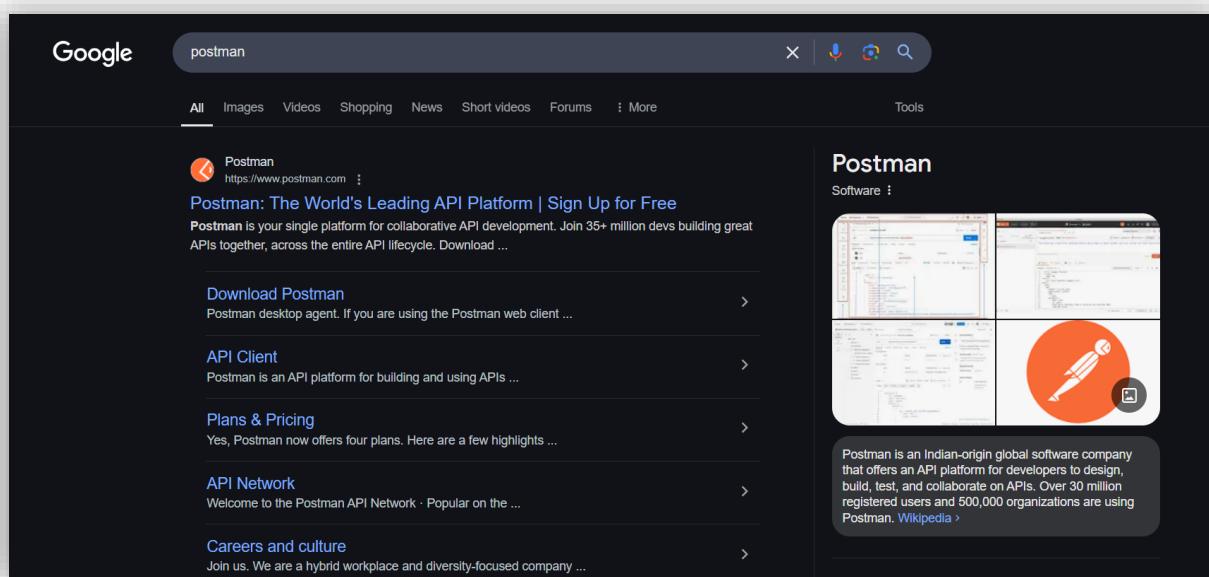


Working with Postman

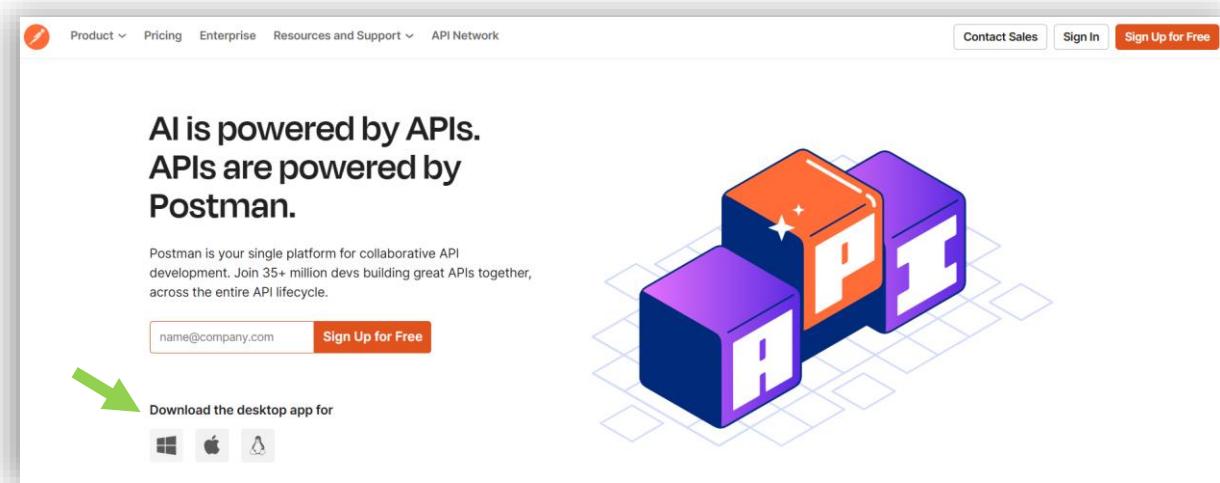
Postman is a powerful tool for testing APIs without writing code. Let's see how to use it:

👉 **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 side-by-side views of Postman. 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 application interface displays a workspace titled 'Notion's Public Workspace' with a request for 'Notion API / Databases / Retrieve a database'. The request details show a GET method, URL 'https://api.notion.com/v1/databases/:id', and various parameters like 'Auth' and 'Headers(10)'.

