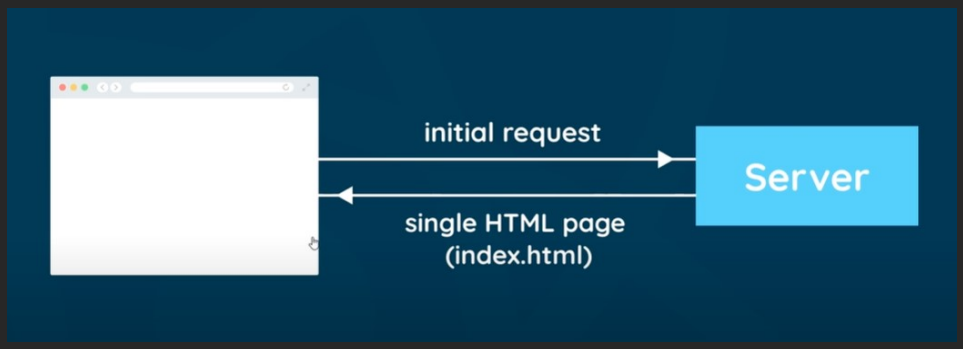
React Js Notes

INTRODUCTION

- Is a java script library used to create websites
-



- Allows us to create Single page apps (SPA)

CREATING A REACT APPLICATION

- npx create-react-app MyBlog => to create a react app
- npm start

INTRODUCTION

- Introduction :

  In : src/App.js

```
import './App.css';

function App() {
  return (
    <div className="App">
      <div className='content'>
          <h1>
              App components
          </h1>
      </div>
    </div>
  );
}

export default App;
```

in the end we always export the component function in order we can use it in the other files .

in src/index.js :

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';


const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

- Dynamic variables :

```
import './App.css';

function App() {

    const title = 'welcome to my blog' ;

    return (
       <div className="App">
         <div className='content'>
            <h1>
               {title}
            </h1>
         </div>
       </div>
    );
}
export default App;
```

also other variables :

```
import './App.css';

function App() {
    const title = 'welcome to my blog' ;
    const likes = 100 ;
    const persone = {
        name: 'ashraf',
        age: 20
    }

    return (
       <div className="App">
         <div className='content'>
             <h1>{title}</h1>
             <p>liked {likes} times</p>
             <p>{persone.name}</p>

             <p>{[1, 5, 6, 8, 7, 12]}</p>
         </div>
       </div>
```

```
    );
}
export default App;
```

- Dynamic attribute :

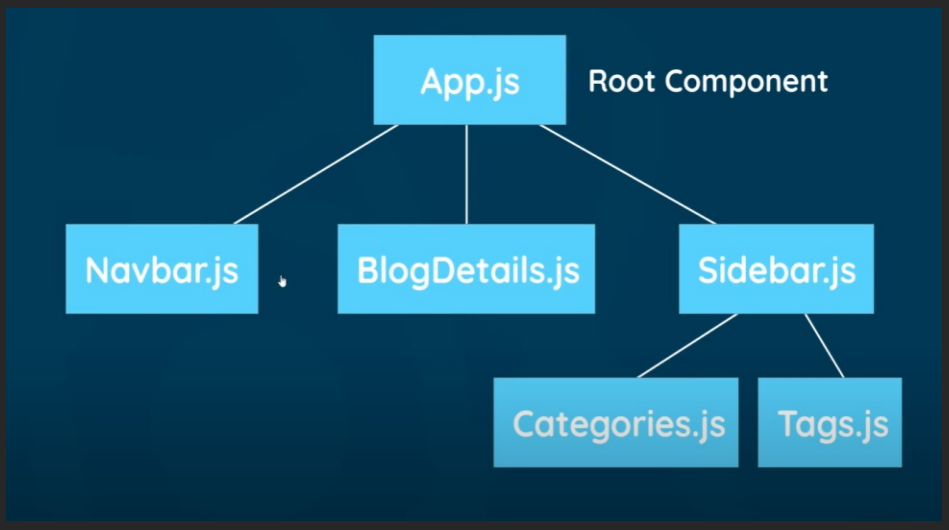
```
import './App.css';

function App() {
    const link = 'https//www.google.com' ;

    return (
       <div className="App">
         <div className='content'>
             <a href={link}></a>
         </div>
       </div>
    );
}
export default App;
```

