11、cas第三方登录

# 接10代码

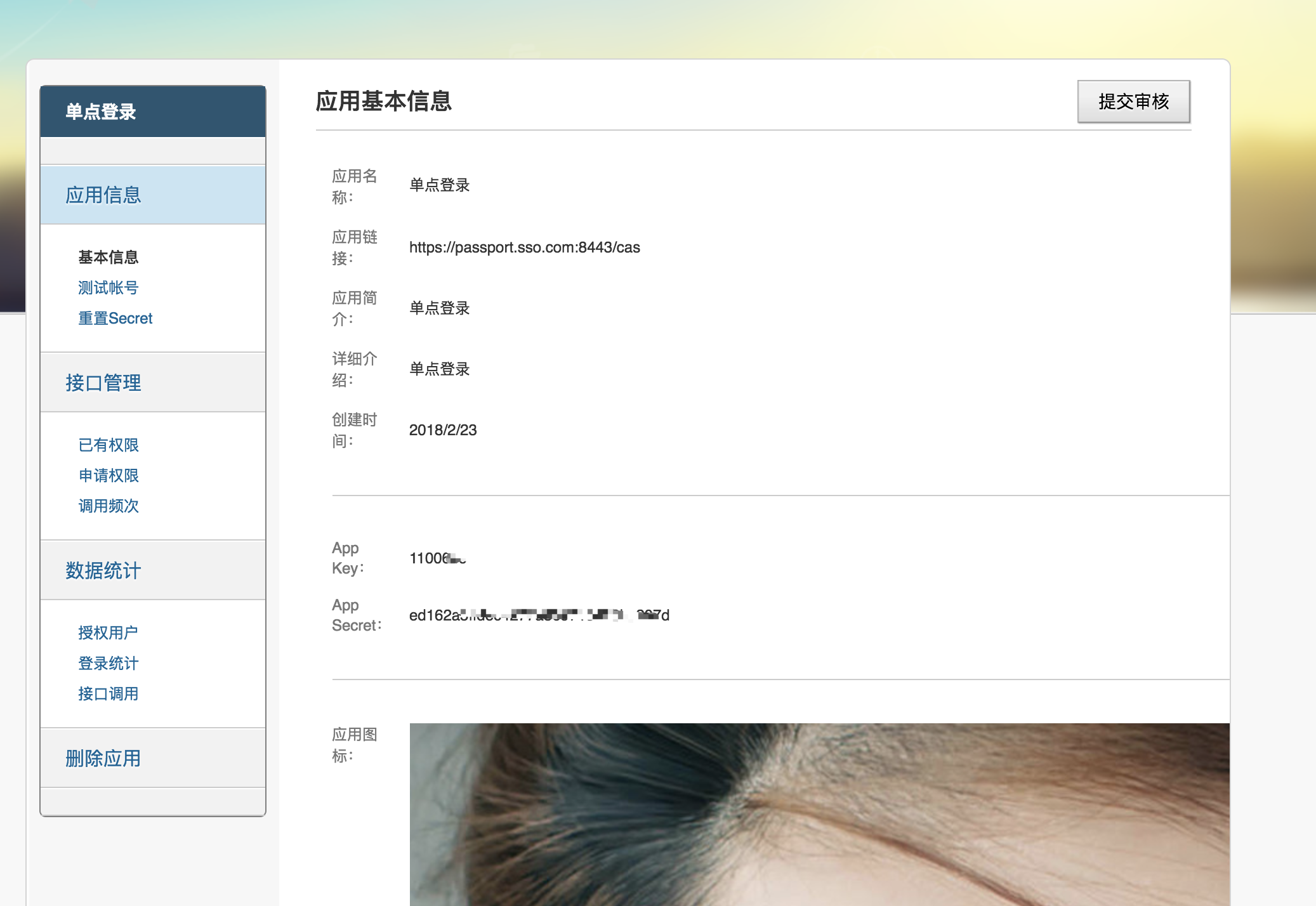
# 注意:第三方登录都是需要第三方平台审核的

# 1、添加依赖

<dependency>  
 <groupId>org.apereo.cas</groupId>  
 <artifactId>cas-server-support-pac4j-webflow</artifactId>  
 <version>${cas.version}</version>  
</dependency>

