

Piyush

.....

Login

localhost:5173

New tab Mail - PIYUSH NA... Imported From...

Dimensions: Responsive 381 x 600 100% No throttling 'Save-Data' default

Elements Console Sources Network Performance Assistant

react-dom-client.js?v=geas7f27:20181

Download the React DevTools for a better development experience:
<https://react.dev/link/react-devtools>

Username	App.jsx:15
piyush nanda	App.jsx:16
112233	App.jsx:15
Username: Piyush Nanda	App.jsx:15
Password: 112233	App.jsx:16
Username: Abhishek	App.jsx:15
Password: 99999	App.jsx:16

Piyush Nanda
.....
Login

Piyush Nanda's Workspace

New Import

Get DEL New F Experiment POST Add

No environment

+

Search collections

Experiment_6.1

Experiment_8

POST Add Product

GET Get Products

DEL Delete Product

PUT Update

HTTP Experiment_8 / Add Product

Save Share

POST http://localhost:3000/login

Params Auth Headers (9) Body Scripts Settings Cookies

raw JSON

1 {
2 "username": "testuser",
3 "password": "password123"
4 }

Body 200 OK 69 ms 415 B

{} JSON Preview Visualize

1 {"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJpZCI6MSwidXNlcj0hbWUiOiJ0ZXN0dXNlciIsImhdCI6MTc2MDgwODk5
MywiZXhwIjoxNzYwODEyNTkzfQ.
SUoGPY310-dNTG8Fho0wBz09WYV4BLVAaOsGn3DpF0A"}

History

Flows

Schema Beautify

This screenshot shows a REST API testing interface. The left sidebar displays collections, environments, flows, and history. The main workspace shows an experiment named 'Experiment_8' with a sub-section 'Add Product'. A POST request is being made to the URL 'http://localhost:3000/login'. The 'Body' tab is selected, showing a JSON payload with 'username' and 'password' fields. The response status is '200 OK' with a response time of 69 ms and a size of 415 B. The response body is a JSON object containing a single key 'token' with a long string value.

Piyush Nanda's Workspace

New Import

Experiment_8 / Get Products

GET http://localhost:3000/protected

Auth Type: Bearer Token

Token: [REDACTED]

Body: { } JSON

200 OK 11 ms 358 B

```
1: "message": "You have accessed a protected route!",  
2: "user": {  
3:     "id": 1,  
4:     "username": "testuser",  
5:     "iat": 1760809261,  
6:     "exp": 1760812861
```

This screenshot shows the Postman application interface. The left sidebar displays collections, environments, flows, and history. The main workspace shows an experiment named 'Experiment_8' with a collection named 'Get Products'. A GET request is defined with the URL 'http://localhost:3000/protected'. The 'Auth' tab is selected, showing 'Bearer Token' as the type with a redacted token value. The response body is displayed as JSON, indicating success with a message and a user object containing an ID, username, and JWT tokens.

Piyush Nanda's Workspace

New Import

Experiment_8 / Get Products

GET http://localhost:3000/protected

Auth Type: Bearer Token

Token: [REDACTED]

403 Forbidden

Body: {"message": "Invalid token"}

This screenshot shows a REST API testing interface. The left sidebar contains collections, environments, flows, and history. The main area shows an experiment named 'Experiment_8' with a 'Get Products' endpoint. The method is set to GET and the URL is http://localhost:3000/protected. The 'Auth' tab is selected, showing 'Bearer Token' as the type and a redacted token value. The response status is 403 Forbidden, with the message 'Invalid token' in the body. The interface includes tabs for Params, Headers, Body, Scripts, Settings, and Cookies.

