

Aquarius Technical Test

Java Version : 1.8

Port: 8080

H2 DB access : <http://localhost:8080/h2-console>

Username : adminUser

Password : password

Start Spring boot application.

Task 1 and Task 2

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

The screenshot shows the Swagger UI interface for the 'main-controller' (Main Controller). The selected endpoint is 'GET /api/getPriceInfo' with the operation name 'getPriceInfo'. There are no parameters. The 'Execute' button is highlighted in blue. The 'Responses' section shows a 200 status code. The 'Response body' is displayed as a JSON array with two elements:

```
[{"id": 4, "symbol": "DUSKBTC", "quoteCurrency": null, "baseCurrency": null, "askPrice": 0.0000039, "bidPrice": 0.00000389, "bidSource": "Binance", "askSource": "Binance"}, {"id": 5, "symbol": "SMTUSD1", "quoteCurrency": null, "baseCurrency": null, "askPrice": 0.023073, "bidPrice": 0.02358}
```

Coin pairs are fetched from Binance and Houbi and best price is stored in DB running in 10 seconds retrieval intervals

Response Body will show all the coin pairs and their source data for the prices

API : <http://localhost:8080/api/getPriceInfo>

HttpMethod : GET

Task 3 , 4 and 5

API : <http://localhost:8080/api/transaction>

HttpMethod : POST

Purpose of this API is for user to Buy or Sell Coins

POST /api/transaction makeTransaction

Parameters

transaction * required (body)

Description: transaction

Example Value | Model

```
{  "amount": 1,  "symbol": "BTCUSDT",  "transaction": "B",  "userId": 1}
```

Parameter content type: application/json

Execute Clear

Responses

Response content type: */*

Curl

RequestBody

key	Value	type	Remarks
<u>Amount</u>	<u>E.g 1 , 2 ,3</u>	<u>Long</u>	
<u>Symbol</u>	<u>E.g ETHUSDT,BTCUSDT</u>	<u>String</u>	
<u>Transaction</u>	<u>B/S</u>	<u>String</u>	<u>B for Buy , S for Sell</u>
<u>userId</u>	<u>1</u>	<u>Long</u>	<u>Defaulted to 1 for the only pre created user</u>

API : <http://localhost:8080/api/getWalletInfo>

HttpMethod : GET

Purpose of this API is for user to view his existing wallet

The screenshot displays a REST client interface with the following sections:

- GET** `/api/getWalletInfo` `getWalletInfo`
- Parameters**: No parameters. A **Cancel** button is present.
- Execute** (blue button) and **Clear** (grey button) buttons.
- Responses**: Response content type is set to `*/*`.
- Curl**:

```
curl -X GET "http://localhost:8080/api/getWalletInfo" -H "accept: */*"
```
- Request URL**: `http://localhost:8080/api/getWalletInfo`
- Server response**:

Code	Details
200	<div>Response body<pre>{ "id": 1, "name": "admin", "wallet": { "id": 2, "walletHolding": [{ "id": 3, "amount": 19713.17, "symbol": "USDT" }, { "id": 2674, "amount": 1, "symbol": "BTC" }], "activeInd": "Y" }, "transactionList": [{ "id": 2673, "transaction": "b", "price": 30290.9, "quoteCurrency": "BTC" }] }</pre></div> <div>Download</div>

It will show the holdings of the User current coins and their respective amount

API : <http://localhost:8080/api/getTransactionInfo>

HttpMethod : GET

Purpose of this API is for user to view his previous transaction records

The screenshot shows a REST client interface with the following sections:

- GET** `/api/getTransactionInfo` `getTransactionInfo`
- Parameters**: No parameters. A **Cancel** button is present.
- Execute** and **Clear** buttons.
- Responses**: Response content type is `*/*`.
- Curl**:

```
curl -X GET "http://localhost:8080/api/getTransactionInfo" -H "accept: */*"
```
- Request URL**: `http://localhost:8080/api/getTransactionInfo`
- Server response**:

