INTEGRATING REACT WITH WEB API

Tutorial

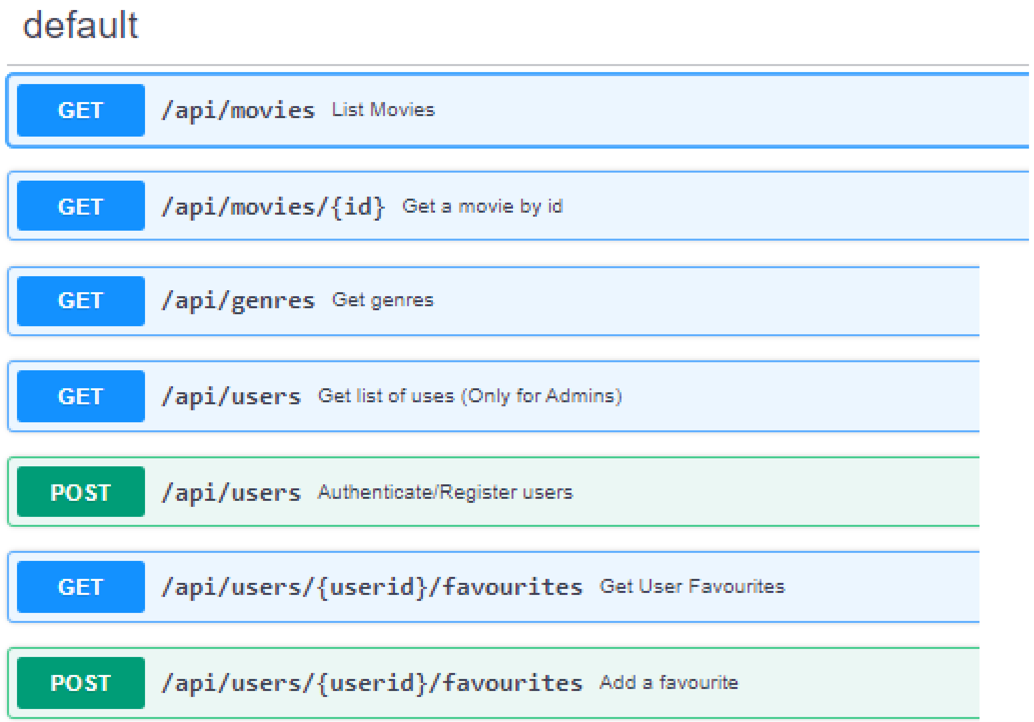
Agenda

- Review API Design
- Develop front end API Module
  - Endpoint
  - Routes
- React Authentication Context
- React Login/Registration Page

Review API Design

Fully understand the API

- Examine Routes
- Examine Request & Response Requirements
  - Body (e.g. properties)
  - Format (e.g. json)
- Examine Authentication
  - JWT/Sessions
- **TEST WITH POSTMAN**
  - Even better, write unit tests!



default

GET /api/movies List Movies

GET /api/movies/{id} Get a movie by id

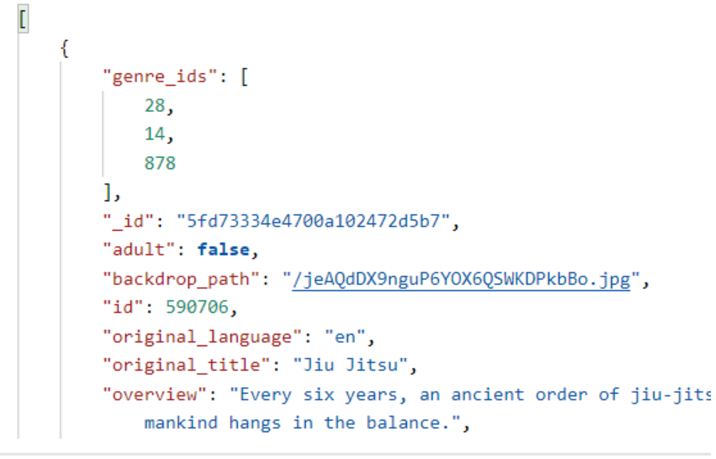
GET /api/genres Get genres

GET /api/users Get list of uses (Only for Admins)

