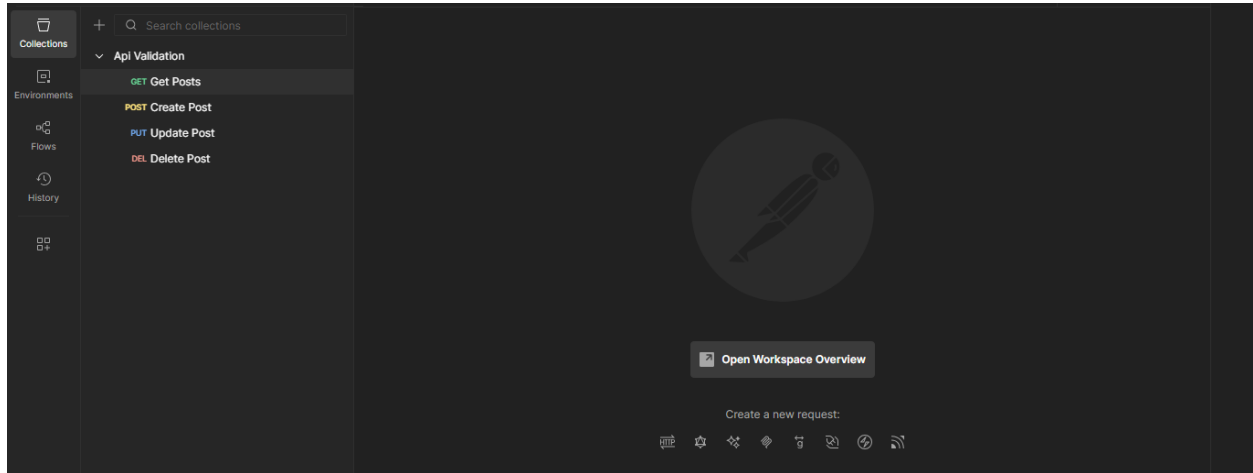


Experiment No. 6

Step 1: Create a Collection

1. Open **Postman**.
2. Create a **New Collection** → Name it API Validation.
3. Inside this collection, we will add 4 requests: GET, POST, PUT, DELETE.



Step 2: GET Request (Read Data)

1. Create a **New Request**.
2. Set method = **GET**.
3. Enter URL:
4. `https://jsonplaceholder.typicode.com/posts`
5. Save as Get Posts inside the collection.
6. Click **Send** → Observe JSON response.

Validation Code (Tests tab):

```
pm.test("Status code is 200", function () {  
    pm.response.to.have.status(200);  
});  
  
pm.test("Response is JSON", function () {  
    pm.response.to.be.json;  
});
```