# 1、csdn

## 1、csdn创建应用 <http://open.csdn.net/apps>



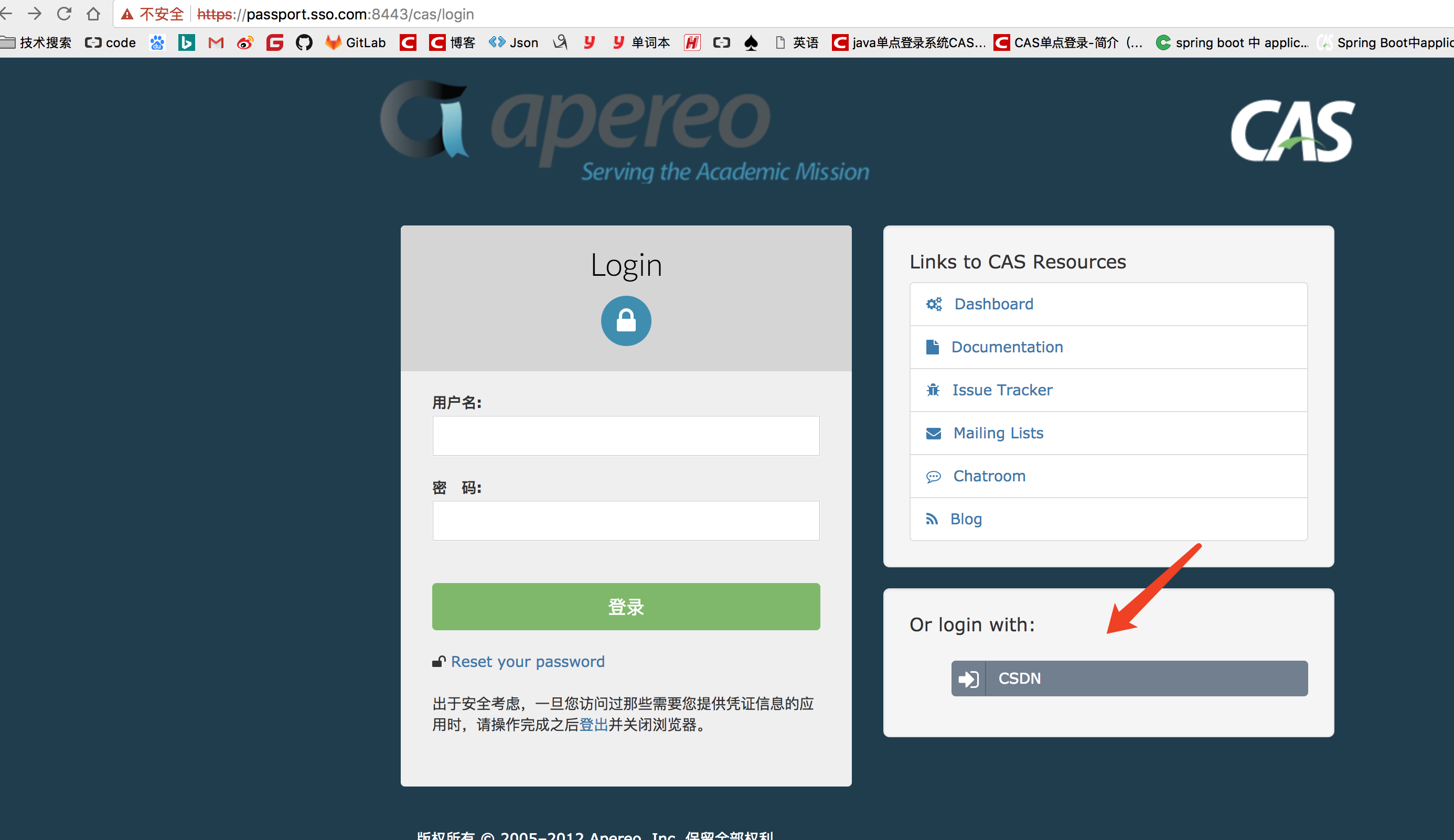
## 2、csdn官方文档

<http://open.csdn.net/wiki/oauth2>

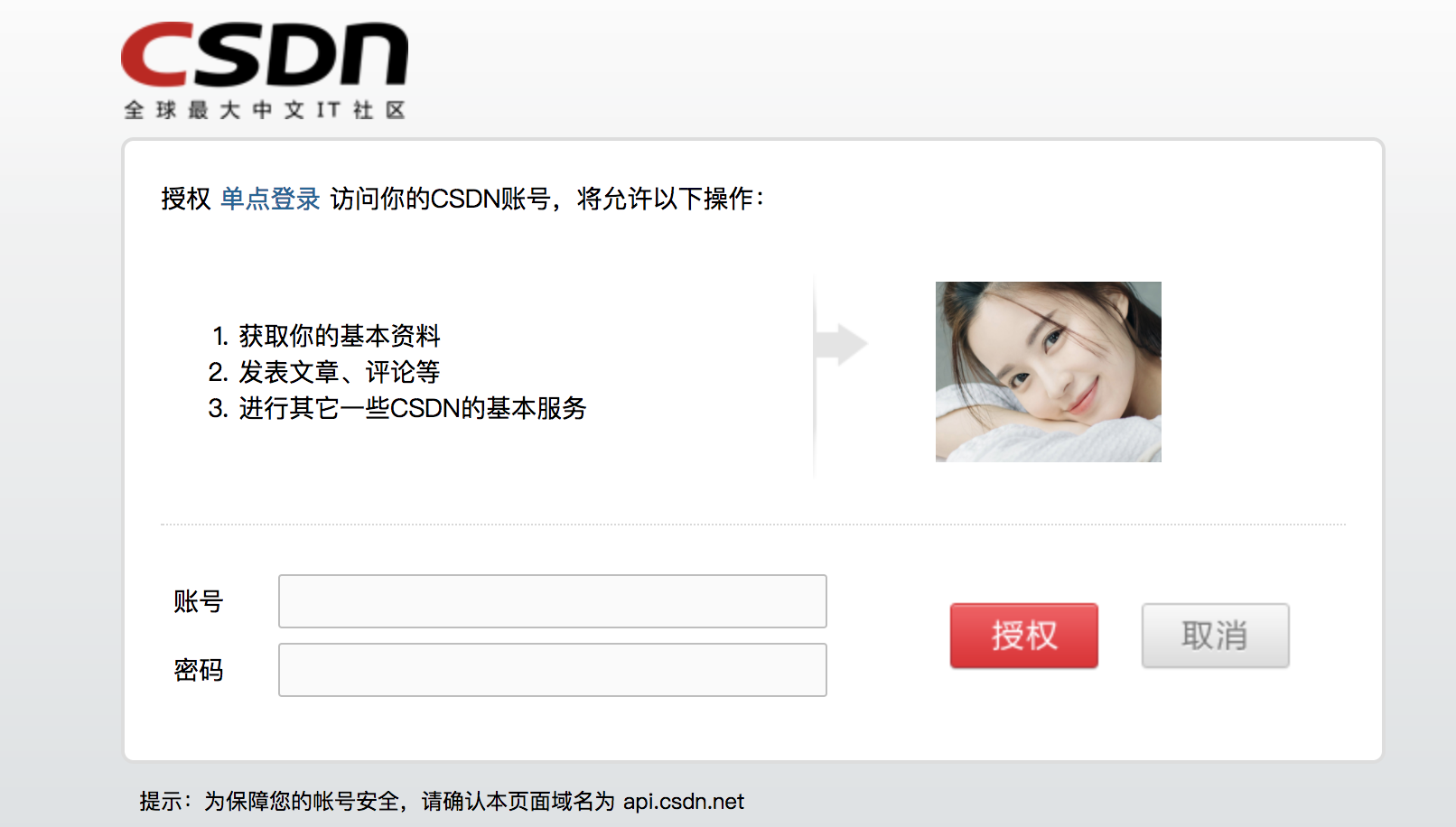
## 3、sso-config 开始配置csdn登录

#CSDN OAuth Login  
cas.authn.pac4j.oauth2[0].id=1100649  
cas.authn.pac4j.oauth2[0].secret=ed162a5ffdec4277a869715428bc207d  
