

## HTTP(41):

1. npm install axios

## HTTP(42):Get request and render the deta:

1.Import axios library on component[import axios from 'axios';]

2.We need to create a state property which will store the list of post.

```
class PostList extends Component {  
  constructor(props){  
    super(props)  
    this.state = {  
      post: {}  
    }  
  }  
}
```

3.use axios to make a get request to the JSON placeholder API the question though is where do we place the code to make the get request the answer is in the component did mount lifecycle method.

```
componentDidMount() {  
  axios.get('https://jsonplaceholder.typicode.com/posts')  
    .then(response => {  
      console.log(response);  
      this.setState({  
        posts: response.data  
      });  
    })  
    .catch(error => {  
      console.log(error);  
    });  
}
```

From the api  
page ,it is a link

```
render() {  
  const { posts } = this.state;  
  return (  

```

```

<div>
  <h2>List of Posts</h2>
  {
    posts.length ? (
      posts.map(post => <div key={post.id}>{post.title}</div>)
    ) : (
      <p>No posts found.</p>
    )
  }
</div>
);
}
}

```

```

File Edit Selection View Go Run Terminal Help my-first-app

EXPLORER
  OPEN EDITORS
    JS App.js src M
    JS PostList.js src\Compo... U
  MY-FIRST-APP
    src
      Components
        JS FunctionClick.js U
        JS Greet.js U
        JS Hello.js U
        JS InLine.js U
        JS Legend.js U
        JS LifecycleA.js U
        JS LifecycleB.js U
        JS Message.js U
        # myStyle.css U
        JS NameList.js U
        JS ParentComponent.js U
        JS PostList.js U
        JS StyleSheet.js U
        JS Table.js U
        JS UserContext.js U
        JS UserGreeting.js U
        JS WelCome.js U
      # App.css
      JS App.js M
      JS App.test.js
      # appStyles.css
      # appStyles.module.css U
      # index.css
      JS index.js
      logo.svg
      JS reportWebVitals.js
      JS setupTests.js
      .gitignore
      package-lock.json M
      package.json M
      README.md

JS App.js M JS PostList.js U X
src > Components > JS PostList.js > PostList > componentDidMount
1 import React, { Component } from 'react';
2 import axios from 'axios';
3
4 class PostList extends Component {
5   constructor(props) {
6     super(props);
7
8     this.state = {
9       posts: [],
10      errorMsg: ''
11    };
12  }
13
14  componentDidMount() {
15    axios.get('https://jsonplaceholder.typicode.com/posts') // fixed endpoint
16      .then(response => {
17        this.setState({ posts: response.data });
18      })
19      .catch(error => {
20        console.log(error);
21        this.setState({ errorMsg: 'Error retrieving data' });
22      });
23  }
24
25  render() {
26    const { posts, errorMsg } = this.state;
27    return (
28      <div>
29        <h2>List of Posts</h2>
30        {posts.length ? (
31          posts.map(post => <div key={post.id}>{post.title}</div>)
32        ) : (
33          <p>No posts found.</p>
34        )}
35        { /* Show error message if exists */
36          errorMsg && <p style={{ color: 'red' }}>{errorMsg}</p>
37        }
38      </div>
39    );
40  }
41
42  export default PostList;
43

```

HTTP(43):HTTP POST

```

JS Appjs M JS PostListjs U JS PostFormjs U
src > Components > JS PostFormjs > ...
1 import React, { Component } from 'react';
2 import axios from 'axios'; // Don't forget to import axios
3
4 class PostForm extends Component {
5   constructor(props) {
6     super(props)
7
8     this.state = {
9       userId: '',
10      title: '',
11      body: ''
12    }
13  }
14
15  changeHandler = (e) => {
16    this.setState({ [e.target.name]: e.target.value });
17  }
18
19  submitHandler = e => {
20    e.preventDefault();
21    console.log(this.state);
22
23    axios.post('https://jsonplaceholder.typicode.com/posts', this.state)
24      .then(response => {
25        console.log('Response:', response);
26        alert('Post submitted successfully!');
27      })
28      .catch(error => {
29        console.error('Error:', error);
30        alert('Something went wrong while submitting!');

```

```

JS Appjs M JS PostListjs U JS PostFormjs U
src > Components > JS PostFormjs > ...
4 class PostForm extends Component {
19   submitHandler = e => {
28     .catch(error => {
29       console.error('Error:', error);
30       alert('Something went wrong while submitting!');
31     });
32   }
33
34   render() {
35     const { userId, title, body } = this.state;
36     return (
37       <div>
38         <h2>Create a New Post</h2>
39         <form onSubmit={this.submitHandler}>
40           <div>
41             <input
42               type="text"
43               name="userId"
44               placeholder="User ID"
45               value={userId}
46               onChange={this.changeHandler}
47             />
48           </div>
49           <div>
50             <input
51               type="text"
52               name="title"
53               placeholder="Title"
54               value={title}
55               onChange={this.changeHandler}

```

```

JS Appjs M JS PostListjs U JS PostFormjs U
src > Components > JS PostFormjs > ...
4 class PostForm extends Component {
34   render() {
43     <div>
50       <input
51         type="text"
52         name="title"
53         placeholder="Title"
54         value={title}
55         onChange={this.changeHandler}
56       />
57     </div>
58     <div>
59       <input
60         type="text"
61         name="body"
62         placeholder="Body"
63         value={body}
64         onChange={this.changeHandler}
65       />
66     </div>
67     <div>
68       <button type="submit">Submit</button>
69     </div>
70   </form>
71 </div>
72 );
73 }
74 }
75
76 export default PostForm;

```