MULTIPLE COMPONENTS

- Component tree :



- Navbar component :

In the file : src/components/Navbar.js :

```
const Navbar = () => {
    return(
```

```
        <nav className="navbar">
            <h1>
                Ashraf's blog
            </h1>
            <div>
                <a href="/">Home</a>
                <a href="/Create">New blog</a>
            </div>
        </nav>
    );
}

export default Navbar;
```

in the file : src/App.js

```
import './App.css';
import Navbar from "./components/Navbar";

function App() {
    const link = 'http://www.google.com' ;

    return (
       <div className="App">

         <Navbar />

         <div className='content'>
             <a href={link}>google</a>
         </div>
       </div>
    );
}
export default App;
```

- Click events :

```
const Home = () => {
    const handleClick = () => {
        alert('Hello, bros');
    }

    return (
        <div className='home'>
            <h2>Home page</h2>
            <button onClick={handleClick}>Click here</button>
        </div>
    ) ;
}

export default Home ;
```

when we write the parenties we invoke the function automatically , we just write the name of function ,it's a reference to the function not the function , and we invoke this function after clicking to the button .

but when we have a function which takes an element as an argument, we do not make the parenties either, but we create an anonymous funtion (it's again an arrow function inside the 'onClick') and we invoke the function there :

```
const Home = () => {
    const handleClick = () => {
        alert('Hello, bros');
    }
    const handleClickAgain = (name) => {
        alert('Hello' + name) ;
    }

    return (
        <div className='home'>
            <h2>Home page</h2>
            <button onClick={handleClick}>Click here</button>
            <button onClick={() => {
                handleClickAgain('Ashraf')
            }}>Click here again</button>
        </div>
    ) ;
}

export default Home ;
```

- Using states :

The variable wich created on the top of main function (Home, Navbar… for example) are called " non reactive variables " , it doesn't change inside the template whene we handle an event , for example :

```
const Home = () => {
    let name = "ashraf" ;

    const handleClick = () => {
        name = "khabar" ;
    }

    return (
        <div className='home'>
            <h2>Home page</h2>
            <p>{name}</p>
            <button onClick={handleClick}>Click here</button>
        </div>
    ) ;
}

export default Home ;
```

the variable name won't change by clicking the button even though we assign it a new value inside the function, that's what we call it " non reactive variable " .

in order to make a variable reactive , we use something called HOOK .

```
import {useState} from "react";

const Home = () => {
    let [name, setName] = useState("ashraf") ;

    const handleClick = () => {
        setName('KHABAR') ;
    }

    return (
        <div className='home'>
            <h2>Home page</h2>
            <p>{name}</p>
            <button onClick={handleClick}>Click here</button>
        </div>
    ) ;
}

export default Home ;
```

the variable name inside the array is now reactive variable with a setFucntion (in our case is setName) .
the hook variable can be an array too, string or Boolean and whatever we want .

REACT DEV TOOLS

- Introduction :

  We gonna use the comande :

  ```
  npm install -g serve
  serve -s build
  ```

  in order to preview the structure and the nesting of templates inside the main function .

OUTPUTTING LISTS

- Creating a input :

  We will create a hook variable which will contains an array of objects (blogs in our case) .

  We gonna use a map function in order to cycle to printing inside the template .

  The map method ballbacks function for each item whereby each item around we want to return a bit of jsx template , and that's going to go inside the parentheses .

  When we output a list using a map method, each root element in the template we return ,must have a key property, this key property is something that react use to keep track of

each item in the DOM as it outputs it, (if we updates, remove or add items, react keep track of those items) .

```
import {useState} from "react";

const Home = () => {
   const [blogs, setGlogs] = useState([
       {title: 'first blog', body: 'we gonna say tha we have ...',
author: 'ashraf khabar', id: 1} ,
       {title: 'second blog', body: 'we gonna also say that we ...',
author: 'sami aouad', id: 2}
   ]);

    return (
        <div className='home'>
            <div>
                {blogs.map( (blog) => (
                    <div className="blog-preview" key={blog.id}>
                        <h2>{blog.title}</h2>
                        <p>Written by {blog.author}</p>
                    </div>
                ))}
            </div>
        </div>
    ) ;
}

export default Home ;
```

- Pros :

In the home page we will show all blogs or some of them, but in search page we will show only blogs which fit with the input in the search field , so the list of blogs have the same structure in both pages , only the filter change (depends on what we want to show) .

