

ReactJS. Front to backend. BT 2022

Project 2 - Github-finder

<https://github.com/PacktPublishing/React-Front-to-Back-2022>

Table of Contents

0 Hints.....	3
1 GitHub Finder project. Start.....	4
1.1 Intro.....	4
1.2 Setup Tailwind and Daisy UI.....	4
1.2.1 Tailwind.....	4
1.2.2 Daisy UI.....	5
1.3 Navbar component.....	6
1.4 Footer component.....	7
1.5 Main Pages and routes.....	8
2 Working with GitHub AP.....	9
2.1 Github API & getting token.....	9
2.1.1 How can you use GitHub API?.....	9
2.1.2 How to get GitHub API token?.....	9
2.1.3 How to use environment variables?.....	9
2.2 UserList component.....	10
2.3 Spinner Component.....	11
2.4 Display Users.....	12
2.5 Setup GitHub Context.....	13
2.6 Setup GitHub Reducer.....	14
2.7 Refactoring code (clean up Fetch Users).....	14
2.8 UserSearch component.....	15
2.9 Search Users.....	16
2.10 Clear functionality.....	17
10 User Profile and Alerts.....	18
10.1 Alert Context and Reducer.....	18

0 Hints

1 GitHub Finder project. Start

1.1 Intro

1.2 Setup Tailwind and Daisy UI

1.2.1 Tailwind

First of all you need to create a new project

CLI=> *npx create-react-app github-finder --use-npm*

Go to the new folder (github-finder)

CLI=> *npm install -D tailwindcss postcss autoprefixer*

CLI=> *npx tailwindcss init -p*

Open file **tailwind.config.js** and paste

```
/** @type {import('tailwindcss').Config} */
module.exports = {
  content: [
    "./src/**/*.{js,jsx,ts,tsx}",
  ],
  theme: {
    extend: {},
  },
  plugins: [],
}
```

Open **index.css**

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

1.2.2 Daisy UI

CLI=> `npm i daisyui`

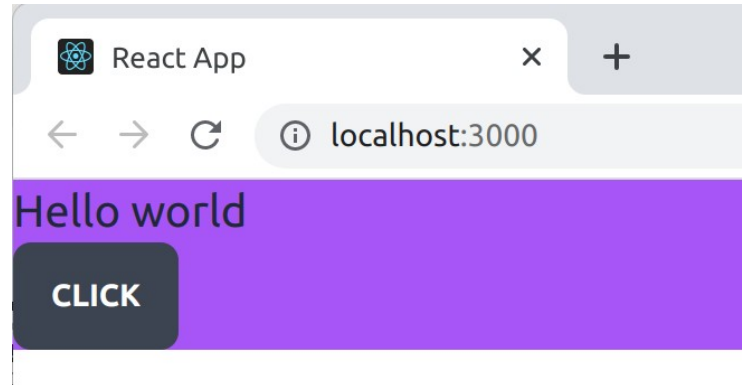
Open file `tailwind.config.js`

plugins: `[require('daisyui')],`

Now you can start your react app using `npm start`.

Change main component `App.js` => `App.jsx` (to be able to work with tag plugin)

```
App.jsx  U  X
github-finder > src > App.jsx > ...
1  import logo from './logo.svg';
2  import './App.css';
3
4  function App() {
5    return (
6      <div className='bg-purple-500'>
7        <h1 className='text-xl'>Hello world</h1>
8        <button className='btn'>Click</button>
9      </div>
10    );
11  }
12
13  export default App;
14
```



1.3 Navbar component

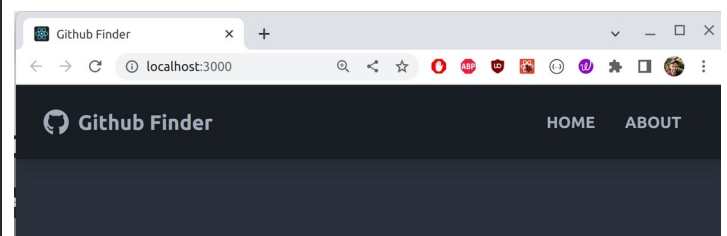
You need to install react-router-dom and react-icons

CLI=> `npm i react-router-dom react-icons`

Create a new **Navbar.jsx** component in github-finder/src/components/layout

```
Navbar.jsx U X
src > components > layout > Navbar.jsx > ...
1 import { FaGithub } from 'react-icons/fa';
2 import { Link } from 'react-router-dom';
3 import PropTypes from 'react';
4
5 const Navbar = ({ title }) => {
6   return (
7     <nav className='navbar mb-12 shadow-lg bg-neutral text-neutral-content'>
8       <div className='container mx-auto'>
9         <div className='flex-none px-2 mx-2'>
10           <FaGithub className='inline pr-2 text-3xl' />
11           <Link to='/' className='text-lg font-bold align-middle'>
12             {title}
13           </Link>
14         </div>
15         <div className='flex-1 px-2 mx-2'>
16           <div className='flex justify-end'>
17             <Link to='/' className='btn btn-ghost btn-sm rounded-btn'>
18               Home
19             </Link>
20             <Link to='/' className='btn btn-ghost btn-sm rounded-btn'>
21               About
22             </Link>
23           </div>
24         </div>
25       </div>
26     </nav>
27   );
28 };
29 Navbar.defaultProps = {
30   title: 'Github Finder',
31 };
32 Navbar.propTypes = {
33   title: PropTypes.string,
34 };
35 export default Navbar;
36
```

