Exploring and Testing APIs with Postman

1. Introduction

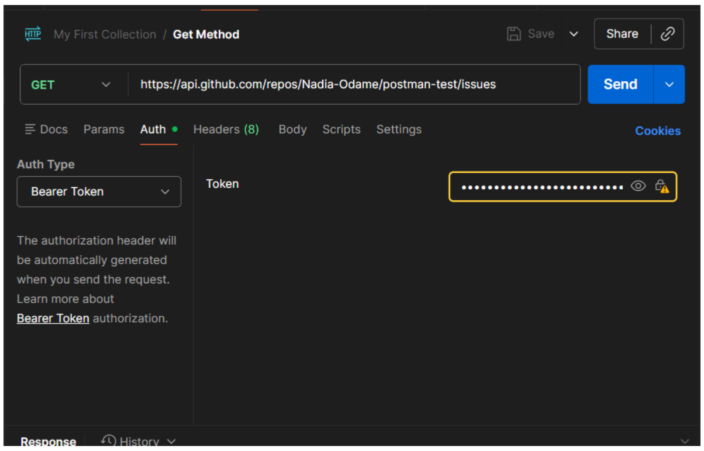
This assignment focussed on exploring and testing a real-world API using Postman with emphasis on authentication, CRUD operations, environment variables and automation. For this task, the **GitHub Rest API** (Issues endpoint) was selected because it supports authenticated access and full CRUD operations making it suitable for practical API testing and documentation.

The work was carried out entirely in Postman using a Personal Access Token (PAT) for authentication. All requests were tested, responses verified and environment variables used to make the workflow dynamic and reusable.

2. API Choice and Base URL

- **API Used**: Github Rest API -Issues
- **Base URL**: https://api.github.com
- **Repository Used**: Nadia-Odame/postman-test
- **Primary Resources**: Issues

This API allows authenticated users to create, read, update, label, comment on and delete issue related resources.


My First Collection / Get Method

GET https://api.github.com/repos/Nadia-Odame/postman-test/issues

Auth Type: Bearer Token

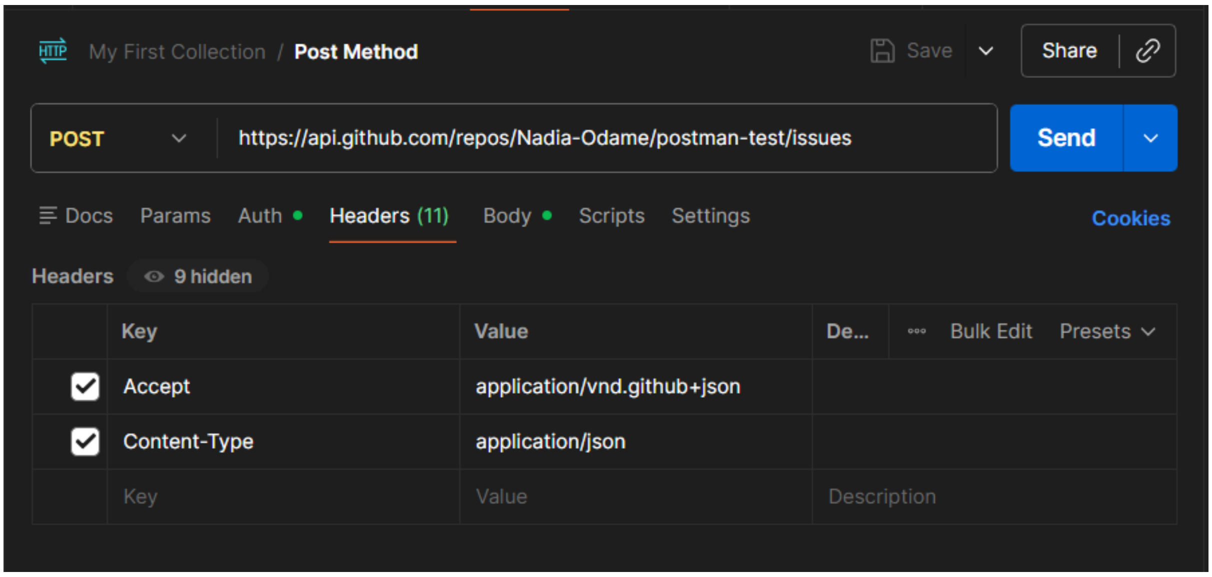
Token: ••••••••••••••••••••••••

The authorization header will be automatically generated when you send the request. Learn more about Bearer Token authorization.