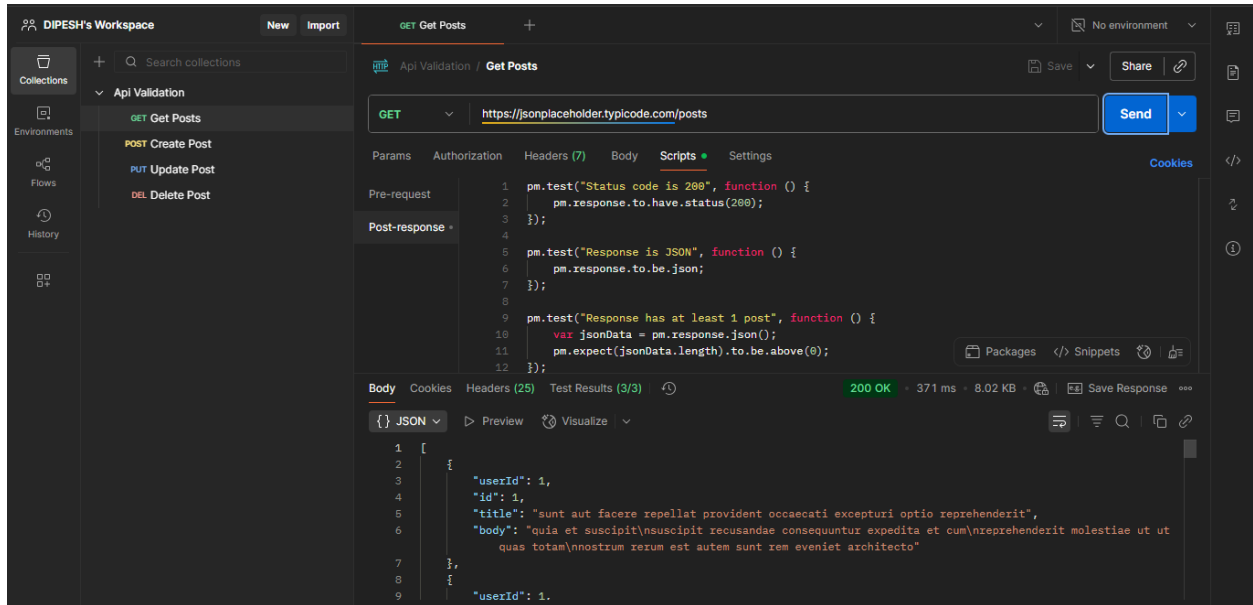
```
pm.test("Response has at least 1 post", function () {
```

Experiment No. 6

```
var jsonData = pm.response.json();

pm.expect(jsonData.length).to.be.above(0);

});
```



Step 3: POST Request (Create Data)

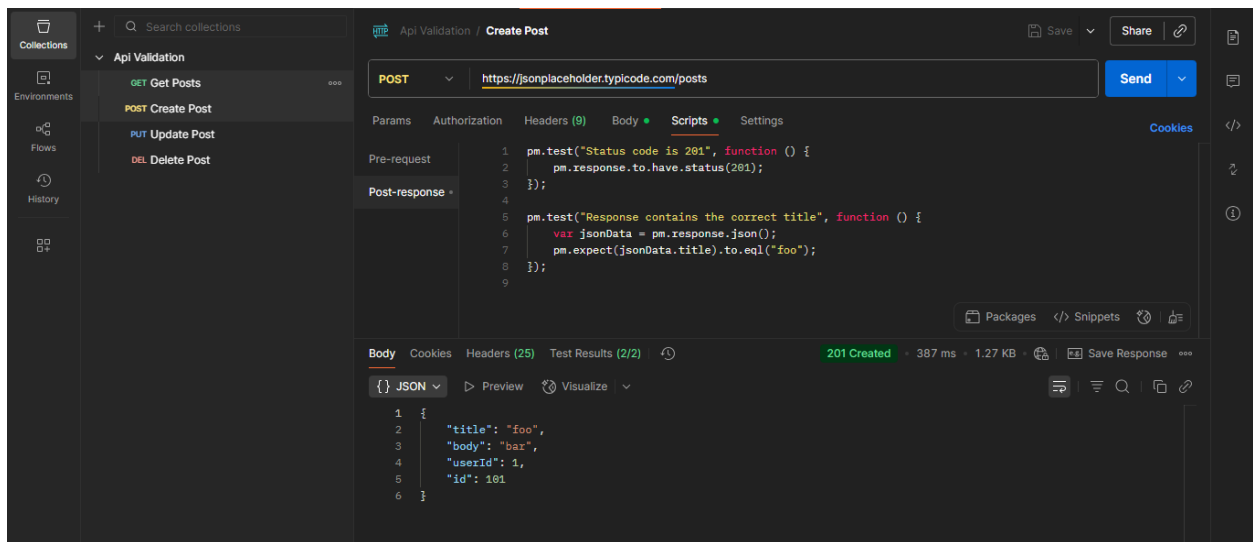
1. Create a **New Request**.
2. Set method = **POST**.
3. Enter URL:
4. <https://jsonplaceholder.typicode.com/posts>
5. Go to **Body** → **raw** → **JSON**.
6. Add JSON body:

```
{
  "title": "foo",
  "body": "bar",
  "userId": 1
}
```
7. Save as Create Post.
8. Click **Send** → Observe new JSON object created.