In src/App.test.js :

```
import {useState} from "react";
import BlogList from "./blogList";

const Home = () => {
   const [blogs, setGlogs] = useState([
       {title: 'first blog', body: 'we gonna say tha we have ...',
author: 'ashraf khabar', id: 1} ,
       {title: 'second blog', body: 'we gonna also say that we ...',
author: 'sami aouad', id: 2}
   ]);

    return (
        <div className='home'>
          <BlogList blogs={blogs}/>// first blogs is the name of
variable / the second blogs is the variable above
        </div>
    ) ;
}

export default Home ;
```

in src/components/Home.js :

```
import {useState} from "react";
import BlogList from "./blogList";

const Home = () => {
   const [blogs, setGlogs] = useState([
       {title: 'first blog', body: 'we gonna say tha we have ...',
author: 'ashraf khabar', id: 1} ,
       {title: 'second blog', body: 'we gonna also say that we ...',
author: 'sami aouad', id: 2} ,
       {title: 'third blog', body: 'we gonna also say that look who we
are ...', author: 'ashraf khabar', id: 3}
   ]);

    return (
        <div className='home'>
           <BlogList blogs={blogs} title="All blogs"/>
           <BlogList blogs={blogs.filter( (blog) => blog.author ==
'ashraf khabar' )} title="Ashraf's blogs"/>
        </div>
    ) ;
}

export default Home ;
```

in src/components/blogList.js :

```
const BlogList = (props) => {
const blogs = props.blogs ;

    return (
        <div className='blog-list'>
                {blogs.map( (blog) => (
                    <div className="blog-preview" key={blog.id}>
                        <h2>{blog.title}</h2>
                        <p>Written by {blog.author}</p>
                    </div>
                ))}
        </div>
    ) ;
}

export default BlogList ;
```

an other way :

```
const BlogList = ({blogs, title}) => {
const blogs = props.blogs ;

    return (
```

```
        <div className='blog-list'>
                {blogs.map( (blog) => (
                    <div className="blog-preview" key={blog.id}>
                        <h2>{blog.title}</h2>
                        <p>Written by {blog.author}</p>
                    </div>
                ))}
        </div>
    ) ;
}

export default BlogList ;
```

- Filter and reusing components :

  We gonna use the filter function which fires a rollback function for each item of the array , if we return true for the item it keeps in the array , false no .

  In src/components/Home.js :

```
import {useState} from "react";
import BlogList from "./blogList";

const Home = () => {
   const [blogs, setGlogs] = useState([
       {title: 'first blog', body: 'we gonna say tha we have ...',
author: 'ashraf khabar', id: 1} ,
       {title: 'second blog', body: 'we gonna also say that we ...',
author: 'sami aouad', id: 2} ,
       {title: 'third blog', body: 'we gonna also say that look who we
are ...', author: 'ashraf khabar', id: 3}
   ]);

    return (
        <div className='home'>
           <BlogList blogs={blogs} title="All blogs"/>
           <BlogList blogs={blogs.filter( (blog) => blog.author ==
'ashraf khabar' )} title="Ashraf's blogs"/>
        </div>
    ) ;
}

export default Home ;
```

- Function as prop:

  What if I want a user to delete a blog or something like this .

