

Icap-auth使用指南

背景：

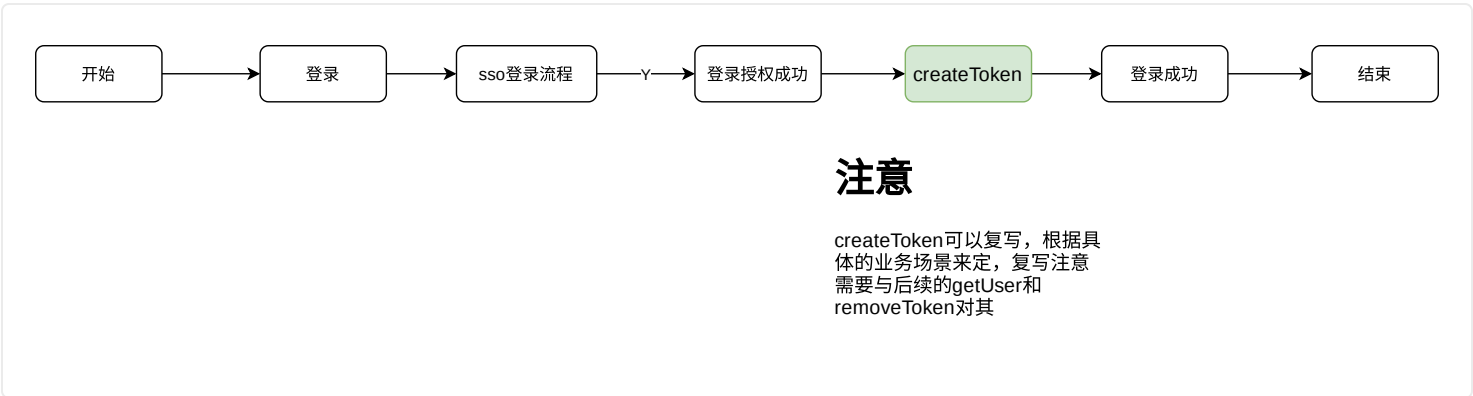
在实际的业务中，B端场景往往需要实现单点登录，那么制品应用现有的登录逻辑便不满足需求了，所以这里在ide3.0版本后制品应用默认引入了Icap-auth依赖库，用来完成登录流程中的token颁发、认证、清除token

实现方式

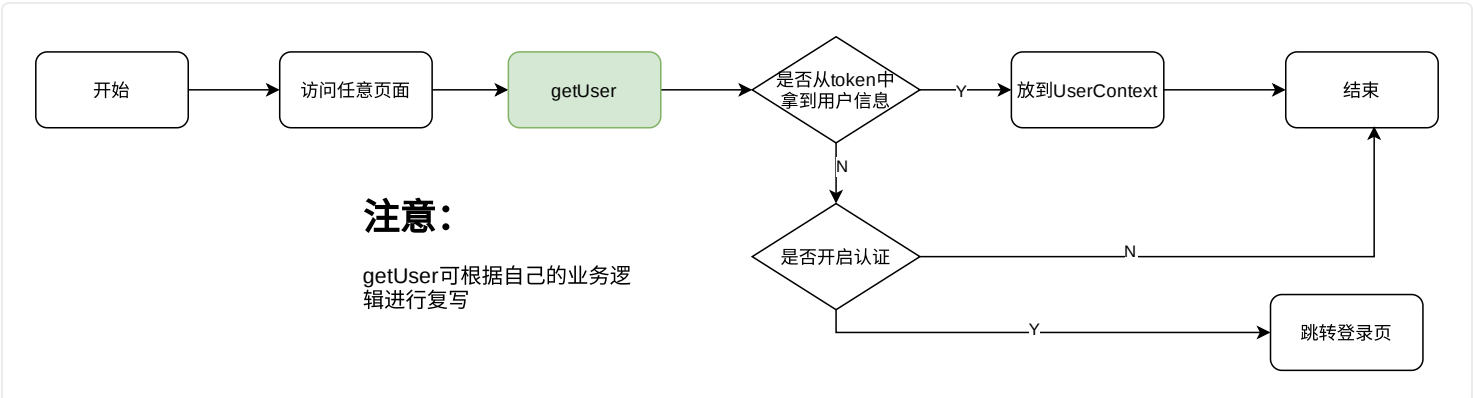
- 制品应用实际上是通过spi的形式引入当前依赖
- 现有的Icap-auth依赖库已经默认实现登录的认证逻辑。开发者只需要在对应地方手动调用createToken、getUser、removeToken即可，而不需要去写里面的任何逻辑
- 如果当前应用的默认认证逻辑满足不了需求，比如当前应用是门户网站，需要管理所有子应用的登录、用户信息。那么这个时候就可以去复写createToken、getUser、removeToken这三个方法里面的逻辑，比createToken就需要去手动创建token然后加到同一父域的cookie当中。