POST /api/users Authenticate/Register users

GET /api/users/{userid}/favourites Get User Favourites

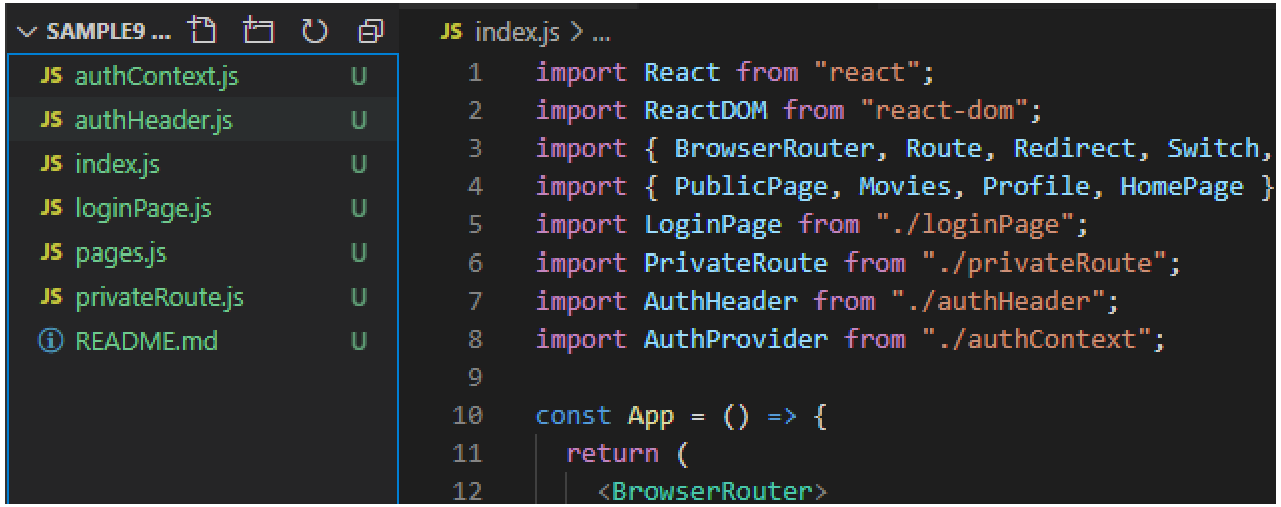
POST /api/users/{userid}/favourites Add a favourite



```
[
  {
    "genre_ids": [
      28,
      14,
      878
    ],
    "_id": "5fd73334e4700a102472d5b7",
    "adult": false,
    "backdrop_path": "/jeAQdDX9nguP6YOX6QSWKDPkbBo.jpg",
    "id": 590706,
    "original_language": "en",
    "original_title": "Jiu Jitsu",
    "overview": "Every six years, an ancient order of jiu-jits
      mankind hangs in the balance.",
```

Design React App

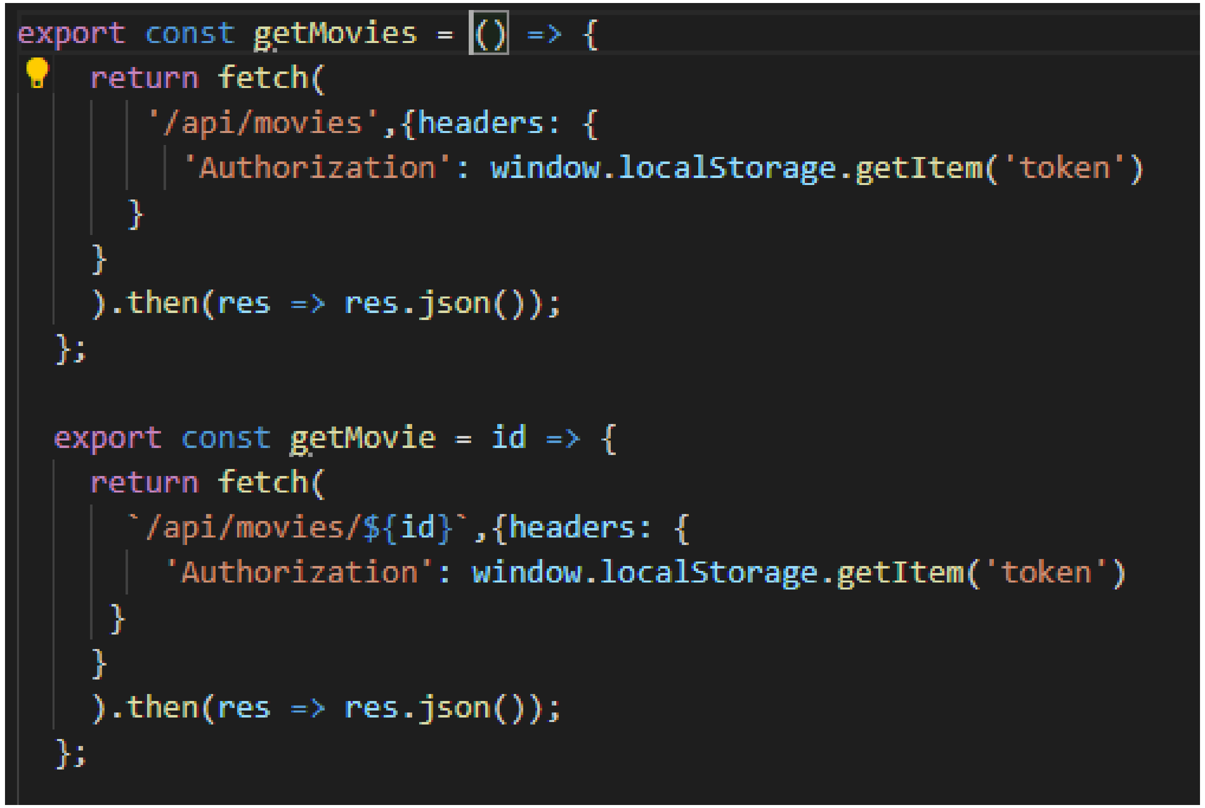
- This can be your Movies App from assignment 1
- Use stub API/TMDB up to now
- Don’t forget authentication(login/register) pages and routes



```
import React from "react";
import ReactDOM from "react-dom";
import { BrowserRouter, Route, Redirect, Switch,
import { PublicPage, Movies, Profile, HomePage }
import LoginPage from "./loginPage";
import PrivateRoute from "./privateRoute";
import AuthHeader from "./authHeader";
import AuthProvider from "./authContext";

const App = () => {
  return (
    <BrowserRouter>
```

