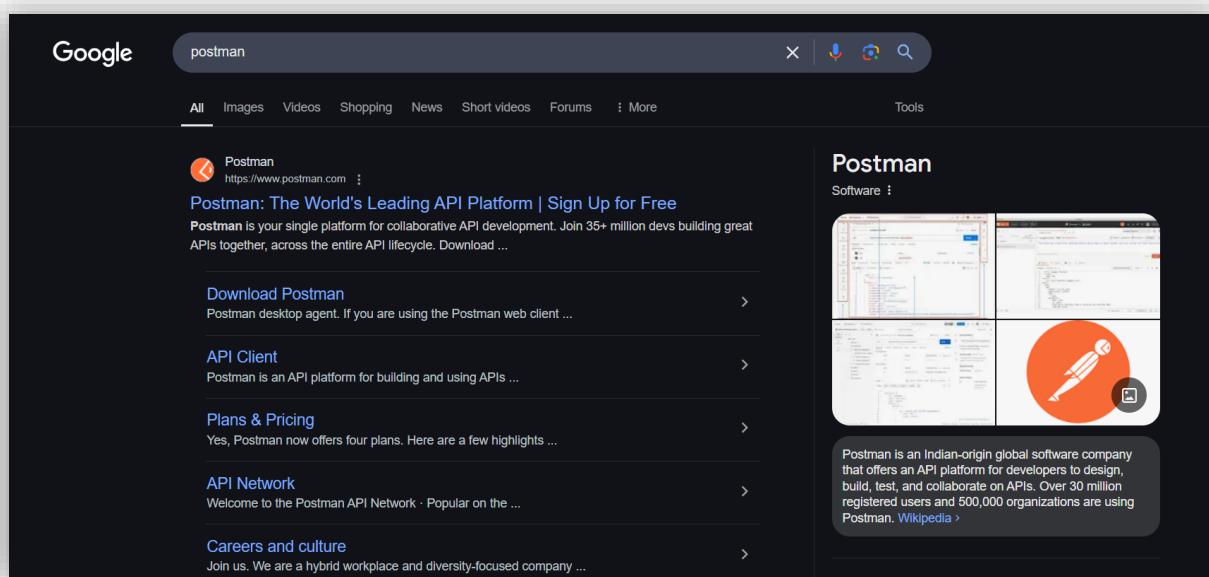


## Working with Postman

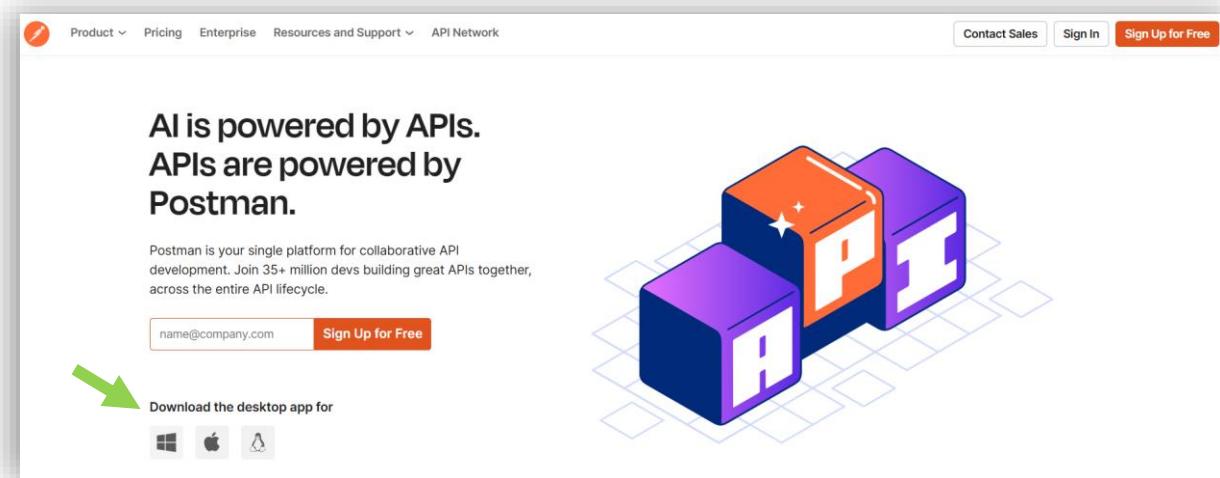
Postman is a powerful tool for testing APIs without writing code. Let's see how to use it:

### 👉 **Downloading Postman**

**Step 1:** Search for "postman download" in your browser and click on the official Postman website.



**Step 2:** Select your operating system (Windows, Mac, or Linux) and click the download button.



**Step 3:** Install the application by following the installation wizard.

The screenshot shows two parts of the Postman interface. On the left, the 'Download Postman' page features a large green arrow pointing to the 'Windows 64-bit' download button. On the right, the Postman web interface shows a workspace named 'Notion's Public Workspace' with a request for 'Notion API / Databases / Retrieve a database'. The request details include a GET method, URL https://api.notion.com/v1/databases/:id, and various parameters like 'Auth' and 'Headers(10)'.