cas.authn.pac4j.oauth2[0].authUrl=http://api.csdn.net/oauth2/authorize  
cas.authn.pac4j.oauth2[0].tokenUrl=http://api.csdn.net/oauth2/access\_token  
cas.authn.pac4j.oauth2[0].profileUrl=http://api.csdn.net/user/getinfo  
cas.authn.pac4j.oauth2[0].profileAttrs.id=username  
cas.authn.pac4j.oauth2[0].clientName=CSDN

## 4、开始测试 启动 sso-config sso-server <https://passport.sso.com:8443/cas/login>



## 5、点击登录 （我这里应用没有通过审核，所以没有成功，）





### 正确情况下



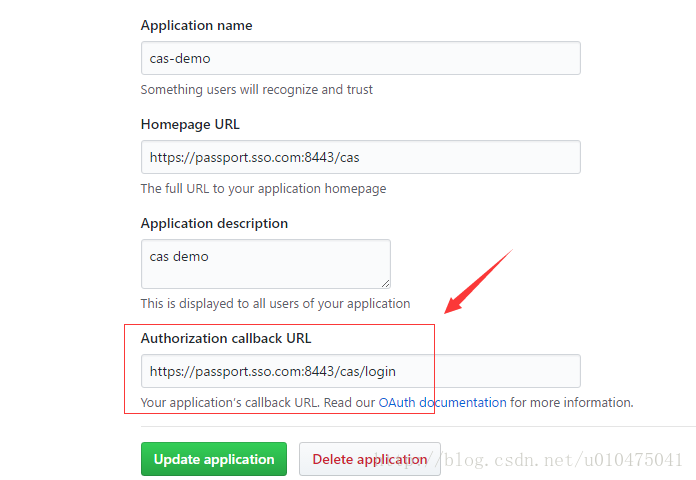
# 2、其他的登录，不详细解释了

## GitHub授权登录

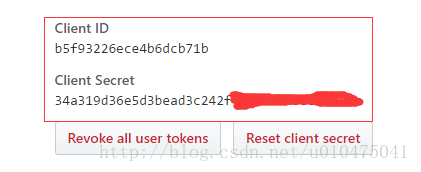
而GitHub的授权登录就更为简单了，举一反三~   
为什么说简单呢，因为cas为我们准备好了接口信息，不用我们去找了，也就是说，到github申请appid以及获取秘钥即可，哈哈哈哈~

### 创建应用

点击[打开创建应用](https://github.com/settings/applications/new)



