

- Now, we'll refactor our codes. Also we use axios for our actions.

GithubActions.js

```
/** For use those function to another file or Component, we have to  
add export just before write function. */
```

```
import axios from "axios";  
const GITHUB_URL = process.env.REACT_APP_GITHUB_URL;  
const GITHUB_TOKEN = process.env.REACT_APP_GITHUB_TOKEN;  
  
const github = axios.create({ // we create new axios here and set  
it.  
  baseURL: GITHUB_URL,  
  headers: { Authorization: `token ${GITHUB_TOKEN}` }  
})
```

```
// Get search Result  
export const searchUsers = async (text) => { / First we delete  
setLoading, we'll add it at UserSearch Component.
```

```
  const params = new URLSearchParams({  
    q: text  
  })  
  
  const response = await github.get(`/search/users?${params}`)  
  
  return response.data.items {/ we'll return our data.  
}
```

```
// Get Single User and Repos
```

```
export const getUserAndRepos = async (login) => {/ with axios we  
merged our user and repos and use them like below  
  const [user, repos] = await Promise.all([  
    github.get(`/users/${login}`), / github.get comes from const  
github.  
    github.get(`/users/${login}/repos`)  
  ])
```

```
    return { user: user.data, repos: repos.data } // we'll return our
data.
}
```

GithubContext.js

```
import { createContext, useReducer } from "react";
import githubReducer from "../GithubReducer";

const GithubContext = createContext();

export const GithubProvider = ({ children }) => {
  const initialState = {
    users: [],
    user: {},
    repos: [],
    loading: false,
  }

  const [state, dispatch] = useReducer(githubReducer, initialState)

  return <GithubContext.Provider value={{
    ...state, / Here we delete other things, because we moved them
to Actions file. Also we add dispatch here and we call another
states with dispatch.
    dispatch,
  }}>
    {children}
  </GithubContext.Provider>
}

export default GithubContext
```

UserSearch.jsx

```
import { useState, useContext } from "react";
import GithubContext from "../../context/github/GithubContext";
import AlertContext from "../../context/alert/AlertContext";
import { searchUsers } from "../../context/github/GithubActions";

function UserSearch() {
  const [text, setText] = useState("");

  const { users, dispatch } = useContext(GithubContext); / We catch
dispatch and users here.

  const { setAlert } = useContext(AlertContext);

  const handleChange = (e) => setText(e.target.value);
  const handleSubmit = async (e) => { / We changed it async.
    e.preventDefault();

    if (text === "") {
      setAlert("Please Enter Something", "error");
    } else {
      dispatch({ type: "SET_LOADING" }); / We use dispatch here,
and it catches case we wrote at GithubReducer
      const users = await searchUsers(text);
      dispatch({ type: "GET_USERS", payload: users }); / We use
dispatch here, and it catches case we wrote at GithubReducer
      setText("");
    }
  };

  return (
    <div className="grid grid-cols-1 xl:grid-cols-2 lg:grid-cols-2
md:grid-cols-2 mb-8 gap-8">
      <div>
        <form onSubmit={handleSubmit}>
          <div className="form-control">
            <div className="relative">
              <input
```

```

        type="text"
        className="w-full pr-40 bg-gray-200 input input-lg
text-black"

        placeholder="Search"
        value={text}
        onChange={handleChange}
      />
      <button
        type="submit"
        className="absolute top-0 right-0 rounded-l-none w-
36 btn btn-lg"
      >
        Go
      </button>
    </div>
  </div>
</form>
</div>
{users.length > 0 && (
  <div>
    <button {/ Like above we catch CLEAR_USERS type when we
click Clear Button.
      onClick={() => dispatch({ type: "CLEAR_USERS" })}
      className="btn btn-ghost btn-lg"
    >
      Clear
    </button>
  </div>
)}
</div>
);
}

export default UserSearch;

