1. Para saber se o token esta indo na requisição é só abrir o INSPECIONAR no google chrome, ir na requisição que deseja, clicar nela, ir em HEADERS e ver em AUTHORIZATION
2. O arquivo api.js é responsável por injetar esse token em todos os serviços
3. import axios from 'axios'
4. const api = axios.create({
5. baseURL: 'http://127.0.0.1:8000/api',
6. })
7. api.interceptors.request.use(async (*config*) => {
8. const token = localStorage.getItem("@token\_petshop")
9. if (token) {
10. *config*.headers['Authorization'] = `Bearer ${token}`;
11. }
12. return *config*;
13. });
14. api.interceptors.response.use(
15. *response* => *response*,
16. *error* => {
17. if (*error*.response.status === 401 && *error*.response.statusText === "Unauthorized") {
18. window.location.href = '/';
19. }
20. });
21. export default api

para chamar o axios agora será dessa forma

import api from "./api";

class RaceService {

    async getAllRaces() {

        const response = await api

        .get('races')

        return response.data

    }

}

export default **new** RaceService();

1. Injetando no código de HTML o token de acesso ao fazer login
2. .then((*data*) => {
3. api.defaults.headers.common['Authorization'] = `Bearer ${*data*.data.token}`
4. É necessário adicionar um interceptador, pois se a tela for recarregada é necessário fazer o login de novo, então se tiver um token salvo no local storage, o sistema vai e pegar aquele token na hora da tela atualizar e injeta de novo na pagina não precisando fazer login
5. api.interceptors.request.use(async (*config*) => {
6. const token = localStorage.getItem("@token\_petshop")
7. if (token) {
8. *config*.headers['Authorization'] = `Bearer ${token}`;
9. }
10. return *config*;
11. });
12. Caso o token expire ou a resposta dê não autorizado, o interceptador redireciona para a tela de login novamente

api.interceptors.response.use(

*response* => *response*,

*error* => {

        if (*error*.response.status === 401 && *error*.response.statusText === "Unauthorized") {

            window.location.href = '/';

        }

    });

1. É importante pegar no local storage as permissões do usuário e converter para um array de string
2. <script>
3. export default {
4. data(){
5. return {
6. permissions: JSON.parse(localStorage.getItem("@permissions\_petshop"))
7. }
8. }
9. }
10. </script>

E para o menu aparecer para aquele usuário, basta colocar um v-if

      <router-link *to*="/veterinarios" v-if="permissions.includes('get-profissionals')">

        <v-list-item *prepend-icon*="mdi-star" *title*="Profissionais" *value*="starred"> </v-list-item>

      </router-link>

Lembrando que essa configuração se faz presente na tela de LOGIN

.then((*data*) => {

                  api.defaults.headers.common['Authorization'] = `Bearer ${*data*.data.token}`

                  localStorage.setItem("@token\_petshop", *data*.data.token) //armazenando o token

                  localStorage.setItem("@permissions\_petshop", JSON.stringify(*data*.data.permissions))

*this*.$router.push("/home") //redireciona para a tela home

              })