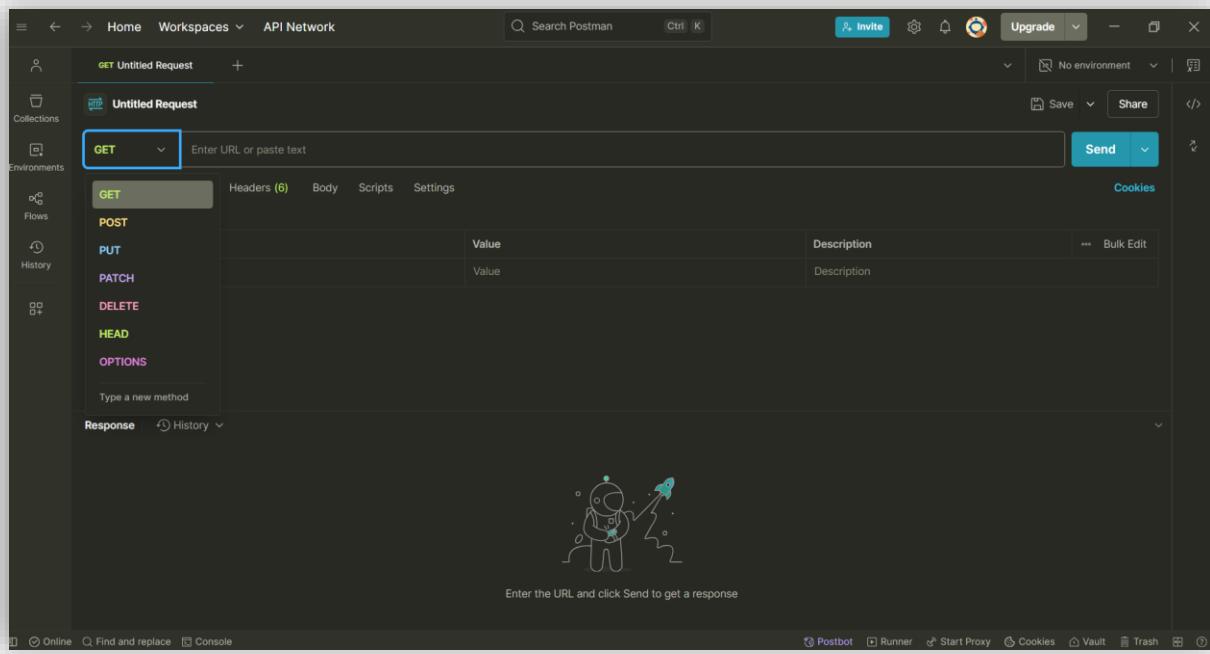
👉 Testing APIs with Postman

➤ Testing GET Requests

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

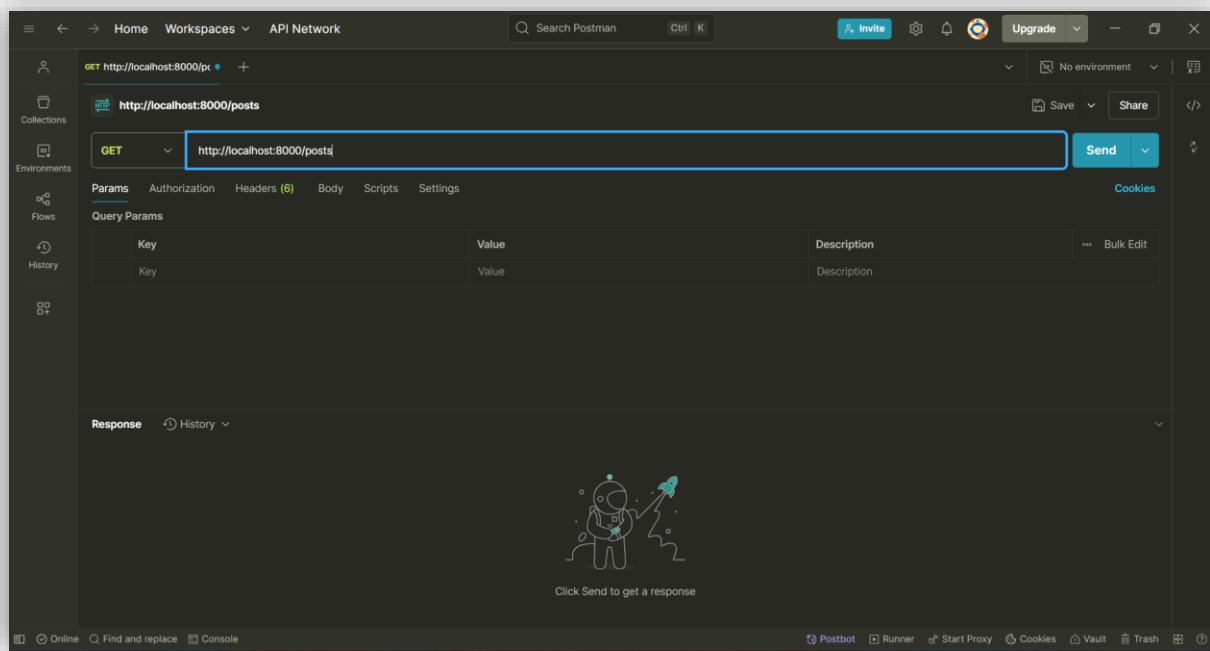
The screenshot shows the Postman application window. A large green arrow points to the '+' button located in the top-left corner of the main workspace area. The workspace itself is dark-themed and features a central icon of a pen writing on a notepad.

