

IT314 Software Engineering Project

Charity Wallet

Group: 8

Unit Testing

Ayush Hirdani (202101139)

NGOs:

1) Register NGO

The screenshot displays the Postman interface for a POST request to `localhost:5000/ngo/register`. The request is labeled `POST registerNgo` and is saved in the `CharityWallet / NGO` collection. The `Tests` tab is active, showing three test scripts:

```
1 pm.test("Successful POST request", function () {
2   pm.expect(pm.response.code).to.be.oneOf([200, 201, 202]);
3 });
4 pm.test("Success Check", function () {
5   var jsonData = pm.response.json();
6   pm.expect(jsonData.success).to.eql(true);
7 });
8 pm.test("OTP Test", function () {
9   var jsonData = pm.response.json();
10  pm.expect(jsonData.message).to.eql("Email for OTP sent to you.");
11 });
```

The right sidebar shows a list of snippets for response body and header checks. The bottom status bar indicates a `200 OK` response with a time of `3.07 s` and a size of `344 B`. The `Test Results (3/3)` section shows all three tests passed:

- PASS** Successful POST request
- PASS** Success Check
- PASS** OTP Test

2) Confirm NGO Registration with Email

First try with Wrong OTP

The screenshot shows the Postman interface for a POST request to `localhost:5000/ngo/confirm-registration`. The request body is a JSON object: `{ "email": "ayushhirdani13@gmail.com", "otp": "123456" }`. The test results show a failure with the message: "Status code is 403 | AssertionError: expected response to have status code 403 but got 500". The test is labeled "Wrong OTP".

Then with the correct OTP.

The screenshot shows the Postman interface for a POST request to `localhost:5000/ngo/confirm-registration`. The request body is a JSON object: `{ "email": "ayushhirdani13@gmail.com", "otp": "267354" }`. The test results show a failure with the message: "Status code is 201 | AssertionError: expected response to have status code 201 but got 200". The test is labeled "Successful. NGO Created."

3) Login NGO

With wrong Password

The screenshot shows the Postman interface for a POST request to `localhost:5000/ngo/login`. The **Tests** tab is active, displaying two test scripts:

```
1 pm.test("Success False Test", function () {  
2   var jsonData = pm.response.json();  
3   pm.expect(jsonData.success).to.eql(false);  
4 });  
5 pm.test("Wrong Password", function () {  
6   var jsonData = pm.response.json();  
7   pm.expect(jsonData.message).to.eql("Invalid email or password.");  
8 });
```

The right sidebar shows snippets for "Response body: JSON value check" and "Response body: Is equal to a string". The bottom status bar indicates a **Status: 400 Bad Request**, **Time: 83 ms**, and **Size: 354 B**. The **Test Results (2/2)** section shows two passed tests:

- PASS** Success False Test
- PASS** Wrong Password

With Correct Password

The screenshot shows the Postman interface for a POST request to `localhost:5000/ngo/login`. The **Tests** tab is active, displaying two test scripts:

```
1 pm.test("Success True", function () {  
2   var jsonData = pm.response.json();  
3   pm.expect(jsonData.success).to.eql(true);  
4 });  
5 pm.test("Status code is 200", function () {  
6   pm.response.to.have.status(200);  
7 });
```

The right sidebar shows snippets for "Status code: Code is 200" and "Response body: Contains string". The bottom status bar indicates a **Status: 200 OK**, **Time: 104 ms**, and **Size: 598 B**. The **Test Results (2/2)** section shows two passed tests:

- PASS** Success True
- PASS** Status code is 200

4) Getting My NGO Profile

While logged in

The screenshot shows the Postman interface for a GET request to `localhost:5000/ngo/myNgo`. The **Tests** tab is active, displaying two test scripts:

```
1 pm.test("Status code is 200", function () {  
2   pm.response.to.have.status(200);  
3 });  
4 pm.test("NGO Data Check", function () {  
5   var jsonData = pm.response.json();  
6   pm.expect(jsonData.ngo).to.be.instanceOf(Object);  
7 });
```

The right sidebar shows a list of snippets, including "Status code: Code is 200", "Response body: Contains string", "Response body: JSON value check", "Response body: Is equal to a string", and "Response headers: Content-Type header check".

