# Spring Boot自己实现简版OAuth2.0

先介绍一下背景，当时跟几个合作方一起给客户做一个系统，A系统是做数据分析的，采集一些数据，然后经过分析，用很多图表展示出来。客户要求其他合作方也能在自己的WEB内展示A系统的页面，我们就提出A可以根据OAuth2.0协议搞一个认证服务。

我自己先用Servlet实现了一个认证服务，现在改成Spring Boot实现。

另外Spring Boot结合Spring Security以及OAuth2.0的插件也能很快的实现，抽空用这个搞一次。

## OAuth2协议改版授权流程

先说明一下，毕竟是我们系统内部用，很多配置都事先协调好了，不像微博那种面向互联网用户的，所以我们在OAuth2协议上做了很多简化（比如我们就没有实现更新AccessToken的功能，返回AccessToken的时候也没有带上有效期），大家在实际工作中还是按照自身的情况做。

1. 客户端Get请求<http://127.0.0.1:6742/oauth2/authorize/>客户端ID 。
2. 授权服务端跳转至登录页面。
3. 用户输入账号密码进行登录。
4. 授权服务端校验登录信息，生成授权码并返回至客户端。
5. 客户端Post请求<http://127.0.0.1:6742/accesstoken> 传入客户端ID、口令、授权码
6. 授权服务端校验客户端ID、口令、授权码，返回AccessToken。

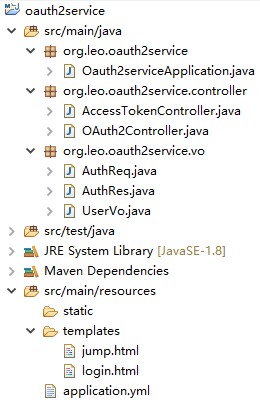
注意：协议必须是HTTPS。

## 工程构建与结构

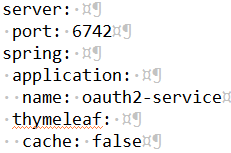
参考以前的文章构建工程，依赖选Web、Thymeleaf（这个是用来做页面的，大家按照实际情况选用即可）。

本例只是作为Demo，所以没有引入Eureka等Spring Cloud组件，在实际工作中请按照实际情况引入相关依赖。

代码结构如下：



application.yml：



## 代码

Oauth2serviceApplication没什么可改的。

OAuth2Controller：

这个Controller主要是与页面配套，authorize方法是跳转至登录页面；login方法是对用户名密码进行校验。



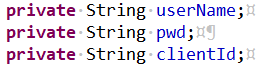
AccessTokenController：

这个Controller主要是提供Restful接口，返回AccessToken。

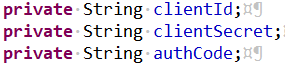


注意：在实际工作中，这个方法肯定会遇到跨域问题，我们可以使用CROS协议来解决此事。后续有空了再写一篇说说这个吧。

UserVo：



AuthReq：



AuthRes：

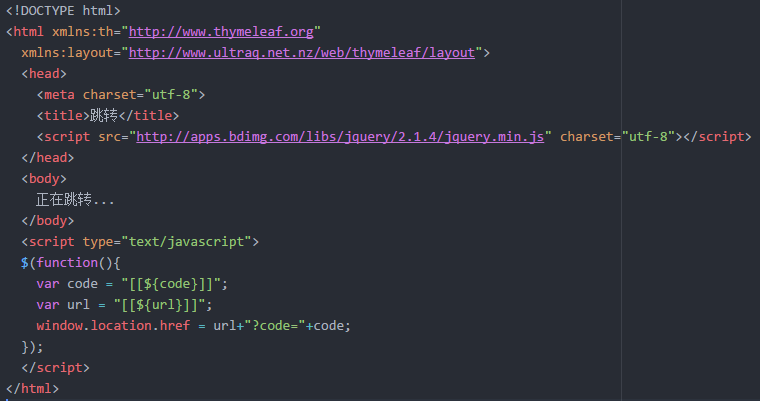


login.html：



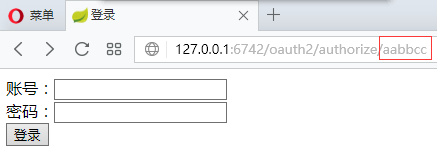
jump.html：

跳转至客户端的URL。



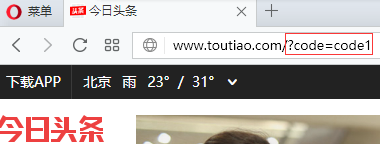
## 结果测试：

1. 浏览器访问授权服务：

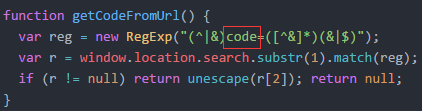


红框内的aabbcc就是客户端ID。

1. 输入账号、密码登录，校验成功后会跳转至实现配置好的客户端URL，本例中是今日头条：



红框内就是返回的授权码，一般来说，客户端的这个页面可以通过各种方式拿到这个code1，比如：



红框内的code就是url的key。

1. 获取AccessToken：

客户端发送Post请求，传客户端ID、口令、刚得到的授权码，获取AccessToken，本例使用Postman。

