Contents

[FOLDER STRUCUTRE: 4](#_Toc75530912)

[App.js: 5](#_Toc75530913)

[Blog.js 5](#_Toc75530914)

[Post.js 6](#_Toc75530915)

[NewPost.js 6](#_Toc75530916)

[FullPost.js 7](#_Toc75530917)

[Styling: 8](#_Toc75530918)

[Blog.css 8](#_Toc75530919)

[Post.css 8](#_Toc75530920)

[NewPost.css 8](#_Toc75530921)

[FullPost.css 9](#_Toc75530922)

[Installation of Axios 11](#_Toc75530923)

[HTTP request to get data 11](#_Toc75530924)

[Blog.js 11](#_Toc75530925)

[RENDERING IN SCREEN: 12](#_Toc75530926)

[Blog.js 12](#_Toc75530927)

[Post.js 13](#_Toc75530928)

[Post.css 13](#_Toc75530929)

[Transform Data 13](#_Toc75530930)

[Blog.js 13](#_Toc75530931)

[Making Selectable Data 14](#_Toc75530932)

[Blog.js 14](#_Toc75530933)

[Post.js 16](#_Toc75530934)

[FullPost.js 16](#_Toc75530935)

[Fetching Data on Update (without Creating Infinite Loops) 17](#_Toc75530936)

[Fullpost.js 17](#_Toc75530937)

[MODIFIED FullPost.js 18](#_Toc75530938)

[POSTING DATA TO SERVER: 20](#_Toc75530939)

[NewPost.js 20](#_Toc75530940)

[SENDING DELETE REQUEST: 21](#_Toc75530941)

[ERRORS HANDLING: 22](#_Toc75530942)

[Blog.js 22](#_Toc75530943)

[Adding Interceptors to Execute Code Globally 24](#_Toc75530944)

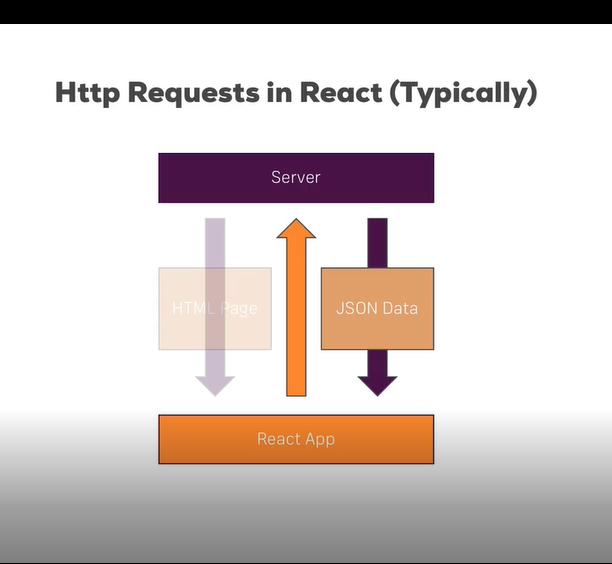
[Index.js 24](#_Toc75530945)

[Setting a Default Global Configuration for Axios 25](#_Toc75530946)

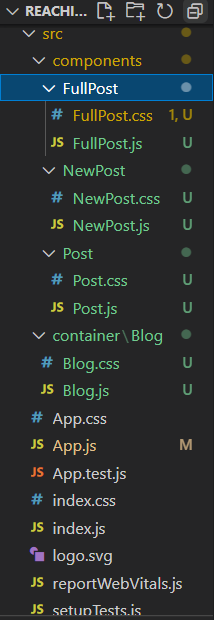
[Index.js 25](#_Toc75530947)

[Blog.js 25](#_Toc75530948)

Reaching out to the Web (Http & Ajax)



# FOLDER STRUCUTRE:



## App.js:

import React, { Component } from 'react';

import Blog from './container/Blog/Blog';

class App extends Component {

  render() {

    return (

      <div className="App">

        <Blog />

      </div>

    );

  }

}

export default App;