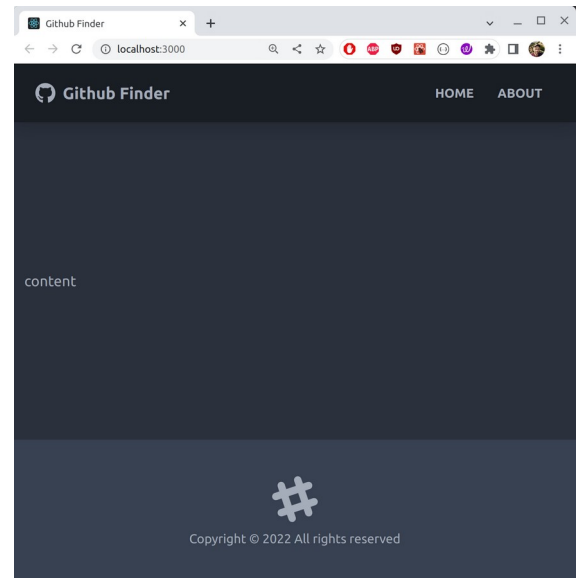
```
App.jsx M X
src > App.jsx > ...
1 import { BrowserRouter as Router, Route } from 'react-router-dom';
2 import Navbar from '../components/layout/Navbar';
3 const App = () => {
4   return (
5     <Router>
6       <div className='flex flex-col justify-between h-screen'>
7         <Navbar />
8         <main>content</main>
9       </div>
10     </Router>
11   );
12 };
13
14 export default App;
```



1.4 Footer component

```
Footer.jsx U X
src > components > layout > Footer.jsx > ...
1  const Footer = props => {
2    const footerYear = new Date().getFullYear();
3    return (
4      <footer className='footer p-10 bg-gray-700 text-primary-content footer-center'>
5        <div>
6          <svg
7            width='50'
8            height='50'
9            viewBox='0 0 24 24'
10           xmlns='http://www.w3.org/2000/svg'
11           fillRule='evenodd'
12           clipRule='evenodd'
13           className='inline-block fill-current'
14         >
15           <path d='M22.672 15.226l-2.432 8.11 8.41 2.515c.33 1.019-.209 2.127-1.23 2.456-1.15.
325-2.148-.321-2.463-1.226l-.84-2.518-5.013 1.677 8.4 2.517c.391 1.203-.434 2.542-1.
831 2.542-.88 0-1.601-.564-1.86-1.314l-.842-2.516-2.431 8.09c-1.135 3.28-2.145-.317-2.
463-1.229-.329-1.018 2.11 2.127 1.231-2.456l2.432-.809-1.621-4.823-2.432 8.08c-1.355.
384-2.558-.59-2.558-1.839 0-.817 5.09-1.582 1.327-1.846l2.433-.809-.842-2.515c-.33-1.
02 2.11-2.129 1.232-2.458 1.02-.329 2.13 2.09 2.461 1.229l.842 2.515 5.011-1.677-.
839-2.517c-.403-1.238 4.84-2.553 1.843-2.553 8.19 0 1.585 5.09 1.85 1.326l.841 2.517 2.
431-.81c1.02-.33 2.131 2.11 2.461 1.229 3.32 1.018-.21 2.126-1.23 2.456l-2.433 8.09 1.
622 4.823 2.433-.809c1.242-.401 2.557 4.84 2.557 1.838 0 .819-.51 1.583-1.328 1.
847m-8.992-6.428l-5.01 1.675 1.619 4.828 5.011-1.674-1.62-4.829'></path>
16         </svg>
17         <p>Copyright &copy; {footerYear} All rights reserved</p>
18       </div>
19     </footer>
20   );
21 };
22
23 export default Footer;
```

```
App.jsx M X
src > App.jsx > ...
1  import { BrowserRouter as Router, Route } from 'react-router-dom';
2  import Navbar from './components/layout/Navbar';
3  import Footer from './components/layout/Footer';
4  const App = () => {
5    return (
6      <Router>
7        <div className='flex flex-col justify-between h-screen'>
8          <Navbar />
9          <main className='container mx-auto px-3 pb-12'>content</main>
10         <Footer />
11       </div>
12     </Router>
13   );
14 };
15
16 export default App;
```



