

Name : SAHANA BHAT

Date : 01/10/2023

Corptec Assignment - Documentation

Execution:

1. **Out of 9** Tasks mentioned in the question (code-challenge-requirement.pdf) I have completed **8** tasks apart from the Reactive Programming due to time constraints.
2. This documentation contains executing the project and you import the postman collection attached down below for testing the API's (I have mentioned the flow of API endpoints in the postman screenshot at the end of this documentation)
3. I have used MYSQL as the RDBMS and the respective configuration is in application.properties file, also attached SQL dump down below in the document
4. For Salesforce Integration I have created a salesforce test account, created the custom object called **CorpCustomer__c** which will be used to synch our application MySQL data. Note the respective authentication username, password, client secrets, and client id will be mentioned in CONSTANTS.JAVA in the com.corptech.com.config package.
5. I have sent the project in zip file also in the **EMAIL** and also find the same in the below githublink Github link: <https://github.com/sahanabhatb/CorpTechAssignment.git>
6. For Spring Boot Web application I have used Springboot with spring security/JWT authentication and authorization, JPA, MYSQL as DB instead of SQL Server, Salesforce Integration, etc.
7. Import the Spring boot Project into Eclipse (which I used) -> Maven Update Project then Right click on project and RUN As Maven build configuration with Goals:"Clean install", uncheck skip test -> then once build success then Run CorptechApplication.java as Java Application.
8. Then go to <http://localhost:8080/swagger-ui/> for swagger documentation, and go to last section for the detailed execution of the api's in the postman, also attached postman collection below



9. Following is the Project folder structure with controller, entity, repositories

```

v corptech (in corptec)
v src/main/java
v com.corptech.com
  > CorptechApplication.java
v com.corptech.com.config
  > Constants.java
  > CustomerConfig.java
  > SalesforceAuthenticator.java
v com.corptech.com.config.JWT
  > CORSFilter.java
  > ExceptionAdvice.java
  > JwtAuthenticationEntryPoint.java
  > JwtRequestFilter.java
  > JwtToken.java
  > WebSecurityConfig.java
v com.corptech.com.controller
  > AuthController.java
  > CustomerController.java
  > UserController.java
v com.corptech.com.model
  > Customer.java
  > JwtRequest.java
  > JwtResponse.java
  > User.java
v com.corptech.com.repository
  > CustomerRepository.java
  > SalesforceDataService.java
  > UserRepository.java
v com.corptech.com.service
  > CustomerService.java
  > JwtUserDetailsService.java
v src/main/resources
  > static
  > templates
  > application.properties
v src/test/java
v com.test.corptech
  > CorptechApplicationTests.java

```

10. Please find the mysql and dump and the postman collection to quickly import.

 **corptec.sql.txt**  **CorpTech.postman_c
ollection.json**

11. Once the project has been successfully run as java application open POSTMAN and check the following api's and the respective responses (Also attached the screenshot of the same at the end of this doc)

- a. Signup API to create new users => **POST /signup**
 - b. Login api to get access token => **POST /auhtenticate**
 - c. listing of customer => **GET /api/ customer** , Both admin and normal user can access
 - d. retrieve single customer => **GET /api/ customer /{Id}**, Both admin and normal user
 - e. create new customer => **POST /api/ customer** { request body in JSON } , ADMIN user
 - f. update customer => **PUT /api/ customer /{Id}** { request body in JSON } , ADMIN user
 - g. delete customer => **DELETE /api/ customer /{Id}** , ADMIN user
 - h. MySQL to Salesforce Data Synch API => **POST /api/customers/sync** , ADMIN user
- Request and Responses are in JSON.