Piyush Nanda's Workspace

New Import < POST Add • GET Get F • DEL C > + - No environment

Collections: + Q Search collections

Environments: Experiment_6.1, Experiment_8

Flows: />

History: ↪

Console

HTTP Experiment_8 / Add Product

POST http://localhost:3000/login Send

Body raw JSON Cookies

```
1 { "username": "adminUser", "password": "admin123" }
```

Body 200 OK 36 ms 468 B

```
{ } JSON > < i
```

```
2 "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwidXNlcm5hbWUiOiJhZG1pbkB1VzZXIiLCJyb2xlIjoiQWRtaW4iLCJpYXQiOjE3NjA40DgwMTIsImV4cCI6MTc2MDg5MTYxMn0.
```

Runner Vault

This screenshot shows the Postman application interface. On the left, there's a sidebar with sections for Collections, Environments, Flows, and History. The main workspace is titled 'Experiment_8 / Add Product'. A POST request is being made to 'http://localhost:3000/login'. The request body is set to 'raw' and 'JSON', containing the following data:

```
1 { "username": "adminUser", "password": "admin123" }
```

The response status is '200 OK' with a response time of '36 ms' and a size of '468 B'. The response body is shown as:

```
{ } JSON > < i
```

```
2 "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwidXNlcm5hbWUiOiJhZG1pbkB1VzZXIiLCJyb2xlIjoiQWRtaW4iLCJpYXQiOjE3NjA40DgwMTIsImV4cCI6MTc2MDg5MTYxMn0.
```

The interface includes tabs for 'Runner' and 'Vault' at the bottom right.

The screenshot shows a REST API testing interface with the following details:

- Overview:** GET Post data, PUT New Request, POST Get data, DEL New Requests, POST Add Product.
- Experiment_8 / Add Product:** HTTP POST http://localhost:3000/login
- Method:** POST
- URL:** http://localhost:3000/login
- Body Type:** raw JSON
- Body Content:**

```
1 {  
2   "username": "admin",  
3   "password": "1234"  
4 }
```
- Response Status:** 200 OK
- Response Time:** 41 ms
- Response Size:** 297 B
- Response Content:**

```
1 {  
2   "message": "Login successful"  
3 }
```

HTTP Experiment_8 / Add Product

Save Share

GET http://localhost:3000/user-profile Send

Params Authorization Headers (10) Body Scripts Settings Cookies

Auth Type: Bearer Token Token 5d-ZsDPUHZHkvgUFOu6D6La01DeM

Body Cookies Headers (7) Test Results 200 OK 11 ms 370 B Save Response

{ } JSON Preview Visualize

```
1 {  
2   "message": "Welcome to your profile, alice",  
3   "user": {  
4     "id": 267,  
5     "username": "alice",  
6     "role": "Moderator",  
7     "iat": 1761117274,  
8     "exp": 1761124474  
}
```

Console Runner Vault

HTTP Experiment_8 / Add Product

POST http://localhost:3000/login

Params Authorization Headers (10) Body Scripts Settings Cookies

Body raw binary GraphQL JSON Schema Beautify

```
1 {  
2   "username": "alice",  
3   "role": "Moderator"  
4 }
```

Body Cookies Headers (7) Test Results 200 OK 12 ms 456 B Save Response

{ } JSON Preview Visualize

```
1 {  
2   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
eyJpZCI6ODEzLCJ1c2VybmFtZSI6ImFsaWNlIiwicm9sZSI6Ik1vZGVyYXRvciiIsImlhCI6MTc2MTExNzM1OCwiZXhwIjoxNzYxM  
Tl0NTU4fQ.R9aeDunpIYIK9W0F2hexmfrAyrYZKN9BiY958xYPifk",  
3   "expiresIn": "2h"
```

The screenshot shows the Postman application interface. On the left, there's a sidebar with icons for Collections, Environments, Flows, and History. The main area has a header bar with 'HTTP Experiment_8 / Add Product' and buttons for Save, Share, and a copy icon.

The main workspace displays a GET request to 'http://localhost:3000/moderator-panel'. The 'Authorization' tab is selected, showing 'Bearer Token' as the auth type and a token value '5d-ZsDPUHZHkvgUFOu6D6La01DeM' with a copy icon.

Below the request, the response details are shown: a green '200 OK' status, a response time of '23 ms', a size of '286 B', and a 'Save Response' button. The response body is displayed as JSON:

```
1 {  
2   "message": "Welcome to the Moderator Panel, alice"  
3 }
```