1.5 Main Pages and routes

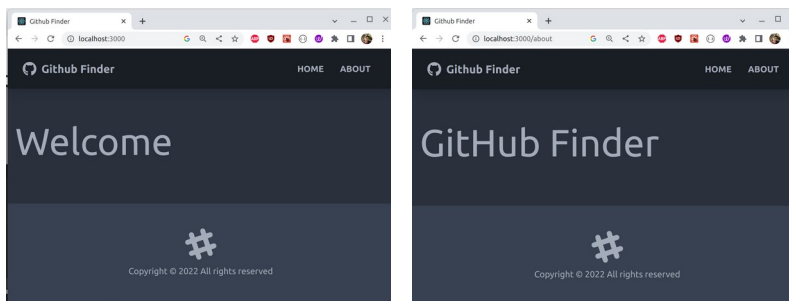
Create a new folder github-finder/src/pages to store components for all the main pages

Create Home.jsx, About.jsx and NotFound.jsx components

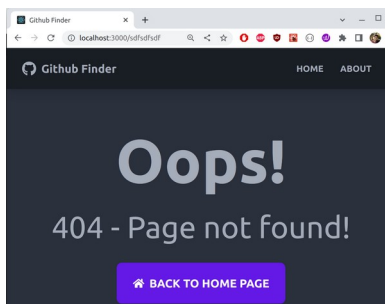
```
App.jsx M X
src > App.jsx > ...
1 import { BrowserRouter as Router, Route, Routes } from 'react-router-dom';
2 import Navbar from './components/layout/Navbar';
3 import Footer from './components/layout/Footer';
4 import Home from './pages/Home';
5 import About from './pages/About';
6 import NotFound from './pages/NotFound';
7
8 const App = () => {
9   return (
10     <Router>
11       <div className='flex flex-col justify-between h-screen'>
12         <Navbar />
13         <main className='container mx-auto px-3 pb-12'>
14           <Routes>
15             <Route path='/' element={ <Home /> } />
16             <Route path='/about' element={ <About /> } />
17             <Route path='/notfound' element={ <NotFound /> } />
18             <Route path='/*' element={ <NotFound /> } />
19           </Routes>
20         </main>
21         <Footer />
22       </div>
23     </Router>
24   );
25 };
26
27 export default App;
28
```

```
About.jsx U X
src > pages > About.jsx > ...
1 const About = props => {
2   return (
3     <div>
4       <h1 className='text-6xl mb-4'>GitHub Finder</h1>
5     </div>
6   );
7 };
8
9 export default About;
10
```

```
Home.jsx U X
src > pages > Home.jsx > ...
1 const Home = props => {
2   return (
3     <div>
4       <h1 className='text-6xl mb-4'>Welcome</h1>
5     </div>
6   );
7 };
8
9 export default Home;
10
```



```
NotFound.jsx U X
src > pages > NotFound.jsx > ...
1 import { FaHome } from 'react-icons/fa';
2 import { Link } from 'react-router-dom';
3
4 const NotFound = props => {
5   return (
6     <div className='hero'>
7       <div className='text-center hero-content'>
8         <div className='max-w-lg'>
9           <h1 className='text-8xl font-bold mb-8'>Oops!</h1>
10          <p className='text-5xl mb-8'>404 - Page not found!</p>
11          <Link to='/' className='btn btn-primary btn-lg'>
12            <FaHome className='mr-2' />
13            BACK to Home Page
14          </Link>
15        </div>
16      </div>
17    </div>
18  );
19 };
20
21 export default NotFound;
22
```



2 Working with GitHub AP

2.1 Github API & getting token

2.1.1 How can you use GitHub API?

You can get information about any github user by ID, search users, get list of repositories and so on.

2.1.2 How to get GitHub API token?

Visit github.com. Go to your profile => settings => developer settings => tokens (classic) => generate new token (classic)

Select user and repo and click on generate token => copy and save **YOUR** token

`ghp_UGuH6CyTtyjQ2ybWJolcyyJza0utl60OTf0x`

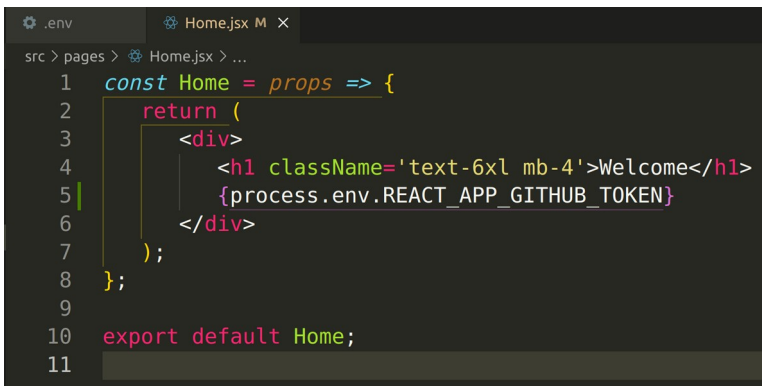
2.1.3 How to use environment variables?