Validation Code (Tests tab):

Experiment No. 6

```
pm.test("Status code is 201", function () {  
    pm.response.to.have.status(201);  
});  
  
pm.test("Response contains the correct title", function () {  
    var jsonData = pm.response.json();  
    pm.expect(jsonData.title).to.eql("foo");  
});
```



Step 4: PUT Request (Update Data)

1. Create a **New Request**.
2. Set method = **PUT**.
3. Enter URL:
4. `https://jsonplaceholder.typicode.com/posts/1`
5. Go to **Body** → **raw** → **JSON**.
6. Add JSON body:

```
{  
    "id": 1,  
    "title": "updated title",  
    "body": "updated body",
```

Experiment No. 6

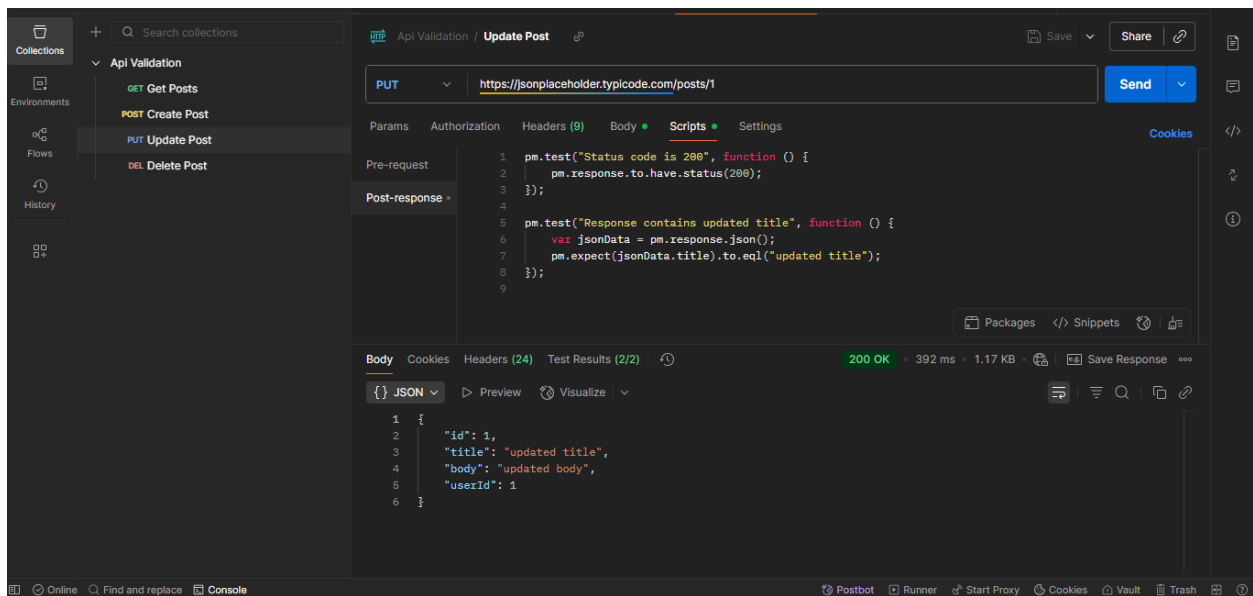
```
"userId": 1
}
```

7. Save as Update Post.
8. Click **Send** → Observe updated JSON object.

Validation Code (Tests tab):

```
pm.test("Status code is 200", function () {
    pm.response.to.have.status(200);
});

pm.test("Response contains updated title", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.title).to.eql("updated title");
});
```



Step 5: DELETE Request (Remove Data)

