

# 使用说明

## 1. 引入(导入到本地)

在main.js中注入(需要导入三个js文件 都放在:github.com/micro-plat/sso/sso-extend中)

```
import {ssoHttpConfig} from './xx/sso.js';
var config = process.env.service;
var ssocfg = ssoHttpConfig(config.apiHost, "localStorage", config.ssoWebHost, config.Ident);

//将sso和http都挂在vue对象中，方便使用
Vue.prototype.$sso = ssocfg.sso;
Vue.prototype.$http = ssocfg.http;
```

参数	类型	说明
apiHost	string	子系统apihost
storagePlace	string	jwt存储方式 [localStorage, sessionStorage],cookie请传空
ssoWebHost	string	sso web系统的host
ident	string	子系统ident

说明：由于将原来的get,post进行了包装,因此要将main.js文件中原来http的方法及引用去掉【切记】如：Vue.prototype.\$post等

## 2. 所有http的交互

在vue页面中使用时有点变化，要调整成这样，原来都是 \$post,\$patch,\$fetch,\$put,\$del(这些都要替换)

```
this.$http.get("/sso/role", {pageIndex:1, pageSize:10})

//其他方法类似
this.$http.post();
this.$http.put();
this.$http.del();
```

## 3. sso web对接相关

### 3.1 增加一个回调页面

登录后sso要回调子系统,同时也要验证刚登录用户的合法性

```
<script>
  export default {
    data () {
      return {
      }
    },
    mounted(){
      this.validSsoLogin();
    },
    methods:{
      validSsoLogin(){
        this.$http.post("xxx/xxx/xxx",{code: this.$route.query.code})
          .then(res =>{
            this.$sso.changeRouteAfterLogin(this.$router, res.user_name, res.role_name);
          }).catch(err => {
            console.log(err);
          });
      }
    }
  }
</script>
```

说明: "xxx/xxx/xxx"是服务端验证地址,记住服务端要将此路径排除登录验证(zookeeper -> auth文件-> exclude中 一定要加上此路

### 3.2 服务端增加一个上面提到的接口 xxx/xxx/xxx

里面调用 `data, err := model.GetSSOClient(u.c).CheckCodeLogin(code)`  
然后处理自己的业务,同时生成子系统的jwt

### 3.3 处理修改密码和退出

找到要修改密码和退出的地方,一般都在menu.vue文件中

修改密码: `this.$sso.changePwd();`

退出: `this.$sso.signOut();`