## Blog.js

import React, { Component } from 'react';

import NewPost from '../../components/NewPost/NewPost';

import Post from '../../components/Post/Post';

import FullPost from '../../components/FullPost/FullPost';

import './Blog.css';

class Blog extends Component {

    render(){

        return(

            <div>

            <section className="Posts">

                <Post />

                <Post />

                <Post />

            </section>

            <section>

                <FullPost />

            </section>

            <section>

                <NewPost />

            </section>

        </div>

        );

    }

}

export default Blog;

## Post.js

import React from 'react';

import './Post.css';

const post = (props) => (

    <article className="Post">

        <h1>Title</h1>

        <div className="Info">

            <div className="Author">Author</div>

        </div>

    </article>

);

export default post;

## NewPost.js

import React, { Component } from 'react';

import './NewPost.css';

class NewPost extends Component {

    state = {

        title: '',

        content: '',

        author: 'Max'

    }

    render () {

        return (

            <div className="NewPost">

                <h1>Add a Post</h1>

                <label>Title</label>

                <input type="text" value={this.state.title} onChange={(event) => this.setState({title: event.target.value})} />

                <label>Content</label>

                <textarea rows="4" value={this.state.content} onChange={(event) => this.setState({content: event.target.value})} />

                <label>Author</label>

                <select value={this.state.author} onChange={(event) => this.setState({author: event.target.value})}>

                    <option value="Max">Max</option>

                    <option value="Manu">Manu</option>

                </select>

                <button>Add Post</button>

            </div>

        );

    }

}

export default NewPost;

## FullPost.js

import React, { Component } from 'react';

import './FullPost.css';

class FullPost extends Component {

    render () {

        let post = <p>Please select a Post!</p>;

        post = (

            <div className="FullPost">

                <h1>Title</h1>

                <p>Content</p>

                <div className="Edit">

                    <button className="Delete">Delete</button>

                </div>

            </div>

        );

        return post;

    }

}

export default FullPost;

# Styling:

## Blog.css

.Posts {

    display: flex;

    flex-flow: row wrap;

    justify-content: center;

    width: 80%;

    margin: auto;

}

## Post.css

.Post {

    width: 100px;

    padding: 16px;

    text-align: center;

    border: 1px solid #eee;

    box-shadow: 0 2px 3px #ccc;

    margin: 10px;

    box-sizing: border-box;

    cursor: pointer;

}

.Post:hover,

.Post:active {

    background-color: #C0DDF5;

}

.Author {

    margin: 16px 0;

    color: #ccc;

}

## NewPost.css

.NewPost {

    width: 80%;

    margin: 20px auto;

    border: 1px solid #eee;

    box-shadow: 0 2px 3px #ccc;

    text-align: center;

}

.NewPost label {

    display: block;

    margin: 10px auto;

    text-align: center;

    font-weight: bold;

}

.NewPost input,

.NewPost textarea,

.NewPost select {

    display: block;

    width: 80%;

    box-sizing: border-box;

    border: 1px solid black;

    outline: none;

    font: inherit;

    margin: auto;

}

.NewPost button {

    margin: 5px 0;

    padding: 10px;

    font: inherit;

    border: 1px solid #fa923f;

    background-color: transparent;

    color: #fa923f;

    cursor: pointer;

}

.NewPost button:hover,

.NewPost button:active {

    color: white;

    background-color: #fa923f;

}

## FullPost.css

.FullPost {

    width: 80%;

    margin: 20px auto;

    border: 1px solid #eee;

    box-shadow: 0 2px 3px #ccc;

    text-align: center;

}

.Edit {

}

.Edit label {

    display: block;

    color: grey;

}

.Edit button {

    border: none;

    background-color: transparent;

    cursor: pointer;

    outline: none;

    color: #fa923f;

}

.Edit button:hover,

.Edit button:active {

    color: #521751;

}

.Edit button.Delete {

    color: red;

}

.Edit button.Delete:hover,

.Edit button.Delete:active {

    color: darkred;

}

# Installation of Axios

Npm install axios –save

# HTTP request to get data

## Blog.js

NOTE: axios.get() will gets data from server and after getting datas then we can console it with then() function.