3. Select "GET" from the dropdown menu



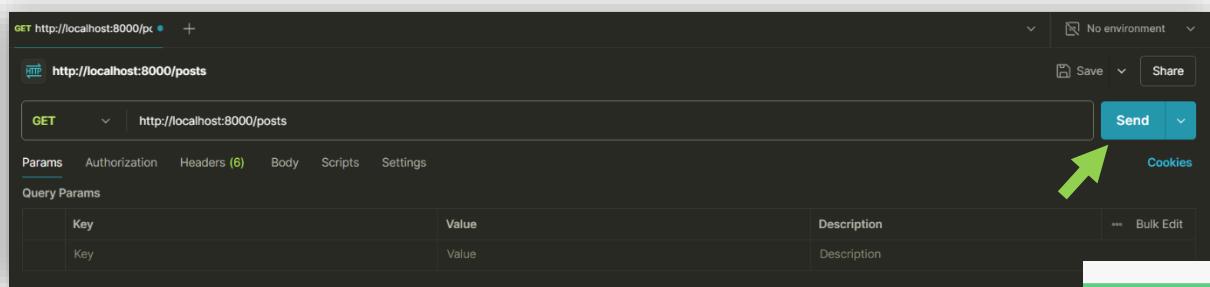
The screenshot shows the Postman application interface. In the top left, there's a sidebar with 'Collections', 'Environments', 'Flows', and 'History'. The main area has a title bar 'GET Untitled Request' and a search bar 'Search Postman'. Below the title bar, there's a dropdown menu set to 'GET', a URL 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' section. 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 person 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, including the sidebar, the list of methods, and the central workspace with the rocket illustration.

5. Click "Send"



This screenshot shows the Postman interface after clicking the 'Send' button. The 'Send' button is now highlighted with a large green arrow pointing to it. The URL input field still shows 'http://localhost:8000/posts'. The rest of the interface remains the same, including the sidebar, the list of methods, and the central workspace with the rocket illustration.

TELUSKO

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 with two elements:

```

1 [
2   {
3     "id": "1",
4     "postProfile": "Java Developer",
5     "postDesc": "should be good",
6     "reqExperience": 5,
7     "postTechStack": [
8       "Java",
9       "Js",
10      "Ts"
11    ]
12  },
13  {
14    "id": "2",
15    "postProfile": "Ts Developer",
16    "postDesc": "should be good",
17    "reqExperience": 5,
18    "postTechStack": [
19      "Ts"
20    ]
21  }
22 ]

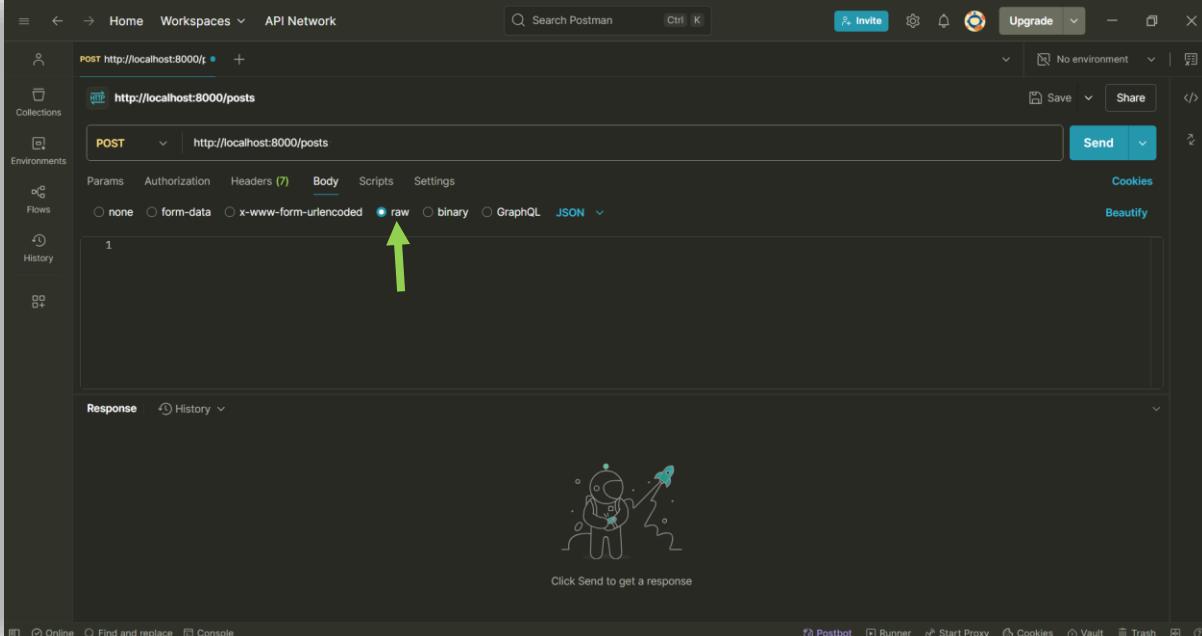
```

