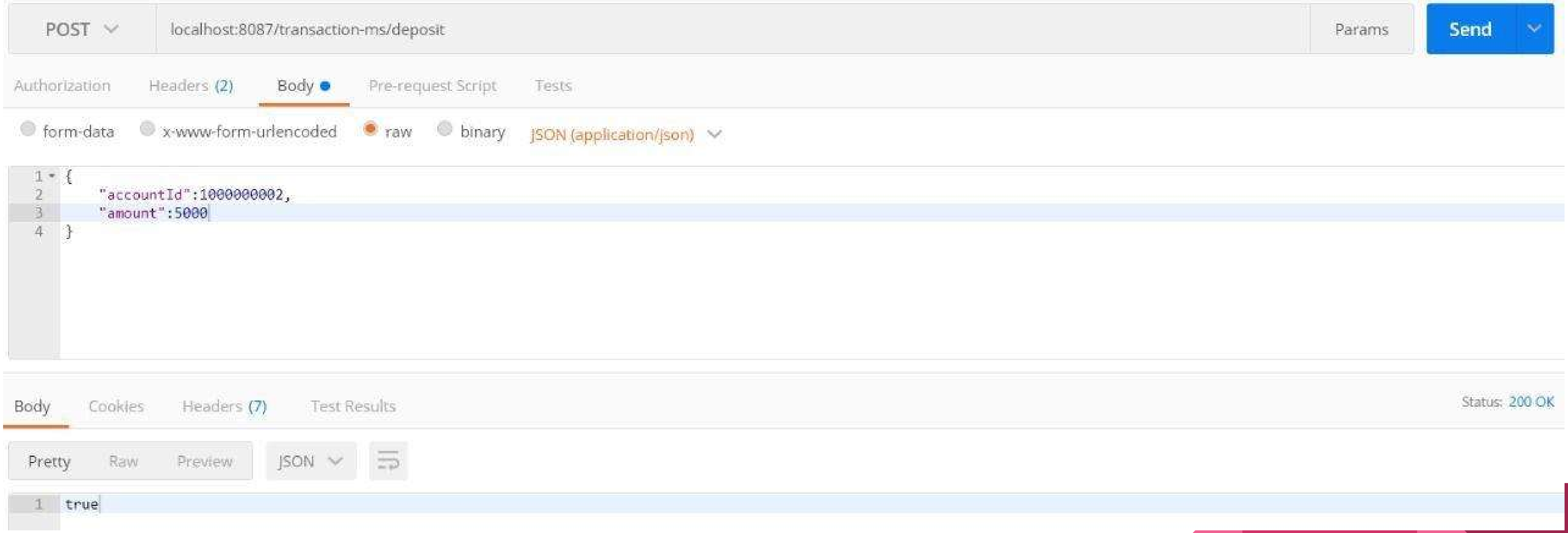
```
{
    "accountId": 1000000002,
    "customerId": "CUSTOMER102",
    "currentBalance": 40000,
    "accountType": "Current",
    "openingDate": "2021-10-08T18:30:00.000+00:00",
    "ownerName": "Albert Einstein",
    "transactions": null
}
```

Transaction Microservice

- Transaction Microservices will perform **operations like:**
  - **Fetching transactions history** for individual account.
  - **Validating the account details** for individual customer.
  - **Checks minimum balance** for the transfer and withdraw account.

- Transactions microservice will **interact** with **Account microservice** to actually **complete the transactions within the accounts.**

Transaction Microservice - Sample request



POST localhost:8087/transaction-ms/deposit

Body: raw, JSON (application/json)

```
{
    "accountId":1000000002,
    "amount":5000
}
```

