# Spring Boot实现OAuth2.0 JWT

## 前言

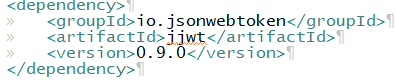
我在工作中很大一部分工作是做后端开发，采用无状态服务方式，那么接口有两个问题需要解决：

1. 用户信息。比如“好友列表”、“个人信息”这些接口，因为不用session了，那么每次请求都要带上用户的唯一标识。我们的解决方案是直接把标识作为共通消息写在请求体里，用POST方法经由HTTS协议发送至服务端，这样投入最少，安全性也相对得到保证。
2. 授权。有些接口，游客身份可以访问，比如“产品列表”，有些就需要用户登录之后才能访问，比如“我的订单”。最常用的肯定就是OAuth2.0协议了。

之前写的《Spring Boot自己实现简版OAuth2.0》只是实现了整个协议的授权流程，但其中关键的Token没有涉及，只是写了个固定值，这样是不够的。本文就是基于JWT来实现Token的签发。

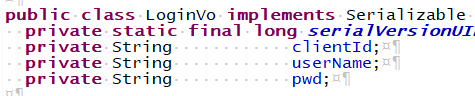
工程可以参看《Spring Boot自己实现简版OAuth2.0》，在这里实现Token的签发，使用Spring Cloud的同学，如果用到了Zuul，可以参看《Spring Cloud网关Zuul&过滤器&集群方案》，在Zuul中实现。

## 实现

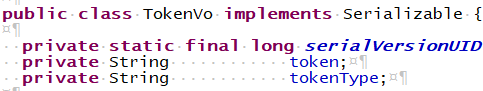


增加jjwt的依赖。

登录请求类：

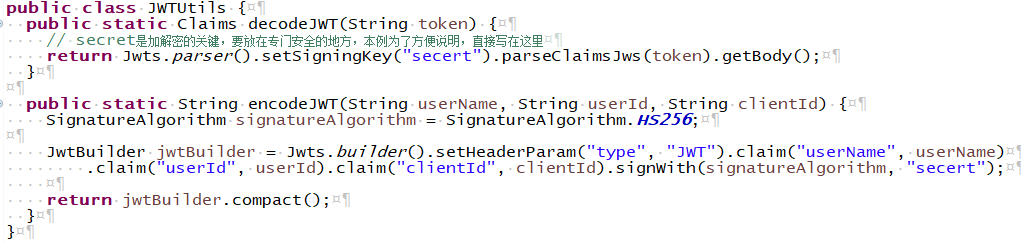


Token信息类：



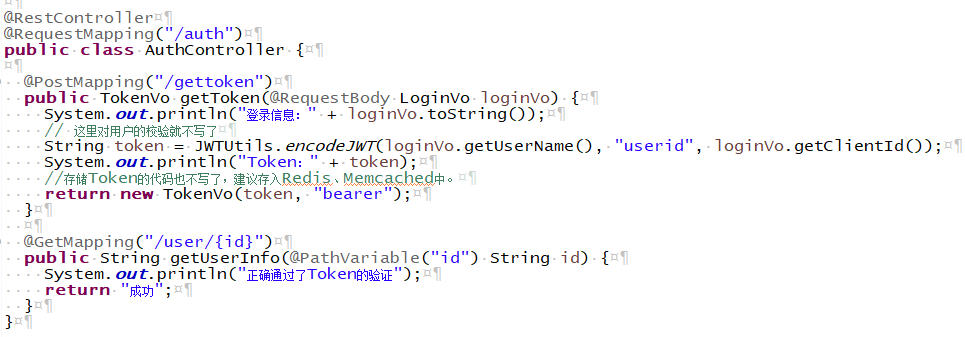
注意，这里的内容请按照实际情况进行修改。

Jwt工具类：



本例只是为了说明JWT的使用，很多细节就不写了，比如过期时间，在实际中还是要进行设置的。

Controller：



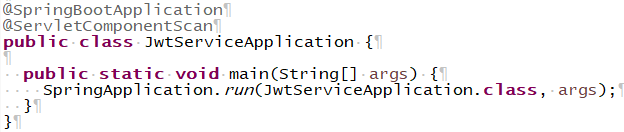
第一个获取Token，第二个测试Token用。

Filter：



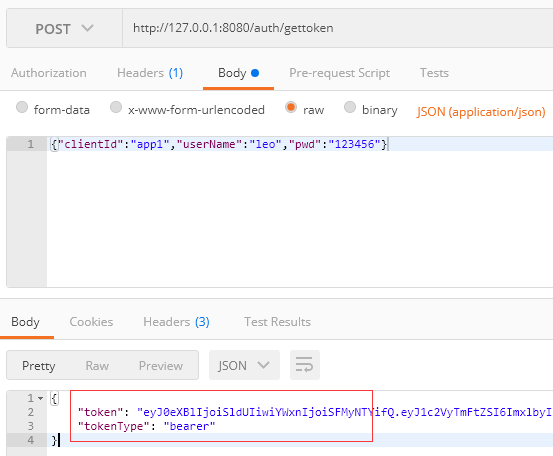
这里我只拦截了那个用来测试的接口，实际工作中接口那么多，哪些需要验证Token，哪些不需要，要进行修改。

Application启动类：

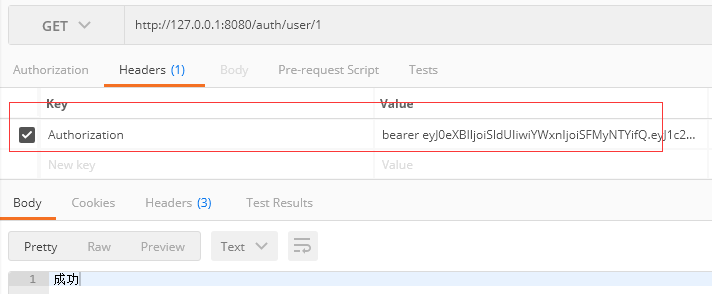


## 测试

使用Postman，获取Token：



使用Token访问接口：



在header中加上刚才获取的Token信息。

后台信息：

