

Setup and Unit Tests

- Sync the project from below address:
<https://github.com/alkimozkan/brokerage-firm-backend-api>

- Import the project into SpringToolsForEclipse
- In SpringToolsForEclipse, run the project as Spring Boot App
- Go to below address in your browser
- localhost:8080/h2-console

Username: admin
Password: 1234

In login page change
Driver Class as org.h2.Driver
JDBC URL as given below
jdbc:h2:file:~/testdb2

Username: admin
Password: 1234

- Run below commands to create pre data on db

```
INSERT INTO side (side_name) VALUES ('BUY');  
INSERT INTO side (side_name) VALUES ('SELL');  
INSERT INTO status (status_name) VALUES ('PENDING');  
INSERT INTO status (status_name) VALUES ('MATCHED');  
INSERT INTO status (status_name) VALUES ('CANCELLED');
```

```
INSERT INTO customer (customer_name) VALUES ('Şükrü Özkan');  
INSERT INTO customer (customer_name) VALUES ('Ahmet Yılmaz');  
INSERT INTO customer (customer_name) VALUES ('Mustafa Deniz');
```

```
INSERT INTO asset (asset_name, size, usable_size, customer_id)  
VALUES ('TRY', '200000', '200000', '1');
```

```
INSERT INTO asset (asset_name, size, usable_size, customer_id)  
VALUES ('USD', '1000', '1000', '1');
```

```
INSERT INTO asset (asset_name, size, usable_size, customer_id)  
VALUES ('TRY', '100000', '100000', '2');
```

```
INSERT INTO asset (asset_name, size, usable_size, customer_id)  
VALUES ('USD', '2000', '2000', '2');
```

```
INSERT INTO asset (asset_name, size, usable_size, customer_id)  
VALUES ('TRY', '50000', '50000', '3');
```

```
INSERT INTO asset (asset_name, size, usable_size, customer_id)  
VALUES ('USD', '3000', '3000', '3');
```

```
INSERT INTO t_order (  
    size,  
    price,
```

```

create_date,
asset_id,
customer_id,
order_side_id,
status_id
) VALUES (
1,
400,
'2025-08-28',
2,
3,
2,
1
);

```

- Now you can stop the spring boot app in SpringToolsForEclipse and then you can run unit tests.

Tables

SELECT * FROM STATUS;

ID	STATUS_NAME
1	PENDING
2	MATCHED
3	CANCELLED

SELECT * FROM SIDE;

ID	SIDE_NAME
1	BUY
2	SELL

SELECT * FROM CUSTOMER;

ID	CUSTOMER_NAME
1	Şükrü Özkan
2	Ahmet Yılmaz
3	Mustafa Deniz

SELECT * FROM ASSET;

CUSTOMER_ID	ID	SIZE	USABLE_SIZE	ASSET_NAME
1	1	200000	200000	TRY
1	2	1000	1000	USD
2	3	100000	100000	TRY
2	4	2000	2000	USD
3	5	50000	50000	TRY
3	6	3000	3000	USD

SELECT * FROM T_ORDER;

ASSET_ID	CREATE_DATE	CUSTOMER_ID	ID	ORDER_SIDE_ID	PRICE	SIZE	STATUS_ID
2	2025-08-28 00:00:00	3	1	2	400	1	1

Endpoints

- Open Postman application then you can create requests as below:

createOrder

Select method type as POST

<http://localhost:8080/rest/api/order/save>

Click authorization and choose Basic Auth then enter username as password.

Then click to Body and select raw and JSON. Json example for request is given below.

```
{
  "size": 1000,
  "price": 40,
  "createDate": "2025-08-28",
  "assetName": "USD",
  "asset": {
    "id": 2
  },
  "customer": {
    "id": 1
  },
  "orderSide": {
    "id": 1,
    "sideName": "BUY"
  },
  "status": {
    "id": 1
  }
}
```

Asset data before the buy command is given below:

SELECT * FROM ASSET;

CUSTOMER_ID	ID	SIZE	USABLE_SIZE	ASSET_NAME
1	1	200000	200000	TRY
1	2	1000	1000	USD
2	3	100000	100000	TRY
2	4	2000	2000	USD
3	5	50000	50000	TRY
3	6	3000	3000	USD