流程图：

登录流程：



认证流程



接口介绍：

getUser：

获取当前的用户信息，拦截器中调用（不需要手动调用，可以重写），每个认证接口都会走当前逻辑，分为以下几个步骤

- 从浏览器中拿到cookie
- 从cookie中拿到token
- 解析token得到用户信息

出参：

参数名	类型	描述	是否必传
userId	String	用户id	是
userName	String	用户名	是
extensionInfos	Map<String,String>	扩展信息	否

CreateToken

sso登录拿到用户信息后调用该逻辑，用于颁发token，分为以下几个步骤

- 拿到用户信息后，颁发token
- 将token塞到cookie中，便于认证

入参：

参数名	类型	描述	是否必传
userId	String	用户id	是
userName	String	用户名	是
extensionInfos	Map<String,String>	扩展信息	否

出参：true/false

removeToken

删除cookie，制品应用退出登录组件默认调用该逻辑

出参：true/false

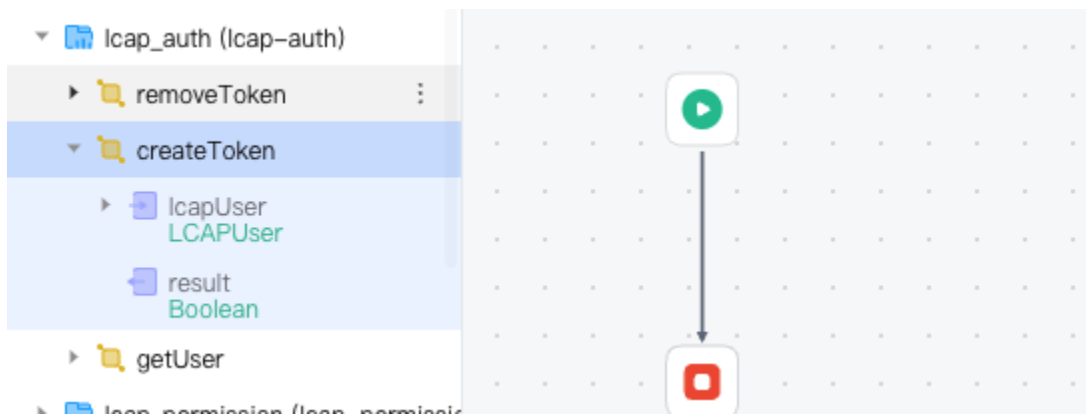
结构体

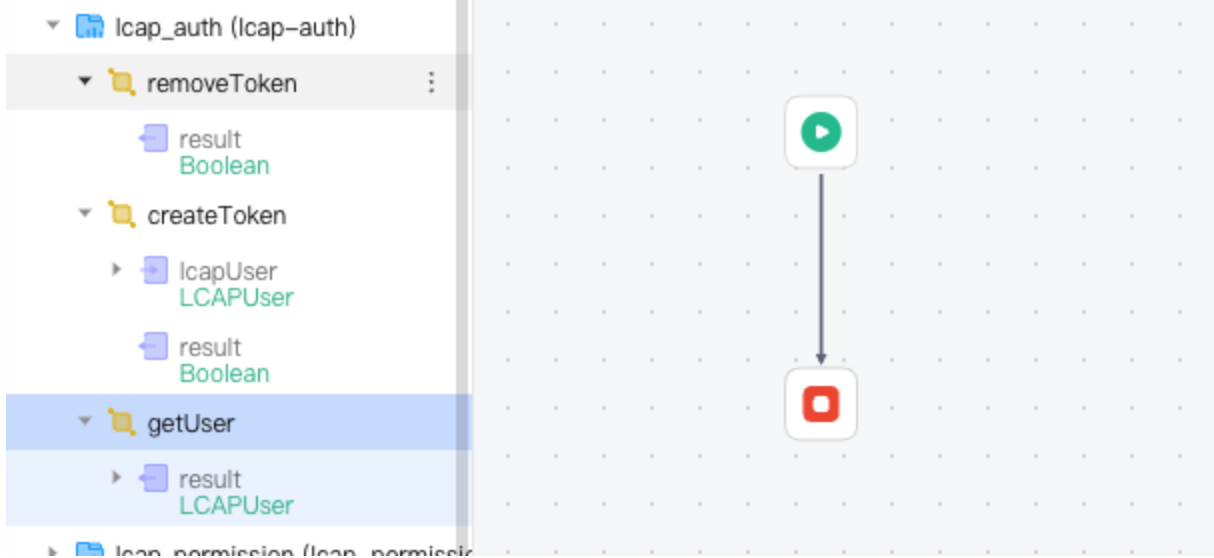
LCAPUser

```
1 public class LCAPUser {
2     @Required
3     public String userId;
4     @Required
5     public String userName;
6     public Map<String,String> extensionInfos;
7 }
```