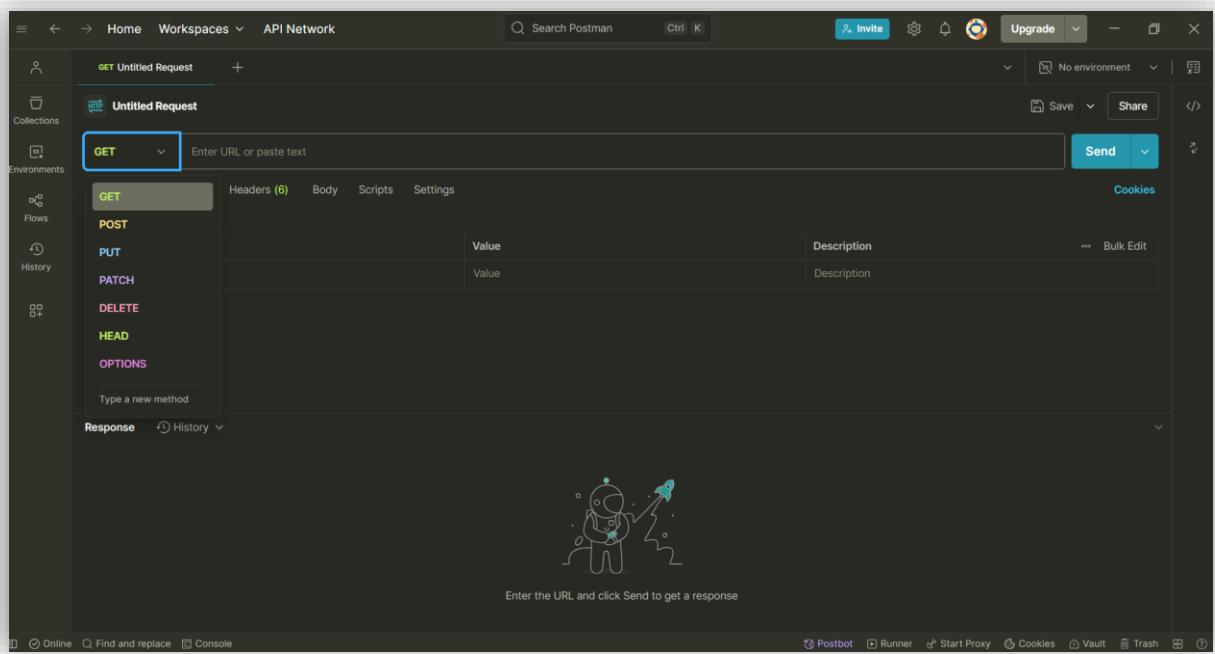
## 👉 Testing APIs with Postman

### ➤ Testing GET Requests

1. Open Postman
2. Click the "+" button to create a new request

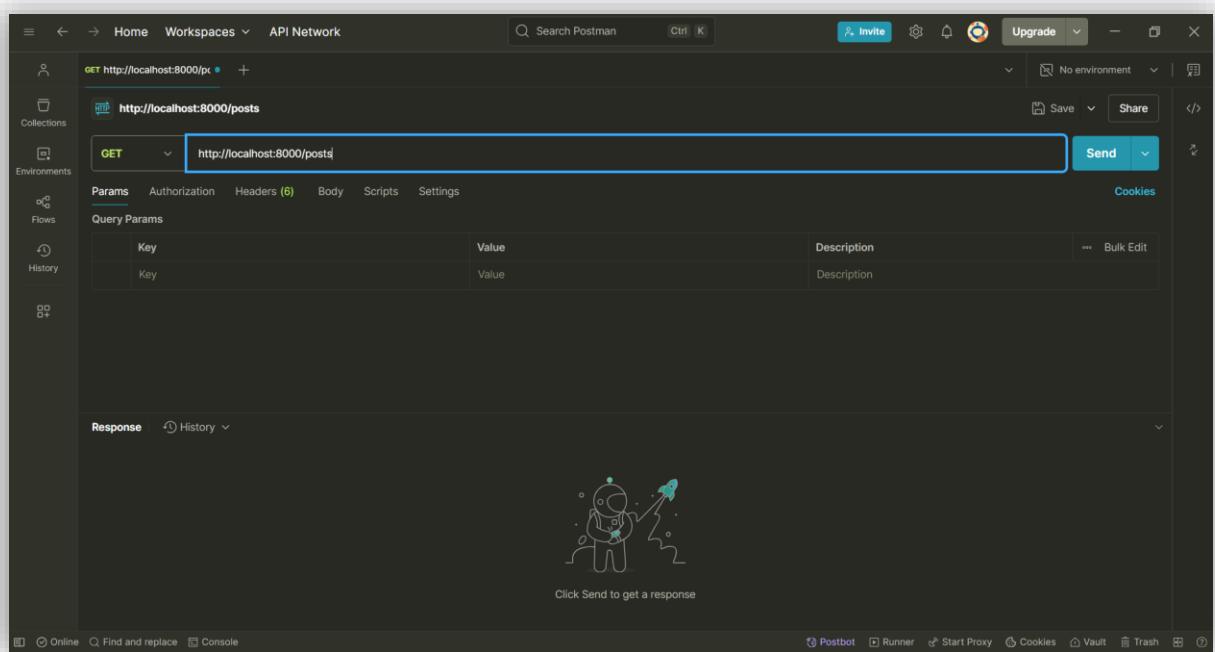
The screenshot shows the Postman web interface with a large green arrow pointing to the '+' button in the top-left corner of the main workspace area. This button is used to create a new API request.

3. Select "GET" from the dropdown menu



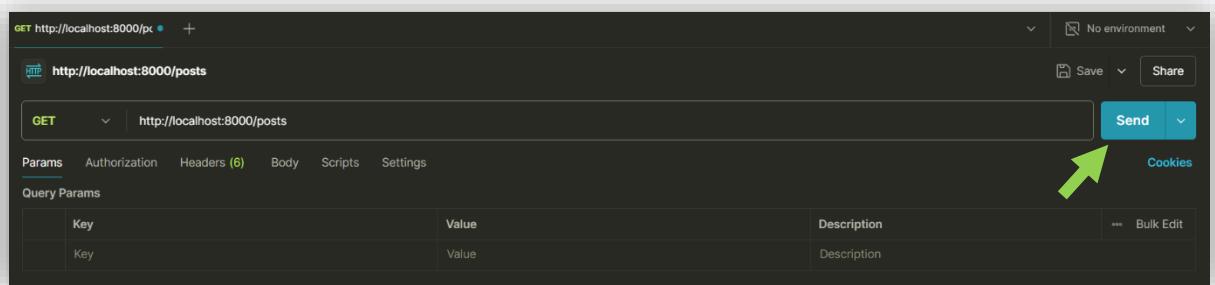
The screenshot shows the Postman application interface. In the top left, there's a sidebar with 'Collections', 'Environments', 'Flows', and 'History'. The main area has a title bar 'GET Untitled Request' and a search bar 'Search Postman'. Below the title bar, there's a dropdown menu set to 'GET', a URL input field 'Enter URL or paste text' containing 'http://localhost:8000/posts', and a 'Send' button. To the right of the URL input is a 'Cookies' button. On the left side of the main area, there's a vertical list of HTTP methods: GET, POST, PUT, PATCH, DELETE, HEAD, and OPTIONS. A small illustration of a character holding a rocket is centered at the bottom. At the very bottom, there are links for 'Online', 'Find and replace', 'Console', and various tools like 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Vault', 'Trash', etc.

