项目前端框架:vue+vuex

需求:给用户配置权限,不同用户权限不同,页面可操作的权限不同

问题:开始用vuex的store存储用户信息,页面刷新之后数据丢失

why:vuex的状态存储是响应式的，当vue组件从store中读取状态的时候，若store中的状态发生变化，那么相应的组件也会得到高效刷新，问题来了，vuex存储的数据只是在页面中，相当于我们定义的全局变量，刷新之后，里面的数据就会恢复到初始化的状态。

解决方法: 浏览器记住用户信息的方法: localStorage,sessionStorage,cookie

/\*这一段是百度的,记住特性,会用就好了\*/

**Cookie**

Cookie 是小甜饼的意思。顾名思义，cookie 确实非常小，它的大小限制为4KB左右。它的主要用途有保存登录信息，比如你登录某个网站市场可以看到“记住密码”，这通常就是通过在 Cookie 中存入一段辨别用户身份的数据来实现的。

**localStorage**

localStorage 是 HTML5 标准中新加入的技术，它并不是什么划时代的新东西。早在 IE 6 时代，就有一个叫 userData 的东西用于本地存储，而当时考虑到浏览器兼容性，更通用的方案是使用 Flash。而如今，localStorage 被大多数浏览器所支持，如果你的网站需要支持 IE6+，那以 userData 作为你的 polyfill 的方案是种不错的选择。

**sessionStorage**

sessionStorage 与 localStorage 的接口类似，但保存数据的生命周期与 localStorage 不同。做过后端开发的同学应该知道 Session 这个词的意思，直译过来是“会话”。而 sessionStorage 是一个前端的概念，它只是可以将一部分数据在当前会话中保存下来，刷新页面数据依旧存在。但当页面关闭后，sessionStorage 中的数据就会被清空。



/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

程序中实现:

因为之前项目有用过cookie,所以我本次采用的是cookie去实现

1.下面这些方法是网上百度的 存,取,删cookie

function setCookie(name, value) {

var Days = 180;

var exp = new Date();

exp.setTime(exp.getTime() + Days \* 24 \* 60 \* 60 \* 1000);

document.cookie = name + "=" + escape(value) + ";expires=" + exp.toGMTString();

}

function getCookie(name) {

var arr, reg = new RegExp("(^| )" + name + "=([^;]\*)(;|$)");

if ( arr = document.cookie.match(reg)){

return unescape(arr[2]);

}else{

return null;

}

}

function delCookie(name) {

var exp = new Date();

exp.setTime(exp.getTime() - 1);

var cval = getCookie(name);

if (cval != null){

document.cookie = name + "=" + cval + ";expires=" + exp.toGMTString();

}

}

export {

setCookie,

getCookie,

delCookie

};

2.页面中引用

1> cookie.setCookie("userInfo", res.data.userInfo.role);  //接口返回的用户信息存到cookie

2> cookie.getCookie("userInfo")  //从cookie中获取用户信息

用了cookie解决了最开始提出的问题