Status: 200 OK

JSON

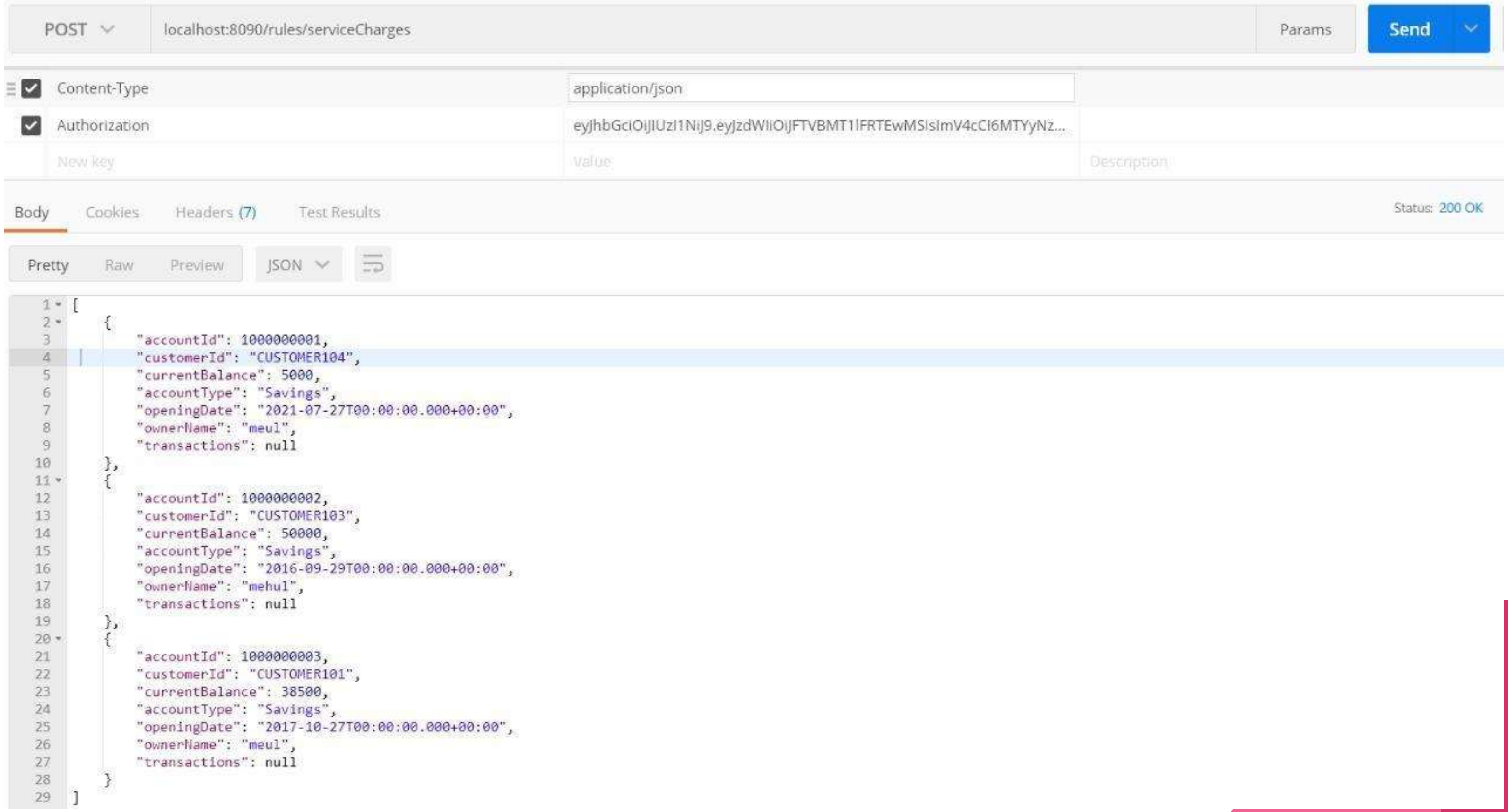
```
true
```

Rules Microservice

- ❖ Rules Microservice will be responsible for **evaluating rules:**
  - **Minimum Account Balance &**
  - **Service Charge Deduction**

- ❖ **Interacts** with **account** and **transaction microservice** for checking rules while **transferring** and **withdrawing** from **source account.**

Rules Microservice - Sample request



POST localhost:8090/rules/serviceCharges

☑ Content-Type: application/json

☑ Authorization: eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJFTVBMT1lFRTEwMSIsImV4cCI6MTYyNz...