```

```

import { FaCodepen, FaStore, FaUserFriends, FaUsers } from "react-
icons/fa";
import { useEffect, useContext } from "react";
import { Link, useParams } from "react-router-dom";
import GithubContext from "../context/github/GithubContext";
import Spinner from "../components/layout/Spinner";
import RepoList from "../components/repos/RepoList";
import { getUserAndRepos } from "../context/github/GithubActions";

function User() { / we delete everythin we don't need, here too.
And we add what we are going to use.
  const { user, loading, repos, dispatch } =
useContext(GithubContext);

  const params = useParams();

  useEffect(() => { / We cant set async here we have to use another
function inside of it
    dispatch({ type: "SET_LOADING" }); / First we call SET_LOADING
case.
    const getUserData = async () => {
      const userData = await getUserAndRepos(params.login);
      dispatch({ type: "GET_USER_AND_REPOS", payload: userData });
      / Here again we call GET_USER_AND_REPOS here. We Set
userData as payload.
    };

    getUserData(); / And after all we call our getUserData
function. And now we can add dependencies, because now, they are
creating when state changes
  }, [dispatch, params.login]);

  const {
    name,
    type,
    avatar_url,
    location,
    bio,
    blog,
    twitter_username,
    login,
    html_url,

```

```

    followers,
    following,
    public_repos,
    public_gists,
    hireable,
  } = user;

  if (loading) {
    return <Spinner />;
  }

  return (
    <>
      <div className="w-full mx-auto lg:w-10/12">
        <div className="mb-4">
          <Link to="/" className="btn btn-ghost">
            Back To Search
          </Link>
        </div>
        <div className="grid grid-cols-1 xl:grid-cols-3 lg:grid-
cols-3 md:grid-cols-3 mb-8 md:gap-8">
          <div className="custom-card-image mb-6 md:mb-0">
            <div className="rounded-lg shadow-xl card image-full">
              <figure>
                <img src={avatar_url} alt="" />
              </figure>
              <div className="card-body justify-end">
                <h2 className="card-title mb-0">{name}</h2>
                <p>{login}</p>
              </div>
            </div>
          </div>
          <div className="col-span-2">
            <div className="mb-6">
              <h1 className="text-3xl card-title">
                {name}
                <div className="ml-2 mr-1 badge badge-success">
{type}</div>
                {hireable && (
                  <div className="mx-1 badge badge-
info">Hireable</div>
                )}
            </div>
          </div>
        </div>
      </div>
    </>
  );

```



```

        rel="noreferrer"
      >
        {twitter_username}
      </a>
    </div>
  </div>
</div>
)}
</div>
</div>
</div>
<div className="w-full py-5 mb-6 rounded-lg shadow-md bg-
base-100 stats">
  <div className="stat">
    <div className="stat-figure text-secondary">
      <FaUsers className="text-3xl md:text-5xl" />
    </div>
    <div className="stat-title pr-5">Followers</div>
    <div className="stat-value pr-5 text-3xl md:text-4xl">
      {followers}
    </div>
  </div>
  <div className="stat">
    <div className="stat-figure text-secondary">
      <FaUserFriends className="text-3xl md:text-5xl" />
    </div>
    <div className="stat-title pr-5">Following</div>
    <div className="stat-value pr-5 text-3xl md:text-4xl">
      {following}
    </div>
  </div>
  <div className="stat">
    <div className="stat-figure text-secondary">
      <FaCodepen className="text-3xl md:text-5xl" />
    </div>
    <div className="stat-title pr-5">Public Repos</div>
    <div className="stat-value pr-5 text-3xl md:text-4xl">
      {public_repos}
    </div>
  </div>
  <div className="stat">
    <div className="stat-figure text-secondary">
      <FaStore className="text-3xl md:text-5xl" />

```



```

        </div>
        <div className="stat-title pr-5">Public Gists</div>
        <div className="stat-value pr-5 text-3xl md:text-4xl">
            {public_gists}
        </div>
    </div>
</div>
<RepoList repos={repos} />
</>
);
}

export default User;