当然啦，创建成功会给我们分配秘钥的



### SSO配置

这些就不多说啦~

application.properties

#GitHub OAuth Login

cas.authn.pac4j.github.id=b5f93226ece4b6dcb71b

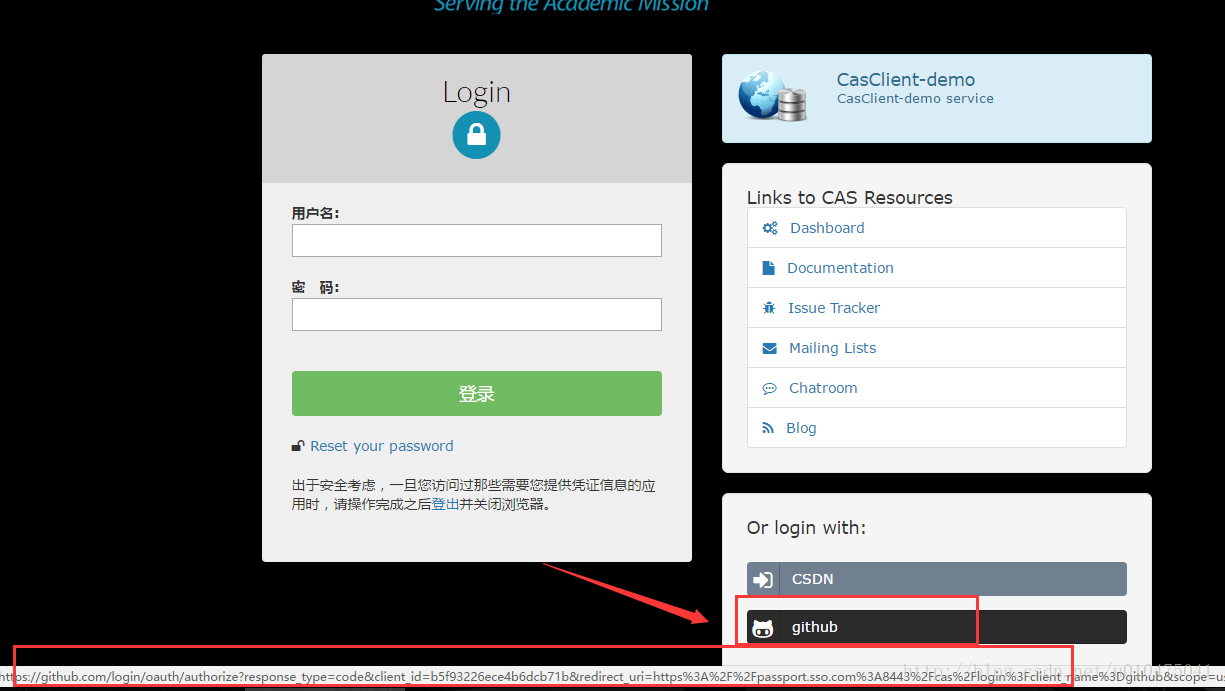
cas.authn.pac4j.github.secret=34a319d36e5d3bead3c242ffb5172d49511b48fe

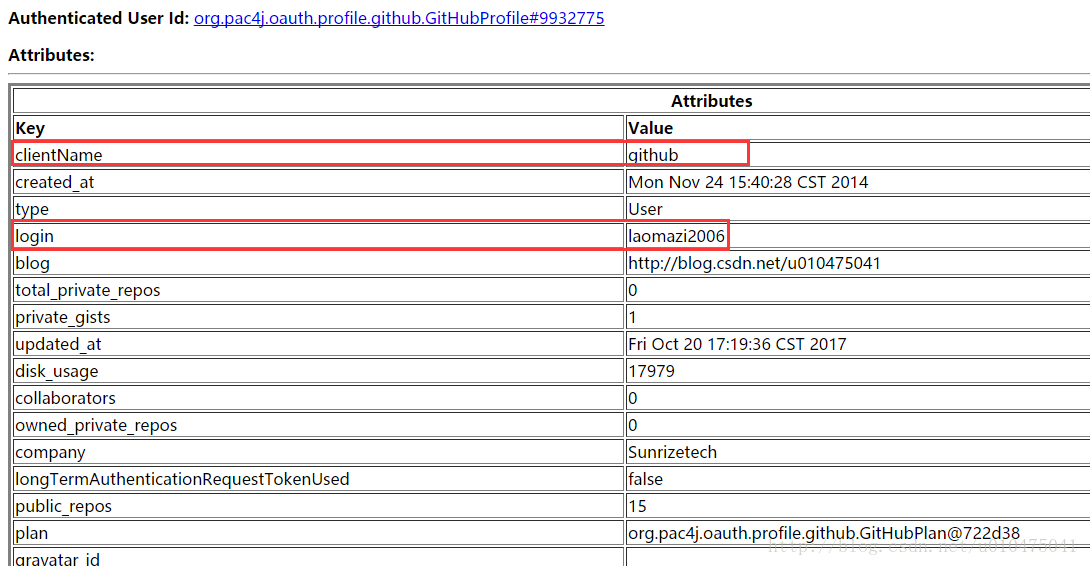
cas.authn.pac4j.github.profileAttrs.id=id