(6 rows, 2 ms)

Asset data after the buy command is given below:

SELECT * FROM ASSET;

CUSTOMER_ID	ID	SIZE	USABLE_SIZE	ASSET_NAME
1	1	200000	160000	TRY
1	2	1000	1000	USD
2	3	100000	100000	TRY
2	4	2000	2000	USD
3	5	50000	50000	TRY
3	6	3000	3000	USD

Asset data after the sell command is given below

SELECT * FROM ASSET;

CUSTOMER_ID	ID	SIZE	USABLE_SIZE	ASSET_NAME
1	1	200000	160000	TRY
1	2	1000	500	USD
2	3	100000	100000	TRY
2	4	2000	2000	USD
3	5	50000	50000	TRY
3	6	3000	3000	USD

```
{
  "size": 500,
  "price": 40,
  "createDate": "2025-08-28",
  "assetName": "USD",
  "asset": {
    "id": 2
  },
  "customer": {
    "id": 1
  },
  "orderSide": {
    "id": 2,
    "sideName": "SELL"
  },
  "status": {
    "id": 1
  }
}
```

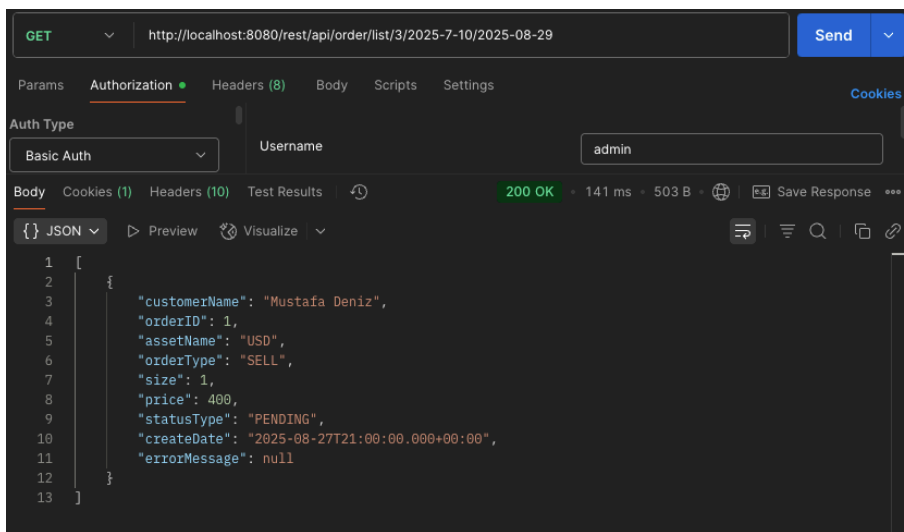
listOrders

Select method type as GET

<http://localhost:8080/rest/api/order/list/customerId/startDate/endDate>

Usage example

<http://localhost:8080/rest/api/order/list/3/2025-7-10/2025-08-29>



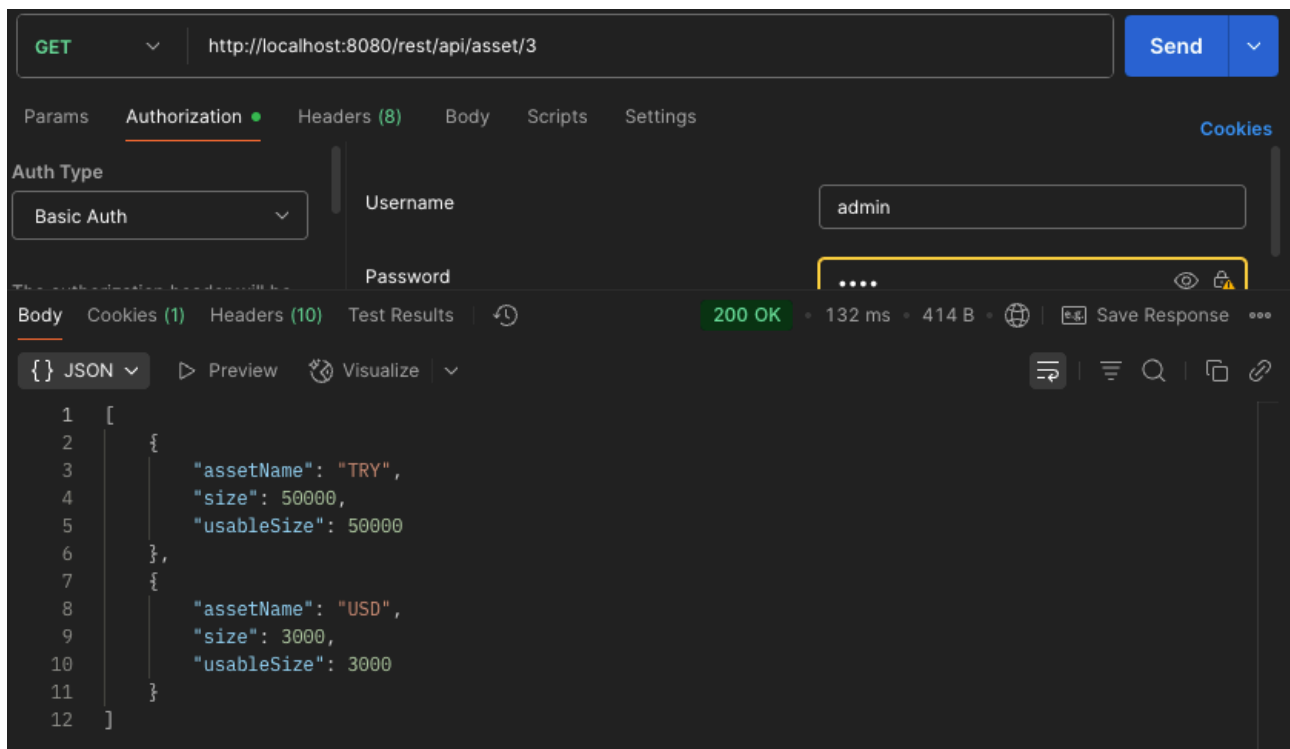
listAssets

Select method type as GET

<http://localhost:8080/rest/api/asset/customerId>

Usage example