4. Enter the URL: <http://localhost:8000/posts>



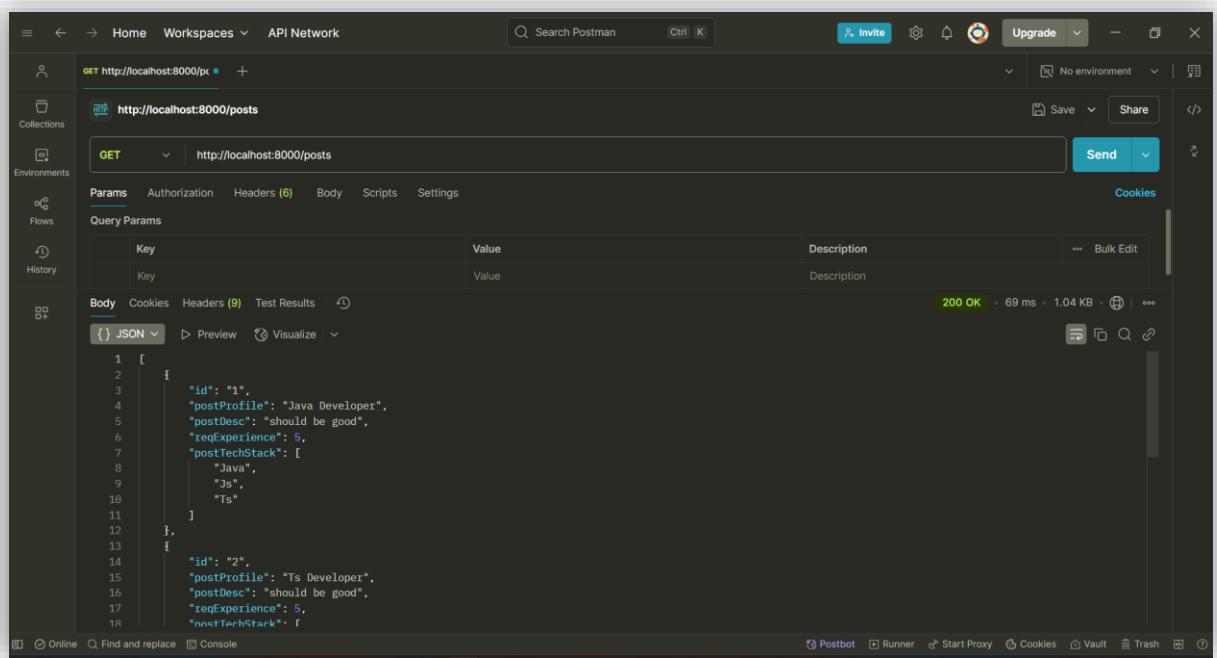
This screenshot shows the Postman interface after entering the URL. The title bar now says 'GET http://localhost:8000/posts'. The URL input field contains 'http://localhost:8000/posts'. The 'Send' button is highlighted with a green arrow pointing to it. The rest of the interface is similar to the previous screenshot, with the same sidebar, methods list, and bottom navigation.

5. Click "Send"



This screenshot shows the Postman interface after clicking the 'Send' button. The URL input field still shows 'http://localhost:8000/posts'. The 'Send' button is now greyed out. A large green arrow points to the 'Send' button. The rest of the interface remains consistent with the previous screenshots.

