3、Jeecg Boot 前端项目对接CAS步骤

> Jeecg Boot前端加入对应代码，支持单点登录（这块已经做了封装，集成非常简单）
>
> 思路： 前端vue层面判断登录状态，未登录直接在vue层面跳转到CAS服务器，CAS登录成功调回vue页面带着参数ticket，vue再用ticket进行模拟登录。

1、从网盘下载代码，拷贝到前端工程目录下

```
代码下载地址：
链接：https://pan.baidu.com/s/1uylDHry0WWgkEvvSqD8ITA 提取码：6iya
```

2、下载代码解压后，把sso目录下代码拷贝到前端工程/src目录下

3、index.html页面增加CAS服务地址配置

该地址与第二步与后端对接时配置的地址一致

```
  <script>
    window._CONFIG = {};
    window._CONFIG['casPrefixUrl'] = 'https://cas.test.com:8443/cas';
  </script>
```

4、 src/store/modules/user.js文件中增加验证登录方法

（1）js引入

```
import { getAction } from '@/api/manage'
```

(2)actions中增加如下方法

```
// CAS验证登录
    ValidateLogin({ commit }, userInfo) {
      return new Promise((resolve, reject) => {
        getAction("/cas/client/validateLogin",userInfo).then(response => {
          console.log("----cas 登录--------",response);
          if(response.success){
            const result = response.result
            const userInfo = result.userInfo
            Vue.ls.set(ACCESS_TOKEN, result.token, 7 * 24 * 60 * 60 * 1000)
            Vue.ls.set(USER_NAME, userInfo.username, 7 * 24 * 60 * 60 * 1000)
            Vue.ls.set(USER_INFO, userInfo, 7 * 24 * 60 * 60 * 1000)
            commit('SET_TOKEN', result.token)
            commit('SET_INFO', userInfo)
            commit('SET_NAME', { username: userInfo.username,realname: userInfo.realname, welcome: wel
            commit('SET_AVATAR', userInfo.avatar)
            resolve(response)
          }else{
            resolve(response)
          }
        }).catch(error => {
          reject(error)
        })
      })
    },
```

（3）登出方法改造

```
// 登出
    Logout({ commit, state }) {
      return new Promise((resolve) => {
        let logoutToken = state.token;
        commit('SET_TOKEN', '')
        commit('SET_PERMISSIONLIST', [])
        Vue.ls.remove(ACCESS_TOKEN)
        //console.log('logoutToken: '+ logoutToken)
        logout(logoutToken).then(() => {
          resolve()
        }).catch(() => {
          resolve()
        })
      })
    },
```

改造为：

```
// 登出
    Logout({ commit, state }) {
      return new Promise((resolve) => {
        let logoutToken = state.token;
        commit('SET_TOKEN', '')
        commit('SET_PERMISSIONLIST', [])
        Vue.ls.remove(ACCESS_TOKEN)
        //console.log('logoutToken: '+ logoutToken)
        logout(logoutToken).then(() => {
          var sevice = "http://"+window.location.host+"/";
          var serviceUrl = encodeURIComponent(sevice);
          window.location.href = window._CONFIG['casPrefixUrl']+"/logout?service="+serviceUrl;
          //resolve()
        }).catch(() => {
          resolve()
        })
      })
    },
```

5、改造src/main.js代码，增加如下代码

（1） 引入js

```
import SSO from '@/sso/sso.js'
```

（2）改造下面代码

```
new Vue({
    router,
    store,
    mounted () {
      store.commit('SET_SIDEBAR_TYPE', Vue.ls.get(SIDEBAR_TYPE, true))
      store.commit('TOGGLE_THEME', Vue.ls.get(DEFAULT_THEME, config.navTheme))
      store.commit('TOGGLE_LAYOUT_MODE', Vue.ls.get(DEFAULT_LAYOUT_MODE, config.layout))
      store.commit('TOGGLE_FIXED_HEADER', Vue.ls.get(DEFAULT_FIXED_HEADER, config.fixedHeader))
      store.commit('TOGGLE_FIXED_SIDERBAR', Vue.ls.get(DEFAULT_FIXED_SIDEMENU, config.fixSiderbar))
      store.commit('TOGGLE_CONTENT_WIDTH', Vue.ls.get(DEFAULT_CONTENT_WIDTH_TYPE, config.contentWidth)
      store.commit('TOGGLE_FIXED_HEADER_HIDDEN', Vue.ls.get(DEFAULT_FIXED_HEADER_HIDDEN, config.autoHi
      store.commit('TOGGLE_WEAK', Vue.ls.get(DEFAULT_COLOR_WEAK, config.colorWeak))
      store.commit('TOGGLE_COLOR', Vue.ls.get(DEFAULT_COLOR, config.primaryColor))
      store.commit('SET_TOKEN', Vue.ls.get(ACCESS_TOKEN))
      store.commit('SET_MULTI_PAGE',Vue.ls.get(DEFAULT_MULTI_PAGE,true))
    },
    render: h => h(App)
  }).$mount('#app')
```

改造为：

```
SSO.init(() => {
  main();
```

```
});
function main() {
  new Vue({
    router,
    store,
    mounted () {
      store.commit('SET_SIDEBAR_TYPE', Vue.ls.get(SIDEBAR_TYPE, true))
      store.commit('TOGGLE_THEME', Vue.ls.get(DEFAULT_THEME, config.navTheme))
      store.commit('TOGGLE_LAYOUT_MODE', Vue.ls.get(DEFAULT_LAYOUT_MODE, config.layout))
      store.commit('TOGGLE_FIXED_HEADER', Vue.ls.get(DEFAULT_FIXED_HEADER, config.fixedHeader))
      store.commit('TOGGLE_FIXED_SIDERBAR', Vue.ls.get(DEFAULT_FIXED_SIDEMENU, config.fixSiderbar))
      store.commit('TOGGLE_CONTENT_WIDTH', Vue.ls.get(DEFAULT_CONTENT_WIDTH_TYPE, config.contentWidth)
      store.commit('TOGGLE_FIXED_HEADER_HIDDEN', Vue.ls.get(DEFAULT_FIXED_HEADER_HIDDEN, config.autoHi
      store.commit('TOGGLE_WEAK', Vue.ls.get(DEFAULT_COLOR_WEAK, config.colorWeak))
      store.commit('TOGGLE_COLOR', Vue.ls.get(DEFAULT_COLOR, config.primaryColor))
      store.commit('SET_TOKEN', Vue.ls.get(ACCESS_TOKEN))
      store.commit('SET_MULTI_PAGE',Vue.ls.get(DEFAULT_MULTI_PAGE,true))
    },
    render: h => h(App)
  }).$mount('#app')
}
```

6、组织机构登录切换 src/components/tools/UserMenu.vue 增加 mounted()

```
mounted(){
      let depart = this.userInfo().orgCode;
      if(!depart){
        this.updateCurrentDepart();
      }
  },
```

以上步骤完成后，CAS对接完成，启动前端项目 访问http://localhost:3000 即可

上一篇：2.JeecgBoot后端对接CAS步骤

下一篇：工作流开发