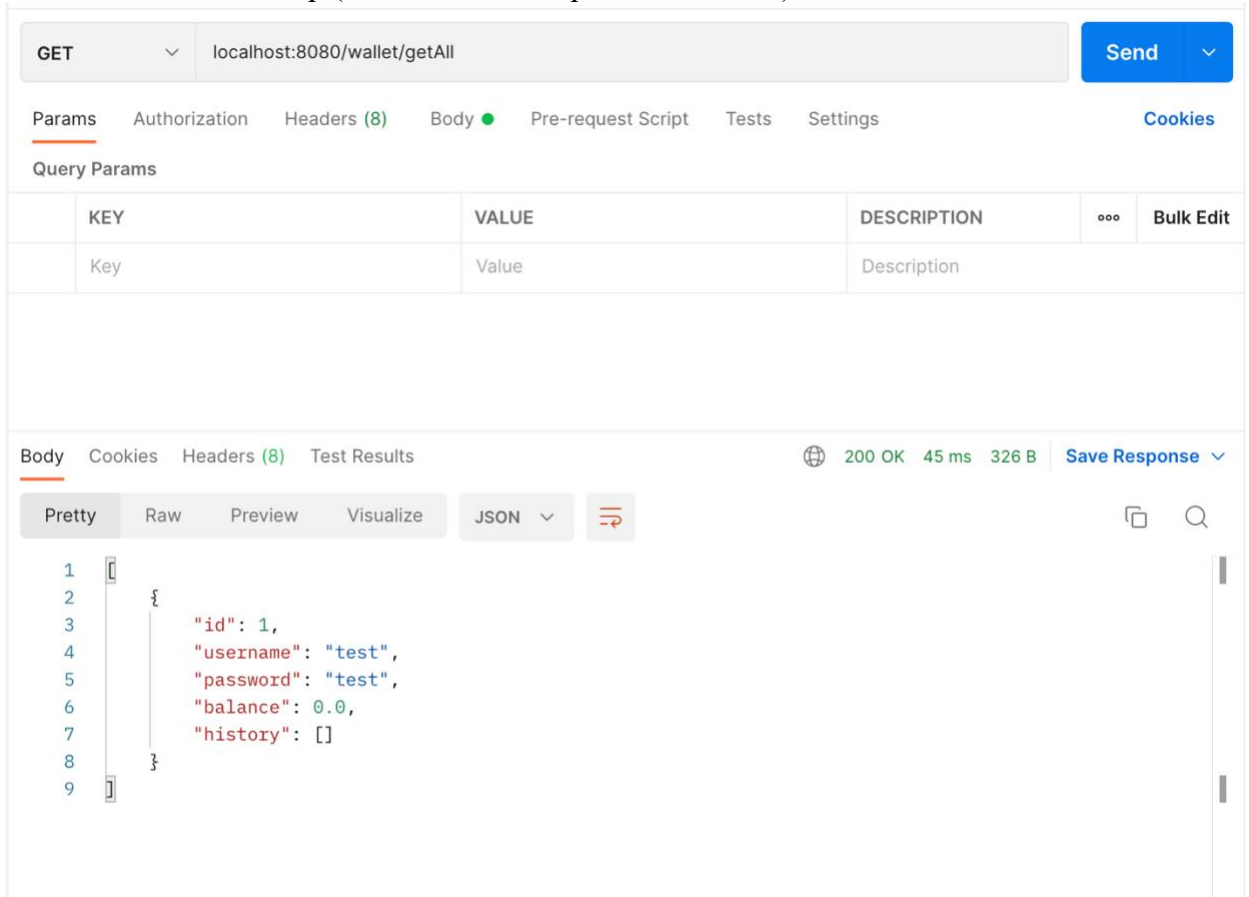


Wallet Testing

Note: Testing is done via Postman and Junit/Rest Assured. A test user wallet is created when the server boots up (username: “test”, password: “test”)



Testing functionalities:

1. `/wallet/register` – to create a user wallet
Created a user wallet with username: “rushal” and password: “pwd”

POST localhost:8080/wallet/register 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": "rushal",
3   "password": "pwd"
4 }
```

Body Cookies Headers (8) Test Results 200 OK 77 ms 304 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Wallet setup successful",
3   "errorCode": 0
4 }
```

2. /wallet/login – to log into a user wallet
Retrieving the aforementioned wallet

POST localhost:8080/wallet/login 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": "rushal",
3   "password": "pwd"
4 }
```

Body Cookies Headers (8) Test Results 200 OK 120 ms 297 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Login successful",
3   "errorCode": 0
4 }
```

3. /wallet/getAll – to retrieve all user wallets (more like an admin thing)

The screenshot shows a REST client interface. The request is a GET to `localhost:8080/wallet/getAll`. The body is a JSON object with `username: "rushal"` and `password: "pwd"`. The response is a 200 OK with a 7 ms latency and 399 B body. The response body is a JSON array of two wallet objects.

```
{
  "id": 1,
  "username": "test",
  "password": "test",
  "balance": 0.0,
  "history": []
},
{
  "id": 2,
  "username": "rushal",
  "password": "pwd",
  "balance": 0.0,
  "history": []
}
```

4. /wallet/balance– to get a user wallet’s balance (“username” as param)

The screenshot shows a REST client interface. The request is a GET to `localhost:8080/wallet/balance?username=rushal`. The response is a 200 OK with a 30 ms latency and 247 B body. The response body is the value `0.0`.