  In src/components/Home.js :

```
import {useState} from "react";
import BlogList from "./blogList";
import blogList from "./blogList";
```

```
const Home = () => {
   const [blogs, setGlogs] = useState([
       {title: 'first blog', body: 'we gonna say tha we have ...',
author: 'ashraf khabar', id: 1} ,
       {title: 'second blog', body: 'we gonna also say that we ...',
author: 'sami aouad', id: 2} ,
       {title: 'third blog', body: 'we gonna also say that look who we
are ...', author: 'ashraf khabar', id: 3}
   ]);

   const handleDelete = (id) => {
        const newBlog = blogs.filter(blog => blog.id !== id);
        setGlogs(newBlog);
   }

    return (
        <div className='home'>
           <BlogList blogs={blogs} title="All blogs"
handleDelete={handleDelete}/>
           <BlogList blogs={blogs.filter( (blog) => blog.author ===
'ashraf khabar' )} title="Ashraf's blogs"/>
        </div>
    ) ;
}

export default Home ;
```

in src/components/blogList.js :

```
const BlogList = (props) => {
    const blogs = props.blogs ;
    const title = props.title ;
    const handleDelete = props.handleDelete ;

    return (
        <div className='blog-list'>
                <h2>{title}</h2>
                {blogs.map( (blog) => (
                    <div className="blog-preview" key={blog.id}>
                        <h2>{blog.title}</h2>
                        <p>Written by {blog.author}</p>
                        <button onClick={() =>
handleDelete(blog.id)}>Delete blog</button>
                    </div>
                ))}
        </div>
    ) ;
}

export default BlogList ;
```

- useEffect hook :

  this hook runs a function every render of the components .

  PS : rendering means showing the output in the browser , because Javascript uses the document object model (DOM) to manipulate the DOM elements

```
import {useState, useEffect} from "react";
import BlogList from "./blogList";
import blogList from "./blogList";

const Home = () => {
   const [blogs, setGlogs] = useState([
       {title: 'first blog', body: 'we gonna say tha we have ...',
author: 'ashraf khabar', id: 1} ,
       {title: 'second blog', body: 'we gonna also say that we ...',
author: 'sami aouad', id: 2} ,
       {title: 'third blog', body: 'we gonna also say that look who we
are ...', author: 'ashraf khabar', id: 3}
   ]);

   const handleDelete = (id) => {
        const newBlog = blogs.filter(blog => blog.id !== id);
        setGlogs(newBlog);
   }

   useEffect( () => {
      alert('use effect run');
   });

    return (
        <div className='home'>
           <BlogList blogs={blogs} title="All blogs"
handleDelete={handleDelete}/>
           <BlogList blogs={blogs.filter( (blog) => blog.author ===
'ashraf khabar' )} title="Ashraf's blogs"/>
        </div>
    ) ;
}

export default Home ;
```

- **Dependencies of useEffect :**

  When we want only useEffect function rendered after a certain data had chaged not all the data inside the main function .

  For example we have an array which inglobe an objects , and we have a string value , these variables are both useHook variables (reactable variables) , I want a useEffect function rerandered after the data changed, but only the String value , so I will pass a dependency as a second argument of that function .

```
import {useState, useEffect} from "react";
import BlogList from "./blogList";
import blogList from "./blogList";

const Home = () => {
   const [blogs, setGlogs] = useState([
       {title: 'first blog', body: 'we gonna say tha we have ...',
author: 'ashraf khabar', id: 1} ,
       {title: 'second blog', body: 'we gonna also say that we ...',
author: 'sami aouad', id: 2} ,
       {title: 'third blog', body: 'we gonna also say that look who we
```

```
are ...', author: 'ashraf khabar', id: 3}
   ]);

   const [name, setName] = useState('ashraf') ;

   const handleDelete = (id) => {
        const newBlog = blogs.filter(blog => blog.id !== id);
        setGlogs(newBlog);
   }

   useEffect( () => {
      alert('use effect run');
   }, [name]);

    return (
        <div className='home'>
           <BlogList blogs={blogs} title="All blogs"
handleDelete={handleDelete}/>
           <BlogList blogs={blogs.filter( (blog) => blog.author ===
'ashraf khabar' )} title="Ashraf's blogs"/>
            <button onClick={() => setName('khabar')}>change</button>
        </div>
    ) ;
}

export default Home ;
```

JSON SERVER

- What is exactly :

  We gonna use a json server to build a fake API to generate a database , because mainly we do not use a local variables like we did above, but we use a structured database .