Below the test scripts, the **Test Results (2/2)** section shows the status of the tests:

- PASS** Status code is 200
- PASS** NGO Data Check

The overall status bar at the bottom indicates: Status: 200 OK, Time: 41 ms, Size: 735 B. A "View Templates" button is also visible.

Without Login

The screenshot shows the Postman interface for a GET request to `localhost:5000/ngo/myNgo`. The **Tests** tab is active, displaying three test scripts:

```
1 pm.test('Status code is 200', function () {  
2   pm.response.to.have.status(200);  
3 });  
4  
5 pm.test('NGO Data Check', function () {  
6   var jsonData = pm.response.json();  
7   pm.expect(jsonData.ngo).to.be.instanceOf(Object);  
8 });  
9  
10 pm.test("Ask for Login First", function () {  
11   var jsonData = pm.response.json();  
12   pm.expect(jsonData.message).to.eql("Login First");  
13 });  
14  
15
```

The right sidebar shows a list of snippets, including "Status code: Code is 200", "Response body: Contains string", "Response body: JSON value check", "Response body: Is equal to a string", and "Response headers: Content-Type header check".

Below the test scripts, the **Test Results (1/3)** section shows the status of the tests:

- FAIL** Status code is 200 | AssertionError: expected response to have status code 200 but got 404
- FAIL** NGO Data Check | AssertionError: expected undefined to be an instance of Object
- PASS** Ask for Login First

The overall status bar at the bottom indicates: Status: 404 Not Found, Time: 6 ms, Size: 337 B. A "View Templates" button is also visible.

5) Update Profile Along with Images.

The screenshot shows a Postman interface for a PATCH request to `localhost:5000/ngo/myNgo`. The request is configured with form-data, and the body contains three fields: `vision` with the value `Updated Star NGO Vision`, `logo` with the file `star.jpg`, and `gallery` with `2 files selected`. The response status is 200 OK, with a time of 5.63 s and a size of 957 B. The test results show two passed assertions: "Status code is 200" and "Success Check".

Key	Value	Description
<input checked="" type="checkbox"/> vision	Updated Star NGO Vision	
<input checked="" type="checkbox"/> logo	star.jpg	
<input checked="" type="checkbox"/> gallery	2 files selected	

Test Results (2/2)

- PASS Status code is 200
- PASS Success Check

6) Add a New Campaign

The screenshot shows a Postman interface for a POST request to `localhost:5000/ngo/addCampaign`. The request is configured with form-data, and the body contains four fields: `title` with the value `Sample Campaign`, `vision` with `Sample Vision`, `description` with `Sample Description`, and `cover` with the file `img2.jpg`. The response status is 201 Created, with a time of 2.20 s and a size of 1.22 KB. The test results show one failed assertion: "Status code is 200 | AssertionError: expected response to have status code 200 but got 201", and one passed assertion: "Success Check".

Key	Value	Description
<input checked="" type="checkbox"/> title	Sample Campaign	
<input checked="" type="checkbox"/> vision	Sample Vision	
<input checked="" type="checkbox"/> description	Sample Description	
<input checked="" type="checkbox"/> cover	img2.jpg	

Test Results (1/2)

- FAIL Status code is 200 | AssertionError: expected response to have status code 200 but got 201
- PASS Success Check

7) Update Campaign

The screenshot shows the 'updateMyCampaign' endpoint in a testing tool. The request method is 'PATCH' and the URL is 'localhost:5000/ngo/updateCampaign?campaignAlias=sample_campaign'. The 'Body' tab is selected, showing a table with one row: 'cover' with value 'img3.jpg'. The status is '200 OK', time is '2.47 s', and size is '449 B'. The test results show two 'PASS' items: 'Status code is 200' and 'Success Check'.