<http://localhost:8080/rest/api/asset/3>



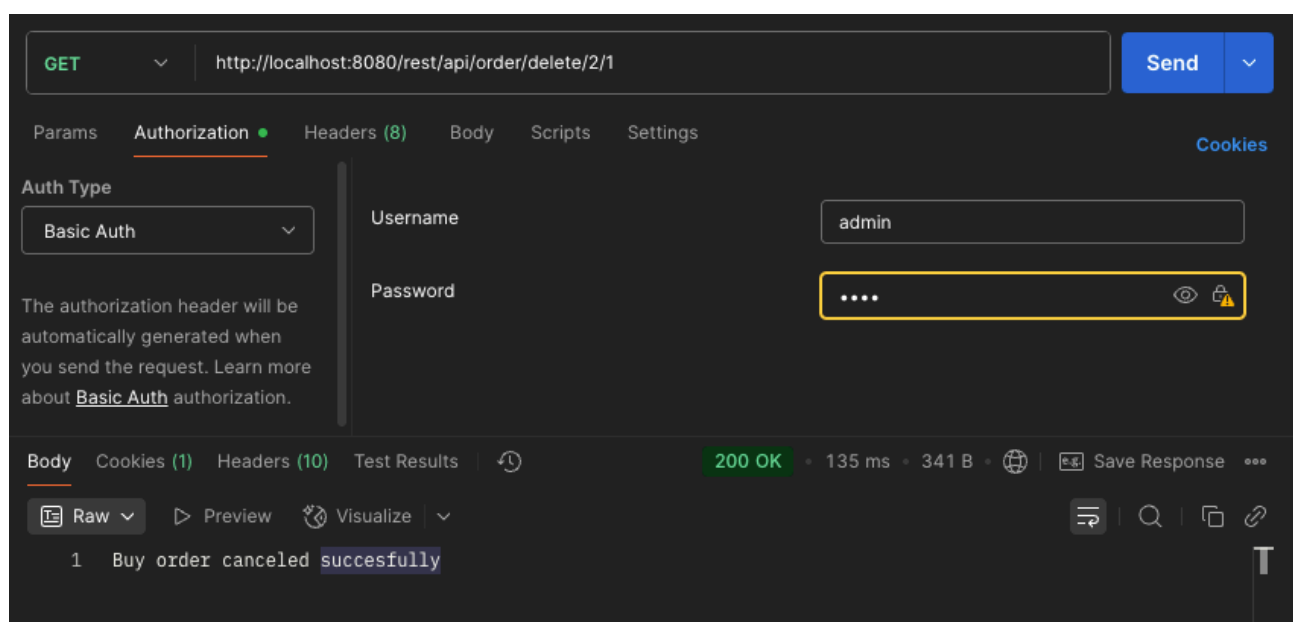
deleteOrder

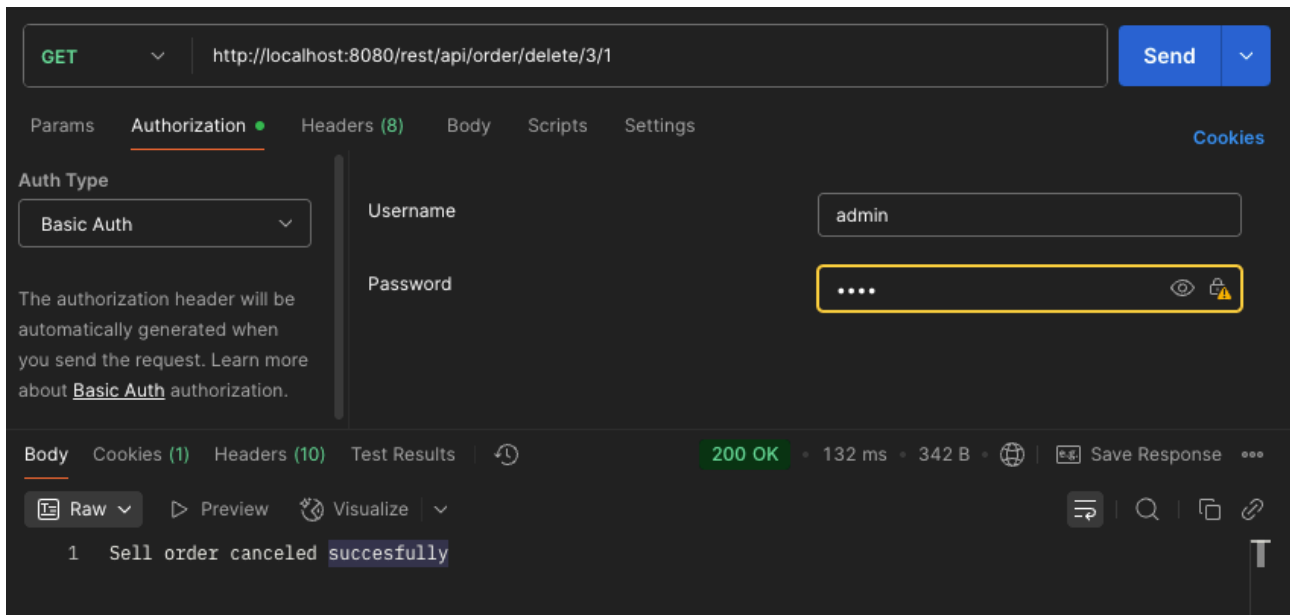
Select method type as GET

<http://localhost:8080/rest/api/order/delete/orderId/customerId>

Usage example

<http://localhost:8080/rest/api/order/delete/1/3>





Asset table is as given below after deleting the orders:

SELECT * FROM ASSET;

CUSTOMER_ID	ID	SIZE	USABLE_SIZE	ASSET_NAME
1	1	200000	200000	TRY
1	2	1000	1000	USD
2	3	100000	100000	TRY
2	4	2000	2000	USD
3	5	50000	50000	TRY
3	6	3000	3000	USD

(6 rows, 0 ms)