- Use it :

  - npx json-server --watch data/db.json --port 8000

  - we gonna use the endPoints :

/blogs	GET	Fetch all blogs
/blogs/{id}	GET	Fetch a single blog
/blogs	POST	Add a new blog
/Blogs/{id}	DELETE	Delete a blog

- Fetch data :

  In src/components/Home.js :

```
import {useState, useEffect} from "react";
import BlogList from "./blogList";
```

```
const Home = () => {
   const [blogs, setGlogs] = useState(null);

   const [name, setName] = useState('ashraf') ;

   useEffect( () => {
      fetch('http://localhost:8000/blogs')
          .then(res => {
              return res.json()
          })
          .then(data => {
              console.log(data) ;
              setGlogs(data);
          })
   }, []);

    return (
        <div className='home'>
            {blogs && <BlogList blogs={blogs} title="All blogs"/>}
        </div>
    ) ;
}

export default Home ;
```

we add blogs && … because we initialize the array with null , and we will map into null first when we run the server and it s not allowed .

- {blogs && <BlogList blogs={blogs} title="All blogs" handleDelete={handleDelete}/>} => we first evaluate blogs , if it fulse we doesn't go to the next statement , but if it true we go to the next statement and run it .

We can add a loading while we didn't get the data yet :

```
import {useState, useEffect} from "react";
import BlogList from "./blogList";

const Home = () => {
   const [blogs, setGlogs] = useState(null);
   const [isPending, setIsPending] = useState(true);

   useEffect( () => {
      fetch('http://localhost:8000/blogs')
          .then(res => {
              return res.json()
          })
          .then(data => {
              console.log(data) ;
              setGlogs(data);
              setIsPending(false) ;
          })
   }, []);

    return (
        <div className='home'>
            {isPending && <div>loading...</div>}
            {blogs && <BlogList blogs={blogs} title="All blogs"/>}
```

```
        </div>
    ) ;
}

export default Home ;
```

- Errors :

```
import {useState, useEffect} from "react";
import BlogList from "./blogList";

const Home = () => {
   const [blogs, setGlogs] = useState(null);
   const [isPending, setIsPending] = useState(true);

   useEffect( () => {
      fetch('http://localhost:8000/blogs')
          .then(res => {
              if(res.ok) {
                  throw Error('couldn t fetch the data') ;
              }
              return res.json()
          })
          .then((data) => {
              console.log(data) ;
              setGlogs(data);
              setIsPending(false) ;
          })
          .then((e) => {
              console.log(e.message);
          })
   }, []);

    return (
        <div className='home'>
            {isPending && <div>loading...</div>}
            {blogs && <BlogList blogs={blogs} title="All blogs"/>}
        </div>
    ) ;
}

export default Home ;
```

when we through an error with a specific message we catch it from the catch promise below .

we can create a custom error by using hooks in order to render it into the browser. Again we gonna use the conditions like before using just the logical conditions in java script.

- Custom hook :

In src/useFetch.js :

```
import {useEffect, useState} from "react";

const useFetch = (url) => {
    const [data, setData] = useState(null);
    const [isPending, setIsPending] = useState(true);
    const [error, setError] = useState(null);

    useEffect( () => {
        fetch(url)
            .then(res => {
                if(!res.ok) {
                    throw Error('couldn t fetch the data') ;
                }
                return res.json()
            })
                .then(data => {
                    setData(data);
                    setIsPending(false) ;
                    setError(null) ;
                })
                .then(e => {
                    setIsPending(false) ;
                    setError(e.message);
            })
    }, [url]);

    return {
        data,
        isPending,
        error
    }
}

export default useFetch ;
```

in src/components/Home.js :

```
import BlogList from "./blogList";
import useFetch from "../useFetch";

const Home = () => {
    const {blogs, isPending, error} =
useFetch('http://localhost:8000/blogs') ;

    return (
        <div className='home'>
            {error && <div>{error}</div>}
            {isPending && <div>loading...</div>}
            {blogs && <BlogList blogs={blogs} title="All blogs"/>}
        </div>
    ) ;
}

export default Home ;
```

THE REACT ROUTER

- Installation :

  - npm install react-router-dom@5
  - importing something like :

```
import {BrowserRouter as Router, Route, Switch } from "react-router-dom";
```

- implementation :

  in order to surround the entire application with the router we need first to surround the code in app.js with a router tag .

  also where we want our page content to go when we go to different pages , we gonna use the switch tag .

