**Axios in React**

* In React communicating with the backend server is done via HTTP protocol using Axios library.
* Axios, which is a popular library is mainly used to send asynchronous HTTP requests to REST endpoints. This library is very useful to perform CRUD operations in React apps.
* This popular library is used to communicate with the backend. Axios supports the Promise API, native to JS ES6.
* Using Axios we make API requests in our application. Once the request is made, we get the data in Return, and then we use this data in our project.
* This library is very popular among developers.
* Install Axios library using the command given below…
  + **npm install axios**
* After Axios installation, you can import this library into your file and use it to make an HTTP request(Get,Post,Put,Delete).
* Axios allows you to communicate with the APIs in your React project. The same tasks can also be performed by using AJAX, but Axios provide you more functionality and features and that helps you in building your application quickly.
* Axios is a promise-based library, so you need to implement some promise-based asynchronous HTTP requests. jQuery and AJAX also perform the same job but in React project React handles each and everything in its own virtual DOM, so there is no need to use jQuery at all.
* example to fetch the customer’s data using Axios…

|  |
| --- |
| const getCustomersData = () => {    axios    .get("<https://jsonplaceholder.typicode.com/customers>")    .then(data => console.log(data.data))    .**catch**(error => console.log(error));    };   getCustomersData(); |

**GET Request with Axios:**

import React from 'react';

 import axios from 'axios';

 export **default** class MyList extends React.Component {

  state = {

    blogs: []

  }

  componentDidMount() {

    axios.get(`https://jsonplaceholder.typicode.com/posts`)

    .then(response => {

      const posts = response.data;

**this**.setState ({posts});

    })

  }

 render() {

**return** (

   < ul >

     {**this**.state.posts.map (post =>  {post.title} )}

   < /ul >

  )}

 }

**POST Request with Axios:**

import React from 'react';

import axios from 'axios';

export **default** class AddPost extends React.Component {

   state = {

     postTitle: '',

   }

   handleChange = event => {

**this**.setState({ postTitle: event.target.value });

   }

   handleSubmit = event => {

     event.preventDefault();

       const post = {

         postName: **this**.state.postName

       };

     axios.post(

`https://jsonplaceholder.typicode.com/posts`, { post })

       .then(res => {

         console.log(res);

         console.log(res.data);

     })

     [Text Wrapping Break]

     }

    render() {

**return** (

         <div>

           <form onSubmit="{this.handleSubmit}">

             <label>

               Post Name:

               <input type="text" name="name"

                     onChange="{this.handleChange}" />

             </label>

             <button type="submit">Add</button>

          </form>

        </div>

 )}}

**Delete Request With Axios:**

|  |
| --- |
| handleSubmit = event => {    event.preventDefault();    axios.**delete**(  `https://jsonplaceholder.typicode.com/posts/${this.state.postName}`)    .then(res => {    console.log(res);    console.log(res.data);    })  } |

**Response Objects in Axios:**When you send a request to the server, you receive a response object from the server with the properties given below…

* **data:**You receive data from the server in payload form. This data is returned in JSON form and parse back into a JavaScript object to you.
* **status:** You get the HTTP code returned from the server.
* **statusText:**HTTP status message returned by the server.
* **headers:** All the headers are sent back by the server.
* **config:**original request configuration.
* **request:**actual XMLHttpRequest object.

**Error Object:**You will get an error object if there will be a problem with the request. Promise will be rejected with an error object with the properties given

* **message:** Error message text.
* **response:** Response object (if received).
* **request:** Actual XMLHttpRequest object (when running in a browser).
* **config:**Original request configuration.

**Features of Axios Library**

* JSON data is transformed automatically.
* It transforms the request and response data.
* Useful in making HTTP requests from Node.js
* It makes XMLHttpRequests from the browser.
* Provide client-side support for protecting against XSRF.
* Supports promise API.
* Ability to cancel the request.

**Shorthand Methods in Axios:**Below are some shorthand methods of Axios…

* axios.request(config)
* axios.head(url[, config])
* axios.get(url[, config])
* axios.post(url[, data[, config]])
* axios.put(url[, data[, config]])
* axios.delete(url[, config])
* axios.options(url[, config])
* axios.patch(url[, data[, config]])