cas.authn.pac4j.github.client-name=github

* 1
* 2
* 3
* 4
* 5

马上重启试试~



再次恭喜你~   


## 微信授权登录

重头戏，这个就稍微跟前面的有点不一样了~

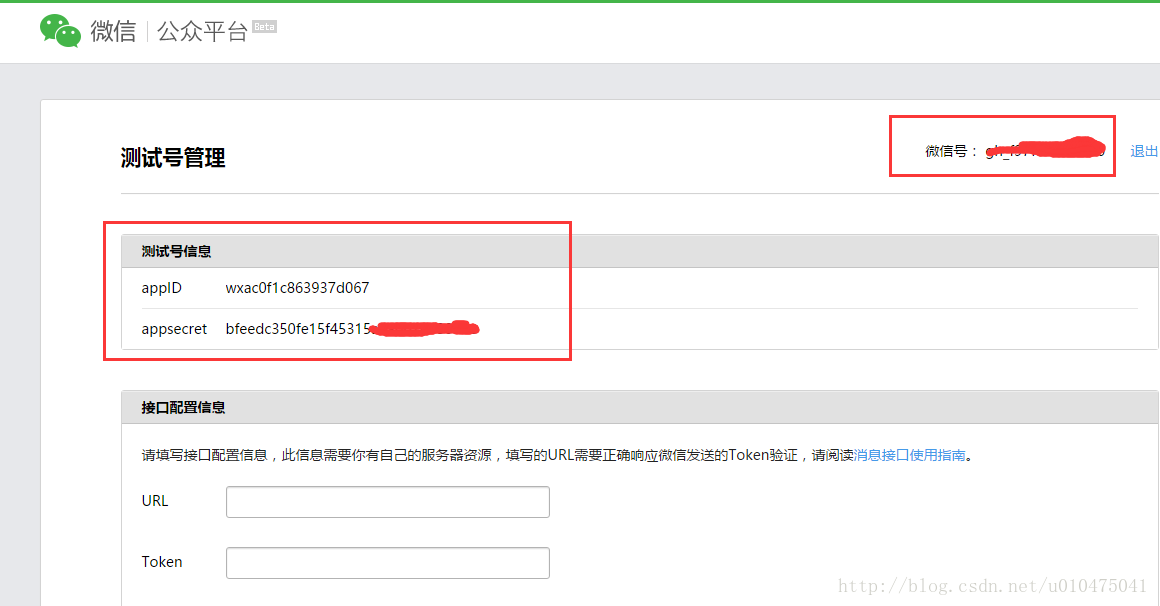
### 创建应用

授权到测试公众号平台[申请测试号](https://mp.weixin.qq.com/debug/cgi-bin/sandbox?t=sandbox/login)，当然有正式的更好。。。



其实我们就是要这个效果。。。

我们登录了，测试平台就默认给我们创建了一个应用~



### 设置回调域名





### 接口信息

我要打开[微信扫描登录文档](https://open.weixin.qq.com/cgi-bin/showdocument?action=dir_list&t=resource/res_list&verify=1&id=open1419316505&token=&lang=zh_CN)