import React, { Component } from 'react';

import NewPost from '../../components/NewPost/NewPost';

import Post from '../../components/Post/Post';

import FullPost from '../../components/FullPost/FullPost';

import './Blog.css';

import axios from 'axios';

class Blog extends Component {

    componentDidMount() {

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

        .then(response => {

            console.log(response);

        });

    }

    render(){

        return(

            <div>

            <section className="Posts">

                <Post />

                <Post />

                <Post />

            </section>

            <section>

                <FullPost />

            </section>

            <section>

                <NewPost />

            </section>

        </div>

        );

    }

}

export default Blog;

# RENDERING IN SCREEN:

## Blog.js

import React, { Component } from 'react';

import NewPost from '../../components/NewPost/NewPost';

import Post from '../../components/Post/Post';

import FullPost from '../../components/FullPost/FullPost';

import './Blog.css';

import axios from 'axios';

class Blog extends Component {

    state = {

        posts: []

    }

    componentDidMount() {

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

        .then(response => {

            this.setState({posts: response.data})

            // console.log(response);

        });

    }

    render(){

        const posts = this.state.posts.map(

            post => {

                return <Post key={post.id} title={post.title} />

            }

        );

        //map provides new array

        return(

            <div>

            <section className="Posts">

                {posts}

            </section>

            <section>

                <FullPost />

            </section>

            <section>

                <NewPost />

            </section>

        </div>

        );

    }

}

export default Blog;

## Post.js

<h1>{props.title}</h1>

## Post.css

.Post {

    width: 250px;

# Transform Data

## Blog.js

import React, { Component } from 'react';

import NewPost from '../../components/NewPost/NewPost';

import Post from '../../components/Post/Post';

import FullPost from '../../components/FullPost/FullPost';

import './Blog.css';

import axios from 'axios';

class Blog extends Component {

    state = {

        posts: []

    }

    componentDidMount() {

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

        .then(response => {

            //transform data [from 1 to 4]

            const posts = response.data.slice(0,4);

            const updatedPosts = posts.map(

                post => {

                    return{

                        ...post,

                        author: 'Max'

                }

                }

            )

            this.setState({posts: updatedPosts})

            // console.log(response);

        });

    }

    render(){

        const posts = this.state.posts.map(

            post => {

                return <Post key={post.id} title={post.title} author={post.author}/>

            }

        );

        //map provides new array

        return(

            <div>

            <section className="Posts">

                {posts}

            </section>

            <section>

                <FullPost />

            </section>

            <section>

                <NewPost />

            </section>

        </div>

        );

    }

}

export default Blog;

# Making Selectable Data

## Blog.js

NOTE: selectedPostId vane state banayim jasko value chai null cha. Post component bata kunai post click vayera tyo post ko id aauxa ani postSelectedHandler vane function ma id as parameter vayera call hunxa ani setState bata selectedPostId ko state value jun click vayeko thiyo tei ma change hunxa jun chai hamile FullPost component ma pathauxum as props.

import React, { Component } from 'react';

import NewPost from '../../components/NewPost/NewPost';

import Post from '../../components/Post/Post';

import FullPost from '../../components/FullPost/FullPost';

import './Blog.css';

import axios from 'axios';

class Blog extends Component {

    state = {

        posts: [],

        selectedPostId: null,

    }

    componentDidMount() {

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

        .then(response => {

            //transform data [from 1 to 4]

            const posts = response.data.slice(0,4);

            const updatedPosts = posts.map(

                post => {

                    return{

                        ...post,

                        author: 'Max'

                }

                }

            )

            this.setState({posts: updatedPosts})

            // console.log(response);

        });

    }

    postSelectedHandler = (id) => {

        this.setState({selectedPostId: id});

    }

    render(){

        const posts = this.state.posts.map(

            post => {

                return <Post key={post.id}

clicked={() => this.postSelectedHandler(post.id)}

title={post.title} author={post.author}/>

            }

        );