Create a new file `.env` (in root folder of your react project/folder/app – github-finder)

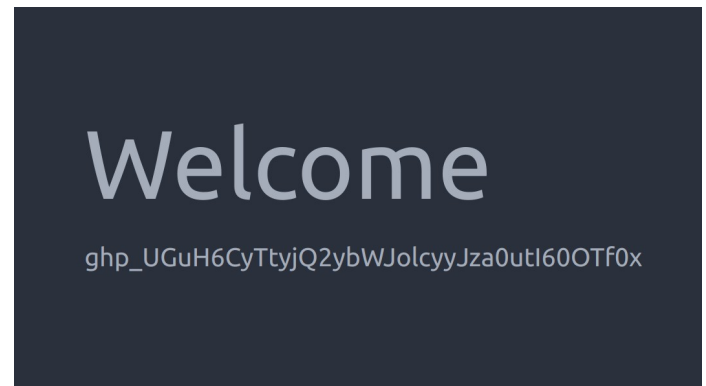
Add 2 variables:

```
REACT_APP_GITHUB_TOKEN="ghp_UGuH6CyTtyjQ2ybWJolcyyJza0utl60OTf0x"
REACT_APP_GITHUB_URL="https://api.github.com"
```

How you can check if it works:



```
.env
Home.jsx M X
src > pages > Home.jsx > ...
1  const Home = props => {
2    return (
3      <div>
4        <h1 className='text-6xl mb-4'>Welcome</h1>
5        {process.env.REACT_APP_GITHUB_TOKEN}
6      </div>
7    );
8  };
9
10 export default Home;
11
```



```
Welcome
ghp_UGuH6CyTtyjQ2ybWJolcyyJza0utl60OTf0x
```

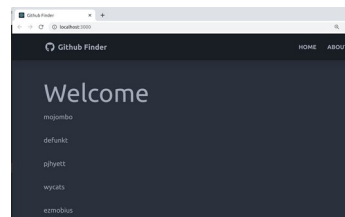
2.2 UserList component

Create a new folder to store all the components that related to User entity

github-finder/src/components/users. There create a new component UserResults.jsx

```
src > components > users > UserResults.jsx > ...
1 import { useEffect, useState } from 'react';
2
3 const UserResults = props => {
4   const [users, setUsers] = useState([]);
5   const [loading, setLoading] = useState(true);
6   useEffect(() => {
7     fetchUsers();
8   }, []);
9
10  const fetchUsers = async () => {
11    const response = await fetch(`${process.env.REACT_APP_GITHUB_URL}/users`, {
12      headers: {
13        Authorization: `token ${process.env.REACT_APP_GITHUB_TOKEN}`,
14      },
15    });
16    const data = await response.json();
17    setUsers(data);
18    setLoading(false);
19  };
20  if (!loading) {
21    return (
22      <div className='grid grid-cols-1 gap-8 xl:grid-cols-4 lg:grid-cols-3 md:grid-col-2'>
23        {users.map(user => (
24          <h3>{user.login}</h3>
25        ))}
26      </div>
27    );
28  } else {
29    return <h3>Loading ...</h3>;
30  }
31 };
32
33 export default UserResults;
34
```

```
src > pages > Home.jsx > ...
1 import UserResults from '../components/users/UserResults';
2 const Home = props => {
3   return (
4     <div>
5       <h1 className='text-6xl mb-4'>Welcome</h1>
6       { /* TODO: SEARCH COMPONENT */ }
7       <UserResults />
8     </div>
9   );
10 };
11
12 export default Home;
13
```



2.3 Spinner Component

Create a new folder to store additional files like a spinner. github-finder/src/components/layout/assets and add there a spinner.

Create a new component Spinner.jsx in layout folder

```
src > components > users > @ UserResults.jsx > ...
1 import { useEffect, useState } from 'react';
2 import Spinner from '../layout/Spinner';
3
4 const UserResults = props => {
5   const [users, setUsers] = useState([]);
6   const [loading, setLoading] = useState(true);
7   useEffect(() => {
8     fetchUsers();
9   }, []);
10
11   const fetchUsers = async () => {
12     const response = await fetch(`${process.env.REACT_APP_GITHUB_URL}/users`, {
13       headers: {
14         Authorization: `token ${process.env.REACT_APP_GITHUB_TOKEN}`,
15       },
16     });
17     const data = await response.json();
18     setUsers(data);
19     setLoading(false);
20   };
21   if (!loading) {
22     return (
23       <div className='grid grid-cols-1 gap-8 xl:grid-cols-4 lg:grid-cols-3 md:grid-col-2'>
24         {users.map(user => (
25           <h3>{user.login}</h3>
26         ))}
27       </div>
28     );
29   } else {
30     return <Spinner />;
31   }
32 };
33
34 export default UserResults;
```

