**ROTEIRO DE AULA – Autenticação JWT em React JS**

**Tema:** Aula 1.1 – Autenticação JWT em React JS

**Data: 21/11/2019**

**Semana:**

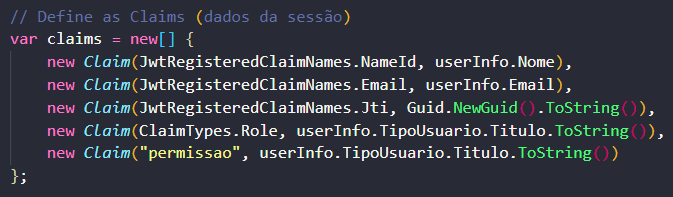
**Tempo estimado:** 4 horas.

**Requisitos e ferramentas para aula:**

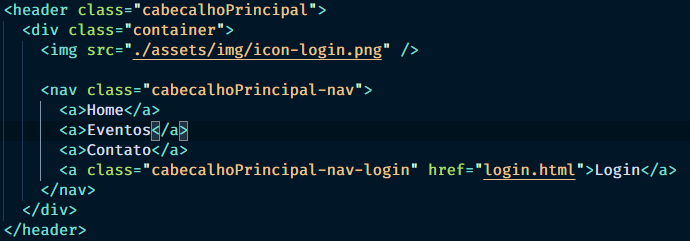
VsCode, arquivos gufos\_base (HTML + CSS), projeto gufos\_ui (React JS).

**Descrever roteiro de aula:**

1. Adicionar uma Claim personalizada no Login Controller da API, para que a propriedade Tipo Usuário seja acessada no Front-End.



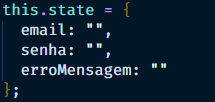
1. Componentizar o cabeçalho da aplicação.

**

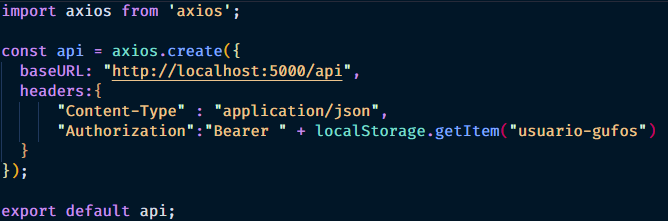
1. Importar os arquivos CSS da aplicação.

**

1. Implementar a tela de Login. **\*\*Se não já tiver sido implementada. Colocar os inputs e form.**
2. Pegar os dados do form.



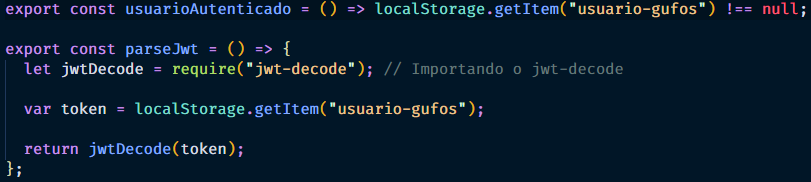
1. Introduzir os alunos ao service de API, dizendo: Agora precisamos fazer a requisição com a API, para isso iremos criar um serviço que facilite a comunicação com ela, passando todas configurações necessárias de uma só vez.
2. **\*\*Opcional: Instalar as dependências do ‘axios’ para utilizá-lo no projeto.**
   1. **‘yarn add –-dev axios’ para instalar via yarn.**
   2. **‘npm install –-save axios’.**
3. **\*\*Opcional: Criar um *service* de *API para fazer as requisições.***

******

1. Fazer a requisição utilizando o service API, caso tenho sido implementado. Somente colocar o token no LocalStorage e enviar para a página de cadastro de Eventos. Não fazer validações ainda.



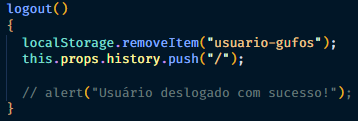
1. Testar o código, mostrando que o token está sendo colocado no LocalStorage do navegador.
2. Validar se o usuário está logado no cabeçalho. Perguntar para os alunos, como podemos saber qual usuário está logado com o token? “Utilizando a Claim ‘permissao’” e para isso vamos criar um service que decodifique o token.
3. Para fazermos a validação será necessário instalar o JWT Decode. Por isso todos devem executar: “npm install –save jwt-decode”



1. Colocar um console.log no Login.js quando este trouxer o token, para verificar se o token foi decodificado.
2. Fazer as validações no Cabecalho.js.



1. Fazer método para deslogar.



1. Demonstrar que o login funciona, mas ainda assim, um usuário comum pode acessar páginas que somente um usuário administrador poderia, por isso devemos adicionar a validação no Routes.js.



**Descrever atividades:**

Os alunos deverão acompanhar a exemplificação tirando dúvidas quando for necessário.

**Tempo atividades: 4hrs**

**Descrever dinâmica:**

**Tempo de dinâmica:**

**Descrever atividade extra:**