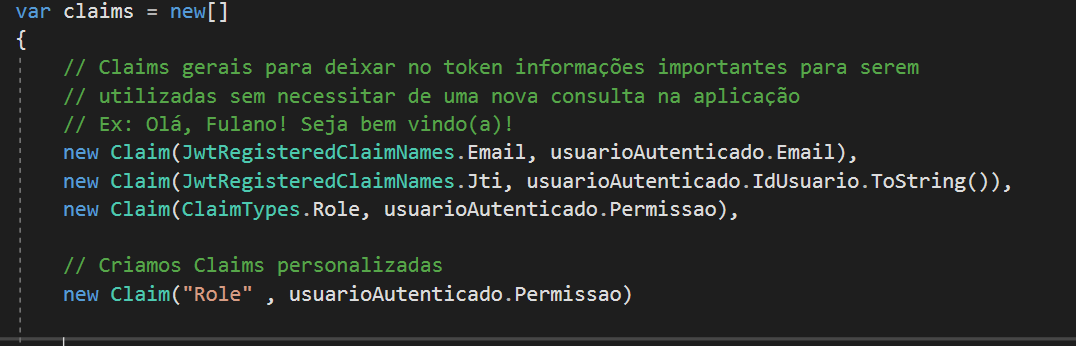
Na API, ContaControllers.cs adicione uma Claim personalizada para facilitar no código front-end:



No projeto react, na página SRC crie uma pasta services e dentro desta para um arquivo auth.tsx:

// Métodos de descriptografia dos tokens

// Verfica se usuário está autenticado ou não

// e retorna um booleano

export const usuarioAutenticado = () => localStorage.getItem("token-filmes") !== null;

// pega o token do localStorage e vai descriptografar

// retorna todos o objeto do token (jwt.io)

//define a constante parseJWT que retorna o payload do usurio convertido em JSON

export const parseJwt = () => {

var token = localStorage.getItem("token-filmes");

if(token){

//codificação para transferência de conteúdo - tipo de criptografia do jwt

//a chave [1] faz com que seja armazenado na variável apenas o payload, que é o que interessa agora

//linha do split, basicamente tá pegando todo o token e cortando em partes separadas pelo ponto, aí separa o token em 3: header, payload e signature

var base64Url = token.split('.')[1];

// EXPERESSÕES REGULARES - https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Regular\_Expressions

//replace substitui uma sequência de caracteres por outra

var base64 = base64Url.replace(/-/g, '+').replace(/\_/g, '/');

//A função window.atob(base64) ou WindowBase64.atob()

//decodifica uma string de dados que foi codificada através da codificação base-64

//decodifica a base64 para string, através do método atob

//e converte a string para JSON

return JSON.parse(window.atob(base64));

}

}

No componente Header adicione mais uma condicional para criação de mais um menu para o usuário Comum:

import React, { useEffect } from 'react';

import logo from '../../assets/images/logo.png';

import './styles.css';

//Importa o useHistory do react-router-dom

import { Link, useHistory } from 'react-router-dom';

**//Importamos a verificação do Token**

**import {parseJwt} from '../../services/auth';**

interface HeaderProps {

description: string;

}

const Header: React.FC<HeaderProps> = (props) => {

//variável que chama o componente History

let history = useHistory();

//criando a função logout, removendo o token do localStorage e chamando a página do usuário deslogado

const logout = () => {

localStorage.removeItem('token-filmes');

history.push('/');

}

// criando a função menu() com a estrutura condicional que retornará o menu para usuário logado e deslogado

const menu = () => {

// a variável token recebe o conteúdo do local storage

const token = localStorage.getItem('token-filmes');

//se o valor de token estiver indefinido ou nulo, chama o menu deslogado, se não chama o menu para quando o usuário estiver logado

if (token === undefined || token === null) {

// Deslogado

return (

<ul className="menu">

<li><Link to="/" className="link">Home</Link></li>

<li><Link to="/login" className="link">Login</Link></li>

<li><Link to="/cadastro" className="link">Cadastro</Link></li>

</ul>

)

} **else {**

**// Logado como admin**

**if(parseJwt().Role === "Administrador"){**

**// Mostra o menu de administrador**

**return (**

<ul className="menu">

<li><Link to="/" className="link">Home</Link></li>

<li><Link to="/perfil" className="link">Perfil</Link></li>

<li><Link to="/genero" className="link">Genero</Link></li>

<li><Link to="/filme" className="link">Filmes</Link></li>

{/\* Link vazio que ao ser clicado chama a função logout() que remove o token do localStorage \*/}

<li><Link to='' onClick={(event) => {

event.preventDefault();

logout();

}}>Logout</Link></li> <br/>

</ul>

)

}**else{**

**// Logado como usuário comum: mostra o menu padrão**

**return (**

**<ul className="menu">**

**<li><Link to="/" className="link">Home</Link></li>**

**<li><Link to="/filme" className="link">Filmes</Link></li>**

**{/\* //Link vazio que ao ser clicado chama a função logout() que remove o token do localStorage \*/}**

**<li><Link to='' onClick={(event) => {**

**event.preventDefault();**

**logout();**

**}}>Logout</Link></li> <br/>**

**</ul>**

)

}

}

}

return (

<div className="header">

<div className="align">

<nav>

<Link to="/"><img src={logo} alt='logo filmes collection' /></Link>

{/\* // chamando a função menu(), criada acima, para dentro da nav \*/}

{menu()}

</nav>

<h3>{props.description}</h3>

</div>

</div>

);

}

export default Header;

Proteção das Rotas. No Arquivo routers:

import React from 'react';

import { BrowserRouter, Route, **Redirect** } from 'react-router-dom';

import Home from './pages/home';

import Login from './pages/login';

import Cadastro from './pages/cadastro';

import Genero from './pages/genero';

import Filme from './pages/filmes';

import Perfil from './pages/perfil';

import {parseJwt} from './services/auth'

function Routers() {

**const RotaPrivadaComum = ({ Component, ...rest } : any) => (**

**<Route**

**{...rest}**

**render={props =>**

**localStorage.getItem('token-filmes') !== null && parseJwt().Role === 'Comum' ? (**

**// Se sim, renderiza de acordo com a rota solicitada e permitida**

**<Component {...props} />**

**) : (**

**// Se não, redireciona para a página de login**

**<Redirect**

**to={{ pathname: "/login"}}**

**/>**

**)**

**}**

**/>**

**);**

**const RotaPrivadaAdm = ({ Component, ...rest } : any) => (**

**<Route**

**{...rest}**

**render={props =>**

**localStorage.getItem('token-filmes') !== null && parseJwt().Role === 'Administrador' ? (**

**// Se sim, renderiza de acordo com a rota solicitada e permitida**

**<Component {...props} />**

**) : (**

**// Se não, redireciona para a página de login**

**<Redirect**

**to={{ pathname: "/login"}}**

**/>**

**)**

**}**

**/>**

);

return (

<BrowserRouter>

<Route path="/" exact component={Home} />

<Route path="/login" component={Login} />

<Route path="/cadastro" component={Cadastro} />

**<RotaPrivadaAdm path="/perfil" Component={Perfil} />**

**<RotaPrivadaAdm path="/genero" Component={Genero} />**

**<RotaPrivadaComum path="/filme" Component={Filme} />**

</BrowserRouter>

);

}

export default Routers;