

sso api交互

SSO Client 就是在包装相关的api调用(服务端使用)，简化用户调用

1. 使用方法

```
1.1 在init()方法中注入sso client
import "github.com/micro-plat/sso/sso"

ssoCleint, err := sso.New(conf.SsoApiHost, conf.Ident, conf.Secret)
if err != nil {
    return err
}
model.SaveSSOClient(c, ssoCleint) //将sso client 保存起来
```

```
1.2 保存和获取 sso client 实例
//SaveSSOClient 保存sso client
func SaveSSOClient(c component.IContainer, m *sso.Client) {
    c.Set("__SsoClient__", m)
}

//GetSSOClient 获取sso client
func GetSSOClient(c component.IContainer) *sso.Client {
    return c.Get("__SsoClient__").(*sso.Client)
}
```

```
1.3 使用sso client 实例
data, err := model.GetSSOClient(u.c).CheckCodeLogin(code)
```

2. 相关方法的说明(输入输出)

2.1 初始化

```
New(apiHost, ident, secret string)
```

输入参数

参数	类型	说明
apiHost	string	sso api host
ident	string	子系统标识(相当于英文名称)
secret	string	子系统秘钥

返回ssoClient 对象

2.2 用户登录验证以及返回用户信息(跳转登录后)

CheckCodeLogin(code string)

由于现在是跳转登录的方式,因此sso回调子系统地址后，子系统要验证登录的合性

输入参数

参数	类型	说明
code	string	调转登录返回给子系统的code

输出参数

参数	类型	说明
UserID	string	用户标识
UserName	string	用户名称
RoleName	string	角色名称
SystemID	string	系统编号
ExtParams	string	用户扩展参数(一个json对象)

2.3 根据用户名获取用户信息

GetUserInfoByName(userName string)

输入参数

参数	类型	说明
userName	string	用户名称

输出参数

参数	类型	说明
userName	string	用户名称
WxOpID	string	微信openID
ExtParams	string	用户扩展参数(一个json对象)
UserID	string	用户标识

2.4 获取用户在某个子系统下的菜单数据

GetUserMenu(userID int)

输入参数

参数	类型	说明
userID	int	用户标识

输出参数

参数	类型	说明
ID	string	菜单标识
Name	string	菜单名称
Level	string	级次
IsOpen	string	是否展开
Icon	string	图标
SystemID	string	系统标识
Parent	string	父级编号
Path	string	路由地址
Sortrank	string	排序编号
Children	对象数组	子菜单

2.5 获取子系统数据数据

GetSystemInfo()

输入参数(无)

输出参数

参数	类型	说明
ID	string	系统标识
Ident	string	系统ident(英文名称)
Name	string	系统名称
Theme	string	主题样式
IndexUrl	string	sso登录后子系统回调地址
Layout	string	页面布局样式
Logo	string	系统图标地址