3. Authentication Method

Authentication Type: Personal Access Token (Bearer Token)

Authentication was handled using a GitHub Personal Access Token stored securely as an environment variable in Postman.



POST https://api.github.com/repos/Nadia-Odame/postman-test/issues

Headers

	Key	Value	De...
✓	Accept	application/vnd.github+json	
✓	Content-Type	application/json	
	Key	Value	Description

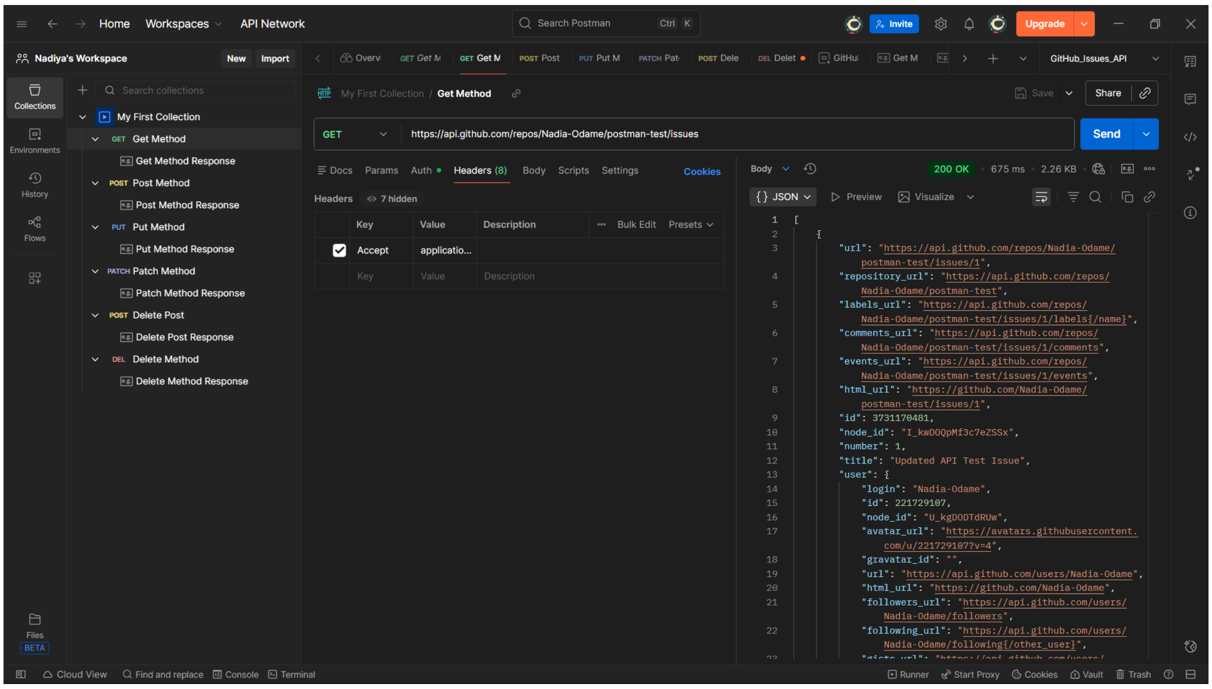
Headers Used in All Requests:

- **Authorization**: Bearer {{auth_token}}
- **Accept**: application/vnd.github+json
- **Content-Type**: application/json

The token was manually generated from GitHub and stored in the Postman environment as auth_token. This token was then reused across all requests causing variable interpolation. (Token automation using pre-request script was not implemented because GitHub PATs are generated externally. However, token reuse via environment variables was successfully demonstrated.)

