

## **Axios & Postman**

### **Axios**

Axios is a promise-based HTTP client for the browser and Node.js. It allows you to make HTTP requests (like GET, POST, PUT, DELETE) from JavaScript code to interact with APIs and servers. It is commonly used in web development, particularly in React or other front-end applications, to communicate with back-end services.

### **Postman**

Postman is a popular API development and testing tool used to send HTTP requests and view responses. It provides an intuitive graphical user interface (GUI) for working with APIs. Postman allows you to test, develop, and debug APIs by sending various types of requests (GET, POST, PUT, DELETE) and inspecting the results.

### **Methods:**

1. GET Method: The GET method is used to retrieve data from a server.
2. POST Method: The POST method is used to send data to a server to create a new resource.
3. PUT Method: The PUT method is used to update an existing resource on the server.
4. PATCH Method: The PATCH method is used to partially update a resource (only modifies the provided fields).
5. DELETE Method: The DELETE method is used to delete a resource on the server.

### **GET Request — Fetch All Posts:**

i) VS Code:

```
// Step 1: Import axios
const axios = require('axios');

// Step 2: Make a GET request
axios.get('https://jsonplaceholder.typicode.com/posts')
  .then((response) => {
    // Step 3: Print response data
    console.log("Response Data:", response.data); // data received from server
  })
  .catch((error) => {
    // Step 4: If there's an error
    console.log("Error:", error);
  });
```

Console output:




```
Response Data: App.jsx:9
(100) [{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...},
{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...},
{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...},
{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...},
{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...},
{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...},
{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...},
{...}, {...}, {...}, {...}]
```


ii) Postman:

### Postman Steps (GET)

1. Open Postman
2. Method: GET
3. URL: `https://jsonplaceholder.typicode.com/posts`
4. Click Send

✓ You'll get the same JSON output as above.


<https://jsonplaceholder.typicode.com/posts>

 Save
 

▼

Share

GET

▼

https://jsonplaceholder.typicode.com/posts

Send

▼

output:


Body

Cookies

Headers (25)

Test Results

↺






200 OK • 77 ms • 8.01 KB •  ...

{ } JSON

▼

▶ Preview

🖼 Visualization

```

1  [
2    {
3      "userId": 1,
4      "id": 1,
5      "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
6      "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"
7    },
8    {
9      "userId": 1,
10     "id": 2,

```

## POST Request — Add a Post

i) VS Code:

```

const axios = require('axios');

axios.post('https://jsonplaceholder.typicode.com/posts', {
  title: 'Learn Axios',
  body: 'Axios is awesome!',
  userId: 1
})
.then((response) => {
  console.log("POST response:", response.data);
})
.catch((error) => {
  console.log("Error:", error);
});

```

Console Output:

```

POST response: App.jsx:21
▶ {title: 'Learn Axios', body: 'Axios is awesome!', userId: 1, id: 101}

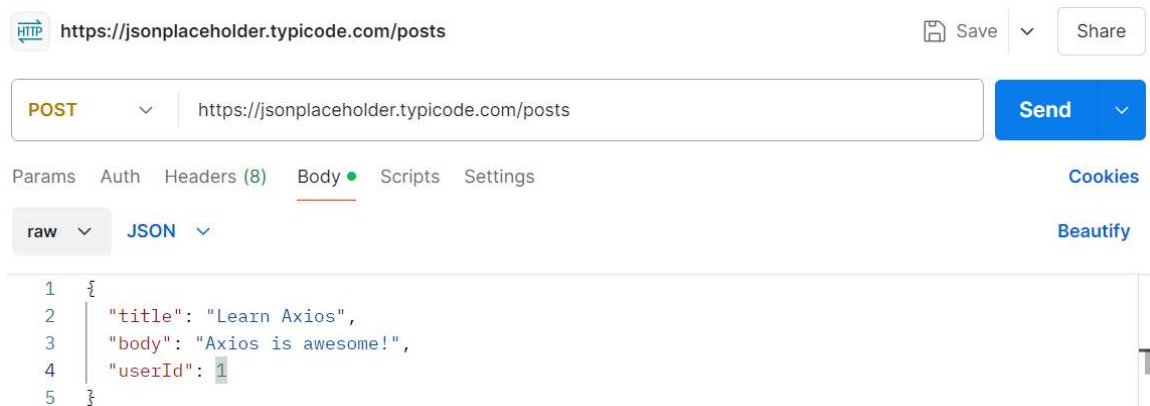
```

ii) Postman:

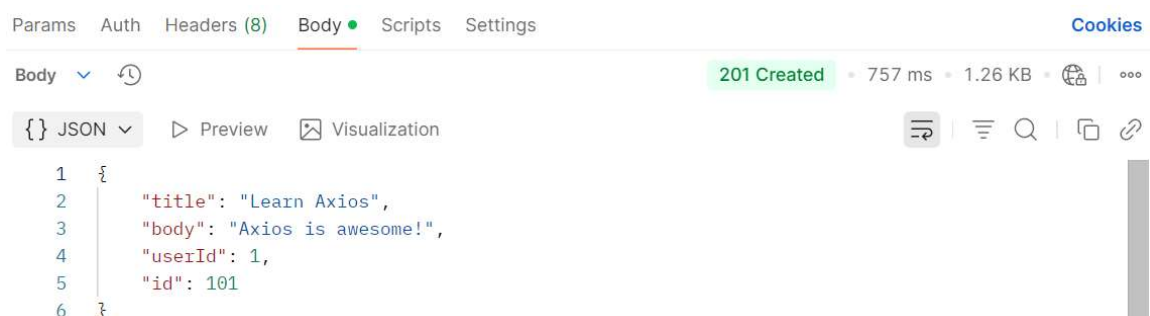
## Postman Steps (POST)

1. Method: **POST**
2. URL: **https://jsonplaceholder.typicode.com/posts**
3. Click **Body** > **raw** > **JSON**
4. Paste this:

```
{
  "title": "Learn Axios",
  "body": "Axios is awesome!",
  "userId": 1
}
```



### Output:



### PUT Request — Update a Post Fully:

#### i)VS Code:

```
const axios = require('axios');

axios.put('https://jsonplaceholder.typicode.com/posts/1', {
  id: 1,
  title: 'Updated Title',
  body: 'Updated Body Content',
  userId: 1
})
.then(response => {
  console.log("PUT response:", response.data);
})
.catch(error => {
  console.log("Error:", error);
});
```

Console output:

```
PUT response: App.jsx:36
▶ {id: 1, title: 'Updated Title', body: 'Updated Body Content', userId: 1}
```

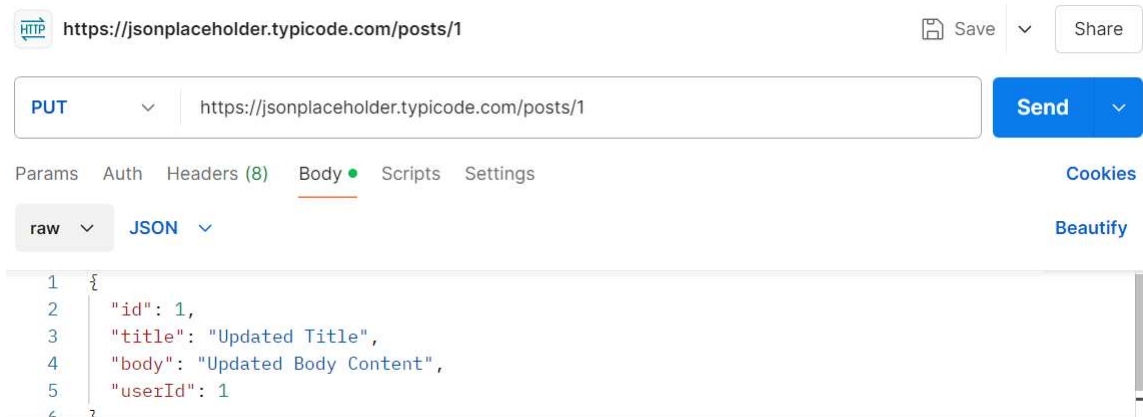
ii) Postman:

### Postman Steps (PUT)

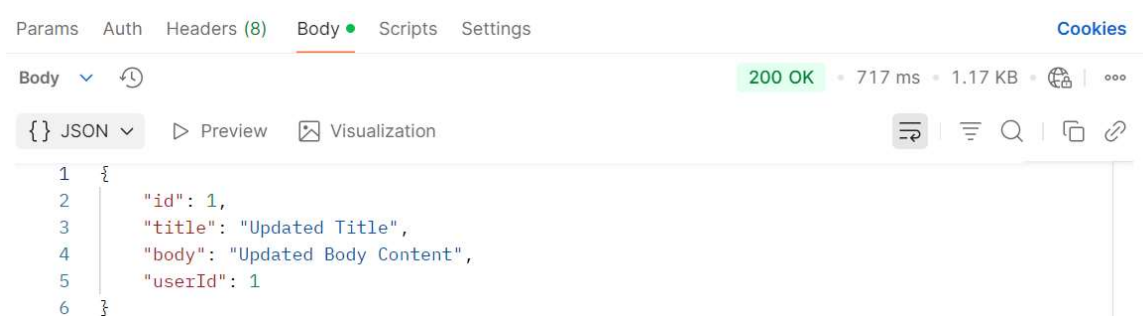
1. Method: PUT
2. URL: `https://jsonplaceholder.typicode.com/posts/1`
3. Body > raw > JSON:

```
{
  "id": 1,
  "title": "Updated Title",
  "body": "Updated Body Content",
  "userId": 1
}
```

4. Click Send



### Output:



### PATCH Request — Update Part of a Post:

#### i) VS Code:

```
const axios = require('axios');

axios.patch('https://jsonplaceholder.typicode.com/posts/1', {
  title: 'Only Title Updated'
})
.then(response => {
  console.log("PATCH response:", response.data);
})
.catch(error => {
  console.log("Error:", error);
});
```