  in src/App.js :

```
import Navbar from "./components/Navbar";
import Home from "./components/Home";
import {BrowserRouter as Router, Route, Switch } from "react-router-dom";
import Create from "./components/Create";

function App() {

    return (
       <Router>
           <div className="App">
               <Navbar />
               <div className='content'>
                   <Switch>
                       <Route exact path='/'>
                            <Home/>
                       </Route>
                       <Route path='/Create'>
                           <Create/>
                       </Route>
                   </Switch>
               </div>
           </div>
       </Router>
    );
}
export default App;
```

- Router links :

```
import {Link} from "react-router-dom";

const Navbar = () => {
    return(
        <nav className="navbar">
            <h1>
                BloGOSS
            </h1>
            <div>
```

```
                <Link to="/">Home</Link>
                <Link to="/Create">New blog</Link>
            </div>
        </nav>
    );
}

export default Navbar;
```

- **useEffect cleanup :**

```
import {useEffect, useState} from "react";

const useFetch = (url) => {
    const [data, setData] = useState(null);
    const [isPending, setIsPending] = useState(true);
    const [error, setError] = useState(null);

    useEffect( () => {
        const abortCont = new AbortController() ;
        fetch(url, {signal: abortCont.signal})
            .then(res => {
                if(!res.ok) {
                    throw Error('couldn t fetch the data') ;
                }
                return res.json()
            })
                .then(data => {
                    setData(data);
                    setIsPending(false) ;
                    setError(null) ;
                })
                .catch(err => {
                    if (err.name === 'AbortError'){
                        console.log('fetch aborted')
                    } else {
                        setIsPending(false) ;
                        setError(err.message);
                    }
            })
        return () => abortCont.abort() ;
    }, [url]);

    return {
        data,
        isPending,
        error
    }
}

export default useFetch ;
```

- **parameters in the router :**

we gonna use a new hook name useParams : *allows us to grab paramters from the url.*

In src/App.js :

```
import Navbar from "./components/Navbar";
import Home from "./components/Home";
import {BrowserRouter as Router, Route, Switch } from "react-router-dom";
import Create from "./components/Create";
import BlogDetails from "./components/BlogDetails";

function App() {

    return (
       <Router>
           <div className="App">
               <Navbar />
               <div className='content'>
                   <Switch>
                       <Route exact path='/'>
                            <Home/>
                       </Route>
                       <Route path='/Create'>
                           <Create/>
                       </Route>
                       <Route path='/Blogs/:id'>
                           <BlogDetails/>
                       </Route>
                   </Switch>
               </div>
           </div>
       </Router>
    );
}
export default App;
```

in src/components/BlogDetails.js :

```
import {useParams} from "react-router-dom";

const BlogDetails = () => {
    const {id} = useParams()  // allows us to grab paramters from the url

    return (
        <div className='blog-details'>
            <h2>Blog details - {id}</h2>
        </div>
    );
}

export default BlogDetails ;
```

in src/components/blogList.js :

```
import {Link} from "react-router-dom";
```

```
const BlogList = (props) => {
    const blogs = props.blogs ;
    const title = props.title ;

    return (
        <div className='blog-list'>
                <h2>{title}</h2>
                {blogs.map( (blog) => (
                    <div className="blog-preview" key={blog.id}>
                        <Link to={"/Blogs/"+blog.id}
style={{textDecoration: 'none'}}>
                            <h2>{blog.title}</h2>
                            <p>Written by {blog.author}</p>
                            <br></br>
                        </Link>
                        <div className="blog-preview">
                            <button className="buttons" >Delete
blog</button>
                        </div>
                    </div>
                ))}
        </div>
    ) ;
}

export default BlogList ;
```

- reuse the custom hook :

  in src/components/BlogDetails.js :

```
import {useParams} from "react-router-dom";
import useFetch from "../useFetch";

const BlogDetails = () => {
    const {id} = useParams()  // allows us to grab paramters from the url
    const {data : blog, isPending, error} =
useFetch('http://localhost:8000/blogs/'+id);

    return (
        <div className='blog-details'>
            {isPending && <div>Loading...</div>}
            {error && <div>{error.message}</div>}
            {blog && (
                <article>
                    <h2>{blog.title}</h2>
                    <br></br>
                    <p>Written by <strong>{blog.author}</strong></p>
                    <br></br>
                    <div>{blog.body}</div>
                </article>
            )}
        </div>
    );
}

export default BlogDetails ;
```