Execution of API's Step by step output screenshots (POSTMAN) – PostmanCollection Attached

1. First need to create an user into database using the /signup api, the user-roles can be given as ADMIN or USER, If given Admin role then user can access all the api's whereas if given USER role then can access only GetAllCustomer and GetCustomerById apis only

The screenshot displays the Postman interface for a POST request. The top bar shows the collection 'CorpTech' and the endpoint 'signup'. The request method is 'POST' and the URL is 'http://localhost:8080/signup'. The 'Body' tab is selected, showing a JSON payload with the following fields: 'username' (sahana), 'password' (1234), 'name' (Sahana Bhat), 'emailid' (sahana@gmail.com), and 'role' (ADMIN). The 'Headers' tab is also visible, showing 11 headers. The 'Test Results' tab is selected, showing a single test result with the value 'true'.


```
1 {
2   "username": "sahana",
3   "password": "1234",
4   "name": "Sahana Bhat",
5   "emailid": "sahana@gmail.com",
6   "role": "ADMIN"
7 }
8 }
```


Body Cookies Headers (11) Test Results



Pretty Raw Preview Visualize JSON








```
1 true
```

2. After signup use the same username password to /authenticate to the app, which will provide the jwt_token that will be used to authenticate the other api's


CorpTech / authenticate&GetToken 



POST  http://localhost:8080/authenticate

Params Authorization Headers (8) **Body**  Pre-request Script Tests  Settings

 none  form-data  x-www-form-urlencoded  raw  binary  GraphQL **JSON** 


```
1 {
2   ... "username": "sahana",
3   ... "password": "1234"
4 }
```


Body Cookies Headers (14) Test Results  Status: 200 OK



Pretty Raw Preview Visualize **JSON**  

```
1 {
2   "token": "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJzYW5hbmEiLCJleHAiOiJlE20TYyNTQ1MjUsImhhdCI6MTY5NjIzNjUyNX0.1qwV3YHaV50bJSR07CpgiYiPPti9E-Ug4-6UvqwQTreA00v0r0onGuilin_Jom86_Pb1z2LCsA4-Rko7kiIQoQ"
3 }
```

Actually we need to Copy this token and use in below apis , but to make it automatically after hitting the authenticate the token will be stored automatically to the jwt_token global variable of the postman, the respective script is written in the Tests section of authenticate request. So no need to manually copy paste to other api headers. Just hit authenticate api and go to next api


CorpTech / authenticate&GetToken 


POST  http://localhost:8080/authenticate


Params Authorization Headers (8) Body  Pre-request Script **Tests**  Settings

```
1 postman.setGlobalVariable("jwt_token", (pm.response.json()).token);
```

3. Create Customer API – POST /api/customers

CorpTech / CustomerAPIs / CreateCustomer 



POST  http://localhost:8080/api/customers

Params Authorization Headers (9) **Body**  Pre-request Script

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary


```
1  {
2    "name": "CustomerName",
3    "email": "customeremail@gmail.com",
4    "phone": "9787897974"
5  }
```


Body Cookies Headers (14) Test Results


Pretty Raw Preview Visualize JSON  


```
1  {
2    "id": "266061e9-5ce7-467c-bde9-d6480fa79ce8",
3    "name": "CustomerName",
4    "email": "customeremail@gmail.com",
5    "phone": "9787897974"
6  }
```

4. Edit/update customer: [PUT /api/customers/{id}](#)

CorpTech / CustomerAPIs / UpdateCustomer 



PUT  http://localhost:8080/api/customers/266061e9-5ce7-467c-bde9-d6480fa79ce8

Params Authorization Headers (9) **Body**  Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** 


```
1 {
2   ... "name": "EDITEDCustomerName",
3   ... "email": "editcustomer@gmail.com",
4   ... "phone": "9797979799"
5 }
```


Body Cookies Headers (14) Test Results

Pretty Raw Preview Visualize **JSON**  


```
1 {
2   "id": "266061e9-5ce7-467c-bde9-d6480fa79ce8",
3   "name": "EDITEDCustomerName",
4   "email": "editcustomer@gmail.com",
5   "phone": "9797979799"
6 }
```

5. Get particular customer by ID [GET - /api/customer/{id}](#)

CorpTech / CustomerAPIs / GetCustomerByID 



GET  http://localhost:8080/api/customers/266061e9-5ce7-467c-bde9-d6480fa79ce8

Params Authorization **Headers (7)** Body Pre-request Script Tests Settings

Headers  6 hidden


	Key	Value
<input checked="" type="checkbox"/>	Authorization	Bearer {{jwt_token}}
	Key	Value