使用方法

- 当前默认认证逻辑满足需求，不需要重写createToken
 - 未重写createToken、getUser、removeToken

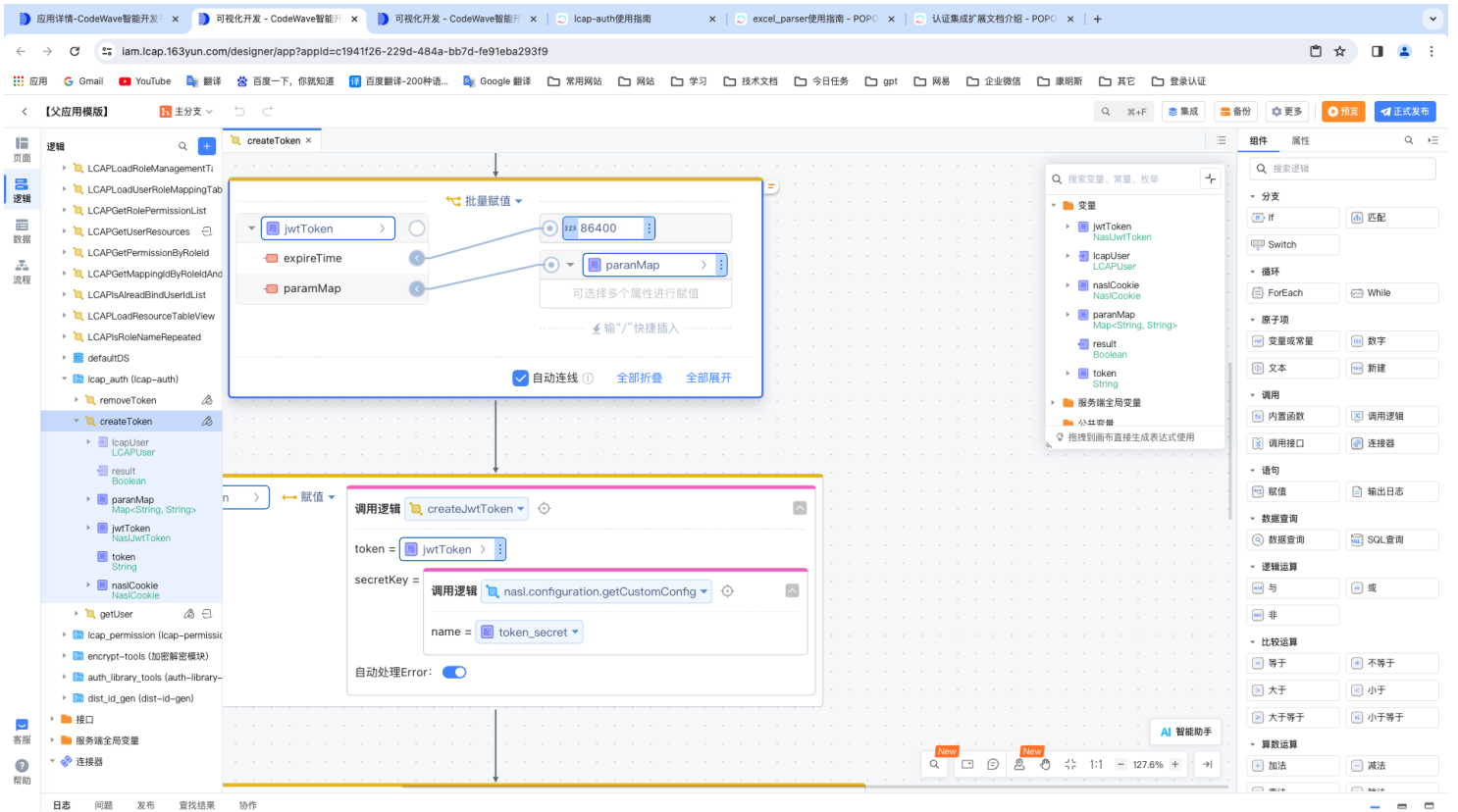
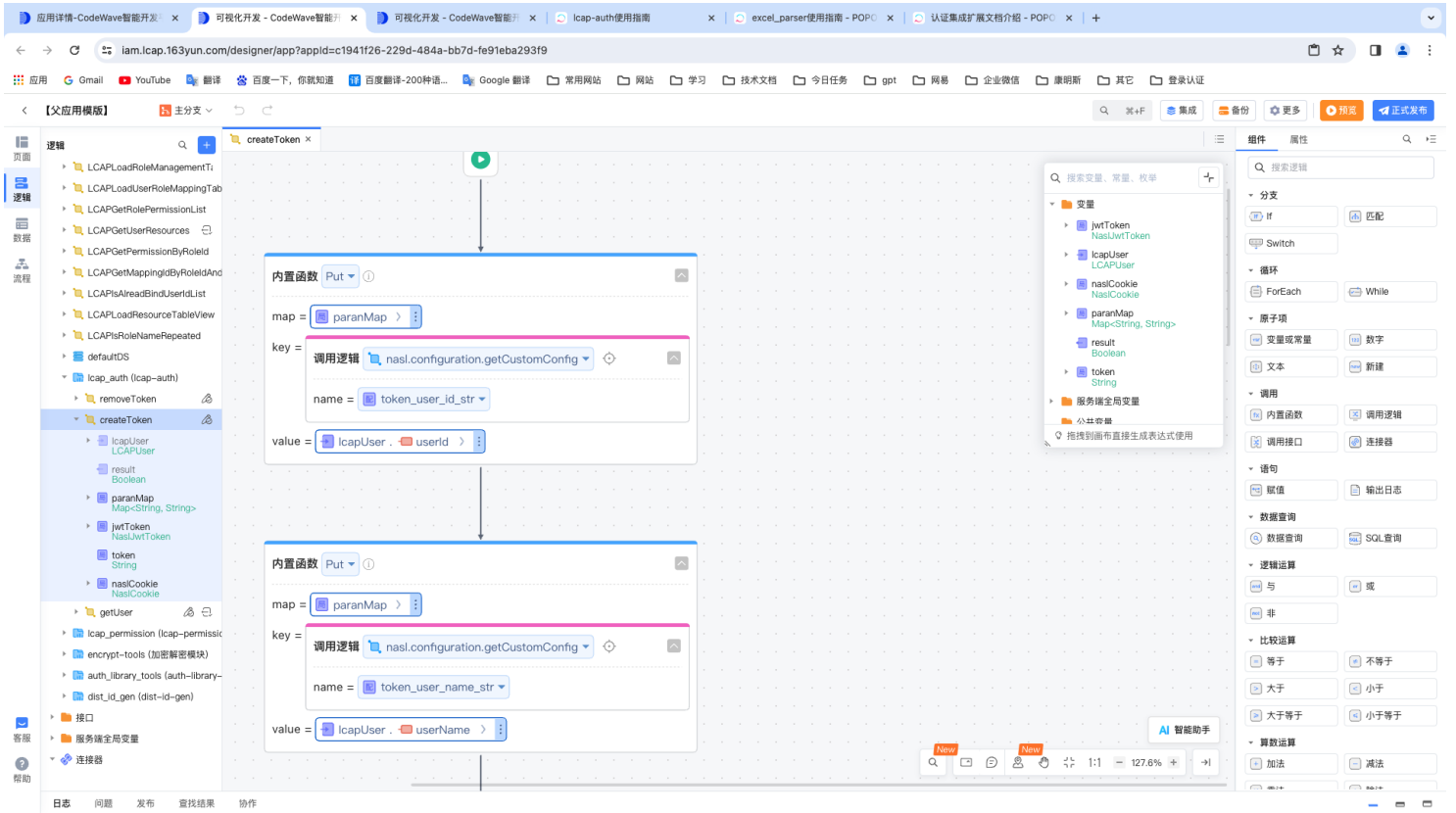