#### Console output:

```
PATCH response: App.jsx:34
{userId: 1, id: 1, title: 'Only Title Updated', body: 'quia et suscipit\nuscupit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto'}
```

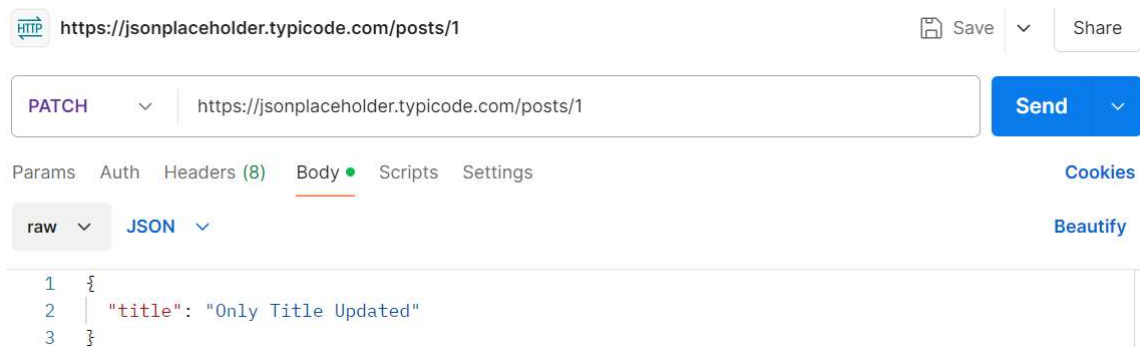
## ii) Postman:

### Postman Steps (PATCH)

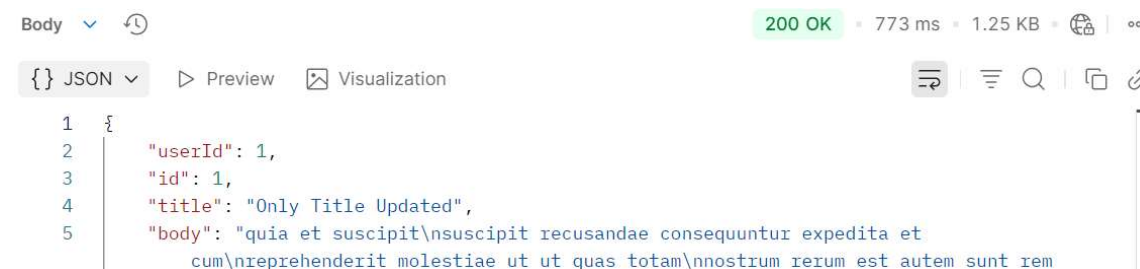
1. Method: PATCH
2. URL: `https://jsonplaceholder.typicode.com/posts/1`
3. Body > raw > JSON:

```
{
  "title": "Only Title Updated"
}
```

#### 4. Click Send



## Output:



## DELETE Request — Remove a Post:

### i)VS Code:

```
const axios = require('axios');

axios.delete('https://jsonplaceholder.typicode.com/posts/1')
  .then(response => {
    console.log("DELETE response:", response.data);
  })
  .catch(error => {
    console.log("Error:", error);
  });
```

### Console Output:

```
DELETE response: ▶ {}
```

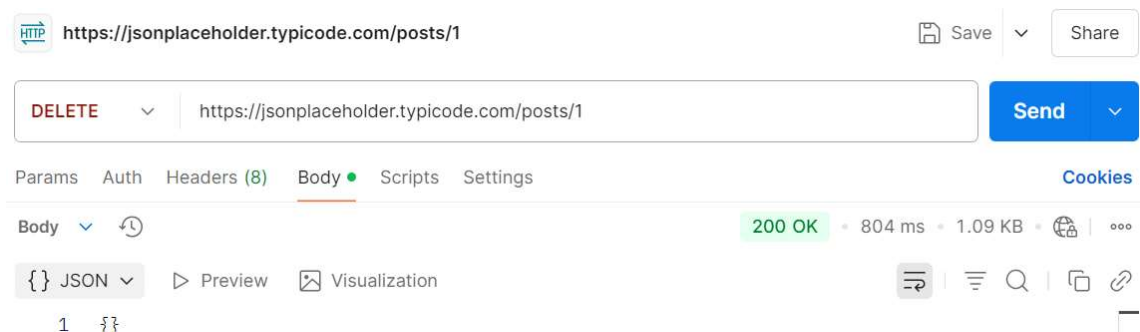
### ii) Postman:

#### Postman Steps (DELETE)

1. Method: DELETE
2. URL: `https://jsonplaceholder.typicode.com/posts/1`
3. Click Send

✓ You'll get an empty object `{}` showing it was deleted.

### Output:



The screenshot shows the Postman interface for a DELETE request. At the top, the URL bar contains `https://jsonplaceholder.typicode.com/posts/1`. Below it, the method is set to `DELETE` and the same URL is entered in the request body field. The `Send` button is visible. The bottom section shows the response details: `200 OK` with a status of `200 OK`, a response time of `804 ms`, and a size of `1.09 KB`. The response body is displayed as `{}` in the JSON view.