#### 授权接口



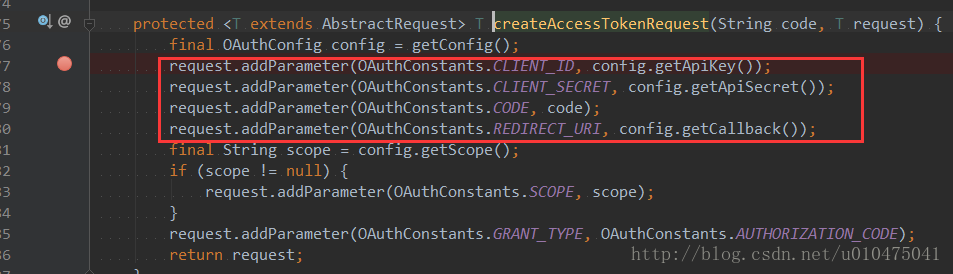
由上图得知：   
1. 授权路径为：https://open.weixin.qq.com/connect/qrconnect   
2. appid参数为：appidcas默认为client\_id，需要做调整   
3. 网页授权登录scope=snsapi\_login需要做调整

#### access\_token接口



由上图得知：

1. access\_token路径为：https://api.weixin.qq.com/sns/oauth2/access\_token
2. appid参数为：appidcas默认为client\_id，需要做调整
3. secret参数为：secretcas默认为client\_secret,需要做调整

经过源码com.github.scribejava.core.oauth.OAuth20Service#createAccessTokenRequest的分析   


很可惜不符合pac4j的默认请求，那么有以下几个解决办法：

1. 添加一个中转程序，把access\_token路径让中转程序接管，中转程序请求到微信，再返回给sso程序即可（推荐）
2. 修改cas以及pac4j源码，添加自己的部分逻辑（不推荐）

#### 用户信息接口



由上图得知：   
1. 用户信息接口为https://api.weixin.qq.com/sns/userinfo

### SSO配置

application.properties

#WeChat OAuth Login

cas.authn.pac4j.oauth2[1].id=wxac0f1c863937d067

#由于微信的不为clinet\_id，为appid

cas.authn.pac4j.oauth2[1].customParams.appid=wxac0f1c863937d067

#微信scope登录为snsapi\_login

cas.authn.pac4j.oauth2[1].customParams.scope=snsapi\_login

cas.authn.pac4j.oauth2[1].secret=bfeedc350fe15f45315a0ab67643085e

cas.authn.pac4j.oauth2[1].authUrl=https://open.weixin.qq.com/connect/qrconnect

cas.authn.pac4j.oauth2[1].tokenUrl=https://api.weixin.qq.com/sns/oauth2/access\_token

cas.authn.pac4j.oauth2[1].profileUrl=https://api.weixin.qq.com/sns/userinfo

cas.authn.pac4j.oauth2[1].clientName=WeChat

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11



哎呀，对不起大家，

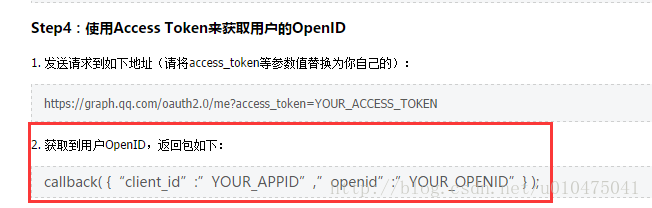
* 获取不到能够成功测试的应用，非常抱歉
* 没有测试到access\_token，非常抱歉
* 没有提取到用户信息，非常抱歉

## QQ授权登录

[申请成为开发者](https://connect.qq.com/manage.html#/)   
[接入文档](http://wiki.connect.qq.com/%E4%BD%BF%E7%94%A8authorization_code%E8%8E%B7%E5%8F%96access_token)

qq登录更奇葩，恐怕在cas中做的话就比较吃力了，除非cas开放了自定义OAuth2的配置定义

奇葩1: 开发者审核需要拍照上传。。。   
奇葩2: 用户信息返回jsonp(解决办法，通过中转程序来做这个用户信息的适配)   
奇葩3: access\_token获取报文不是json