- **Controlled inputs :**

```
import {useState} from "react";

const Create = () => {
    const [title, setTitle] = useState('') ;
    const [body, setBody] = useState('') ;
    const [author, setAuthor] = useState('ashraf khabar') ;

    return (
        <div className="create">
            <h2>Add a new blog</h2>
            <form>
                <label>Blog title : </label>
                <input
                    type='text'
                    required
                    value={title}
                    onChange={(e) => setTitle(e.target.value)}
                />
                <label>Blog body : </label>
                <textarea
                    required
                    value={body}
                    onChange={(e) => setBody(e.target.value)}
                ></textarea>
                <label>Blog author : </label>
                <select
                    value={author}
                    onChange={(e) => setAuthor(e.target.value)}
                >
                    <option value='ashraf khabar'>ashraf khabar</option>
                    <option value='sami aouad'>sami aoaud</option>
                </select>
                <button>Add blog</button>
            </form>
            <div>{title}</div>
        </div>

    ) ;
}

export default Create ;
```

- **Submit event :**

```
import {useState} from "react";

const Create = () => {
    const [title, setTitle] = useState('') ;
    const [body, setBody] = useState('') ;
    const [author, setAuthor] = useState('ashraf khabar') ;

    const handleSubmit = (e) => {
        e.preventDefault();
        const blog = {title, body, author};
    }

    return (
```

```
        <div className="create">
            <h2>Add a new blog</h2>
            <form onSubmit={handleSubmit}>
                <label>Blog title : </label>
                <input
                    type='text'
                    required
                    value={title}
                    onChange={(e) => setTitle(e.target.value)}
                />
                <label>Blog body : </label>
                <textarea
                    required
                    value={body}
                    onChange={(e) => setBody(e.target.value)}
                ></textarea>
                <label>Blog author : </label>
                <select
                    value={author}
                    onChange={(e) => setAuthor(e.target.value)}
                >
                    <option value='ashraf khabar'>ashraf khabar</option>
                    <option value='sami aouad'>sami aoaud</option>
                </select>
                <button>Add blog</button>
            </form>
            <div>{title}</div>
        </div>

    ) ;
}

export default Create ;
```

- Making a POST request :

We gonna use a fetch api and this time it gonna have a second argument where we gonna tack on the data and also define the type of request (in this case is POST) .

Also a headers is the content type that is being sent, in this case we gonna make (application/json) , in layman language we gonna telling the server that we are sending Jason data .

And the body property, is the actual data sending by this request , and because we want it to be Jason object we should convert it to a json object not a js-object .

And because the fetch ipi function is asynchronous and return a promis ,we need to add then then option, which fire a function when this is complete wich gonna have (CONSOLE.LOG('NEW BLOG ADDED')) .

```
import {useState} from "react";

const Create = () => {
    const [title, setTitle] = useState('') ;
```

```
    const [body, setBody] = useState('') ;
    const [author, setAuthor] = useState('ashraf khabar') ;
    const [isPending, setIsPending] = useState(false) ;

    const handleSubmit = (e) => {
        e.preventDefault();
        setIsPending(true) ;

        const blog = {title, body, author};
        fetch('http://localhost:8000/blogs', {
            method: 'POST',
            headers: {'Content-type': "application/json"},
            body: JSON.stringify(blog)
        }).then(() => {
            console.log('New blog added');
            setIsPending(false) ;
        })
    }

    return (
        <div className="create">
            <h2>Add a new blog</h2>
            <form onSubmit={handleSubmit}>
                <label>Blog title : </label>
                <input
                    type='text'
                    required

          value={title}
                    onChange={(e) => setTitle(e.target.value)}
                />
                <label>Blog body : </label>
                <textarea
                    required
                    value={body}
                    onChange={(e) => setBody(e.target.value)}
                ></textarea>
                <label>Blog author : </label>
                <select
                    value={author}
                    onChange={(e) => setAuthor(e.target.value)}
                >
                    <option value='ashraf khabar'>ashraf khabar</option>
                    <option value='sami aouad'>sami aoaud</option>
                </select>
                {!isPending && <button>Add blog</button>}
                {isPending && <button disabled>Adding blog ... </button>}
            </form>
        </div>

    ) ;
}

export default Create ;
```

