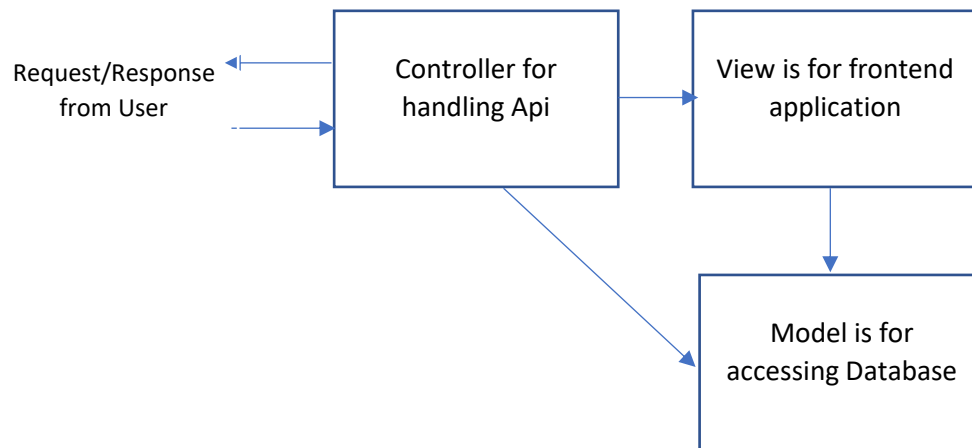


This is the documentation of the given Project (“Wallet Application”) . Here I have described the whole project. Please read this documentation before run the project .

I have used Spring Boot to build the project. MVC (Model – View – Controller) design pattern is used here.



Pic: - MVC design pattern

For Api documentation, I have used Swagger. The Swagger-ui url is “[Swagger UI](#)”.

The required steps –

**Step 1** – For this project purpose , I have used mysql database. In mysql database , please create a database with the name “walleddb”.

**Step 2** - Next, Run the Main class of spring boot application.

**Step 3**- Next , Goto the “[Swagger UI](#)” and select “wallet-controller”.

**Step 4**- You need to create an user account to access the wallet. So, Goto the Api – “createWallet” and create an user.

**POST** /api/v1/wallet/createAccount saveWallet

In this Api – username , “password” field is mandatory. Please give those field value. Do no give the “walletNo” . Spring automatically create the value.

POST ▼ http://localhost:8080/api/v1/wallet/createAccount Send ▼

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

none form-data x-www-form-urlencoded raw binary GraphQL JSON ▼ Beautify

```

1  {
2    "firstName": "Anupal",
3    "lastName": "Patra",
4    "address": "Kolkata",
5    "userName": "anupal",
6    "password": "anupal123",
7    "balance": "0"
8  }

```

Body Cookies Headers (5) Test Results 200 OK 611 ms 295 B Save Response ▼

**Pretty** Raw Preview Visualize JSON ▼ 🔍

```

1  {
2    "walletNo": 2,
3    "firstName": "Anupal",
4    "lastName": "Patra",
5    "address": "Kolkata",
6    "balance": 0.0,
7    "userName": "anupal",
8    "password": "anupal123"
9  }

```

**Step 5-** Next, You can check Account Details , Balance check from that. For checking account details, you can visit – “accountDetails” api . Please provide “userName” and “password”.

POST ▼ /api/v1/wallet/accountDetails getWallet

POST ▼ http://localhost:8080/api/v1/wallet/accountDetails Send ▼

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

none form-data x-www-form-urlencoded raw binary GraphQL JSON ▼ Beautify

```

1  {
2    "userName": "anupal",
3    "password": "anupal123"
4  }

```

Body Cookies Headers (5) Test Results 200 OK 246 ms 295 B Save Response ▼

**Pretty** Raw Preview Visualize JSON ▼ 🔍

```

1  {
2    "walletNo": 2,
3    "firstName": "Anupal",
4    "lastName": "Patra",
5    "address": "Kolkata",
6    "balance": 0.0,
7    "userName": "anupal",
8    "password": "anupal123"
9  }

```

For checking balance, you can visit – “checkBalance” api.

**POST** `/api/v1/wallet/checkBalance` getBalance

POST `http://localhost:8080/api/v1/wallet/checkBalance` Send

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

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

```
1 {
2   ... "userName" : "anupal",
3   ... "password" : "anupal123"
4 }
```

Body Cookies Headers (5) Test Results 200 OK 34 ms 167 B Save Response

Pretty Raw Preview Visualize **JSON**

```
1 0.0
```

**Step 6** – For amount deposit transaction, you can use – “deposit” api.

Please provide the values of “username” , “password” and “amount” field.