CharityWallet / NGO / **updateMyCampaign**

PATCH localhost:5000/ngo/updateCampaign?campaignAlias=sample_campaign Send

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

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

Key	Value	Description
<input checked="" type="checkbox"/> cover	img3.jpg	
Key	Value	Description

Body Cookies (1) Headers (9) **Test Results (2/2)** Status: 200 OK Time: 2.47 s Size: 449 B Save as example

Perform API tests faster with templates for integration testing, regression testing, and more. View Templates

All Passed Skipped Failed

PASS Status code is 200

PASS Success Check

8) Delete Campaign

The screenshot shows the 'deleteMyCampaign' endpoint in a testing tool. The request method is 'DELETE' and the URL is 'localhost:5000/ngo/deleteCampaign?campaignAlias=sample_campaign'. The 'Params' tab is selected, showing a table with one row: 'campaignAlias' with value 'sample_campaign'. The status is '200 OK', time is '84 ms', and size is '348 B'. The test results show two 'PASS' items: 'Status code is 200' and 'Success Check'.

CharityWallet / NGO / **deleteMyCampaign**

DELETE localhost:5000/ngo/deleteCampaign?campaignAlias=sample_campaign Send

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

Query Params

Key	Value	Description
<input checked="" type="checkbox"/> campaignAlias	sample_campaign	
Key	Value	Description

Body Cookies (1) Headers (9) **Test Results (2/2)** Status: 200 OK Time: 84 ms Size: 348 B Save as example

Perform API tests faster with templates for integration testing, regression testing, and more. View Templates

All Passed Skipped Failed

PASS Status code is 200

PASS Success Check

Tried deleting same campaign again

CharityWallet / NGO / deleteMyCampaign

DELETE localhost:5000/ngo/deleteCampaign?campaignAlias=sample_campaign

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

Query Params

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	campaignAlias	sample_campaign			
	Key	Value	Description		

Body Cookies (1) Headers (9) Test Results (0/2) Status: 404 Not Found Time: 117 ms Size: 345 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "success": false,
3   "message": "Campaign Not Found."
4 }
```

9) Change Password

Passwords only changed after valid OTP.

CharityWallet / NGO / changePasswordConfirmation

POST localhost:5000/ngo/changePasswordConfirm

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

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

Beautiful

```
1 {
2   "email": "ayushhirdani113@gmail.com",
3   "otp": "619997",
4   "password": "Sample123"
5 }
```

Body Cookies (1) Headers (10) Test Results (2/2) Status: 200 OK Time: 157 ms Size: 608 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "success": true,
3   "message": "NGO Password Updated Successfully."
4 }
```

10) Logout

The screenshot shows a REST client interface with a dark theme. At the top, a list of requests includes 'GET Logout'. The main area shows a GET request to 'localhost:5000/ngo/logout'. Below the request bar, tabs for 'Params', 'Authorization', 'Headers (6)', 'Body', 'Pre-request Script', 'Tests', and 'Settings' are visible. The 'Body' tab is selected, displaying a JSON response in 'Pretty' format:

```
{  "success": true,  "message": "Logged Out Successfully."}
```

. The status bar at the bottom indicates 'Status: 200 OK', 'Time: 94 ms', and 'Size: 412 B'.

Key	Value	Description
Key	Value	Description

Try Logout without Login:

The screenshot shows the same REST client interface, but the response is a 404 Not Found. The status bar at the bottom indicates 'Status: 404 Not Found', 'Time: 7 ms', and 'Size: 337 B'. The JSON response in the 'Body' tab is:

```
{  "success": false,  "message": "Login First"}
```

.

Key	Value	Description
Key	Value	Description

Campaigns:

1) getCampaigns

CharityWallet / Campaigns / **getCampaigns**

GET localhost:5000/campaign Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (9) **Test Results (2/2)** Status: 200 OK Time: 22 ms Size: 1.16 KB Save as example

Perform API tests faster with templates for integration testing, regression testing, and more. View Templates

All Passed Skipped Failed

PASS Status code is 200

PASS Success Check

2) Get single campaign by its alias

GET localhost:5000/campaign/dashboard?campaignAlias=star_campaign Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> campaignAlias	star_campaign			
Key	Value	Description		

Body Cookies Headers (9) **Test Results (2/2)** Status: 200 OK Time: 36 ms Size: 587 B Save as example

