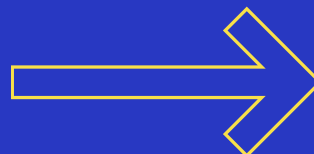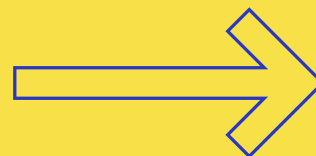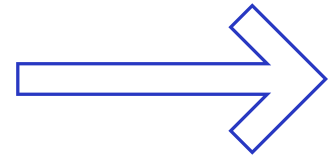# Protected routes in React

React

react-router-dom v6

Intro

Protected routes in React.js are important for building secure and authenticated web applications. They allow you to control access to certain routes based on the user's authentication status

Setup

1. Create useAuth file to check user can access or not.
2. Create PrivateRoute file to the wrap the components which we want to make secure.
3. Create PublicRoute file to the wrap the components which we want to make public.
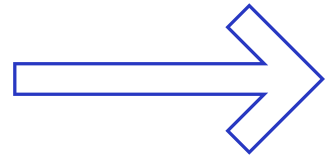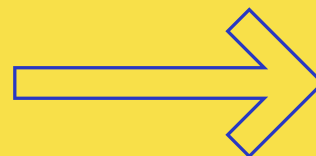
useAuth.js

```javascript
//useAuth.js

export const useAuth = () => {
    const user = localStorage.getItem('token')
    if (user) {
        return true;
    } else {
        return false
    }
};
```
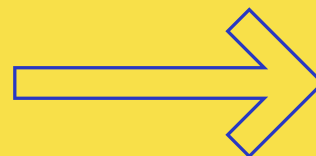
privateRoutes.js

```javascript
import { Outlet, Navigate } from 'react-router-dom'
import { useAuth } from './useAuth'

function PrivateRoutes() {
    const token = useAuth()
    return token ? <Outlet/> : <Navigate to='/login'/>
}

export default PrivateRoutes
```
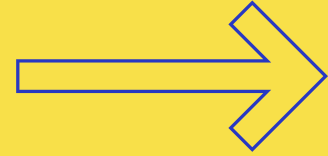
publicRoutes.js

```javascript
//publicRoutes.js

import { Outlet, Navigate } from 'react-router-dom'
import { useAuth } from './useAuth'

function PublicRoutes() {
    const token = useAuth()
    return token ? <Navigate to='/' /> : <Outlet />
}

export default PublicRoutes
```

App.js

```js
//App.js

import { Routes, Route } from 'react-router-dom'
import PrivateRoutes from "./privateRoutes"
import PublicRoutes from './publicRoutes'

function App() {
  return (
    <Routes>
      <Route element={<PrivateRoutes />}>
        //All Private Components
      </Route>
      <Route element={<PublicRoutes />}>
        //All Public Components
      </Route>
    </Routes>
  )
}

export default App
```

For more information about Protected Routing in React

https://dev.to/sundarbadagala081/protected-routes-in-react-47b1