        //map provides new array

        return(

            <div>

            <section className="Posts">

                {posts}

            </section>

            <section>

                <FullPost id={this.state.selectedPostId}/>

            </section>

            <section>

                <NewPost />

            </section>

        </div>

        );

    }

}

export default Blog;

## Post.js

Note: Jabo particular post click hunxa ne clicked props call hunxa ra jun post click vayo tesako id postSelectedHandler function ma janxa

import React from 'react';

import './Post.css';

const post = (props) => (

    <article className="Post"  onClick={props.clicked}>

        <h1>{props.title}</h1>

        <div className="Info">

            <div className="Author">{props.author}</div>

        </div>

    </article>

);

export default post;

## FullPost.js

NOTE: props id null cha vane false hunxa yedi kunai id cha vane true hunxa

import React, { Component } from 'react';

import './FullPost.css';

class FullPost extends Component {

    render () {

        let post = <p style={{textAlign:'center'}}>Please select a Post!</p>;

        if(this.props.id){

            post = (

                <div className="FullPost">

                    <h1>Title</h1>

                    <p>Content</p>

                    <div className="Edit">

                        <button className="Delete">Delete</button>

                    </div>

                </div>

            );

        }

        return post;

    }

}

export default FullPost;

# Fetching Data on Update (without Creating Infinite Loops)

## Fullpost.js

import React, { Component } from 'react';

import axios from 'axios';

import './FullPost.css';

class FullPost extends Component {

    componentDidUpdate(){

        if(this.props.id){

            axios.get('https://jsonplaceholder.typicode.com/posts/'+this.props.id)

            .then(resposne => {

                console.log(resposne)

            });

        }

    }

## MODIFIED FullPost.js

import React, { Component } from 'react';

import axios from 'axios';

import './FullPost.css';

class FullPost extends Component {

    state = {

        loadedPost: null,

    }

    componentDidUpdate(){

        if(this.props.id){

            if( !this.state.loadedPost ||

this.state.loadedPost &&

this.state.loadedPost.id !== this.props.id)

{

       axios.get('https://jsonplaceholder.typicode.com/posts/'+this.props.id)

                .then(response => {

                    // console.log(resposne)

                    this.setState({loadedPost: response.data});

                });

            }

        }

    }

    // IN console website, look network where we have request infinte time i.e. infinite loops because of

    // this.setState({loadedPost: response.data});

    // we are continously updating state

    render () {

        let post = <p style={{textAlign:'center'}}>Please select a Post!</p>;

        //ERROR REMOVING

        if(this.props.id){

            post = <p style={{textAlign:'center'}}>Loading...</p>;

        }

        if(this.state.loadedPost){

            post = (

                <div className="FullPost">

                    <h1>{this.state.loadedPost.title}</h1>

                    <p>{this.state.loadedPost.body}</p>

                    <div className="Edit">

                        <button className="Delete">Delete</button>

                    </div>

                </div>

            );

        }

        //NULL TITLE ERROR KINA AYO VANE... componentDidUpdate() vanda aagi nai

        //yo tala ko if wala statement run vayo jun ma {this.state.loadedPost.title}, loadedPost null nai hunxa

        // if(this.props.id){

        //     post = (

        //         <div className="FullPost">

        //             <h1>{this.state.loadedPost.title}</h1>

        //             <p>l{this.state.loadedPost.content}</p>

        //             <div className="Edit">

        //                 <button className="Delete">Delete</button>

        //             </div>

        //         </div>

        //     );

        // }

        return post;

    }

}

export default FullPost;

# POSTING DATA TO SERVER:

## NewPost.js

import React, { Component } from 'react';

import axios from 'axios';

import './NewPost.css';

class NewPost extends Component {

    state = {

        title: '',

        content: '',

        author: 'Max'

    }

    postDataHandler = () => {

        // JS Object

        const post = {

            title: this.state.title,

            body: this.state.content,

            author: this.state.author,

        };

        axios.post('https://jsonplaceholder.typicode.com/posts', post)

        .then( response => {

            console.log(response);

        });

