# **Customer Service Portal**

## **Problem Statement**

The problem statement contains below three parts

1. Build a Spring Micro-services application with three different services.
2. Implement a Jenkins pipeline to extract this application from GIT, build it and deploy it.
3. Implement services monitoring.

## **Solution:**

Customer Service portal has three main services.

PFB the structure:

**Customer Service**

**DB Service**

**Admin Service**

### **Customer Service:**

This service exposes four rest end points. All the endpoints are secured with basic authentication.

Please find below sample API’s details:

|  |  |  |  |
| --- | --- | --- | --- |
| API Name | API URL | Sample Request | Sample Response |
| Retrieve Customer | (GET)<http://192.168.99.101:8081/customer-service/customer/1> |  | {  "firstName": "Satya",  "lastName": "Sudheer",  "password": "pass123",  "accounts": [          {              "type": "Saving",              "balance": 3000.22          }      ],  "dateOfBirth": "1992-06-06"  } |
| Create account | (POST)<http://192.168.99.101:8081/customer-service/account?customerId=1> | {  "type":"Saving",  "balance":1000  } | Created |
| Update Balance | (PUT)http://192.168.99.101:8081/customer-service/account/3?balance=3000.22 |  | {      "type": "Saving",      "balance": 3000.22  } |
| Delete account | (DELETE)http://192.168.99.101:8081/customer-service/account/2 |  | Deleted |

Sample API details can be verified in Swagger UI.

PFB Swagger URL : [http://192.168.99.101:8081/swagger-ui.html#](http://192.168.99.101:8081/swagger-ui.html)

### **Admin Service**

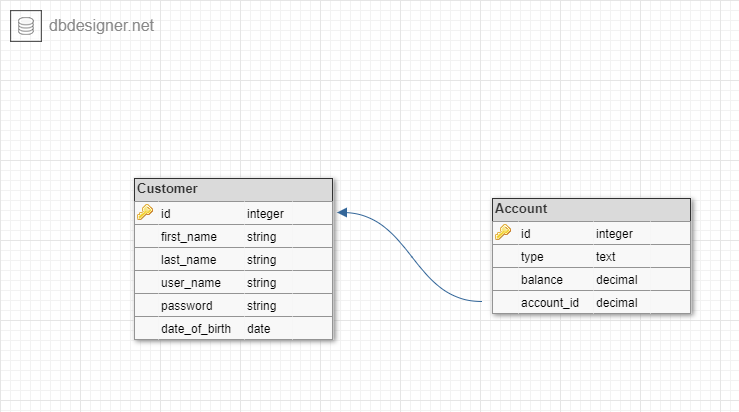
This service exposes four rest end points. Please find below sample the API’s details:

|  |  |  |  |
| --- | --- | --- | --- |
| API Name | API URL | Sample Request | Sample Response |
| Create Customer | (POST)http://192.168.99.101: 8083/ admin-service/customer | {  "firstName": "Satya",  "lastName": "Sudheer",  "password": "pass123",  "dateOfBirth": "1992-06-06"  } | {  "firstName": "Satya",  "lastName": "Sudheer",  "password": "pass123",  "dateOfBirth": "1992-06-06"  } |
| Retrieve all Customers | (GET)http://192.168.99.101: 8083/ admin-service/customers |  | [      {          "firstName": "Satya123",          "lastName": "Sudheer",          "password": "pass123",          "dateOfBirth": "1992-06-06"      },      {          "firstName": "Jai",          "lastName": "Henry",          "password": "jai123",          "dateOfBirth": "1992-06-10"      }  ] |
| Modify Customer | (PUT)http://192.168.99.101: 8083/ admin-service/customer/1 | {  "firstName":"Satya",  "lastName":"Kolipaka",  "password":"pass123",  "dateOfBirth":"1992-06-06"    } | {  "firstName":"Satya",  "lastName":"Kolipaka",  "password":"pass123",  "dateOfBirth":"1992-06-06"    } |
| Delete Customer | (DELETE)http://192.168.99.101: 8083/ admin-service/customer/1 |  | True |

### **DB Service**

Above services will be connected to DB-service and perform all the operations in Database.

The Database contains two tables. PFB DB Model:



### Technical Details:

All the three services are implemented using spring boot

1. Spring Boot 2.2.5
2. Java 1.8
3. H2 Database
4. Maven for Build
5. ControllerAdvice for exception handling (only customer service)
6. Swagger2 for API specification (only customer service)
7. Spring Security for basic authentication (only customer service)
8. Docker and Jenkins for CI/CD
9. Shell script in Jenkins job

### Deployment:

All three services are deployed as Docker containers in Docker toolbox using Jenkins CI/CD pipeline

GIT Hub

Deploy build files in VM using

Docker VM

192.168.99.101

Shell script

Jenkins Pipeline

Maven Build

* Jenkins pipeline URL: <http://192.168.99.101:8080/>

### Monitoring:

To monitoring all three services, we are configuring Prometheus and Grpahna (as a Docker images). Spring boot actuator end points will share the application level information

DB Service

Grpahna

Prometheus

Customer Service

Admin Service

* Prometheus URL: <http://192.168.99.101:9090/>
* Grpahna URL: <http://192.168.99.101:3000/>