Code	Details
200	<p>Response body</p> <pre>[{ "id": 2673, "transaction": "b", "price": 30290.9, "quoteCurrency": "BTC", "baseCurrency": "USD", "amount": 1, "dateTime": [2023, 7, 9, 16, 26, 57, 686800000] }, { "id": 2678, "transaction": "b", "price": 30290.9, "quoteCurrency": "BTC",</pre>

It will show the transaction type B (Buy) / S (Sell) and the price at the time for transaction ,date time and amount

Test Case

User Buys 2 coin of ETH and sell 1 ETH

First check Wallet and have 50000 worth of USDT

The screenshot shows a REST client interface with the following sections:

- Parameters:** GET /api/getWalletInfo getWalletInfo. A "Cancel" button is present.
- Responses:** A dropdown menu for "Response content type" is set to "*/*".
- Curl:** A text area containing the command: `curl -X GET "http://localhost:8080/api/getWalletInfo" -H "accept: */*"`
- Request URL:** A text area containing: `http://localhost:8080/api/getWalletInfo`
- Server response:**
 - Code:** 200
 - Details:**
 - Response body:** A JSON object:

```
{  "id": 1,  "name": "admin",  "wallet": {    "id": 2,    "walletHolding": [      {        "id": 3,        "amount": 50000,        "symbol": "USDT"      }    ],    "activeInd": "Y"  },  "transactionList": []}
```

 A "Download" button is next to it.
 - Response headers:**

```
connection: keep-alive  content-type: application/json  date: Sun, 09 Jul 2023 08:59:17 GMT  keep-alive: timeout=60  transfer-encoding: chunked
```
- Responses:** A label at the bottom left.

Returned JSON :

```
{ "id": 1,
  "name": "admin",
  "wallet": {
    "id": 2,
    "walletHolding": [
      {
        "id": 3,
        "amount": 50000,
```

```
    "symbol": "USDT"
  },
  "activeInd": "Y"
},
"transactionList": []
}
```

Buys 2 amount of ETH and check wallet again

POST

/api/transaction makeTransaction

Cancel

Parameters

Name	Description
transaction * required <small>(body)</small>	transaction <div>Example Value Model</div> <div><pre>{ "amount": 2, "symbol": "ETHUSD", "transaction": "B", "userId": 1 }</pre></div>

Cancel

Parameter content type
application/json

Execute

Clear

Responses

Response content type */*

GET

/api/getWalletInfo getWalletInfo

Cancel

Parameters

No parameters

ExecuteClear

Responses

Response content type */*

Curl

curl -X GET "http://localhost:8080/api/getWalletInfo" -H "accept: */*"

Request URL

http://localhost:8080/api/getWalletInfo

Server response

Code	Details
200	<div>Response body</div> <pre>{ "id": 1, "name": "admin", "wallet": { "id": 2, "walletHolding": [{ "id": 3, "amount": 46263.16, "symbol": "USDT" }, { "id": 2674, "amount": 2, "symbol": "ETH" }], "activeInd": "Y" }, "transactionList": [{ "id": 2673, "transaction": "B", "price": 1868.42, "quoteCurrency": "ETH", </pre> <div>Download</div>

Response headers

Response Body

```
{
  "id": 1,
  "name": "admin",
  "wallet": {
    "id": 2,
    "walletHolding": [
      {
        "id": 3,
        "amount": 46263.16,
        "symbol": "USDT"
      },
      {
        "id": 2674,
```

```
    "amount": 2,
    "symbol": "ETH"
  }
],
"activeInd": "Y"
},
"transactionList": [
  {
    "id": 2673,
    "transaction": "B",
    "price": 1868.42,
    "quoteCurrency": "ETH",
    "baseCurrency": "USDT",
    "amount": 2,
    "dateTime": [
      2023,
      7,
      9,
      17,
      2,
      3,
      767000000
    ]
  }
]
```