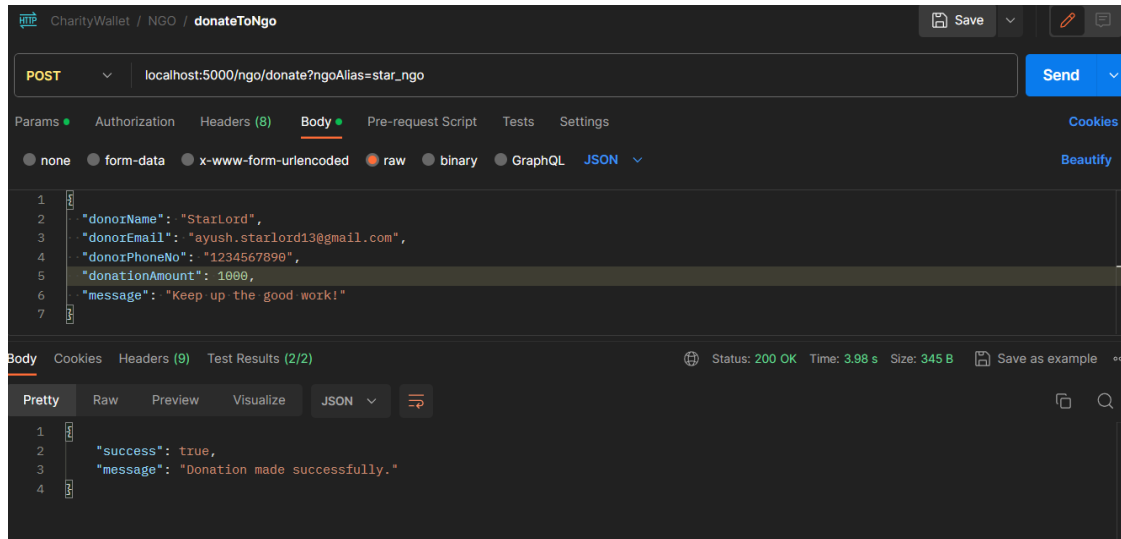
Pretty Raw Preview Visualize JSON

```
1  {
2    "success": true,
3    "data": {
4      "title": "Star Campaign",
5      "vision": "",
6      "description": "",
7      "organizerType": "NGO",
8      "organizerId": "655d9d162a9db07b508cbef3",
9      "alias": "star_campaign",
10     "cover": "1FVzZq0iEht5C3573rnAnLyY7SBe_uE6x",
11     "donationsTillNow": 1200,
12     "verified": true,
13     "createdAt": "2023-12-02T09:21:17.478Z",
14     "__v": 0
15   }
16 }
```

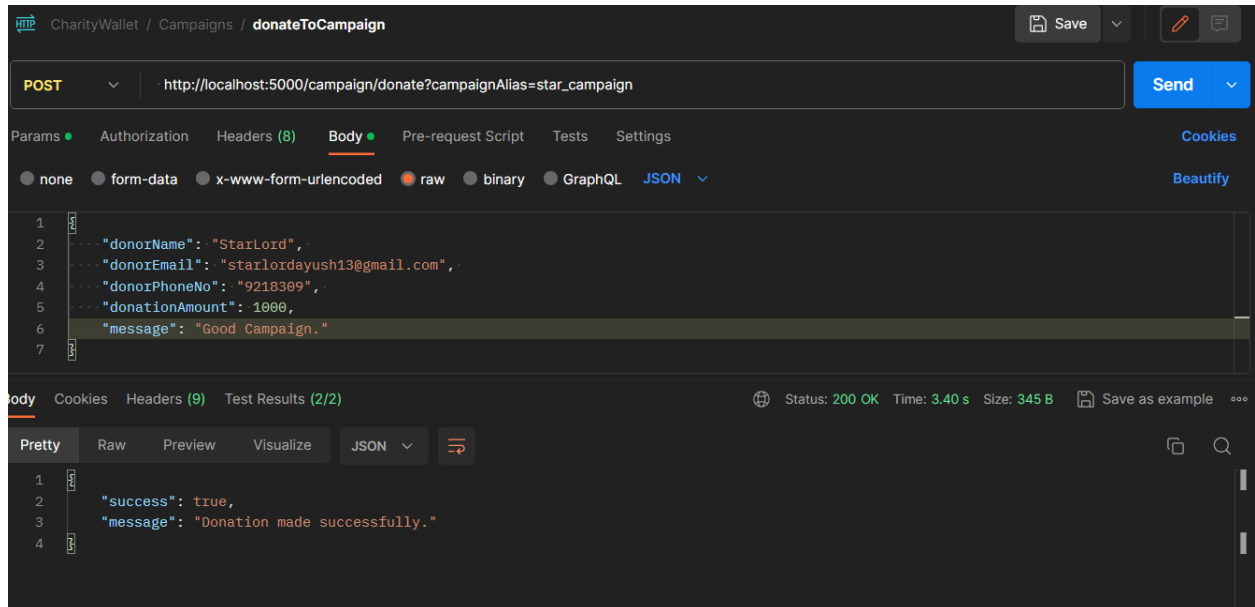
The Login Processes for Organizers and Fundraisers is similar. Due to time constraints, I can't show the processes.