➤ 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

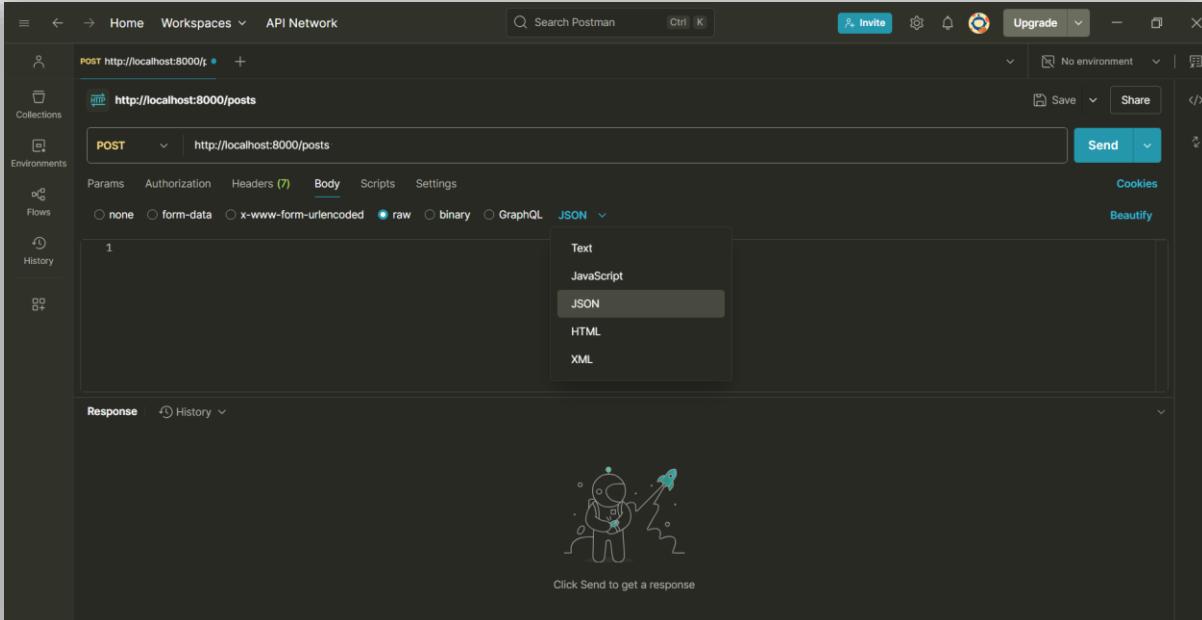
The screenshot shows the Postman interface with a POST request to `http://localhost:8000/posts`. A green arrow points to the "Body" tab. The response section displays the message: "This request does not have a body".

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 dropdown menu now has 'Text', 'JavaScript', 'JSON', 'HTML', and 'XML'. The 'JSON' option is highlighted with a dark grey background, indicating it is selected. The other options ('Text', 'JavaScript', 'HTML', 'XML') are in a lighter grey. The rest of the interface is identical to the first screenshot, including the POST request to 'http://localhost:8000/posts' and the 'raw' body content.

- 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, API Network, and a search bar labeled 'Search Postman'. On the right side of the header, there are buttons for Invite, Upgrade, Share, and a 'No environment' dropdown. Below the header, 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 }
```

The 'Params', 'Authorization', 'Headers', 'Scripts', and 'Settings' tabs are also visible. On the left sidebar, there are sections for Collections, Environments, Flows, and History. At the bottom of the interface, there are buttons for Online, Find and replace, Console, and a 'Send' button.

- 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