6. You'll see the JSON data from your fake backend appear in the response section

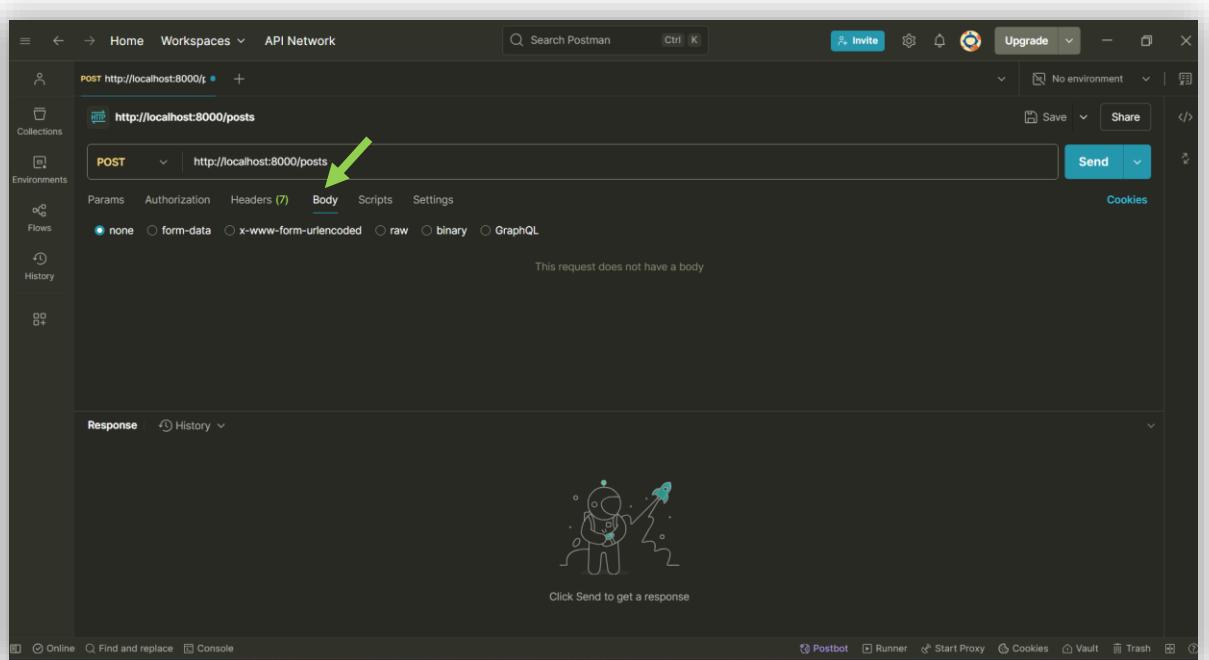


The screenshot shows the Postman interface with a successful GET request to `http://localhost:8000/posts`. The response body is a JSON array containing two posts:

```
[{"id": "1", "postProfile": "Java Developer", "postDesc": "should be good", "reqExperience": 5, "postTechStack": ["Java", "Js", "Ts"]}, {"id": "2", "postProfile": "Ts Developer", "postDesc": "should be good", "reqExperience": 5, "postTechStack": ["Ts"]}]
```

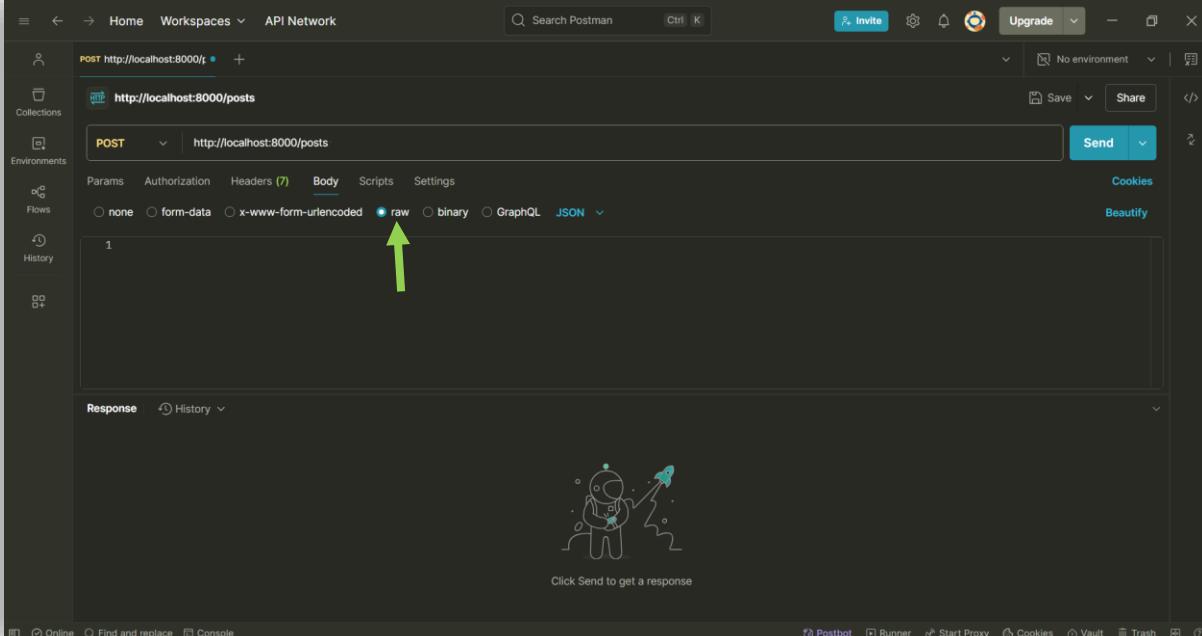
## ➤ Testing POST Requests

1. Create a new request with the "+" button
2. Select "POST" from the dropdown menu
3. Enter the same URL: <http://localhost:8000/posts>
4. Go to the "Body" tab



