

GithubContext.js

`/* Firstly we change name of our fetchUsers function to searchUsers`

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

```
const GithubContext = createContext();
```

`/ At below we define a couple var,for use shortway to our url and token`

```
const GITHUB_URL = process.env.REACT_APP_GITHUB_URL;
const GITHUB_TOKEN = process.env.REACT_APP_GITHUB_TOKEN;
```

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

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

`// Get search Result`

```
const searchUsers = async (text) => {
  setLoading();

  const params = new URLSearchParams({
    q: text
  })
  /* At below we add search parameters to our fetch
  const response = await fetch(`${GITHUB_URL}/search/users?
  ${params}`, {
    headers:{
      Authorization: `token ${GITHUB_TOKEN}`,
    },
  });
  const { items } = await response.json();

  dispatch({
```

```

        type: "GET_USERS", / We don't need to change it cause we
doing same thing at here.
        payload: items,

    })
}

// Clear Users from Result

const clearUsers = () => { / We sent it to UserSearch with use
useContext
    dispatch({ / We set new dispatch type for Clear Users for our
button
        type: "CLEAR_USERS",
    })
}

// Set Loading

const setLoading = () => dispatch({ type: "SET_LOADING" })

return <GithubContext.Provider value={{
    users: state.users,
    loading: state.loading,
    searchUsers,{/ We pass here and we'll catch at UserSearch
component
    clearUsers {/ We pass here and we'll catch at UserSearch
component
    }}>
    {children}
</GithubContext.Provider>
}

export default GithubContext

```

```

import { useState, useContext } from "react";

```

```

import GithubContext from "../../context/github/GithubContext";

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

  const { users, searchUsers, clearUsers } =
    useContext(GithubContext); // we catch searchUsers and clearUsers

  const handleChange = (e) => setText(e.target.value);
  const handleSubmit = (e) => {
    e.preventDefault();

    if (text === "") {
      alert("Please Enter Something");
    } else {
      searchUsers(text);
      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}> {/ we add handleSubmit and
define it above
          <div className="form-control">
            <div className="relative">
              <input
                type="text"
                className="w-full pr-40 bg-gray-200 input-lg
text-black"
                placeholder="Search"
                value={text}
                onChange={handleChange} {/we add handleChange and
define it above
              />
              <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 onClick={clearUsers} className="btn btn-ghost
btn-lg">
      {/ clearUsers comes from our GithubContext
    Clear
  </button>
</div>
  )}
</div>
);
}

export default UserSearch;

```

```

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

```

```
    case "CLEAR_USERS":
      return { / Here we took our state and set users as empty
array, so our users will be not show at UI
        ...state,
        users: []
      }
    default:
      return state
  }
}

export default githubReducer
```