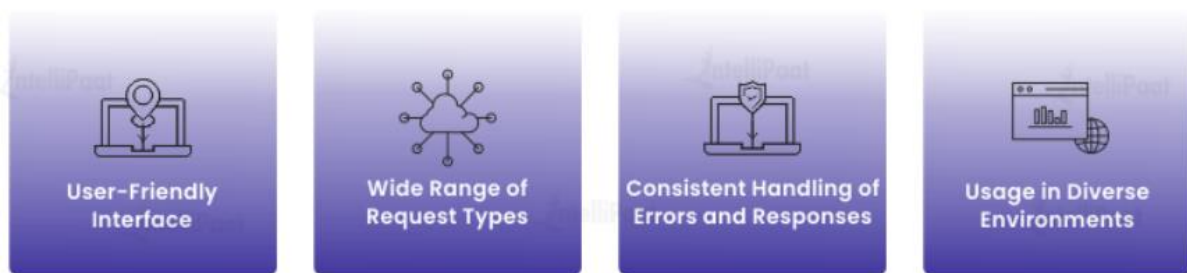# What is Axios?

Axios is lightweight package and use to make HTTP requests in Any Javascript Library like React, Angular or Vue. Axios makes it easy to send asynchronous HTTP requests to REST endpoints and perform CRUD operations. If you use Fetch method in Javascript, Axios is the "Easy to use" Version of Fetch.

Need of Axios in React

- Axios offers a user-friendly interface that simplifies the process of making HTTP requests.
- It supports a wide range of request types, such as GET, POST, PUT, DELETE, and more, providing flexibility in interacting with servers.
- Axios ensures consistent handling of errors and responses, making it convenient to manage errors and present meaningful messages to users.
- The versatility of Axios shines through its compatibility with both web browsers and Node.js, allowing its usage in diverse environments.

```
PS D:\MERN> npx create-react-app axiosapp
Need to install the following packages:
create-react-app@5.0.1
Ok to proceed? (y) y
```

Installing Axios in React

```
PS D:\MERN\axiosapp> npm install axios

added 3 packages, and audited 1545 packages in 2s
```

We can check in package.json

```json
{} package.json > ...

  "name": "axiosapp",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^5.17.0",
    "@testing-library/react": "^13.4.0",
    "@testing-library/user-event": "^13.5.0",
    "axios": "^1.7.7",
    "react": "^18.3.1",
    "react-dom": "^18.3.1",
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.4"
```

Once installed, import Axios in your React component using the following code:
`import axios from 'axios';`

```javascript
import { useState, useEffect } from "react";
import axios from 'axios';
```

Making GET Requests:

GET requests are used to *retrieve data* from a server.

```javascript
import axios from 'axios';
```

```
import axios from 'axios';

function App() {

  axios.get('https://jsonplaceholder.typicode.com/posts')
  .then(response => {
    // Handle successful response
    console.log(response.data);
  })
  .catch(error => {
    // Handle error
    console.error('Error fetching data:', error);
  });
}

export default App;
```

Making POST Requests:

POST requests are used to *send data to the server* to create a new resource.

```
const postData = {
  title: 'John Doe',
  body: 'Hi welcome'
};
```

```
axios.post('https://jsonplaceholder.typicode.com/posts', postData)
  .then(response => {
    // Handle successful response
    console.log('User created:', response.data);
  })
  .catch(error => {
    // Handle error
    console.error('Error creating user:', error);
  });
```

Making DELETE Requests:

DELETE requests are used to delete a resource on the server.

```javascript
axios.delete('https://jsonplaceholder.typicode.com/posts/1') // Specify the ID of the post to delete
  .then(response => {
    // Handle successful response
    console.log('Post deleted:', response.data);
  })
  .catch(error => {
    // Handle error
    console.error('Error deleting post:', error);
  });
```

```javascript
import './App.css';
import { useState, useEffect } from "react";
import axios from 'axios';

function App() {
  useEffect(()=>{
    axios.get("https://jsonplaceholder.typicode.com/posts")
  .then((res)=>
    console.log("App.jsx useEffect - res",res.data));
},[]);
  return (
    <>
      <h1>Axios Tutorial</h1>
    </>
  );
};

export default App;
```

New format

```javascript
import { useState, useEffect } from "react";
import axios from 'axios';

function App() {
```

```
  const [myData, setMyData] = useState([]);

  useEffect(() => {
    axios.get("https://jsonplaceholder.typicode.com/posts")
      .then((res) => setMyData(res.data));
  }, []);

  return (
    <>
      <h1>Axios Tutorial</h1>
      {myData.map((post) => {
        const { body, id, title } = post;
        return (
          <div className='card' key={id}>
            <h2>{title}</h2>
            <p>{body}</p>
          </div>
        );
      })}
    </>
  );
}

export default App;
```

For Experiment purpose use the MOCK API

Visit mock API and Login with your email account

      Create project of your choice

      Add only one resource in it for demonstration

      It will open the window like this

Schema

Define Resource schema, it will be used to generate mock data.

id	Object ID	
firstName	Faker.js	random.word
lastName	Faker.js	random.word
checkbox	Faker.js	random.word
RollNo	Faker.js	random.word

Close   **Update**

Add fields as per your html form and update it

```
{
  "username": "$internet.userName",
  "knownIps": ["$internet.ip", "$internet.ipv6"],
  "profile": {
    "firstName": "$name.firstName",
    "lastName": "$name.lastName",
    "staticData": [100, 200, 300]
  }
}
```

2) create react project and replace the files with shared one

3) Install dependencies

Npm I axios react-router-dom

4) **Reference for usenavigate** https://www.geekster.in/articles/usenavigate-in-react-router-dom/

Reference for BrowserRouter https://www.geekster.in/articles/browserrouter-routes-route-in-react-router-dom/

Reference for Semantic-UI https://www.geeksforgeeks.org/semantic-ui/

Optional