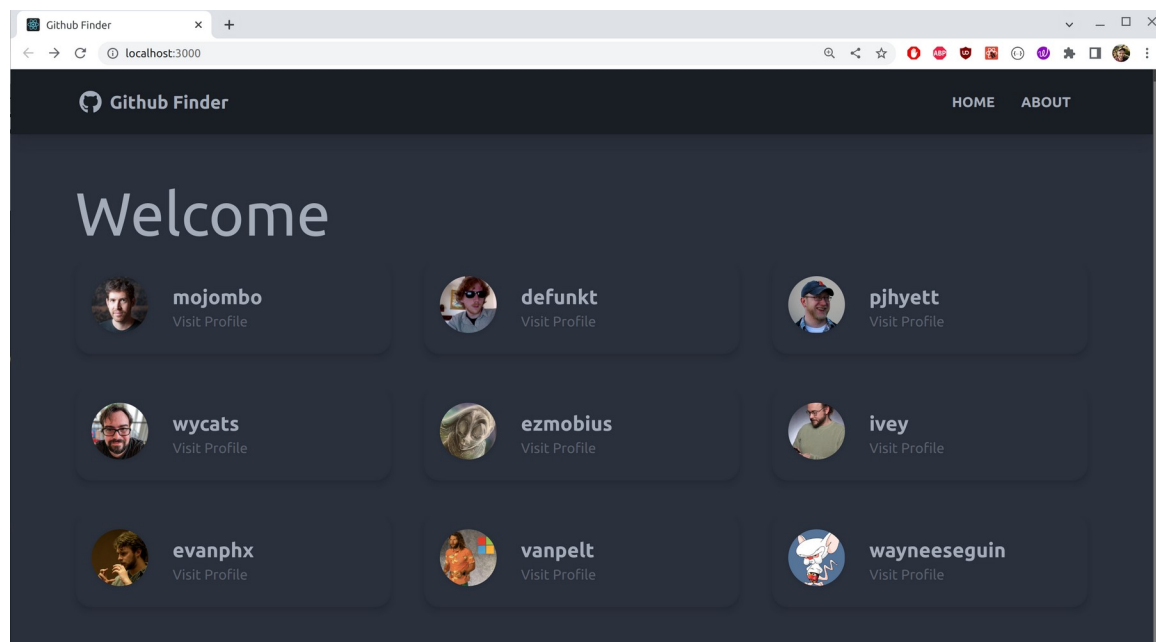
```
src > components > layout > @ Spinner.jsx > ...
1 import spinnerGif from '../assets/spinner.gif';
2 const Spinner = () => {
3   return (
4     <div className='w-100 mt-20'>
5       <img width={180} src={spinnerGif} alt='Loading ...' className='text-center mx-auto' />
6     </div>
7   );
8 };
9
10 export default Spinner;
```

2.4 Display Users

You need to create UserItem.jsx component

```
src > components > users > UserItem.jsx > ...
1 import PropTypes from 'prop-types';
2 import { Link } from 'react-router-dom';
3
4 const UserItem = ({ user: { login, avatar_url } }) => {
5   return (
6     <div className='card shadow-md compact side bg-base-100'>
7       <div className='flex-row items-center space-x-4 card-body'>
8         <div className='avatar'>
9           <div className='rounded-full shadow w-14 h-14'>
10             <img src={avatar_url} alt='Profile' />
11           </div>
12         </div>
13         <div>
14           <h2 className='card-title'>{login}</h2>
15           <Link className='text-base-content text-opacity-40' to={`/${users}/${login}`}>
16             Visit Profile
17           </Link>
18         </div>
19       </div>
20     </div>
21   );
22 };
23
24 UserItem.propTypes = {
25   user: PropTypes.object.isRequired,
26 };
27
28 export default UserItem;
```

```
src > components > users > UserResults.jsx > ...
1 import { useEffect, useState } from 'react';
2 import Spinner from '../layout/Spinner';
3 import UserItem from './UserItem';
4
5 const UserResults = props => {
6   const [users, setUsers] = useState([]);
7   const [loading, setLoading] = useState(true);
8   useEffect(() => {
9     fetchUsers();
10   }, []);
11
12   const fetchUsers = async () => {
13     const response = await fetch(`${process.env.REACT_APP_GITHUB_URL}/users`, {
14       headers: {
15         Authorization: `token ${process.env.REACT_APP_GITHUB_TOKEN}`,
16       },
17     });
18     const data = await response.json();
19     setUsers(data);
20     setLoading(false);
21   };
22   if (!loading) {
23     return (
24       <div className='grid grid-cols-1 gap-8 xl:grid-cols-4 lg:grid-cols-3 md:grid-cols-2'>
25         {users.map(user => (
26           <UserItem key={user.id} user={user} />
27         ))}
28       </div>
29     );
30   } else {
31     return <Spinner />;
32   }
33 };
34
35 export default UserResults;
```