4. CRUD Operations Demonstration

4.1. GET - Retrieve Issues (READ)



My First Collection / Get Method

GET https://api.github.com/repos/Nadia-Odame/postman-test/issues

Headers 7 hidden

	Key	Value	Description
✓	Accept	applicatio...	
	Key	Value	Description

Body 200 OK • 675 ms • 2.26 KB

```
[
    {
        "url": "https://api.github.com/repos/Nadia-Odame/
            postman-test/issues/1",
        "repository_url": "https://api.github.com/repos/
            Nadia-Odame/postman-test",
        "labels_url": "https://api.github.com/repos/
            Nadia-Odame/postman-test/issues/1/labels{/name}",
        "comments_url": "https://api.github.com/repos/
            Nadia-Odame/postman-test/issues/1/comments",
        "events_url": "https://api.github.com/repos/
            Nadia-Odame/postman-test/issues/1/events",
        "html_url": "https://github.com/Nadia-Odame/
            postman-test/issues/1",
        "id": 3731170481,
        "node_id": "I_kwDOQpMf3c7eZSSx",
        "number": 1,
        "title": "Updated API Test Issue",
        "user": {
            "login": "Nadia-Odame",
            "id": 221729107,
            "node_id": "U_kgDODTdRUw",
            "avatar_url": "https://avatars.githubusercontent.
                com/u/221729107?v=4",
            "gravatar_id": "",
            "url": "https://api.github.com/users/Nadia-Odame",
            "html_url": "https://github.com/Nadia-Odame",
            "followers_url": "https://api.github.com/users/
                Nadia-Odame/followers",
            "following_url": "https://api.github.com/users/
                Nadia-Odame/following{/other_user}",
```

Endpoint:

GET /repos/{owner}/{repo}/issues

Full URL:

https://api.github.com/repos/Nadia-Odame/postman-test/issues

Request Body: None

Response:

- Returned an array of issues in JSON format
- Initially returned an empty array ([]) before any issues were created

This request confirmed successful authentication and access to repository data.

4.2. POST - Create an Issue (CREATE)

Endpoint:

POST /repos/{owner}/{repo}/issues

Full URL:

https://api.github.com/repos/Nadia-Odame/postman-test/issues

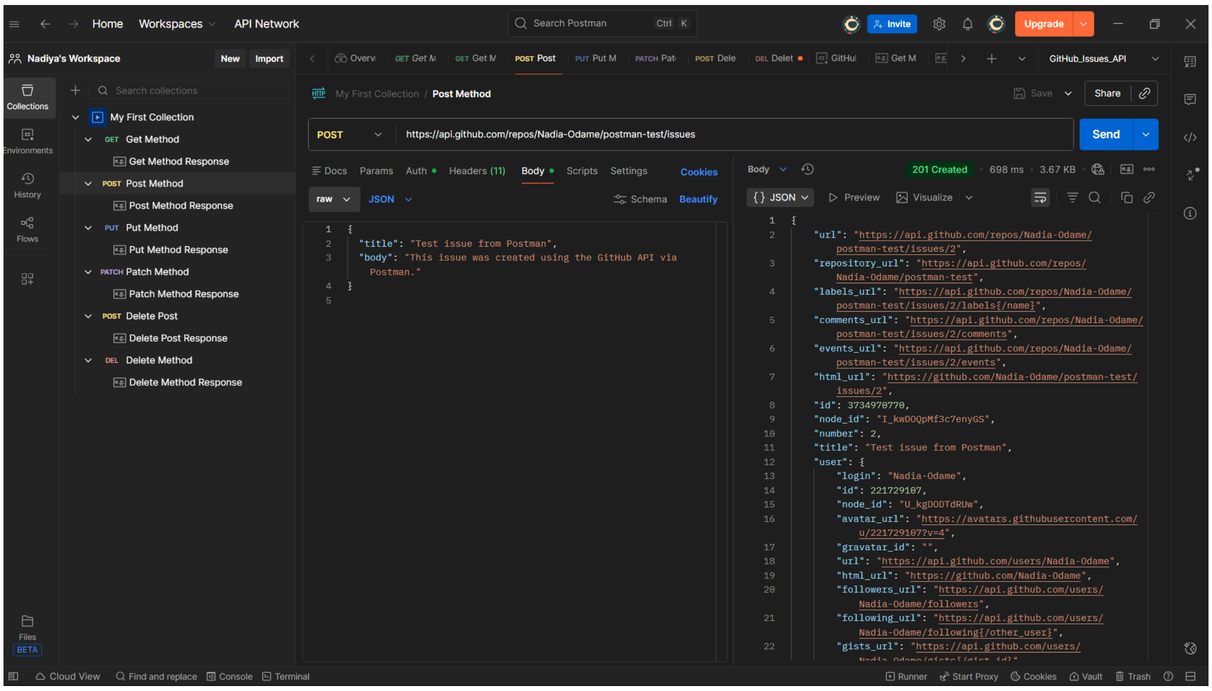
Request Body:

{

"title": "Test issue from Postman",

"body": "This issue was created using the GitHub API via Postman."