Donation Process:

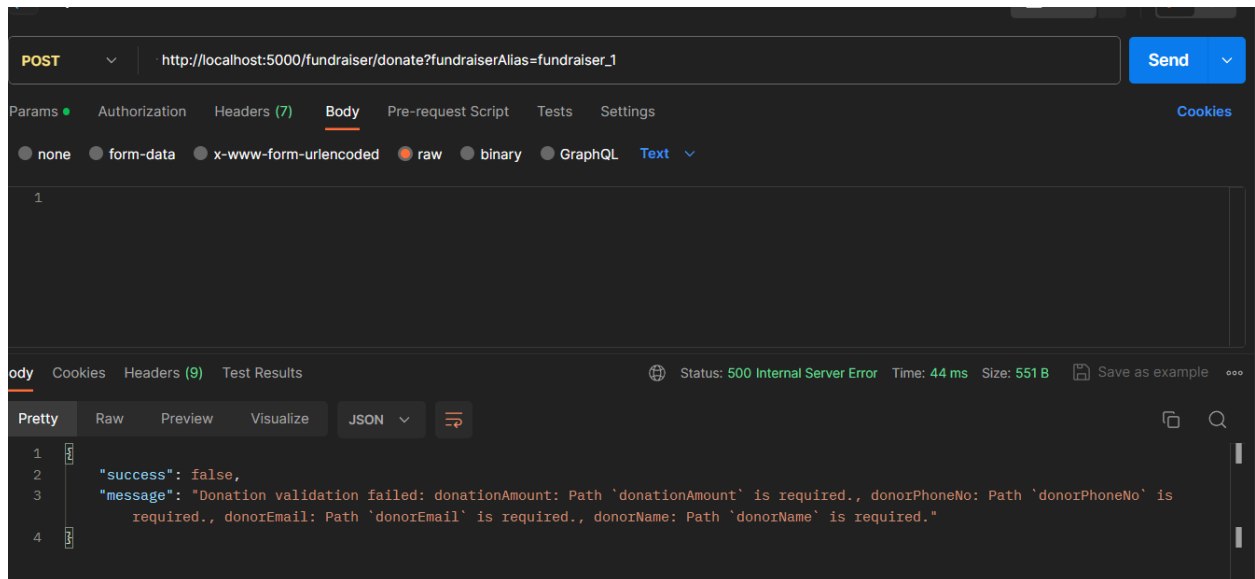
1) Donate to NGO



2) Donate to Campaign

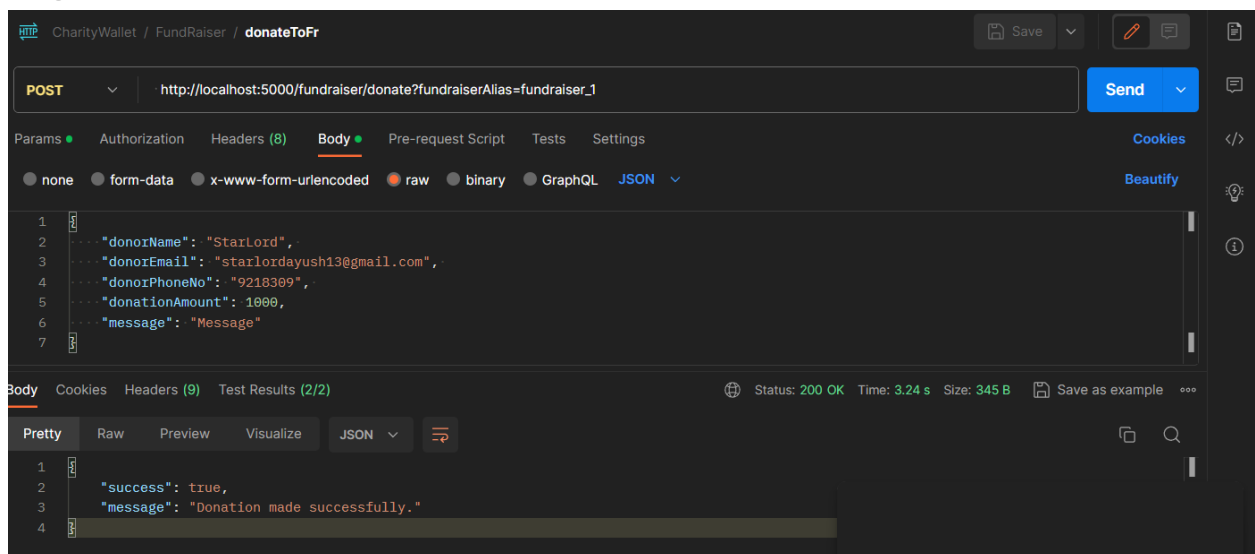


3) Donate to Fundraiser

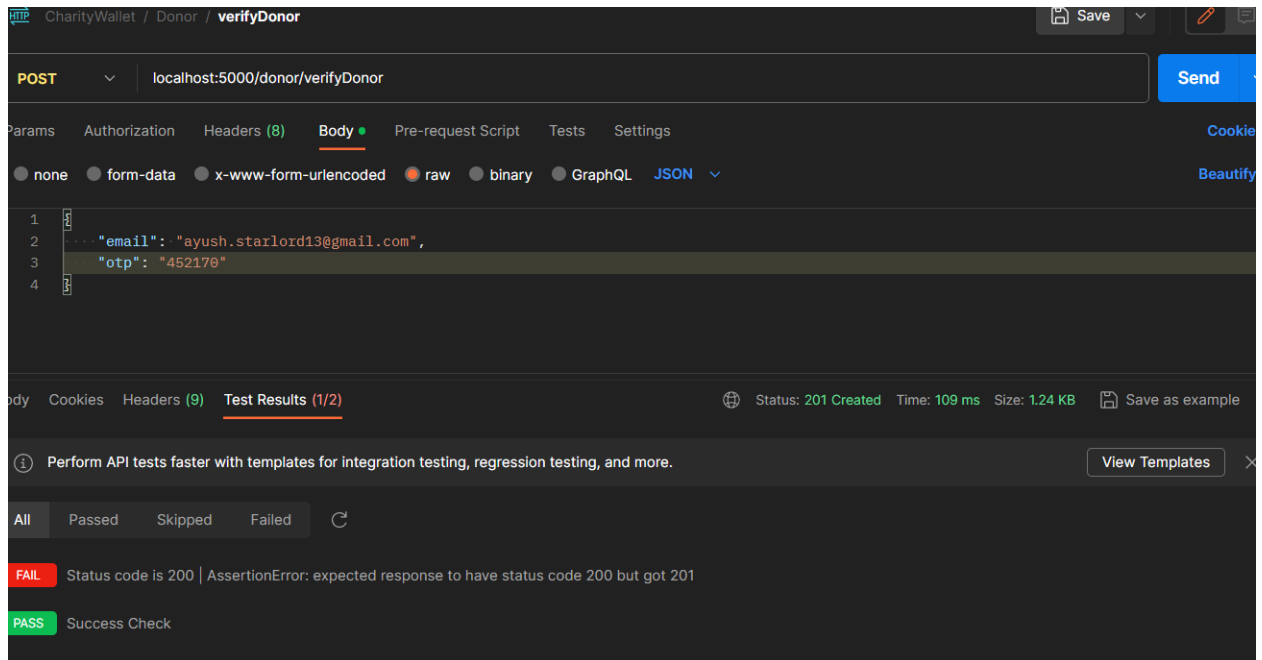


In case of an empty body.

In general Case:



4) View Donor Records



In this route, the body is the donation records of the User.

I also ran a whole environment test with certain routes for which I have attached the file ***'UnitTest.json'***.