Write API Module

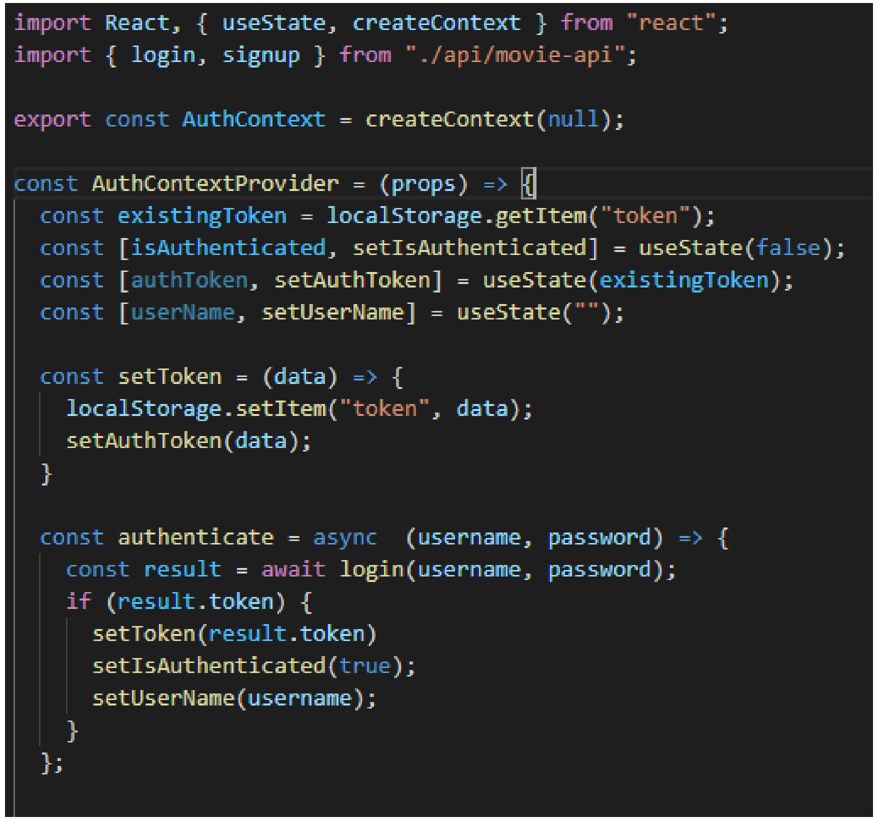
- Abstract API calls to exported functions.



