# **Customer Service Portal**

**Total Time Spent: 5 hours**

## Functionality

As per given requirement, I have implemented below three REST API’s

1. Authenticate the customer

There is no security check for this endpoint. It is post endpoint having JSON body with username and password

1. Retrieve and update the customer details

There is basic authentication security check for these two endpoints.

1. Retrieve 3 customers out of 5, order by date of birth desc

There is basic authentication security check for this endpoint.

Please find below sample API’s details

|  |  |  |  |
| --- | --- | --- | --- |
| API Name | API URL | Sample Request | Sample Response |
| Authenticate customer | (POST)http://localhost:8080/info/authenticate | {  "userName":"satya123",  "password":"pass123"  } | Login Success |
| Retrieve Customer | (GET)http://localhost:8080/info/customer/1 |  | {  "firstName": "Satya",  "lastName": "Sudheer",  "password": "pass123",  "dateOfBirth": "1992-06-06"  } |
| Update Customer | (PUT)http://localhost:8080/info/customer/1 | {  "firstName": "Satya",  "lastName": "Sudheer",  "password": "pass123",  "dateOfBirth": "1992-06-06"  } | {  "firstName": "Satya",  "lastName": "Sudheer",  "password": "pass123",  "dateOfBirth": "1992-06-06"  } |
| Retrieve 3 customers out of 5, order by date of birth desc | (GET)http://localhost:8080/info/customers |  | [  {  "firstName": "jhon",  "lastName": "mh",  "password": "pass123",  "dateOfBirth": "1992-10-06"  },  {  "firstName": "sai",  "lastName": "sd",  "password": "pass123",  "dateOfBirth": "1992-09-06"  },  {  "firstName": "ramesh",  "lastName": "ch",  "password": "pass123",  "dateOfBirth": "1992-08-06"  }  ] |

For above security endpoints, Please provide basic authentication in request header.

Details for basic authentication

Username : admin

Password : password

**Note**: I have implemented two other API’s Add customer and Delete customer. These are my testing purpose, instead of writing Db queries.

Please fine below postman API collection for all API’s



You can also find API’ specification using Swagger UI url

<http://localhost:8080/swagger-ui.html#/>

## Techincal Details

Customer Service portal is developed by using spring boot

1. Spring Boot 2.2.5
2. Java 1.8
3. H2 Database
4. Maven for Build
5. Mocikto and MockMvc for unit testing
6. CntrollerAdvice for exception handling
7. Swagger2 for API specification
8. Spring Security for basic authentication

## Application Start

Build and Run the application using the following commands from the command prompt:

**mvn clean install**

**mvn spring-boot:run**

**Login H2 database**

* Open link <http://localhost:8080/h2-console/>
* Enter below details
  + Driver class : org.h2.Driver
  + JDBC URL : jdbc:h2:mem:testdb
  + User name : test
  + Password : password
* Run Below sample DML queries



## Open question:

I have one open question regarding security implementation.

I have implemented above endpoints (except authenticate endpoint) with basic authentication (username: admin & password: password) having the credentials hard coded in code.

I also thought about to implement authentication and authorization using OAuth 2.0. However, I could not implement due to the time constraint. Here I am providing the approach to implement token-based authentication

1. Import jjwt maven dependency in pom.xml
2. Create the configuration classes to generate and validate token using any **SignatureAlgorithm** provided by jwt dependency
3. Create filter on every request to use validate token passed by header
4. If token validation is success it will forward to original request