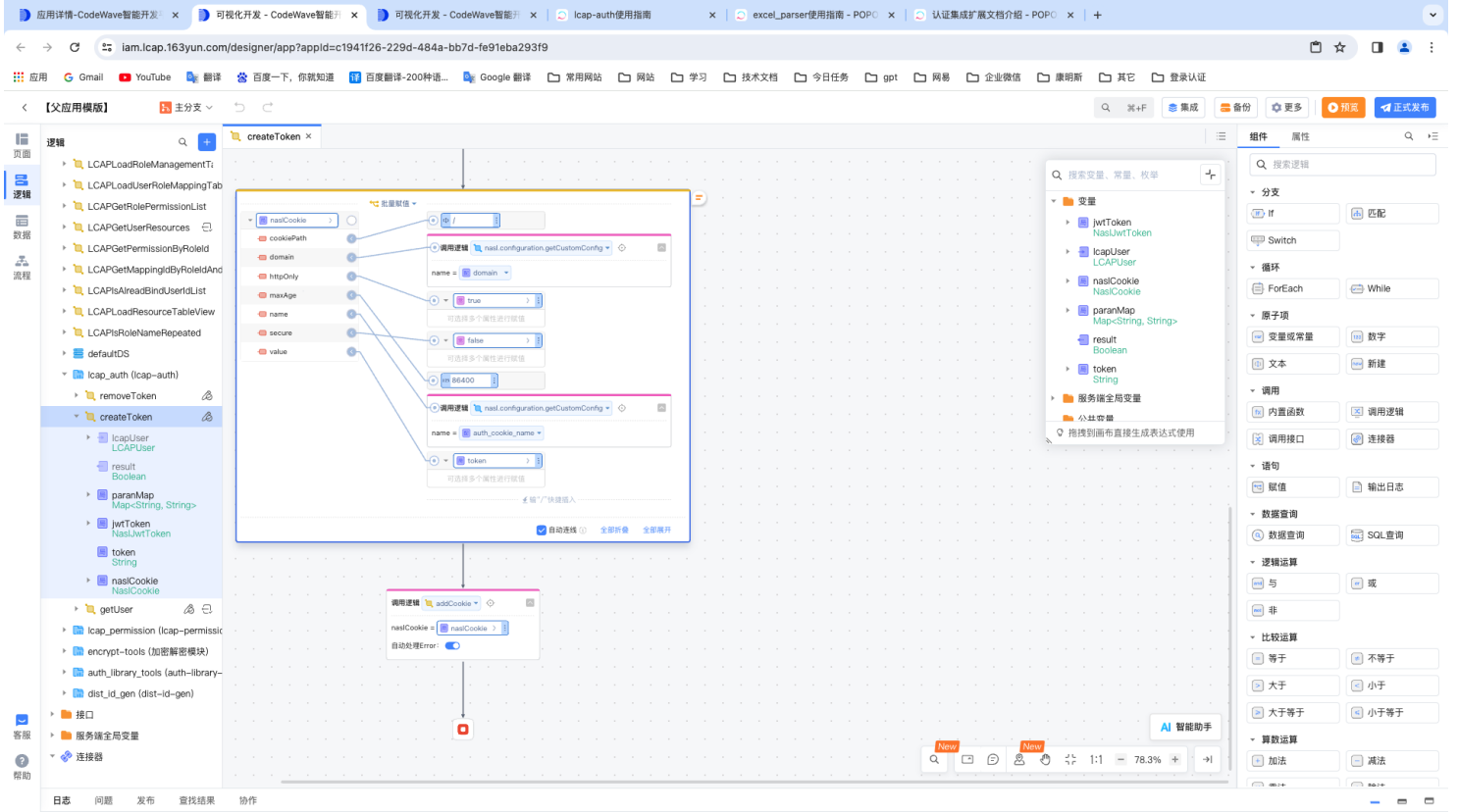


- sso拿到用户后，直接调用createToken，默认逻辑会帮忙创建好token并加到cookie中