### 创建应用



### SSO配置

#QQ OAuth Login

#QQ OAuth Login

cas.authn.pac4j.oauth2[2].id=101433938

cas.authn.pac4j.oauth2[2].secret=ad4fef7e3fb785a51026aaf78121eff0

cas.authn.pac4j.oauth2[2].authUrl=https://graph.qq.com/oauth2.0/authorize

#原路径

#cas.authn.pac4j.oauth2[2].tokenUrl=https://graph.qq.com/oauth2.0/token

#由于返回非json，做代理转发

cas.authn.pac4j.oauth2[2].tokenUrl=http://localhost:8808/token/qq

#原路径

#cas.authn.pac4j.oauth2[2].profileUrl=https://api.weixin.qq.com/sns/userinfo

#中转帮忙做这个jsonp的转发动作，具体看sso-client-demo/sso-client-proxy-demo模块

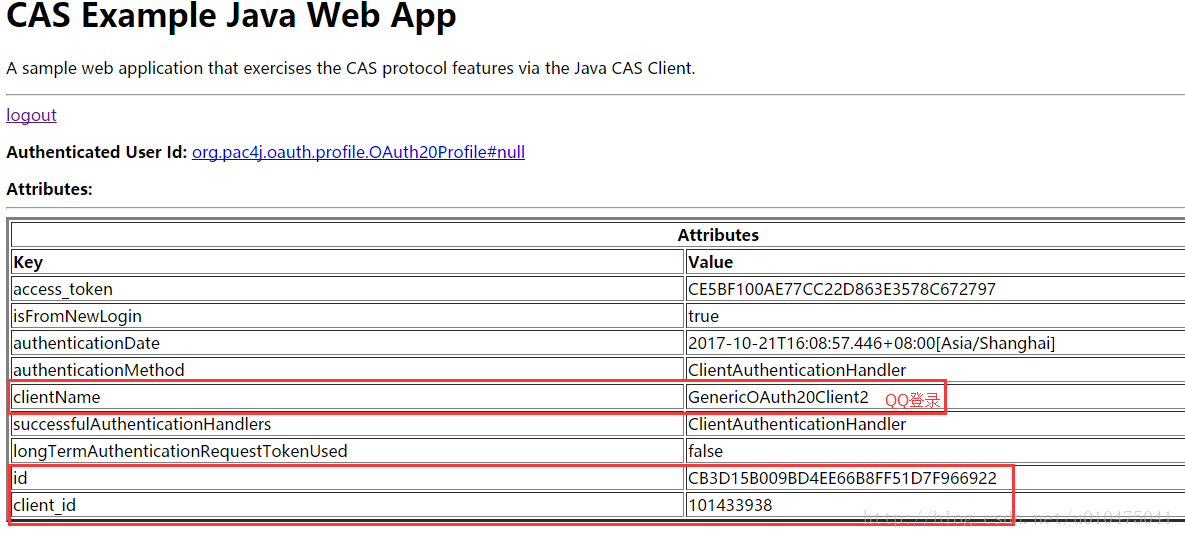
cas.authn.pac4j.oauth2[2].profileUrl=http://localhost:8808/user/qq

cas.authn.pac4j.oauth2[2].profileAttrs.id=openid

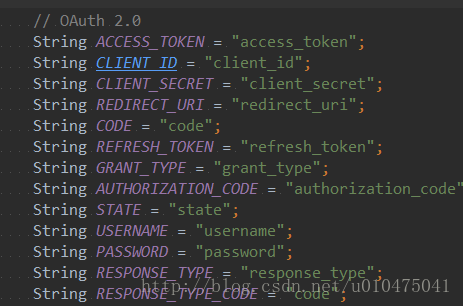
cas.authn.pac4j.oauth2[2].profileAttrs.client\_id=client\_id

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15





# 温馨提示

OAuth2的参数标准：   


**测试成功**：

* CSDN授权登录
* GitHub授权登录
* QQ授权登录

#### 学习 <http://blog.csdn.net/u010475041/article/details/78300586>