```

GithubReducer.js

```

const githubReducer = (state, action) => {
  switch (action.type) {
    case "GET_USERS":
      return {
        ...state,
        users: action.payload,
        loading: false
      }
    case "GET_USER_AND_REPOS":
      return {
        ...state,
        user: action.payload.user,
        repos: action.payload.repos,
        loading: false
      }
    case "SET_LOADING":
      return {
        ...state,

```

```

        loading: true,
      }
    case "CLEAR_USERS":
      return {
        ...state,
        users: []
      }

    default:
      return state
  }
}

export default githubReducer

```

---

Also, if you want to not merge USER and REPOS

```

import { FaCodepen, FaStore, FaUserFriends, FaUsers } from "react-
icons/fa";
import { useEffect, useContext } from "react";
import { Link, useParams } from "react-router-dom";
import GithubContext from "../context/github/GithubContext";
import Spinner from "../components/layout/Spinner";
import RepoList from "../components/repos/RepoList";
import { getUser, getUserRepos } from
"../context/github/GithubActions";
/ We need to import both of them like above.

function User() {
  const { user, loading, repos, dispatch, } =
useContext(GithubContext);

  const params = useParams();

```

```

    useEffect(() => { / We cant set async here we have to use another
function inside of it
    dispatch({ type: "SET_LOADING" });
    const getUserData = async () => { / Here is going to one
request
        const userData = await getUser(params.login);
        dispatch({ type: "GET_USER", payload: userData });

        const userRepoData = await getUserRepos(params.login);
        dispatch({ type: "GET_REPOS", payload: userRepoData });
    };

    getUserData();
}, [dispatch, params.login]);

const {
    name,
    type,
    avatar_url,
    location,
    bio,
    blog,
    twitter_username,
    login,
    html_url,
    followers,
    following,
    public_repos,
    public_gists,
    hireable,
} = user;

if (loading) {
    return <Spinner />;
}

return (
    <>
        <div className="w-full mx-auto lg:w-10/12">
            <div className="mb-4">
                <Link to="/" className="btn btn-ghost">
                    Back To Search
            </div>
        </div>
    </>
)

```





```

        <div className="stat-value pr-5 text-3xl md:text-4xl">
          {followers}
        </div>
      </div>
      <div className="stat">
        <div className="stat-figure text-secondary">
          <FaUserFriends className="text-3xl md:text-5xl" />
        </div>
        <div className="stat-title pr-5">Following</div>
        <div className="stat-value pr-5 text-3xl md:text-4xl">
          {following}
        </div>
      </div>
      <div className="stat">
        <div className="stat-figure text-secondary">
          <FaCodepen className="text-3xl md:text-5xl" />
        </div>
        <div className="stat-title pr-5">Public Repos</div>
        <div className="stat-value pr-5 text-3xl md:text-4xl">
          {public_repos}
        </div>
      </div>
      <div className="stat">
        <div className="stat-figure text-secondary">
          <FaStore className="text-3xl md:text-5xl" />
        </div>
        <div className="stat-title pr-5">Public Gists</div>
        <div className="stat-value pr-5 text-3xl md:text-4xl">
          {public_gists}
        </div>
      </div>
    </div>
  </div>
  <RepoList repos={repos} />
</>
);
}

export default User;

```

```
const githubReducer = (state, action) => { / At here we need to  
make two different case.
```

```
  switch (action.type) {  
    case "GET_USERS":  
      return {  
        ...state,  
        users: action.payload,  
        loading: false  
      }  
    
```

```
    case "GET_USER":  
      return {  
        ...state,  
        user: action.payload,  
        loading: false  
      }  
    
```

```
    case "GET_REPOS":  
      return {  
        ...state,  
        repos: action.payload,  
        loading: false  
      }  
    
```

```
    case "SET_LOADING":  
      return {  
        ...state,  
        loading: true,  
      }  
    
```

```
    case "CLEAR_USERS":  
      return {  
        ...state,  
        users: []  
      }  
    
```

```
    default:  
      return state  
    
```

```
  }  
}
```

```
export default githubReducer
```