}


POST https://api.github.com/repos/Nadia-Odame/postman-test/issues

```
{
  "title": "Test issue from Postman",
  "body": "This issue was created using the GitHub API via
    Postman."
}
```

201 Created · 698 ms · 3.67 KB

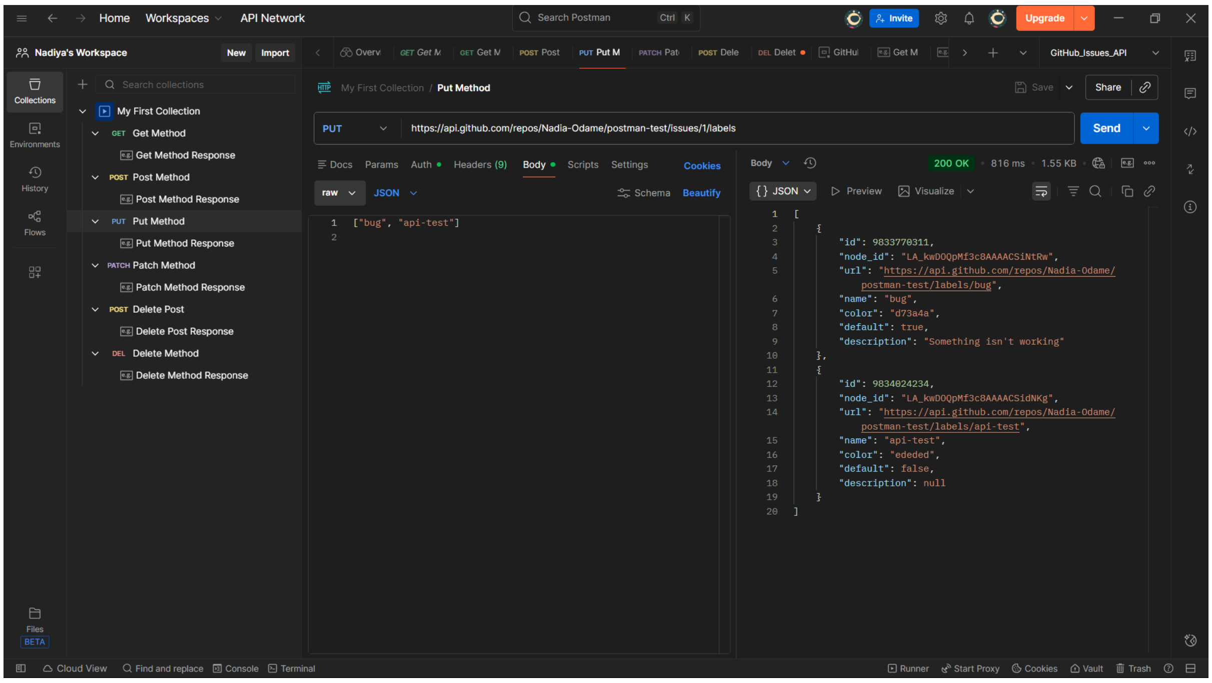
```
{
    "url": "https://api.github.com/repos/Nadia-Odame/
      postman-test/issues/2",
    "repository_url": "https://api.github.com/repos/
      Nadia-Odame/postman-test",
    "labels_url": "https://api.github.com/repos/Nadia-Odame/
      postman-test/issues/2/labels{/name}",
    "comments_url": "https://api.github.com/repos/Nadia-Odame/
      postman-test/issues/2/comments",
    "events_url": "https://api.github.com/repos/Nadia-Odame/
      postman-test/issues/2/events",
    "html_url": "https://github.com/Nadia-Odame/postman-test/
      issues/2",
    "id": 3734970770,
    "node_id": "I_kwDOQpMf3c7enyGS",
    "number": 2,
    "title": "Test issue from Postman",
    "user": {
        "login": "Nadia-Odame",
        "id": 221729107,
        "node_id": "U_kgDODTdRUw",
        "avatar_url": "https://avatars.githubusercontent.com/
          u/221729107?v=4",
        "gravatar_id": "",
        "url": "https://api.github.com/users/Nadia-Odame",
        "html_url": "https://github.com/Nadia-Odame",
        "followers_url": "https://api.github.com/users/
          Nadia-Odame/followers",
        "following_url": "https://api.github.com/users/
          Nadia-Odame/following{/other_user}",
        "gists_url": "https://api.github.com/users/
```

Response Highlights:

- HTTP Status: 201 created
- Issue number automatically generated (#1)
- Issue state set to open
    - Html_url provided for viewing the issue on GitHub

This confirmed successful creation of a new resource via the API.