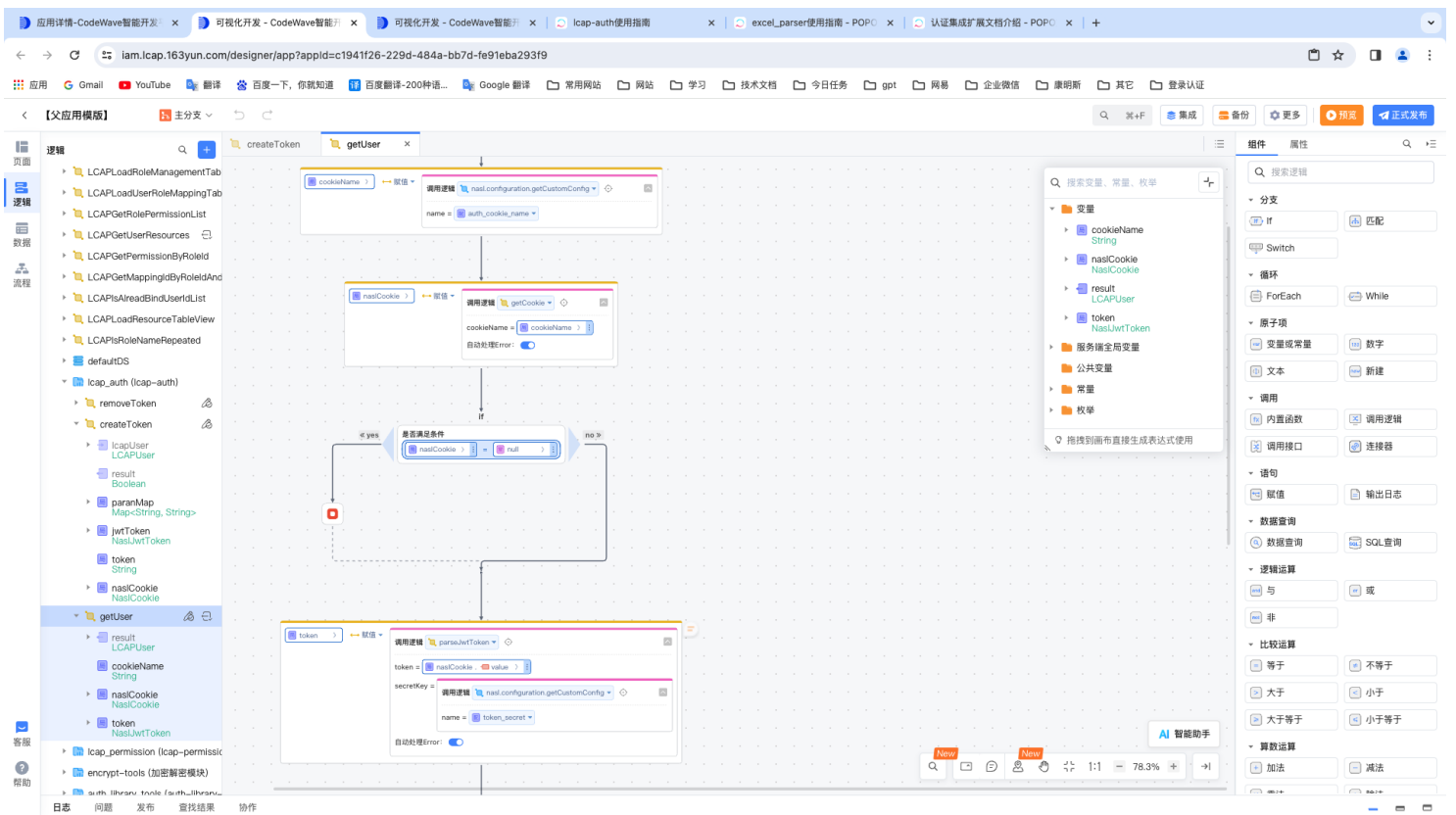


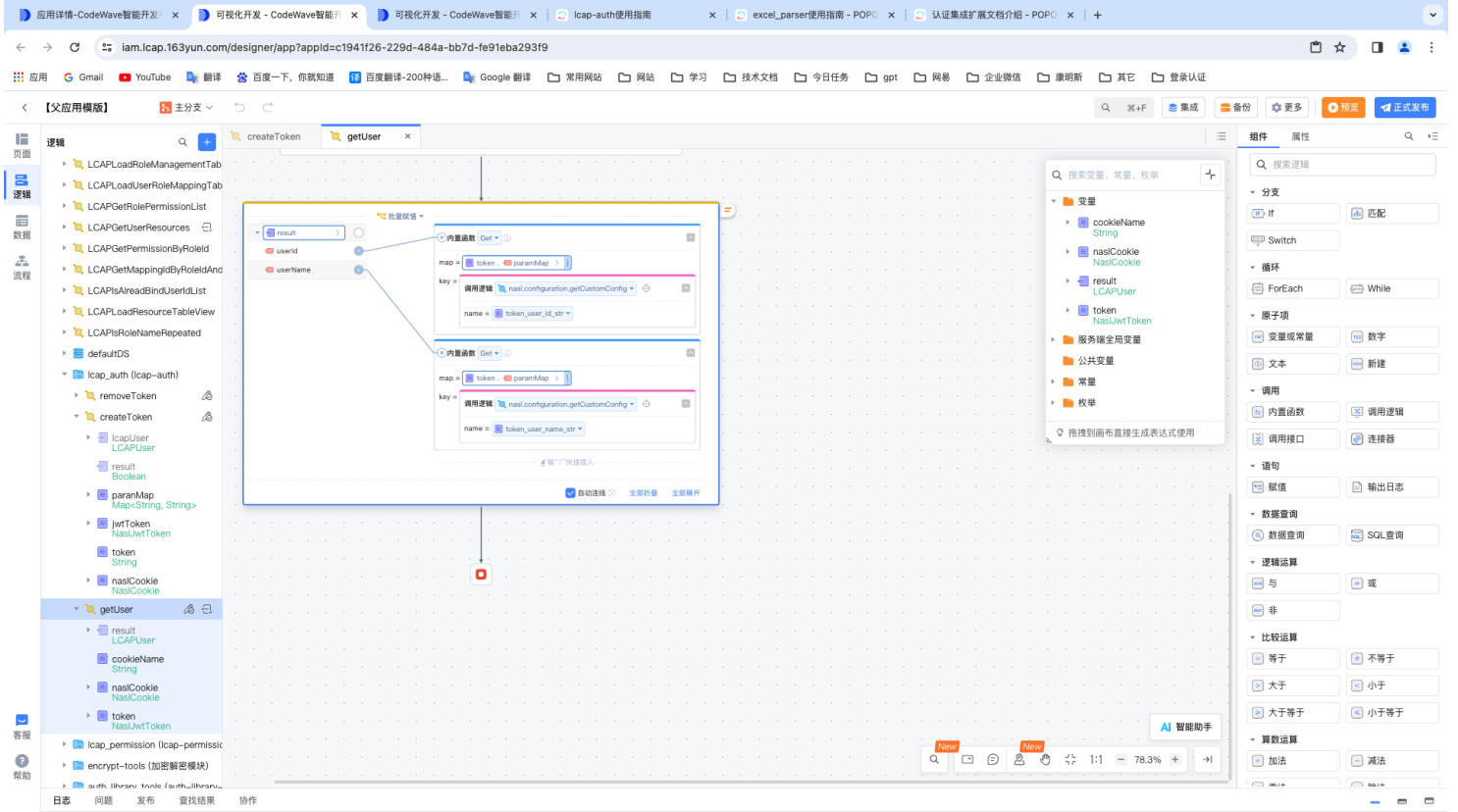
- 制品应用中getUser会在拦截器里面自动调用，不需要开发者手动去调
- 需要重写createToken、getUser、removeToken的场景，以门户网站举例：
 - 重写createToken





。重写getUser





。重写removeToken