2.5 Setup GitHub Context.

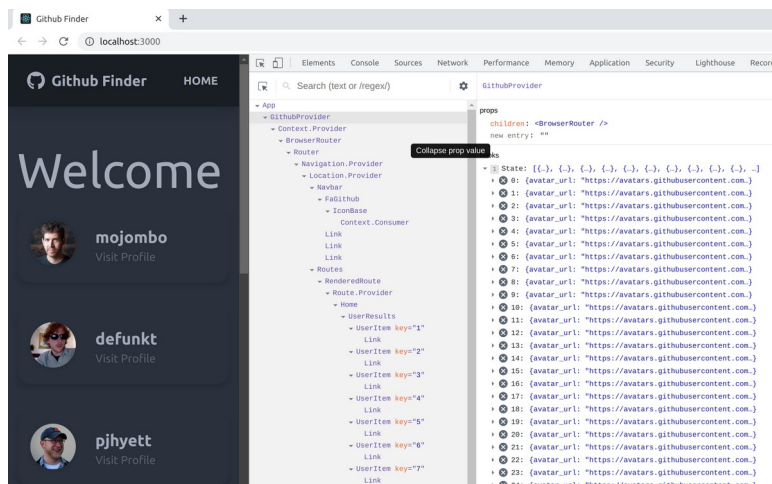
Before this lesson some components had component level states (useState hooks) and it was possible to send some parameters from one component to another using props drilling - what is very difficult in case of big app. Now you need to replace it with Context – it's app level state (you have to import it in main App.js component and wrap everything in that component). If you need to use any functionality from an app level state (context) inside another component – you just need to import context, destructure all the functionality that you are going to use and just reuse them.

Create a new folder github-finder/src/context/github

Create a new context file GithubContext.js

```
1 import { createContext, useState } from 'react';
2
3 const GithubContext = createContext();
4
5 const GITHUB_URL = process.env.REACT_APP_GITHUB_URL;
6 const GITHUB_TOKEN = process.env.REACT_APP_GITHUB_TOKEN;
7
8 export const GithubProvider = ({ children }) => {
9   const [users, setUsers] = useState([]);
10   const [loading, setLoading] = useState(true);
11
12   const fetchUsers = async () => {
13     const response = await fetch(`${GITHUB_URL}/users`, {
14       headers: {
15         Authorization: `token ${GITHUB_TOKEN}`,
16       },
17     });
18     const data = await response.json();
19     setUsers(data);
20     setLoading(false);
21   };
22
23   return (
24     <GithubContext.Provider value={{ users, loading, fetchUsers }}>
25       {children}
26     </GithubContext.Provider>
27   );
28 };
29
30 export default GithubContext;
```

```
1 import { BrowserRouter as Router, Route, Routes } from 'react-router-dom';
2 import Navbar from './components/Layout/Navbar';
3 import Footer from './components/Layout/Footer';
4 import Home from './pages/Home';
5 import About from './pages/About';
6 import NotFound from './pages/NotFound';
7 import { GithubProvider } from './context/github/GithubContext';
8
9 const App = () => {
10   return (
11     <GithubProvider>
12       <Router>
13         <div className='flex flex-col justify-between h-screen'>
14           <Navbar />
15           <main className='container mx-auto px-3 pb-12'>
16             <Routes>
17               <Route path='/' element={ <Home /> } />
18               <Route path='/about' element={ <About /> } />
19               <Route path='/notfound' element={ <NotFound /> } />
20               <Route path='/*' element={ <NotFound /> } />
21             </Routes>
22           </main>
23           <Footer />
24         </div>
25       </Router>
26     </GithubProvider>
27   );
28 };
29
30 export default App;
```



```
1 import { useContext, useEffect } from 'react';
2 import Spinner from './Layout/Spinner';
3 import UserItem from './UserItem';
4 import GithubContext from '../context/github/GithubContext';
5
6 const UserResults = props => {
7   const { users, loading, fetchUsers } = useContext(GithubContext);
8
9   useEffect(() => {
10     fetchUsers();
11   }, []);
12
13   if (!loading) {
14     return (
15       <div className='grid grid-cols-1 gap-8 xl:grid-cols-4 lg:grid-cols-3 md:grid-cols-2'>
16         {users.map(user => (
17           <UserItem key={user.id} user={user} />
18         ))}
19       </div>
20     );
21   } else {
22     return <Spinner />;
23   }
24 };
25
26 export default UserResults;
```

2.6 Setup GitHub Reducer.

Create a new file github-finder/src/context/github/GithubReducer.js

```
1  const githubReducer = (state, action) => {
2    switch (action.type) {
3      case 'GET_USERS':
4        return {
5          ...state,
6          users: action.payload,
7          loading: false,
8        };
9      default: {
10       return state;
11     }
12   }
13 };
14
15 export default githubReducer;
```