**POST** `/api/v1/wallet/deposit` depositAmount

POST `http://localhost:8080/api/v1/wallet/deposit` Send

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

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

```
1 {
2   ... "userName" : "anupal",
3   ... "password" : "anupal123",
4   ... "amount" : "1234.45"
5 }
```

Body Cookies Headers (5) Test Results 200 OK 147 ms 175 B Save Response

Pretty Raw Preview Visualize **Text**

```
1 Success
```

**Step 7** – For amount withdrawal transaction, you can use – “withdrawal” api.

Please provide the values of “username” , “password” and “amount” field.

The screenshot shows a REST client interface with a POST request to `http://localhost:8080/api/v1/wallet/withdrawal`. The request body is a JSON object with the following fields: `username` (anupal), `password` (anupal123), and `amount` (134.45). The response status is 200 OK, and the response is displayed in the 'Body' tab.

```
POST /api/v1/wallet/withdrawal withdrawalAmount
```

POST `http://localhost:8080/api/v1/wallet/withdrawal` Send

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

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

```
1 {
2   "username": "anupal",
3   "password": "anupal123",
4   "amount": "134.45"
5 }
```

Body Cookies Headers (5) Test Results 200 OK 66 ms 175 B Save Response

Pretty Raw Preview Visualize Text

1 Success

**Step 8** – Also you can see all the transaction details . you can use – “transactionList” api.

Please provide the values of “username” , “password” field.

The screenshot shows a REST client interface with a GET request to `http://localhost:8080/api/v1/wallet/transactionList`. The request body is a JSON object with the following fields: `username` (anupal) and `password` (anupal123). The response status is 200 OK, and the response is displayed in the 'Body' tab.

```
GET /api/v1/wallet/transactionList showTransactionList
```

GET `http://localhost:8080/api/v1/wallet/transactionList` Send

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

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

```
1 {
2   "username": "anupal",
3   "password": "anupal123"
4 }
```

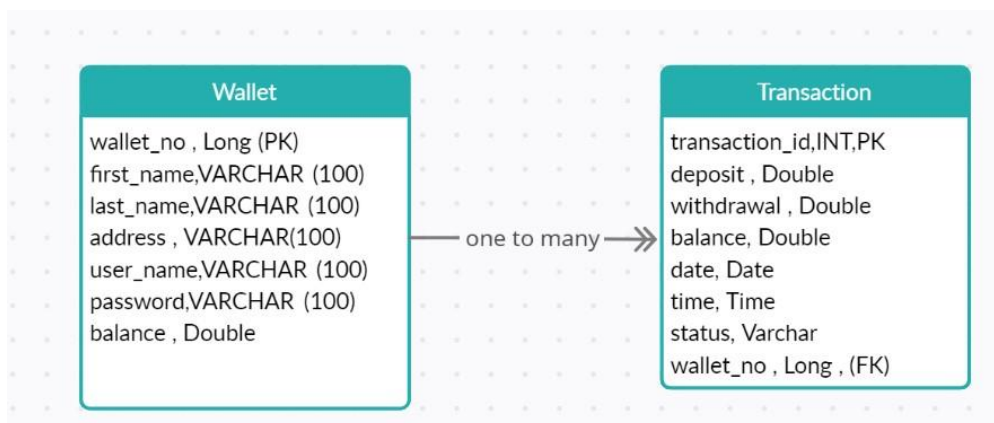
```
[
  {
    "transactionId": 52,
    "deposit": 1234.45,
    "withdrawal": null,
    "balance": 1234.45,
    "date": 1666204200000,
    "time": "16:16:17",
    "status": "Success",
    "wallet": {
      "walletNo": 2,
```

```

        "firstName": "Anupal",
        "lastName": "Patra",
        "address": "Kolkata",
        "balance": 1100.0,
        "userName": "anupal",
        "password": "anupal123"
    }
},
{
    "transactionId": 53,
    "deposit": null,
    "withdrawal": 134.45,
    "balance": 1100.0,
    "date": 1666204200000,
    "time": "16:17:58",
    "status": "Success",
    "wallet": {
        "walletNo": 2,
        "firstName": "Anupal",
        "lastName": "Patra",
        "address": "Kolkata",
        "balance": 1100.0,
        "userName": "anupal",
        "password": "anupal123"
    }
}
]

```

For the project purpose , I have designed the database schema. The picture is given below.



Thank you , for giving me the opportunity. I have completed the project as per your requirement .  
 My email – [patra.anupal@outlook.com](mailto:patra.anupal@outlook.com) and phno.- “8231886809” . If you need any additional details, Let me know.