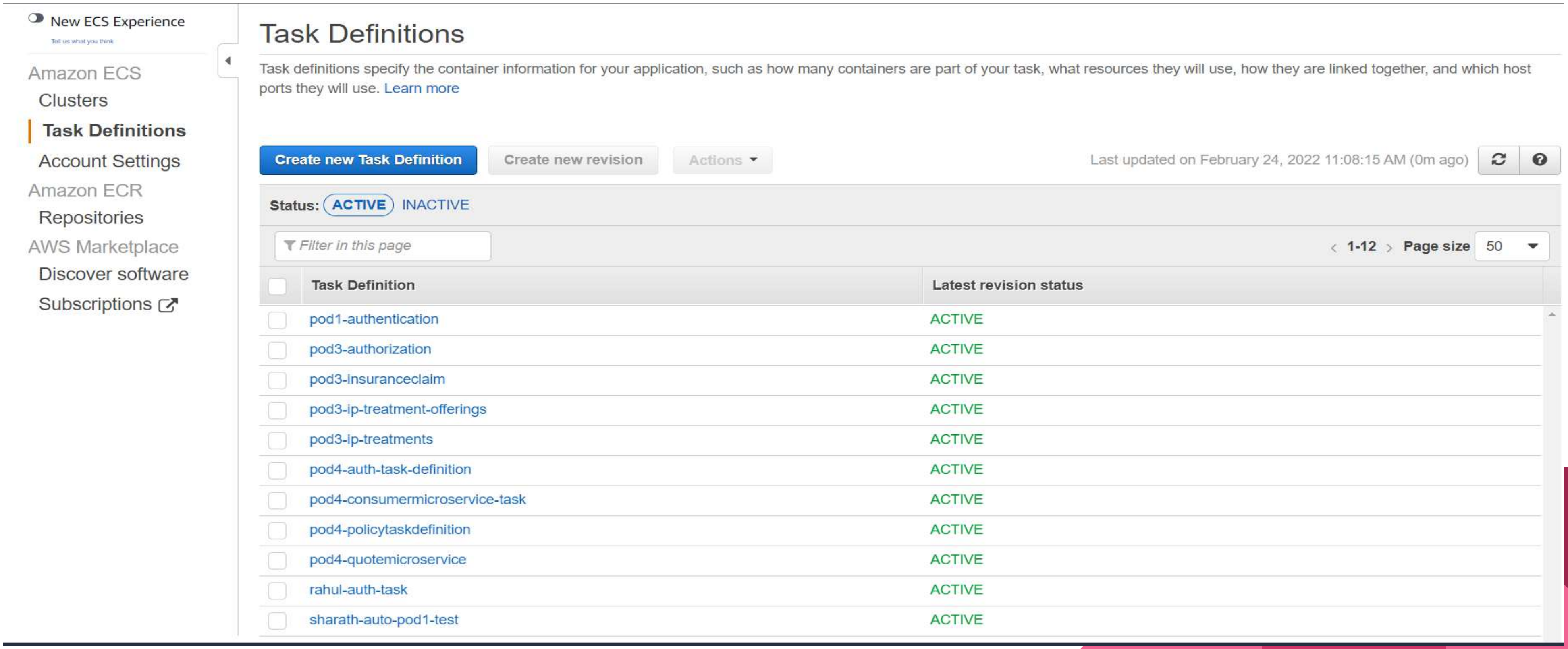
Status: 200 OK

JSON

```
[
    {
        "accountId": 1000000001,
        "customerId": "CUSTOMER104",
        "currentBalance": 5000,
        "accountType": "Savings",
        "openingDate": "2021-07-27T00:00:00.000+00:00",
        "ownerName": "meul",
        "transactions": null
    },
    {
        "accountId": 1000000002,
        "customerId": "CUSTOMER103",
        "currentBalance": 50000,
        "accountType": "Savings",
        "openingDate": "2016-09-29T00:00:00.000+00:00",
        "ownerName": "mehul",
        "transactions": null
    },
    {
        "accountId": 1000000003,
        "customerId": "CUSTOMER101",
        "currentBalance": 38500,
        "accountType": "Savings",
        "openingDate": "2017-10-27T00:00:00.000+00:00",
        "ownerName": "meul",
        "transactions": null
    }
]
```

AWS Deployment

Task Definition



Task Definitions

Task definitions specify the container information for your application, such as how many containers are part of your task, what resources they will use, how they are linked together, and which host ports they will use. Learn more

Last updated on February 24, 2022 11:08:15 AM (0m ago)

□	Task Definition	Latest revision status
□	pod1-authentication	ACTIVE
□	pod3-authorization	ACTIVE
□	pod3-insuranceclaim	ACTIVE
□	pod3-ip-treatment-offerings	ACTIVE
□	pod3-ip-treatments	ACTIVE
□	pod4-auth-task-definition	ACTIVE
□	pod4-consumermicroservice-task	ACTIVE
□	pod4-policytaskdefinition	ACTIVE
□	pod4-quotemicroservice	ACTIVE
□	rahul-auth-task	ACTIVE
□	sharath-auto-pod1-test	ACTIVE

Video Sample:

Link:

https://drive.google.com/file/d/1yJYUHcQva0kM8FZU0zRTBUjw5cChOCWD/view?usp=sharing

Thank You ...!