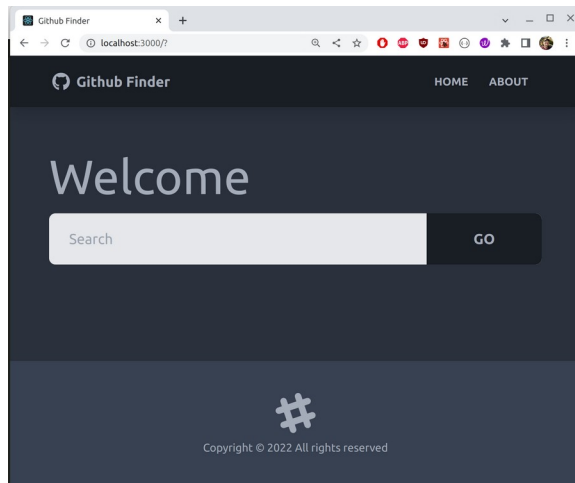
```
1  import { createContext, useReducer } from 'react';
2  import githubReducer from './GithubReducer';
3
4  const GithubContext = createContext();
5
6  const GITHUB_URL = process.env.REACT_APP_GITHUB_URL;
7  const GITHUB_TOKEN = process.env.REACT_APP_GITHUB_TOKEN;
8
9  export const GithubProvider = ({ children }) => {
10    const initialState = {
11      users: [],
12      loading: true,
13    };
14
15    const [state, dispatch] = useReducer(githubReducer, initialState);
16
17    const fetchUsers = async () => {
18      const response = await fetch(`${GITHUB_URL}/users`, {
19        headers: {
20          Authorization: `token ${GITHUB_TOKEN}`,
21        },
22      });
23      const data = await response.json();
24      dispatch({ type: 'GET_USERS', payload: data });
25    };
26
27    return (
28      <GithubContext.Provider value={{ users: state.users, loading: state.loading, fetchUsers }}>
29        {children}
30      </GithubContext.Provider>
31    );
32  };
33
34 export default GithubContext;
```

2.7 Refactoring code (clean up Fetch Users)

2.8 UserSearch component

Create a new component UserSearch.js

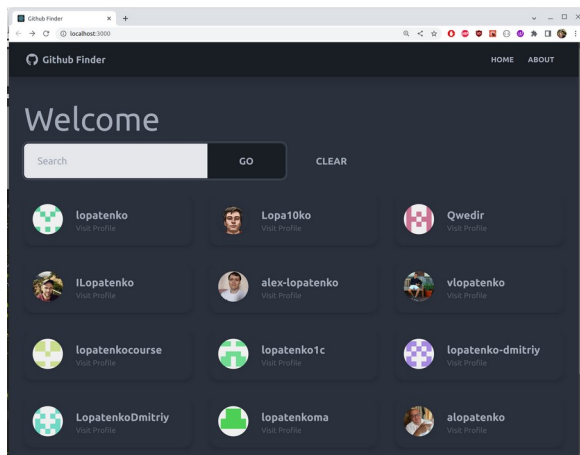
```
src > pages > Home.jsx > ...
1 import UserResults from '../components/users/UserResults';
2 import UserSearch from '../components/users/UserSearch';
3 const Home = props => {
4   return (
5     <div>
6       <h1 className='text-6xl mb-4'>Welcome</h1>
7       <UserSearch />
8       <UserResults />
9     </div>
10   );
11 };
12
13 export default Home;
14
```



```
src > components > users > UserSearch.jsx > ...
1 import { useState, useContext } from 'react';
2 import GithubContext from '../../context/github/GithubContext';
3
4 const UserSearch = () => {
5   const [text, setText] = useState('');
6   const { users } = useContext(GithubContext);
7   const handleChange = e => {
8     setText(e.target.value);
9   };
10
11   const handleSubmit = e => {
12     e.preventDefault();
13     if (text === '') {
14       alert('Please enter something!');
15     } else {
16       /* TODO: search users */
17       setText('');
18     }
19   };
20
21   const handleClear = e => {
22     setText('');
23   };
24
25   return (
26     <div className='grid grid-cols-1 xl:grid-cols-2 lg:grid-cols-2 md:grid-cols-2 mb-8 gap-8'>
27       <div>
28         <form onSubmit={handleSubmit}>
29           <div className='form-control'>
30             <div className='relative'>
31               <input
32                 type='text'
33                 className='w-full pr-40 bg-gray-200 input input-lg text-black'
34                 placeholder='Search'
35                 value={text}
36                 onChange={handleChange}
37               />
38               <button
39                 type='submit'
40                 className='absolute top-0 right-0 rounded-l-none w-36 btn btn-lg'
41               >
42                 Go
43               </button>
44             </div>
45           </div>
46         </form>
47       </div>
48       <div>
49         {(users.length > 0 || text.length !== 0) && (
50           <button
51             className='btn btn-ghost btn-lg'
52             onClick={handleClear}
53           >
54             Clear
55           </button>
56         )}
57       </div>
58     </div>
59   );
60 };
61
62 export default UserSearch;
```