4.3. PUT - Add Labels to an Issue (UPDATE)



PUT https://api.github.com/repos/Nadia-Odame/postman-test/issues/1/labels

```
["bug", "api-test"]
```

200 OK • 816 ms • 1.55 KB

```
[
    {
        "id": 9833770311,
        "node_id": "LA_kwDOQpMf3c8AAAACSiNtRw",
        "url": "https://api.github.com/repos/Nadia-Odame/
            postman-test/labels/bug",
        "name": "bug",
        "color": "d73a4a",
        "default": true,
        "description": "Something isn't working"
    },
    {
        "id": 9834024234,
        "node_id": "LA_kwDOQpMf3c8AAAACSidNKg",
        "url": "https://api.github.com/repos/Nadia-Odame/
            postman-test/labels/api-test",
        "name": "api-test",
        "color": "ededed",
        "default": false,
        "description": null
    }
]
```

Endpoint:

PUT /repos/{owner}/{repo}/issues/{issue_number}/labels

Full URL:

https://api.github.com/repos/Nadia-Odame/postman-test/issues/1/labels

Request Body:

["bug", "api-test"]

Response Highlights:

- Returned an array of label objects
- Confirmed that both labels were successfully attached to the issue

This request demonstrated updating an existing resource by associating metadata (labels).

4.4. PATCH - Update Issue Title (UPDATE)

Endpoint:

PATCH /repos/{owner}/{repo}/issues/{issue_number}

Full URL:

https://api.github.com/repos/Nadia-Odame/postman-test/issues/1

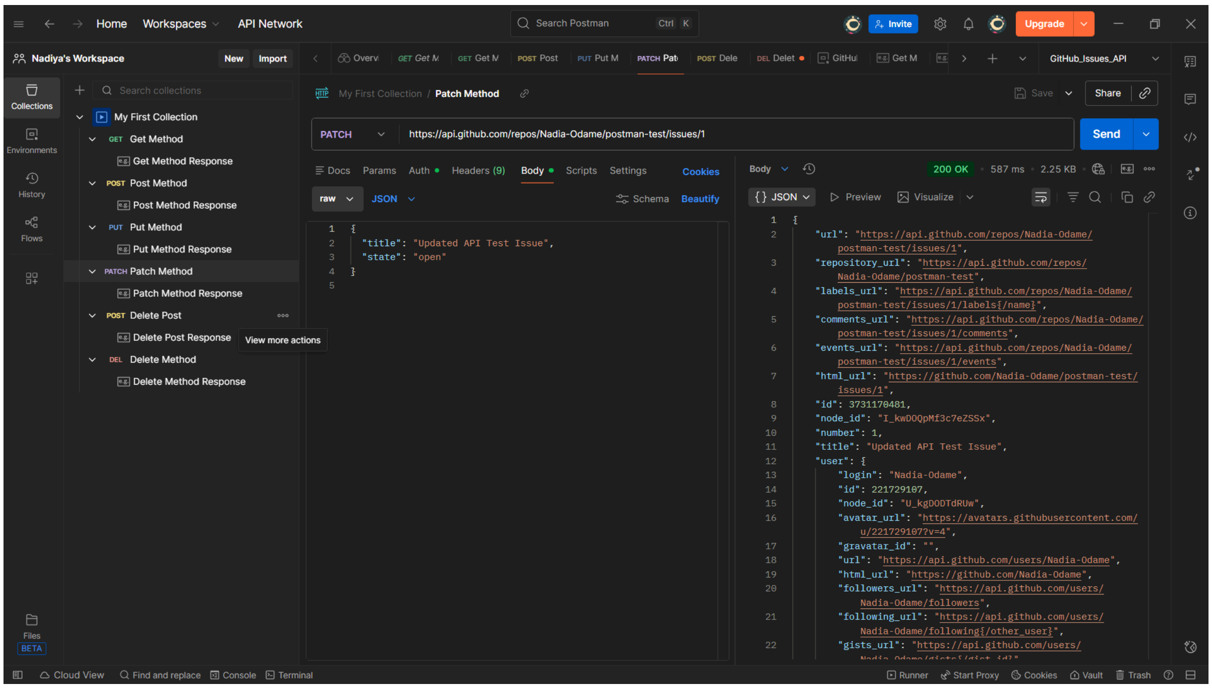
Request body:

{

"title": "Updated API Test Issue",

"state": "open"

}



PATCH https://api.github.com/repos/Nadia-Odame/postman-test/issues/1

```
{
  "title": "Updated API Test Issue",
  "state": "open"
}
```