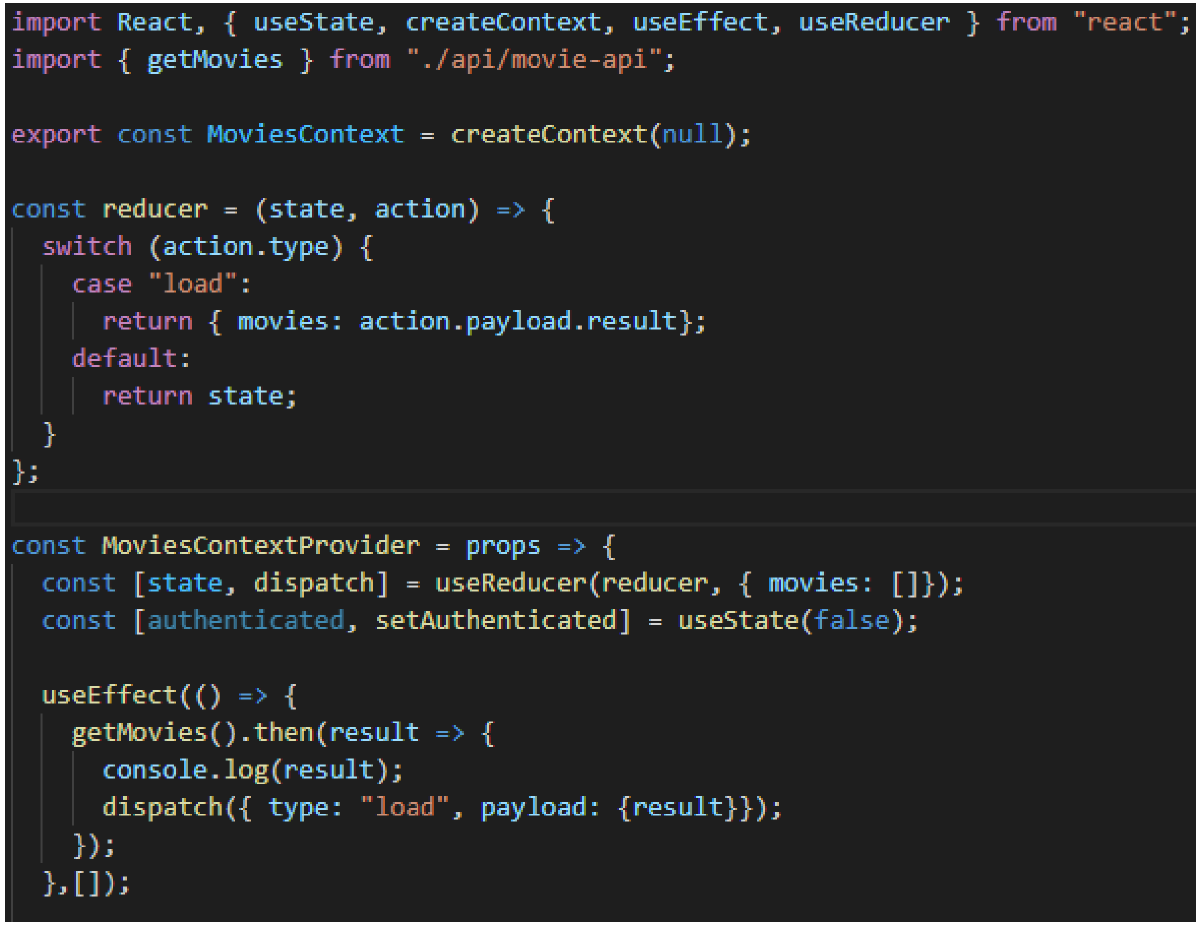
```
export const getMovies = () => {
  return fetch(
     '/api/movies',{headers: {
       'Authorization': window.localStorage.getItem('token')
    }
  }
  ).then(res => res.json());
};

export const getMovie = id => {
  return fetch(
    `/api/movies/${id}`,{headers: {
      'Authorization': window.localStorage.getItem('token')
   }
  }
  ).then(res => res.json());
};
```

Design/Refactor React Contexts to access API

- AuthContext, Movies Context:



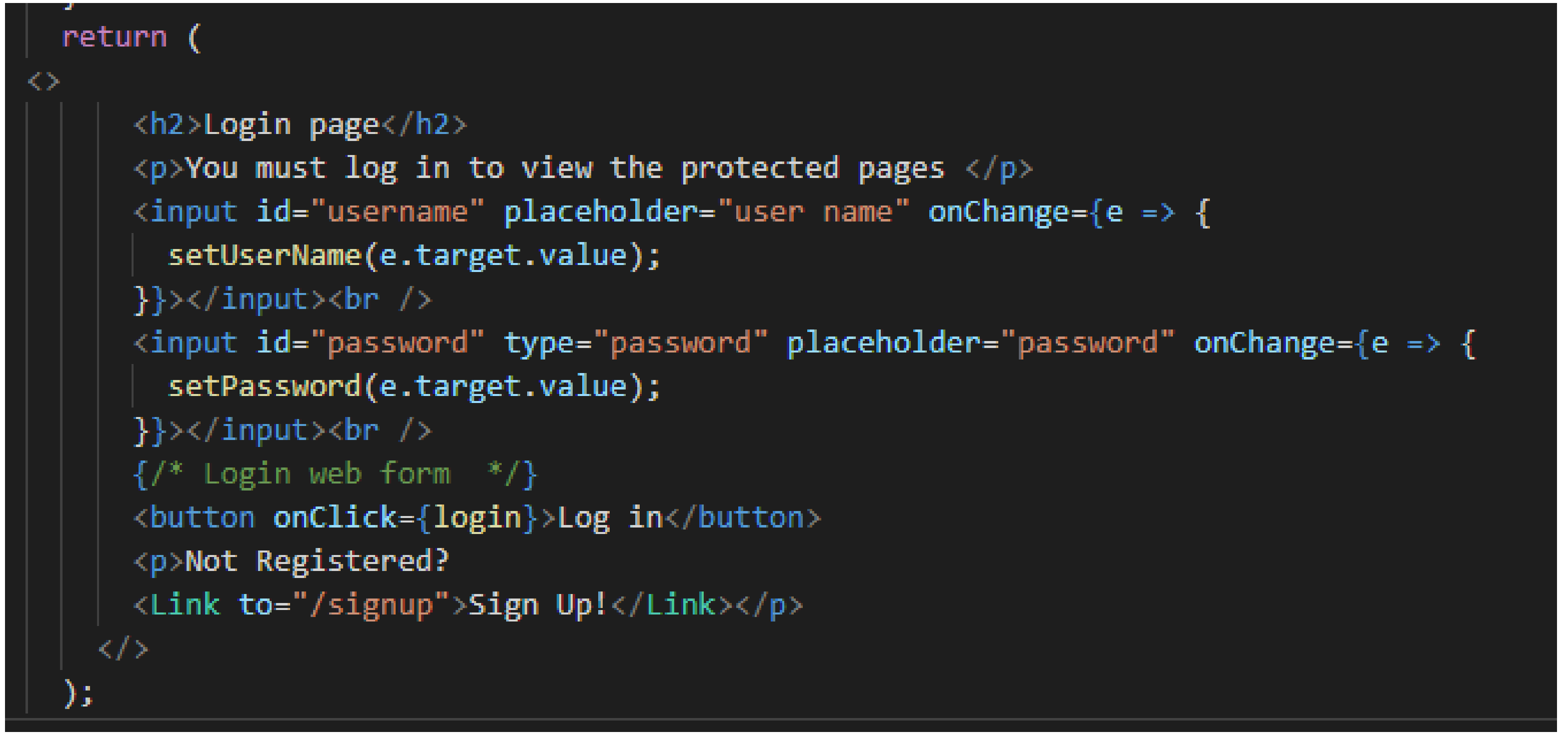
```
import React, { useState, createContext } from "react";
import { login, signup } from "./api/movie-api";

export const AuthContext = createContext(null);

const AuthContextProvider = (props) => {
  const existingToken = localStorage.getItem("token");
  const [isAuthenticated, setIsAuthenticated] = useState(false);
  const [authToken, setAuthToken] = useState(existingToken);
  const [userName, setUserName] = useState("");

  const setToken = (data) => {
    localStorage.setItem("token", data);
    setAuthToken(data);
  }

  const authenticate = async  (username, password) => {
    const result = await login(username, password);
    if (result.token) {
      setToken(result.token)
      setIsAuthenticated(true);
      setUserName(username);
    }
  };
```



```
import React, { useState, createContext, useEffect, useReducer } from "react";
import { getMovies } from "./api/movie-api";

export const MoviesContext = createContext(null);

const reducer = (state, action) => {
  switch (action.type) {
    case "load":
      return { movies: action.payload.result};
    default:
      return state;
  }
};

const MoviesContextProvider = props => {
  const [state, dispatch] = useReducer(reducer, { movies: []});
  const [authenticated, setAuthenticated] = useState(false);

  useEffect(() => {
    getMovies().then(result => {
      console.log(result);
      dispatch({ type: "load", payload: {result}});
    });
  },[]);
```

Update pages to use Contexts

- Login Page:



```
  }
  return (
<>
      <h2>Login page</h2>
      <p>You must log in to view the protected pages </p>
      <input id="username" placeholder="user name" onChange={e => {
        setUserName(e.target.value);
      }}></input><br />
      <input id="password" type="password" placeholder="password" onChange={e => {
        setPassword(e.target.value);
      }}></input><br />
      {/* Login web form  */}
      <button onClick={login}>Log in</button>
      <p>Not Registered?
      <Link to="/signup">Sign Up!</Link></p>
    </>
  );
```