Sell 1 amount of ETH and check wallet again

POST

/api/transaction

makeTransaction

Parameters

Cancel

Name	Description
transaction <small>required</small> <small>(body)</small>	transaction <div>Example Value Model</div> <pre>{ "amount": 1, "symbol": "ETHAUSD", "transaction": "S", "userId": 1}</pre>

Cancel

Parameter content type
application/json

ExecuteClear

Responses

Response content type */*

GET

/api/getWalletInfo

getWalletInfo

Parameters

Cancel

No parameters

ExecuteClear

Responses

Response content type */*

Curl

```
curl -X GET "http://localhost:8080/api/getWalletInfo" -H "accept: */*"
```

Request URL

```
http://localhost:8080/api/getWalletInfo
```

Server response

Code	Details
200	<div>Response body</div> <pre>{ "id": 1, "name": "admin", "wallet": { "id": 2, "walletHolding": [{ "id": 3, "amount": 48131.76, "symbol": "USDT" }, { "id": 2674, "amount": 1, "symbol": "ETH" }] }, "activeInd": "Y", "transactionList": [{ "id": 2673, "transaction": "B", "price": 1868.42, "quoteCurrency": "ETH", }]}</pre> <div>Download</div>

Response Body :

```
{
  "id": 1,
  "name": "admin",
  "wallet": {
    "id": 2,
    "walletHolding": [
      {
        "id": 3,
        "amount": 48131.76,
        "symbol": "USDT"
      },
      {
        "id": 2674,
        "amount": 1,
        "symbol": "ETH"
      }
    ],
    "activeInd": "Y"
  },
  "transactionList": [
    {
      "id": 2673,
      "transaction": "B",
      "price": 1868.42,
      "quoteCurrency": "ETH",
      "baseCurrency": "USDT",
      "amount": 2,
      "dateTime": [
```

```
2023,
7,
9,
17,
2,
3,
767000000
]
},
{
  "id": 2675,
  "transaction": "S",
  "price": 1868.6,
  "quoteCurrency": "USDT",
  "baseCurrency": "ETH",
  "amount": 1,
  "dateTime": [
    2023,
    7,
    9,
    17,
    4,
    44,
    353000000
  ]
}
]
```

Checks Transaction Record of his past actions

The screenshot shows a REST client interface with the following sections:

- Method and URL:** GET /api/getTransactionInfo
- Parameters:** No parameters
- Execute Button:** A blue button labeled "Execute" and a "Clear" button.
- Responses:** A section with a dropdown for "Response content type" set to "*/*".
- Curl:** A text area containing the curl command: `curl -X GET "http://localhost:8080/api/getTransactionInfo" -H "accept: */*"`
- Request URL:** A text area containing the URL: `http://localhost:8080/api/getTransactionInfo`
- Server response:** A table with two columns: "Code" and "Details".
 - Code:** 200
 - Details:** A "Response body" section containing a JSON array of two transaction objects.

```
[
  {
    "id": 2673,
    "transaction": "B",
    "price": 1868.42,
    "quoteCurrency": "ETH",
    "baseCurrency": "USDT",
    "amount": 2,
    "dateTime": [
      2023,
      7,
      9,
      17,
      2,
      3,
      767000000
    ]
  },
  {
    "id": 2675,
    "transaction": "S",
    "price": 1868.6,
    "quoteCurrency": "USDT",
```

Response Body :

```
[
  {
    "id": 2673,
    "transaction": "B",
    "price": 1868.42,
    "quoteCurrency": "ETH",
    "baseCurrency": "USDT",
    "amount": 2,
    "dateTime": [
      2023,
      7,
      9,
```

```
17,  
2,  
3,  
767000000  
]  
,  
{  
  "id": 2675,  
  "transaction": "S",  
  "price": 1868.6,  
  "quoteCurrency": "USDT",  
  "baseCurrency": "ETH",  
  "amount": 1,  
  "dateTime": [  
    2023,  
    7,  
    9,  
    17,  
    4,  
    44,  
    353000000  
  ]  
}  
]
```