200 OK • 587 ms • 2.25 KB

```
{
    "url": "https://api.github.com/repos/Nadia-Odame/
        postman-test/issues/1",
    "repository_url": "https://api.github.com/repos/
        Nadia-Odame/postman-test",
    "labels_url": "https://api.github.com/repos/Nadia-Odame/
        postman-test/issues/1/labels{/name}",
    "comments_url": "https://api.github.com/repos/Nadia-Odame/
        postman-test/issues/1/comments",
    "events_url": "https://api.github.com/repos/Nadia-Odame/
        postman-test/issues/1/events",
    "html_url": "https://github.com/Nadia-Odame/postman-test/
        issues/1",
    "id": 3731170481,
    "node_id": "I_kwDOQpMf3c7eZSSx",
    "number": 1,
    "title": "Updated API Test Issue",
    "user": {
        "login": "Nadia-Odame",
        "id": 221729107,
        "node_id": "U_kgDODTdRUw",
        "avatar_url": "https://avatars.githubusercontent.com/
            u/221729107?v=4",
        "gravatar_id": "",
        "url": "https://api.github.com/users/Nadia-Odame",
        "html_url": "https://github.com/Nadia-Odame",
        "followers_url": "https://api.github.com/users/
            Nadia-Odame/followers",
        "following_url": "https://api.github.com/users/
            Nadia-Odame/following{/other_user}",
        "gists_url": "https://api.github.com/users/
```

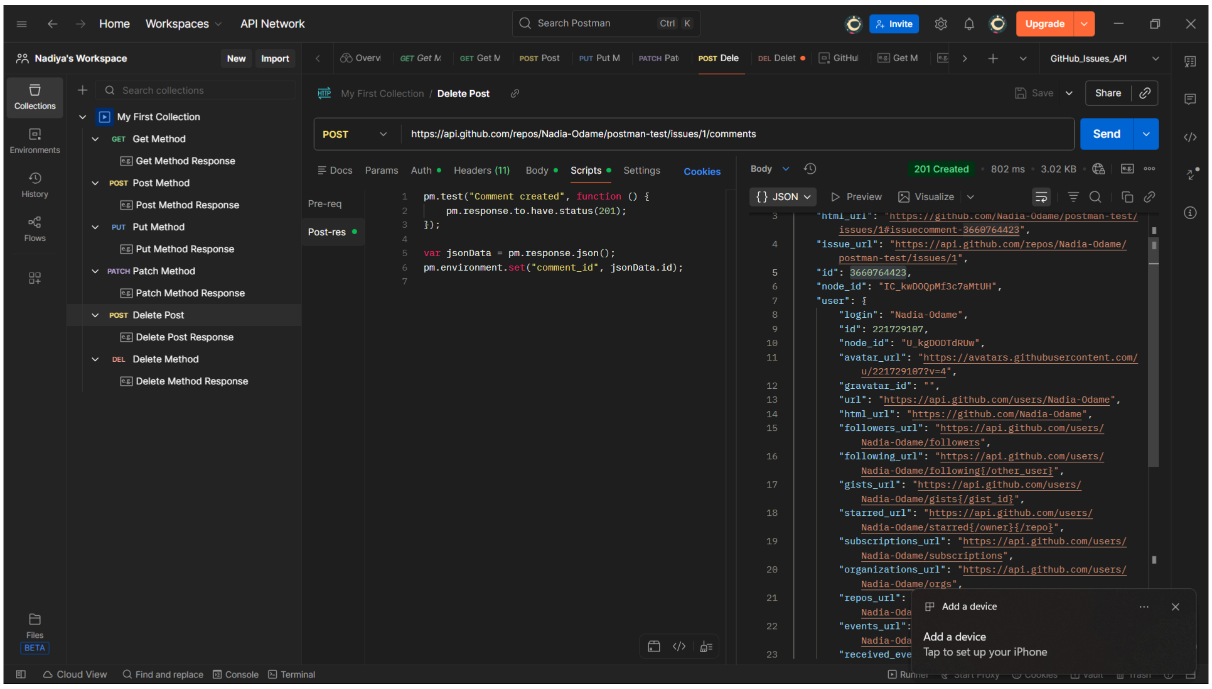
Response Highlights:

- Issue title updated successfully
- Labels remained unchanged
- updated_at timestamp reflected the modification

This showed partial updates using the PATCH method.

4.5. DELETE - Delete a Comment (DELETE)

GitHub does not allow deleting issues via API. Instead, deletion was demonstrated using issue comments which still fulfills the DELETE operation requirement.