2.9 Search Users

```
src > components > users > UserSearch.jsx > ...
1 import { useState, useContext } from 'react';
2 import GithubContext from '../context/github/GithubContext';
3
4 const UserSearch = () => {
5   const [text, setText] = useState('');
6   const { users, searchUsers } = useContext(GithubContext);
7   const handleChange = e => {
8     setText(e.target.value);
9   };
10  const handleSubmit = e => {
11    e.preventDefault();
12    if (text === '') {
13      alert('Please enter something');
14    } else {
15      searchUsers(text);
16      setText('');
17    }
18  };
19  const handleClear = e => { ...
21  };
22  return ( ...
53  );
54 };
55
56 export default UserSearch;
57
```



```
src > context > github > GithubContext.js > ...
1 import { createContext, useReducer } from 'react';
2
3
4 const GithubContext = createContext();
5
6 const GITHUB_URL = process.env.REACT_APP_GITHUB_URL;
7 const GITHUB_TOKEN = process.env.REACT_APP_GITHUB_TOKEN;
8
9 export const GithubProvider = ({ children }) => {
10  const initialState = {
13  };
14
15  const [state, dispatch] = useReducer(githubReducer, initialState);
16
17  const searchUsers = async text => {
18    setLoading();
19    const params = new URLSearchParams({
20      q: text,
21    });
22    const response = await fetch(`${GITHUB_URL}/search/users?${params}`, {
23      headers: {
24        Authorization: `token ${GITHUB_TOKEN}`,
25      },
26    });
27    const { items } = await response.json();
28    dispatch({ type: 'GET_USERS', payload: items });
29  };
30
31  const setLoading = () => dispatch({ type: 'SET_LOADING' });
32  return (
33    <GithubContext.Provider value={{ users: state.users, loading: state.loading, searchUsers }}>
34      {children}
35    </GithubContext.Provider>
36  );
37 };
38
39 export default GithubContext;
40
```


2.10 Clear functionality

```
JS GithubContext.js M X
src > context > github > JS GithubContext.js > ...
1 > import { createContext, useReducer } from 'react'; ...
2
3
4 const GithubContext = createContext();
5
6 const GITHUB_URL = process.env.REACT_APP_GITHUB_URL;
7 const GITHUB_TOKEN = process.env.REACT_APP_GITHUB_TOKEN;
8
9 export const GithubProvider = ({ children }) => {
10 >   const initialState = { ...
11
12   };
13
14   const [state, dispatch] = useReducer(githubReducer, initialState);
15
16   const searchUsers = async text => { ...
17 >   };
18
19   const clearUsers = () => dispatch({ type: 'CLEAR_USERS' });
20
21   const setLoading = () => dispatch({ type: 'SET_LOADING' });
22 >   return ( ...
23
24   );
25 };
26
27 export default GithubContext;
```

```
JS UserSearch.jsx M X
src > components > users > JS UserSearch.jsx > ...
1 import { useState, useContext } from 'react';
2 import GithubContext from '../context/github/GithubContext';
3
4 const UserSearch = () => {
5   const [text, setText] = useState('');
6   const { users, searchUsers, clearUsers } = useContext(GithubContext);
7   const handleChange = e => {
8     setText(e.target.value);
9   };
10 >   const handleSubmit = e => { ...
11
12   };
13   const handleClear = e => {
14     if (users.length !== 0) {
15       clearUsers();
16     }
17     setText('');
18   };
19 >   return ( ...
20
21   );
22 };
23
24 export default UserSearch;
```

```
JS GithubReducer.js M X
src > context > github > JS GithubReducer.js > ...
1 const githubReducer = (state, action) => {
2   switch (action.type) {
3     case 'GET_USERS':
4       return {
5         ...state,
6         users: action.payload,
7         loading: false,
8       };
9     case 'SET_LOADING':
10      return {
11        ...state,
12        loading: true,
13      };
14     case 'CLEAR_USERS':
15      return {
16        ...state,
17        users: [],
18      };
19     default: {
20       return state;
21     }
22   }
23 };
24
25 export default githubReducer;
```

10 User Profile and Alerts

10.1 Alert Context and Reducer

Create a new folder `github-app/src/context/alert`

Create there 2 new files `AlertReducer.js` and `AlertContext.js`

```
JS AlertContext.js U X
src > context > alert > JS AlertContext.js > ...
1 import { createContext, useReducer } from 'react';
2 import alertReducer from './Alert.Reducer';
3
4 const AlertContext = createContext();
5
6 export const AlertProvider = ({ children }) => {
7   const initialState = null;
8
9   const [state, dispatch] = useReducer(alertReducer, initialState);
10
11   const setAlert = (msg, type) => {
12     dispatch({ type: 'SET_ALERT', payload: { msg, type } });
13     setTimeout(() => dispatch({ type: 'REMOVE_ALERT' }), 3000);
14   };
15
16   return (
17     <AlertContext.Provider value={{ alert: state, setAlert }}>{children}</AlertContext.Provider>
18   );
19 };
20
21 export default AlertContext;
```

```
JS AlertContext.js U JS Alert.Reducer.js U X
src > context > alert > JS Alert.Reducer.js > ...
1 const alertReducer = (state, action) => {
2   switch (action.type) {
3     case 'SET_ALERT':
4       return action.payload;
5     case 'REMOVE_ALERT':
6       return null;
7
8     default: {
9       return state;
10    }
11  }
12 };
13
14 export default alertReducer;
15
```