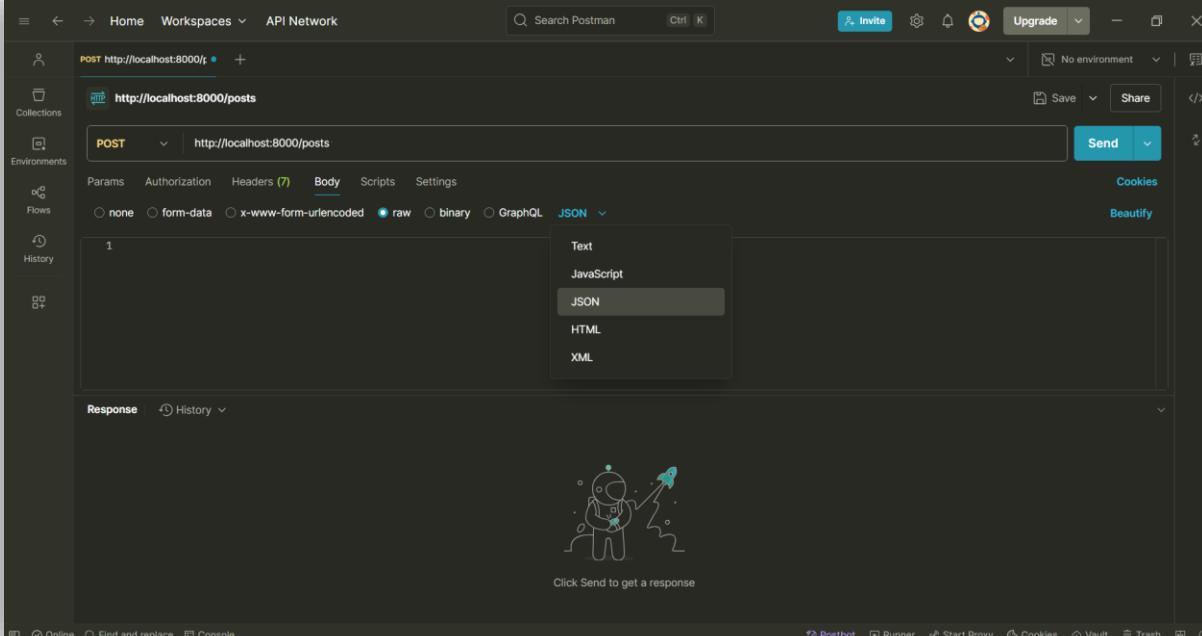
The screenshot shows the Postman interface with a POST request to `http://localhost:8000/posts`. A green arrow points to the "Body" tab, which is currently selected. The "none" option is selected under the Body tab settings. The response section shows a placeholder message: "This request does not have a body".

## 5. Select "raw" option



A screenshot of the Postman application interface. The main window shows a POST request to 'http://localhost:8000/posts'. In the 'Body' tab, there is a dropdown menu with several options: 'none', 'form-data', 'x-www-form-urlencoded', 'raw', 'binary', 'GraphQL', and 'JSON'. A green arrow points upwards from the bottom of the page towards the 'raw' option in the dropdown. The 'raw' option is highlighted with a blue circle. Below the dropdown, there is a text input field containing the number '1'. At the bottom right of the screen, there is a small cartoon character holding a rocket.

## 6. Choose "JSON" from the format dropdown



A screenshot of the Postman application interface, similar to the previous one but with a different selection in the dropdown. The main window shows a POST request to 'http://localhost:8000/posts'. In the 'Body' tab, the dropdown menu is open, and the 'JSON' option is now highlighted with a dark grey background, indicating it has been selected. The other options ('Text', 'JavaScript', 'HTML', 'XML') are visible below it. The rest of the interface, including the text input field containing '1' and the small cartoon character at the bottom right, remains the same.

- Type your JSON data in the editor:

The screenshot shows the Postman application interface. In the top navigation bar, there are links for Home, Workspaces, and API Network. The search bar says "Search Postman". On the right side, there are buttons for Invite, Upgrade, Share, Save, and a dropdown for "No environment". Below the header, there's a sidebar with sections for Collections, Environments, Flows, and History. The main workspace shows a POST request to "http://localhost:8000/posts". The "Body" tab is selected, showing a raw JSON payload:

```
1 {
2   "postProfile": "React Developer",
3   "postDesc": "We are looking for a React developer",
4   "reqExperience": 2,
5   "postTechStack": [
6     "React",
7     "JavaScript",
8     "HTML",
9     "CSS"
10  ]
11 }
```

Below the request, there's a "Response" section which currently displays a cartoon illustration of a person launching a rocket.

- Click "Send"
- The response will show your newly created job post with an ID assigned by the server

## 👉 Remember

- Browsers can only easily send GET requests (by typing URLs)
- Postman lets you test ALL HTTP methods (**GET, POST, PUT, DELETE**)
- It provides a nice interface for adding request bodies, headers, and authentication
- You can save requests for later use and create collections of related requests