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	username	rushal			
	Key	Value	Description		

5. /wallet/add/{amount} – to add a specific amount to a user wallet (“username” as param)
- Add 100 to wallet
 - Add 25 to wallet
- Expected balance is 125

PUTlocalhost:8080/wallet/add/100?username=rushalSend

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettingsCookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	username	rushal			
	Key	Value	Description		

BodyCookiesHeaders (8)Test Results200 OK 39 ms 306 BSave Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   "message": "Amount added successfully",
3   "errorCode": 0
4 }
```

PUTlocalhost:8080/wallet/add/25?username=rushalSend

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettingsCookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	username	rushal			
	Key	Value	Description		

BodyCookiesHeaders (8)Test Results200 OK 12 ms 306 BSave Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   "message": "Amount added successfully",
3   "errorCode": 0
4 }
```

GET localhost:8080/wallet/balance?username=rushal Send

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

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	username	rushal			
	Key	Value	Description		

Body Cookies Headers (8) Test Results 200 OK 27 ms 249 B Save Response

Pretty Raw Preview Visualize JSON

```
1 125.0
```

6. /wallet/withdraw/{amount} – to withdraw a specific amount from a user wallet (“username” as param)
- Withdraw 35
 - Withdraw 65
- Expected balance is 25

PUT localhost:8080/wallet/withdraw/35?username=rushal Send

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

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	username	rushal			
	Key	Value	Description		

Body Cookies Headers (8) Test Results 200 OK 15 ms 310 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Amount withdrawn successfully",
3   "errorCode": 0
4 }
```

PUT

localhost:8080/wallet/withdraw/65?username=rushal

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	username	rushal			
	Key	Value	Description		

Body

Cookies

Headers (8)

Test Results

200 OK 14 ms 310 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

{

"message": "Amount withdrawn successfully",

"errorCode": 0

}

GET

localhost:8080/wallet/balance?username=rushal

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	username	rushal			
	Key	Value	Description		

Body

Cookies

Headers (8)

Test Results

200 OK 7 ms 248 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

25.0

7. /wallet/history – to get a user wallet’s transaction history (“+” for add, “-” for withdraw; “username” as param)

GET localhost:8080/wallet/history?username=rushal Send

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

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	username	rushal			
	Key	Value	Description		

Body Cookies Headers (8) Test Results 200 OK 17 ms 287 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   "+100.0",
3   "+25.0",
4   "-35.0",
5   "-65.0"
6 ]
```

8. /wallet/delete – to delete a user wallet (“username” as param)

DELETE localhost:8080/wallet/delete?username=rushal Send

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

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	username	rushal			
	Key	Value	Description		

Body Cookies Headers (8) Test Results 200 OK 57 ms 298 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Delete Successful",
3   "errorCode": 0
4 }
```

9. Additional testing

a. Invalid credentials

The screenshot shows a REST client interface for a POST request to `localhost:8080/wallet/login`. The request body is a JSON object: `{ "username": "abc", "password": "pwd" }`. The response status is `200 OK` with a response time of `14 ms` and a body size of `300 B`. The response body is a JSON object: `{ "message": "Invalid credentials", "errorCode": 1 }`.

```
POST localhost:8080/wallet/login
```

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": "abc",
3   "password": "pwd"
4 }
```

Body Cookies Headers (8) Test Results 200 OK 14 ms 300 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Invalid credentials",
3   "errorCode": 1
4 }
```

b. Username already taken

The screenshot shows a REST client interface for a POST request to `localhost:8080/wallet/register`. The request body is a JSON object: `{ "username": "rushal", "password": "pwd" }`. The response status is `200 OK` with a response time of `61 ms` and a body size of `313 B`. The response body is a JSON object: `{ "message": "ERROR: Username has to be unique", "errorCode": 1 }`.

```
POST localhost:8080/wallet/register
```

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": "rushal",
3   "password": "pwd"
4 }
```

Body Cookies Headers (8) Test Results 200 OK 61 ms 313 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "ERROR: Username has to be unique",
3   "errorCode": 1
4 }
```


c. Insufficient balance

PUT

localhost:8080/wallet/withdraw/50?username=rushal

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	username	rushal			
	Key	Value	Description		

Body

Cookies

Headers (8)

Test Results

200 OK 26 ms 301 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "message": "Insufficient balance",
3   "errorCode": 1
4 }
```

d. Invalid amount value

PUT

localhost:8080/wallet/add/-50?username=rushal

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	username	rushal			
	Key	Value	Description		

Body

Cookies

Headers (8)

Test Results

200 OK 13 ms 307 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "message": "Enter a non-negative value",
3   "errorCode": 1
4 }
```

10. JUnit System test report – some system tests done using Rest Assured

The screenshot displays an IDE window with the following components:

- Package Explorer:** Shows the project structure with a folder named `TestingSystemTest` containing a file `TestingSystemTest`. The runner is JUnit 5, and it took 2.513 s to execute.
- JUnit Results:** Shows a green bar indicating successful execution. Summary: `Runs: 1/1`, `Errors: 0`, `Failures: 0`. The test finished after 8.241 seconds.
- Code Editor:** Displays the `WalletController.java` file. The code includes a `@Before` method `setUp()` that configures Rest Assured with `port`, `baseURI` (`http://localhost`), and `contentType` (`application/json`). It also includes a `@Test` method `walletTest()` that performs a GET request to `/wallet/balance` and a PUT request to `/wallet/add/{amount}` with a parameter `amount` of 100. The code checks the status code of the PUT request.
- Problems/Console/Coverage:** The Coverage tab is active, showing a table with the following data:

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
Wallet	84.3 %	619	115	734