- **Redirect :**

```
import {useState} from "react";
import {useHistory} from "react-router-dom";
```

```
const Create = () => {
    const [title, setTitle] = useState('') ;
    const [body, setBody] = useState('') ;
    const [author, setAuthor] = useState('ashraf khabar') ;
    const [isPending, setIsPending] = useState(false) ;
    const history = useHistory();

    const handleSubmit = (e) => {
        e.preventDefault();
        setIsPending(true) ;

        const blog = {title, body, author};
        fetch('http://localhost:8000/blogs', {
            method: 'POST',
            headers: {'Content-type': "application/json"},
            body: JSON.stringify(blog)
        }).then(() => {
            console.log('New blog added');
            setIsPending(false) ;
            // history.go(-1)
            history.push('/');
        })
    }

    return (
        <div className="create">
            <h2>Add a new blog</h2>
            <form onSubmit={handleSubmit}>
                <label>Blog title : </label>
                <input
                    type='text'
                    required
                    value={title}
                    onChange={(e) => setTitle(e.target.value)}
                />
                <label>Blog body : </label>
                <textarea
                    required
                    value={body}
                    onChange={(e) => setBody(e.target.value)}
                ></textarea>
                <label>Blog author : </label>
                <select
                    value={author}
                    onChange={(e) => setAuthor(e.target.value)}
                >
                    <option value='ashraf khabar'>ashraf khabar</option>
                    <option value='sami aouad'>sami aoaud</option>
                </select>
                {!isPending && <button>Add blog</button>}
                {isPending && <button disabled>Adding blog ... </button>}
            </form>
        </div>

    ) ;
}

export default Create ;
```

- **Deleting :**

```
import {useHistory, useParams} from "react-router-dom";
import useFetch from "../useFetch";

const BlogDetails = () => {
    const {id} = useParams()  // allows us to grab paramters from the url
    const {data : blog, isPending, error} =
useFetch('http://localhost:8000/blogs/'+id);
    const history = useHistory() ;

    const handleClick = () => {
        fetch('http://localhost:8000/blogs/'+blog.id, {
            methode: 'DELETE'
        }).then(() => {
            history.push('/') ;
        })
    }

    return (
        <div className='blog-details'>
            {isPending && <div>Loading...</div>}
            {error && <div>{error.message}</div>}
            {blog && (
                <article>
                    <h2>{blog.title}</h2>
                    <br></br>
                    <p>Written by <strong>{blog.author}</strong></p>
                    <br></br>
                    <div>{blog.body}</div>
                    <button onClick={handleClick}>Delete</button>
                </article>
            )}
        </div>
    );
}

export default BlogDetails ;
```

- Nor found – 404 error :

  In src/components/NotFound.js :

```
import {Link} from "react-router-dom";

const NotFound = () => {

    return (
        <div className='not-found'>
            <h2>Sorry</h2>
            <p>That page was not found</p>
            <Link to="/">Back to home page ...</Link>
        </div>
    ) ;
}

export default NotFound ;
```

  in src/App.js :

```jsx
import Navbar from "./components/Navbar";
import Home from "./components/Home";
import {BrowserRouter as Router, Route, Switch } from "react-router-dom";
import Create from "./components/Create";
import BlogDetails from "./components/BlogDetails";
import NotFound from "./components/NotFound";

function App() {

    return (
       <Router>
           <div className="App">
               <Navbar />
               <div className='content'>
                   <Switch>
                       <Route exact path='/'>
                            <Home/>
                       </Route>
                       <Route path='/Create'>
                           <Create/>
                       </Route>
                       <Route path='/Blogs/:id'>
                           <BlogDetails/>
                       </Route>
                       <Route path='*'>
                           <NotFound/>
                       </Route>
                   </Switch>
               </div>
           </div>
       </Router>
    );
}
export default App;
```