    }

    render () {

        return (

            <div className="NewPost">

                <h1>Add a Post</h1>

                <label>Title</label>

                <input type="text" value={this.state.title} onChange={(event) => this.setState({title: event.target.value})} />

                <label>Content</label>

                <textarea rows="4" value={this.state.content} onChange={(event) => this.setState({content: event.target.value})} />

                <label>Author</label>

                <select value={this.state.author} onChange={(event) => this.setState({author: event.target.value})}>

                    <option value="Max">Max</option>

                    <option value="Manu">Manu</option>

                </select>

                <button onClick={this.postDataHandler}>Add Post</button>

            </div>

        );

    }

}

export default NewPost;

# SENDING DELETE REQUEST:

deletePostHandler = () => {

        axios.delete('https://jsonplaceholder.typicode.com/posts'+this.props.id)

        .then(response => {

            console.log(response);

        });

    }

    render () {

        let post = <p style={{textAlign:'center'}}>Please select a Post!</p>;

        //ERROR REMOVING

        if(this.props.id){

            post = <p style={{textAlign:'center'}}>Loading...</p>;

        }

        if(this.state.loadedPost){

            post = (

                <div className="FullPost">

                    <h1>{this.state.loadedPost.title}</h1>

                    <p>l{this.state.loadedPost.content}</p>

                    <div className="Edit">

                        <button className="Delete" onClick={this.deletePostHandler}>Delete</button>

                    </div>

                </div>

            );

        }

# ERRORS HANDLING:

## Blog.js

import React, { Component } from 'react';

import NewPost from '../../components/NewPost/NewPost';

import Post from '../../components/Post/Post';

import FullPost from '../../components/FullPost/FullPost';

import './Blog.css';

import axios from 'axios';

class Blog extends Component {

    state = {

        posts: [],

        selectedPostId: null,

        error: false,

    }

    componentDidMount() {

        axios.get('https://jsonpslaceholder.typicode.com/posts')

        .then(response => {

            //transform data [from 1 to 4]

            const posts = response.data.slice(0,4);

            const updatedPosts = posts.map(

                post => {

                    return{

                        ...post,

                        author: 'Max'

                }

                }

            )

            this.setState({posts: updatedPosts})

            // console.log(response);

        })

        .catch(error => {

            // console.log(error)

            this.setState({error: true})

        });

    }

    postSelectedHandler = (id) => {

        this.setState({selectedPostId: id});

    }

    render(){

        let posts = <p>Something wrong</p>

        if(!this.state.error){

            posts = this.state.posts.map(

                post => {

                    return <Post key={post.id}

clicked={() => this.postSelectedHandler(post.id)}

title={post.title}

author={post.author}/>

                }

            );

        }

        //map provides new array

        return(

            <div>

            <section className="Posts">

                {posts}

            </section>

            <section>

                <FullPost id={this.state.selectedPostId}/>

            </section>

            <section>

                <NewPost />

            </section>

        </div>

        );

    }

}

export default Blog;

# Adding Interceptors to Execute Code Globally

## Index.js

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import App from './App';

import reportWebVitals from './reportWebVitals';

import axios from 'axios';

//globally , new interceptors

// axios.interceptors.request.use( request => {

//   console.log(request);

//   //YOU can also edit request config

//   return request;

// });

axios.interceptors.request.use( request => {

  console.log(request);

  //YOU can also edit request config

  return request;

},error => {

  console.log(error);

  return Promise.reject(error);

});

//to handle response

axios.interceptors.response.use(response => {

  console.log(response);

  return response;

},error => {

  console.log(error);

  return Promise.reject(error);

}

)

ReactDOM.render(………

# Setting a Default Global Configuration for Axios

## Index.js

import axios from 'axios';

//GLOBAL CONFIGURATION

//you always have same link of json

axios.defaults.baseURL= 'https://jsonplaceholder.typicode.com';

axios.defaults.headers.common['Authorization'] = 'AUTH TOKEN';

axios.defaults.headers.post['Content-Type'] = 'application/json';

ReactDOM.render(………..

## Blog.js

axios.get('/posts')