POST https://api.github.com/repos/Nadia-Odame/postman-test/issues/1/comments

```
pm.test("Comment created", function () {
    pm.response.to.have.status(201);
});

var jsonData = pm.response.json();
pm.environment.set("comment_id", jsonData.id);
```

```
  "html_url": "https://github.com/Nadia-Odame/postman-test/issues/1#issuecomment-3660764423",
  "issue_url": "https://api.github.com/repos/Nadia-Odame/postman-test/issues/1",
  "id": 3660764423,
  "node_id": "IC_kwDOQpMf3c7aMtUH",
  "user": {
    "login": "Nadia-Odame",
    "id": 221729107,
    "node_id": "U_kgDODTdRUw",
    "avatar_url": "https://avatars.githubusercontent.com/u/221729107?v=4",
    "gravatar_id": "",
    "url": "https://api.github.com/users/Nadia-Odame",
    "html_url": "https://github.com/Nadia-Odame",
    "followers_url": "https://api.github.com/users/Nadia-Odame/followers",
    "following_url": "https://api.github.com/users/Nadia-Odame/following{/other_user}",
    "gists_url": "https://api.github.com/users/Nadia-Odame/gists{/gist_id}",
    "starred_url": "https://api.github.com/users/Nadia-Odame/starred{/owner}{/repo}",
    "subscriptions_url": "https://api.github.com/users/Nadia-Odame/subscriptions",
    "organizations_url": "https://api.github.com/users/Nadia-Odame/orgs",
    "repos_url":
      Nadia-Oda
    "events_url":
      Nadia-Oda
    "received_eve
```

Add a device
Tap to set up your iPhone

Step 1: Create a Comment

Endpoint:

POST /repos/{owner}/{repo}/issues/{issue_number}/comments

Full URL:

https://api.github.com/repos/Nadia-Odame/postman-test/issues/1/comments

Test Script Used:

pm.test("Comment created", function () {

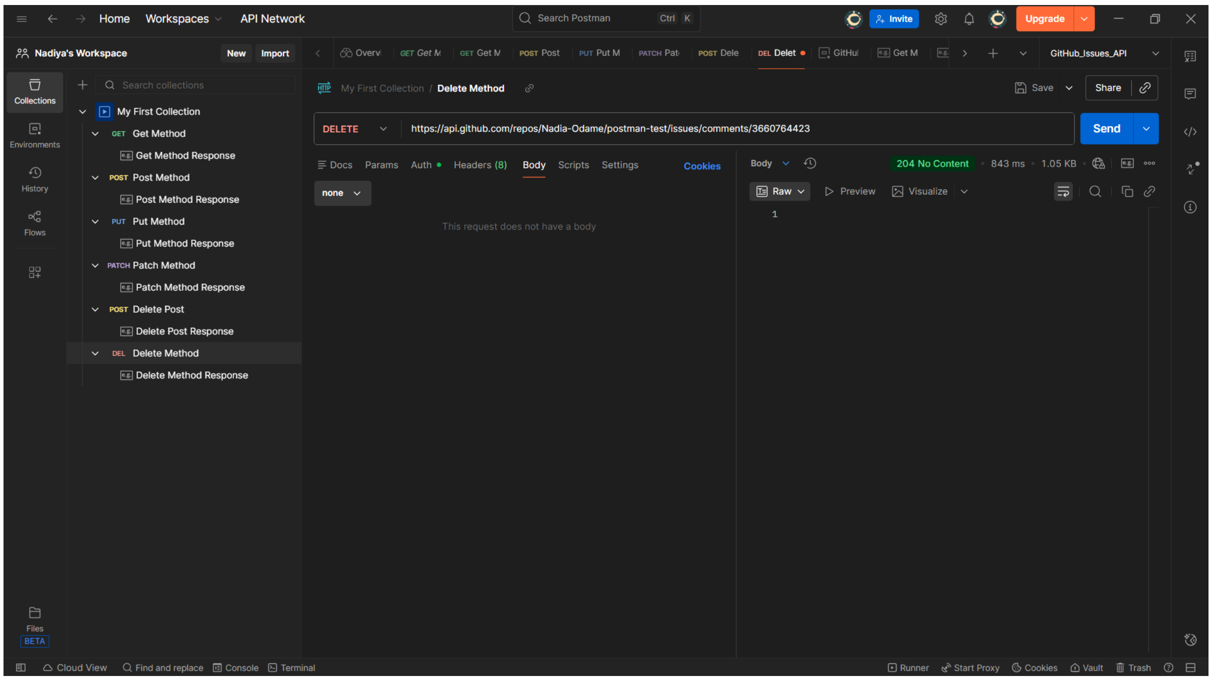
pm.response.to.have.status(201);

});

var jsonData = pm.response.json();

pm.environment.set("comment_id", jsonData.id);

This script verified successful creation and automatically stored the comment_id.