Body Cookies Headers (14) Test Results

Pretty Raw Preview Visualize **JSON**  


```
1 {
2   "id": "266061e9-5ce7-467c-bde9-d6480fa79ce8",
3   "name": "EDITEDCustomerName",
4   "email": "editcustomer@gmail.com",
5   "phone": "9797979799"
6 }
```

6. Delete particular customer by id – DELETE /api/customer/{id}

CorpTech / CustomerAPIs / DeleteCustomer 



DELETE  http://localhost:8080/api/customers/266061e9-5ce7-467c-bde9-d6480fa79ce8

Params Authorization Headers (9) Body ● Pre-request Script Tests Settings

Headers  8 hidden


	Key	Value
<input checked="" type="checkbox"/>	Authorization	Bearer {{jwt_token}}
	Key	Value


Body Cookies Headers (14) Test Results

Pretty Raw Preview Visualize JSON  


```
1 {
2   "deleted": true
3 }
```

7. Get all customer GET /api/customers

CorpTech / CustomerAPIs / GetAllCustomers 



GET  http://localhost:8080/api/customers

Params Authorization Headers (7) Body Pre-request Script Tests Sett

Headers  6 hidden

	Key	Value
<input checked="" type="checkbox"/>	Authorization	Bearer {{jwt_token}}
	Key	Value


Body Cookies Headers (14) Test Results


Pretty Raw Preview Visualize JSON  

```
1 [
2   {
3     "id": "02309ecd-ef68-4d73-bae8-cc66f8886c8a",
4     "name": "CustomerName1",
5     "email": "customer1email@gmail.com",
6     "phone": "9787897974"
7   },
8   {
9     "id": "7e987b52-27b2-4865-8b0a-586c8ab2b82d",
10    "name": "CustomerName2",
11    "email": "customer2email@gmail.com",
12    "phone": "789456577"
13  }
14 ]
```


8. Synch Data from Mysql database to salesforce account , Note that I have created a custom object named **CorpCustomer__c** in the salesforce and the data from customer table will be synched with salesforce in this custom object

POST – api/customers/synch


CorpTech / CustomerAPIs / SynchDBToSF 



POST  http://localhost:8080/api/customers/synch

Params Authorization **Headers (8)** Body Pre-request Script Tests Settings

Headers  7 hidden

	Key	Value
<input checked="" type="checkbox"/>	Authorization	Bearer {{jwt_token}}


Body Cookies Headers (14) Test Results 


Pretty Raw Preview Visualize JSON  

```
1  {
2    "totalSize": 2,
3    "done": true,
4    "records": [
5      {
6        "attributes": {
7          "type": "CorpCustomer__c",
8          "url": "/services/data/v52.0/objects/CorpCustomer__c/a005g00003EkH10AAF"
9        },
10       "Name": "CustomerName1",
11       "Id": "a005g00003EkH10AAF",
12       "Customer_id__c": "02309ecd-ef68-4d73-bae8-cc66f8886c8a",
13       "Email__c": "customer1email@gmail.com",
14       "Phone__c": "9787897974"
15     },
16     {
17       "attributes": {
```



ADDITIONAL USER CREATION

You can create addition user from signup api as USER role and you can access only GET API's and other API's will show as forbidden as below in the screenshot

CorpTech / CustomerAPIs / SynchDBToSF 



POST  http://localhost:8080/api/customers/sync

Params Authorization **Headers (8)** Body Pre-request Script Tests Settings

Headers  7 hidden

	Key	Value
<input checked="" type="checkbox"/>	Authorization	Bearer {{jwt_token}}
	Key	Value

Body Cookies **Headers (14)** Test Results

Pretty Raw Preview Visualize JSON  

```
1 {
2   "timestamp": "2023-10-02T09:29:11.202+00:00",
3   "status": 403,
4   "error": "Forbidden",
5   "message": "",
6   "path": "/api/customers/sync"
7 }
```