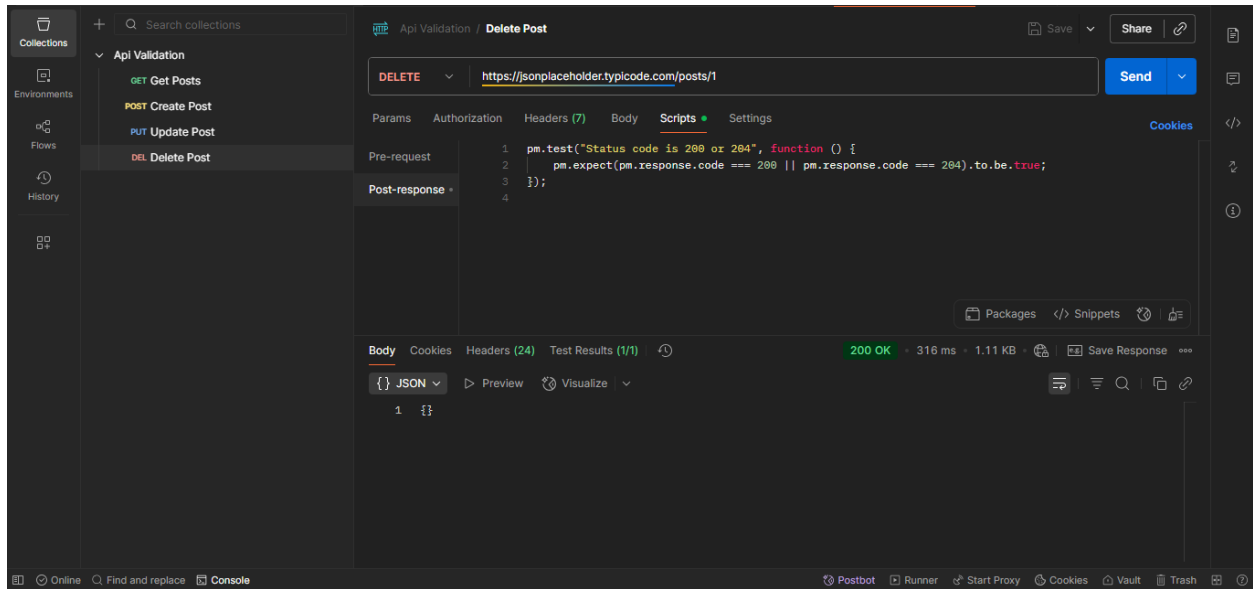
1. Create a **New Request**.
2. Set method = **DELETE**.
3. Enter URL:
4. <https://jsonplaceholder.typicode.com/posts/1>
5. Save as Delete Post.

Experiment No. 6

- Click **Send** → Response should be empty with status 200 OK or 204 No Content.

Validation Code (Tests tab):

```
pm.test("Status code is 200 or 204", function () {  
  
    pm.expect(pm.response.code === 200 || pm.response.code === 204).to.be.true;  
  
});
```



Step 6: Run Collection

- Open **Collection Runner** in Postman.
- Select API Validation collection.
- Run all 4 requests together.
- Check **test results** (green = passed).

Experiment No. 6

Collections

Environments

Flows

History

Search collections

Api Validation

GET Get Posts

POST Create Post

PUT Update Post

DEL Delete Post

Api Validation - Run results

Run today at 10:37:08 PM · View all runs

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	none	1	3s 259ms	8	440 ms

All Tests

Passed (8)

Failed (0)

Skipped (0)

View Summary

PASS

Response is JSON

PASS

Response has at least 1 post

POST

Create Post

https://jsonplaceholder.typicode.com/posts

201 • 272 ms • 1.294 KB • 2

PASS

Status code is 201

PASS

Response contains the correct title

PUT

Update Post

https://jsonplaceholder.typicode.com/posts/1

200 • 1036 ms • 1.199 KB • 2

PASS

Status code is 200

PASS

Response contains updated title

DELETE

Delete Post

https://jsonplaceholder.typicode.com/posts/1

PASS

Status code is 200 or 204

Run completed. All requests executed.