DELETE https://api.github.com/repos/Nadia-Odame/postman-test/issues/comments/3660764423

Body: none

This request does not have a body

204 No Content · 843 ms · 1.05 KB

Step 2: Delete the Comment

Endpoint:

DELETE /repos/{owner}/{repo}/issues/comments/{comment_id}

Full URL:

https://api.github.com/repos/Nadia-Odame/postman-test/issues/comments/{comment_id}

Response Highlights:

- HTTP Status: 204 No content
- Comment successfully deleted

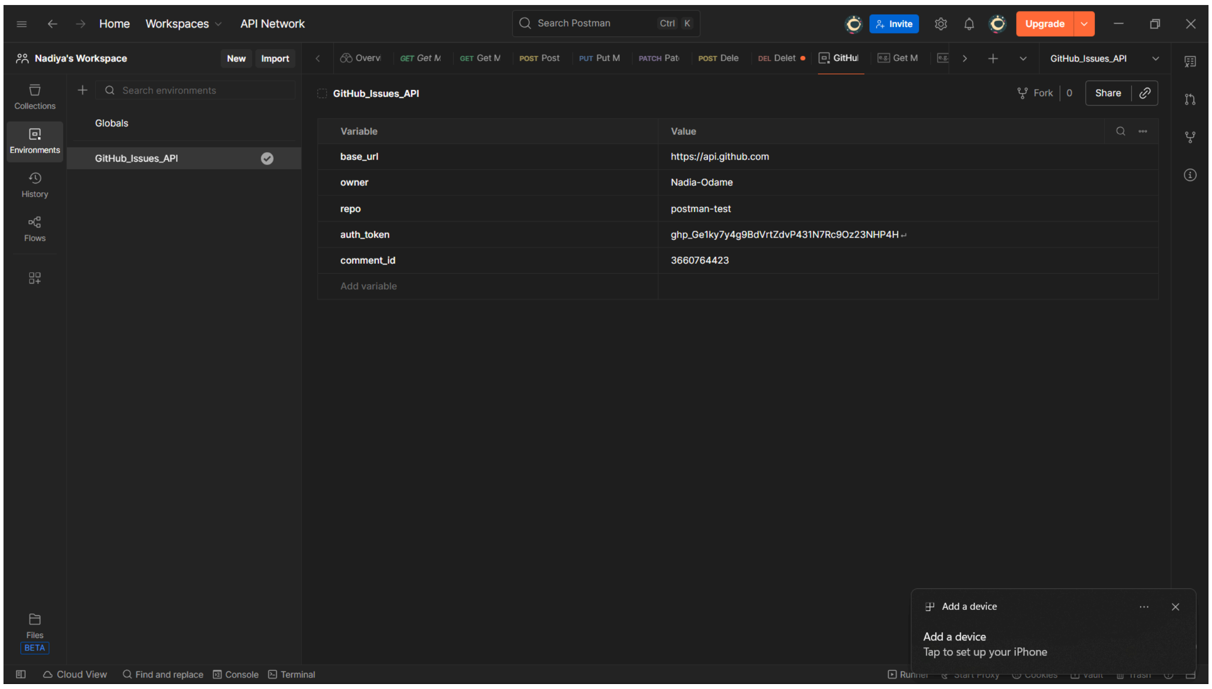
This demonstrated a complete create and delete workflow using dynamic variables.

5. Postman Environment & Variables

A Postman environment (GitHub_Issues_API) was created with the following variables:

Variable Name	**Purpose**
base_url	Stores the Github API base URL

owner	Repository owner
repo	Repository name
auth_token	GitHub Personal Access Token
issue_number	Issue Identifier
comment_id	Comment identifier (auto-filled)


GitHub_Issues_API

Variable	Value
base_url	https://api.github.com
owner	Nadia-Odame
repo	postman-test
auth_token	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
comment_id	3660764423

Using environment variables improved security, reduced duplication and made requests reusable across collections.

6. Automation and Testing

- Basic Test scripts were added to validate response status codes.
- Dynamic data (comment ID) was captured and reused automatically
- Requests were structured to allow execution in sequence using the Collection Runner

7. **Conclusion**

This assignment successfully demonstrated authenticated API interaction using Postman and the GitHub Rest API. All core CRUD operations were implemented and verified using real endpoints. Environment variables and basic automation improved efficiency and showed practical understanding of API testing workflows.