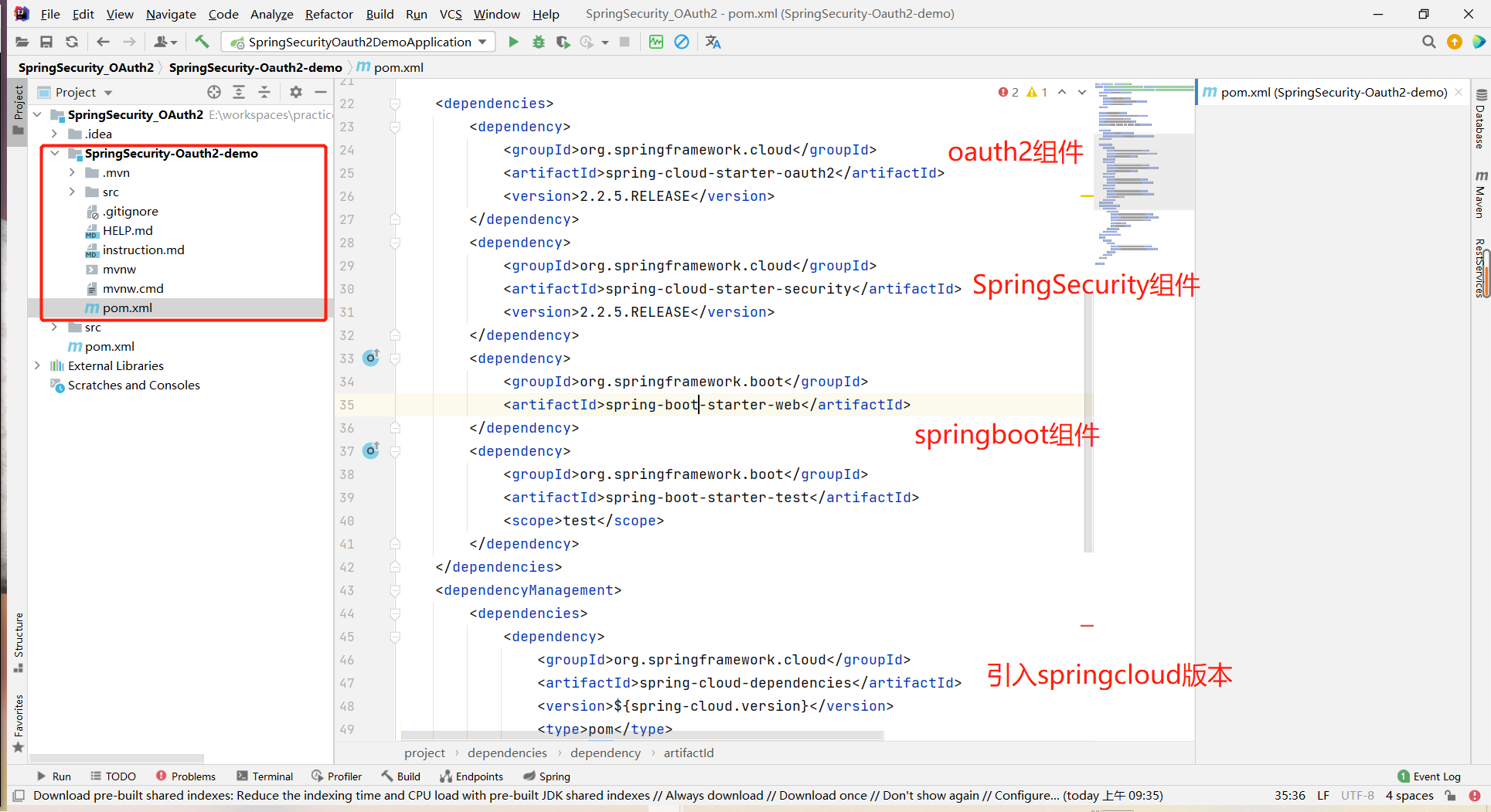
# SpringSecurity整合OAuth2

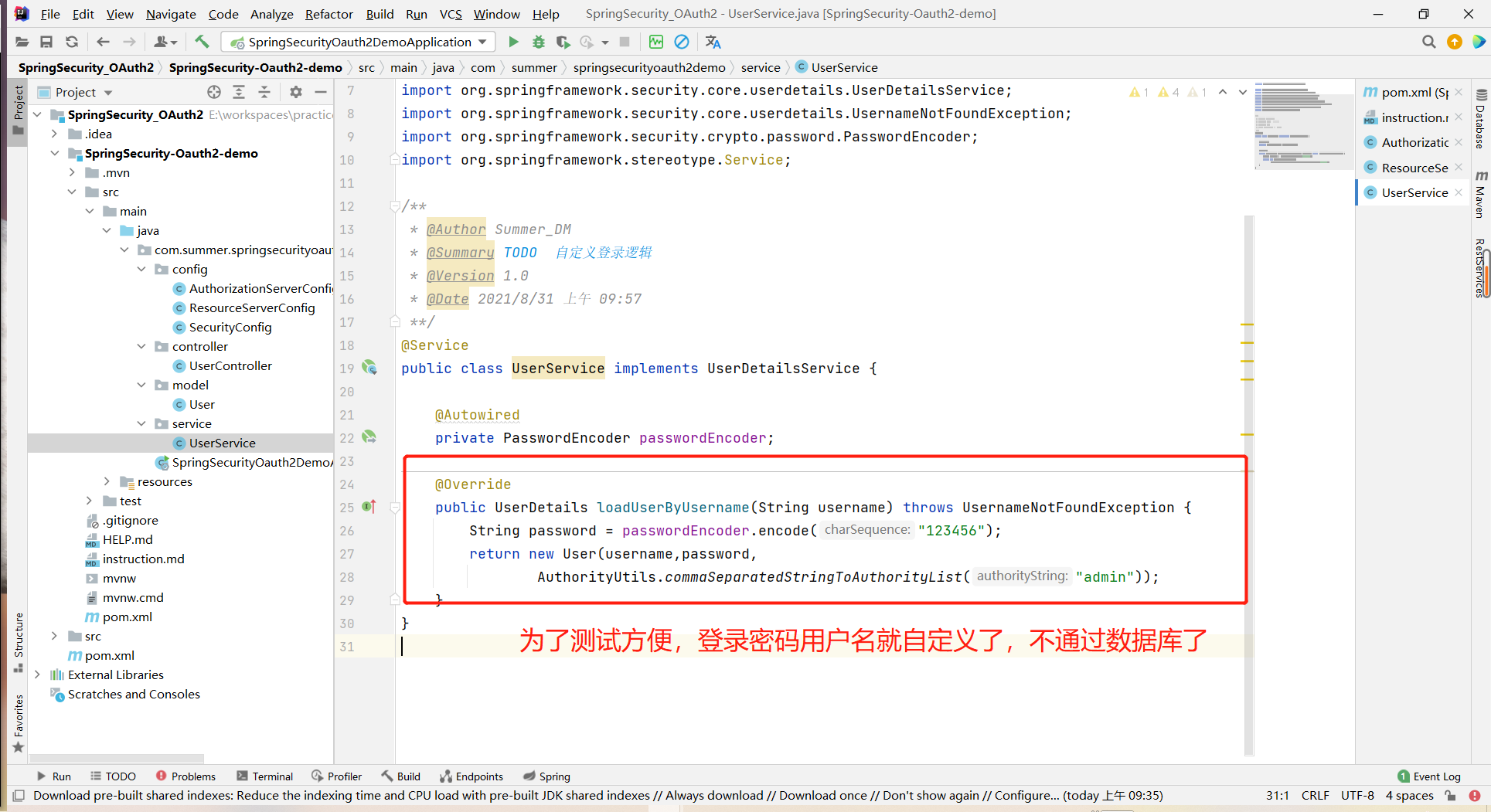
***####说明一下需要准备的模块***1、需要一个自定义的登录逻辑,去实现SpringSecurity底层的UserDetailsService--**UserService**    
2、需要一个自定义的User类来实现SpringSecurity底层的UserDetails--**User**    
3、需要一个SpringSecurity的配置类--**SecurityConfig**    
4、需要一个授权服务器的配置类-**-AuthorizationServerConfig**    
5、需要一个资源服务器的配置类--**ResourceServerConfig**    
6、需要写一个可以访问的资源--**UserController**

***注意：一般情况下，授权服务器和资源服务器不在一台机器，都是单独的系统。***

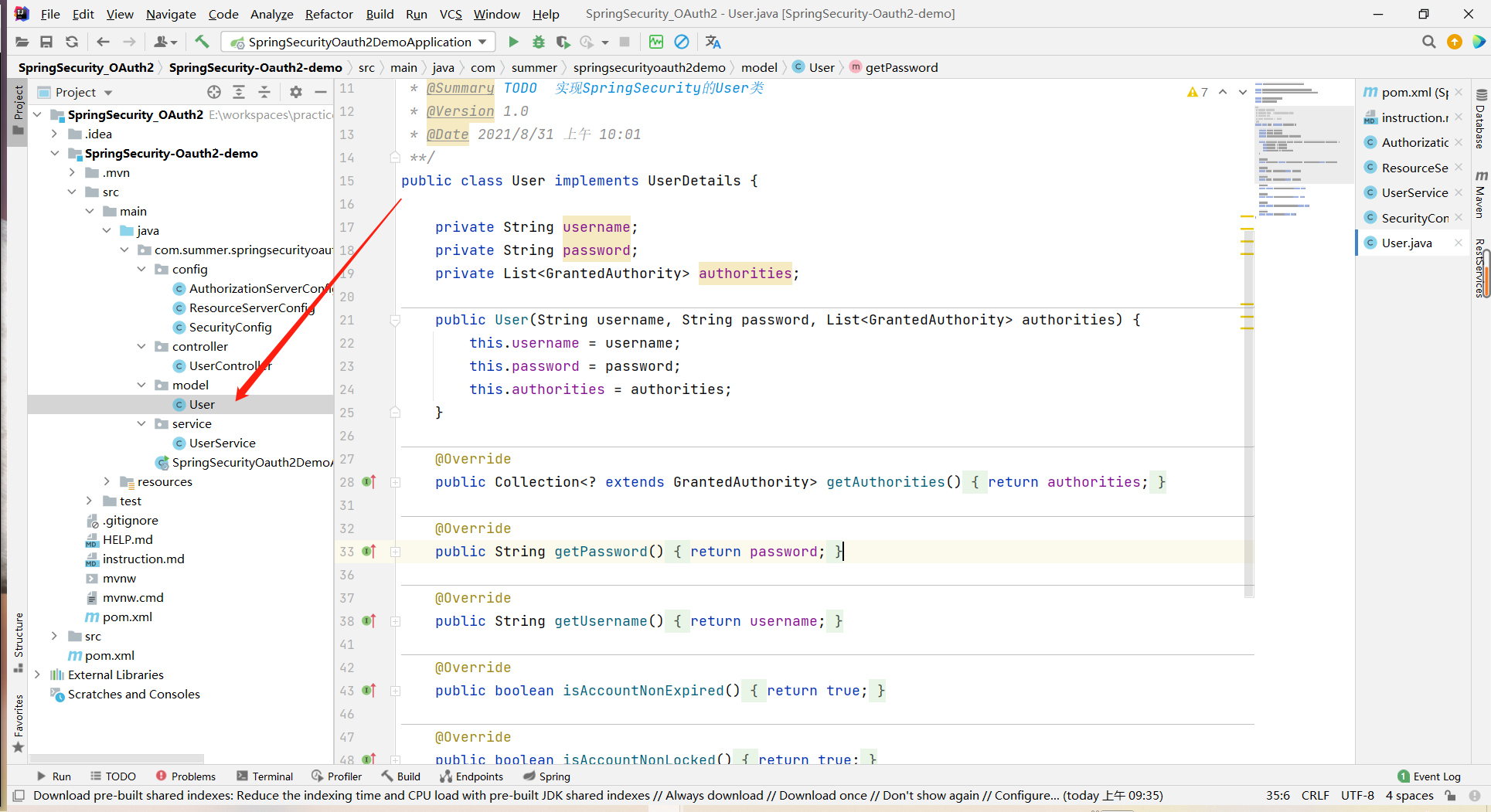
1. 我们需要新建一个工程；



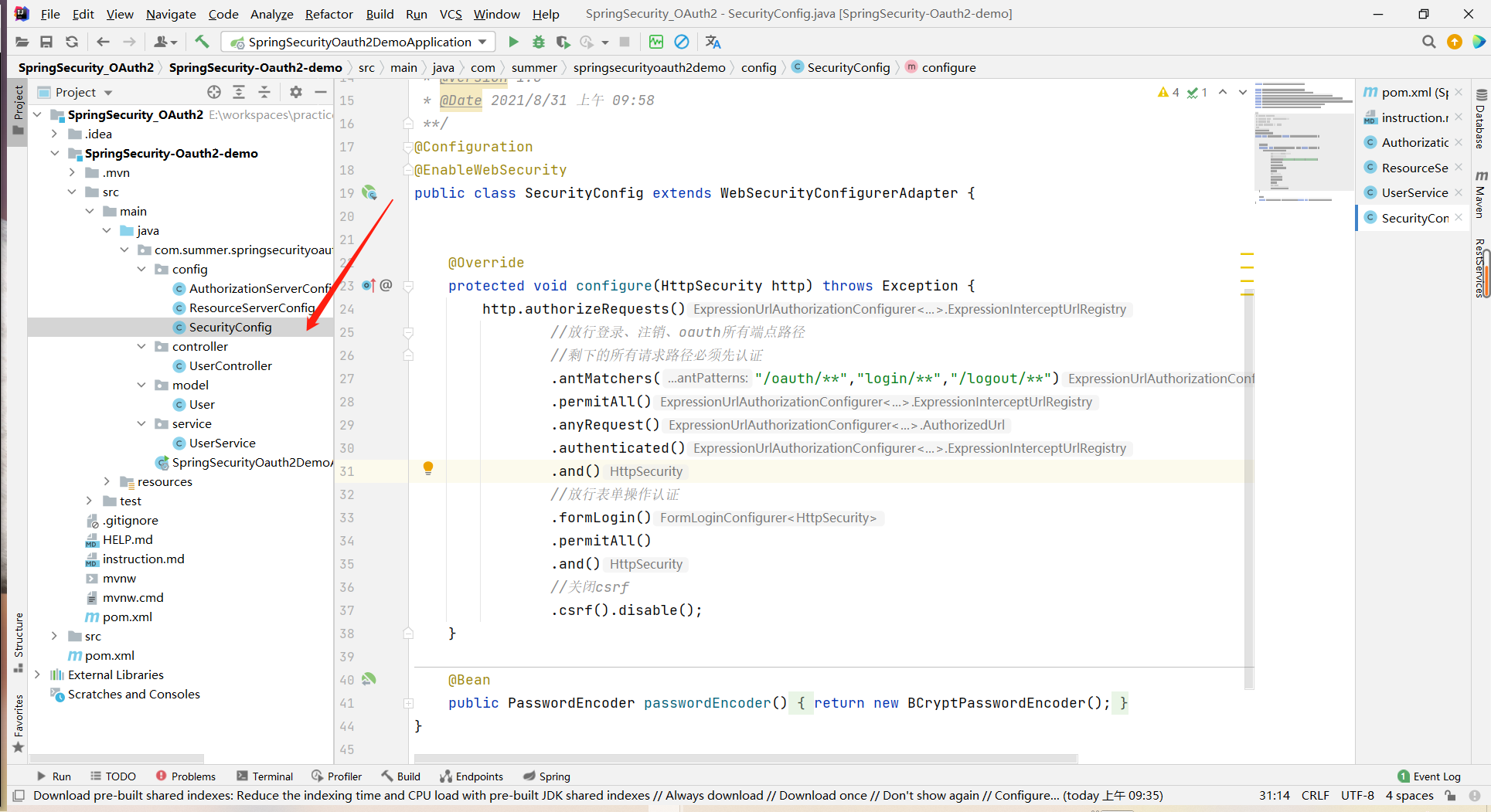
1. 首先我们先写一个自定义登录的逻辑，就相当于是客户端；



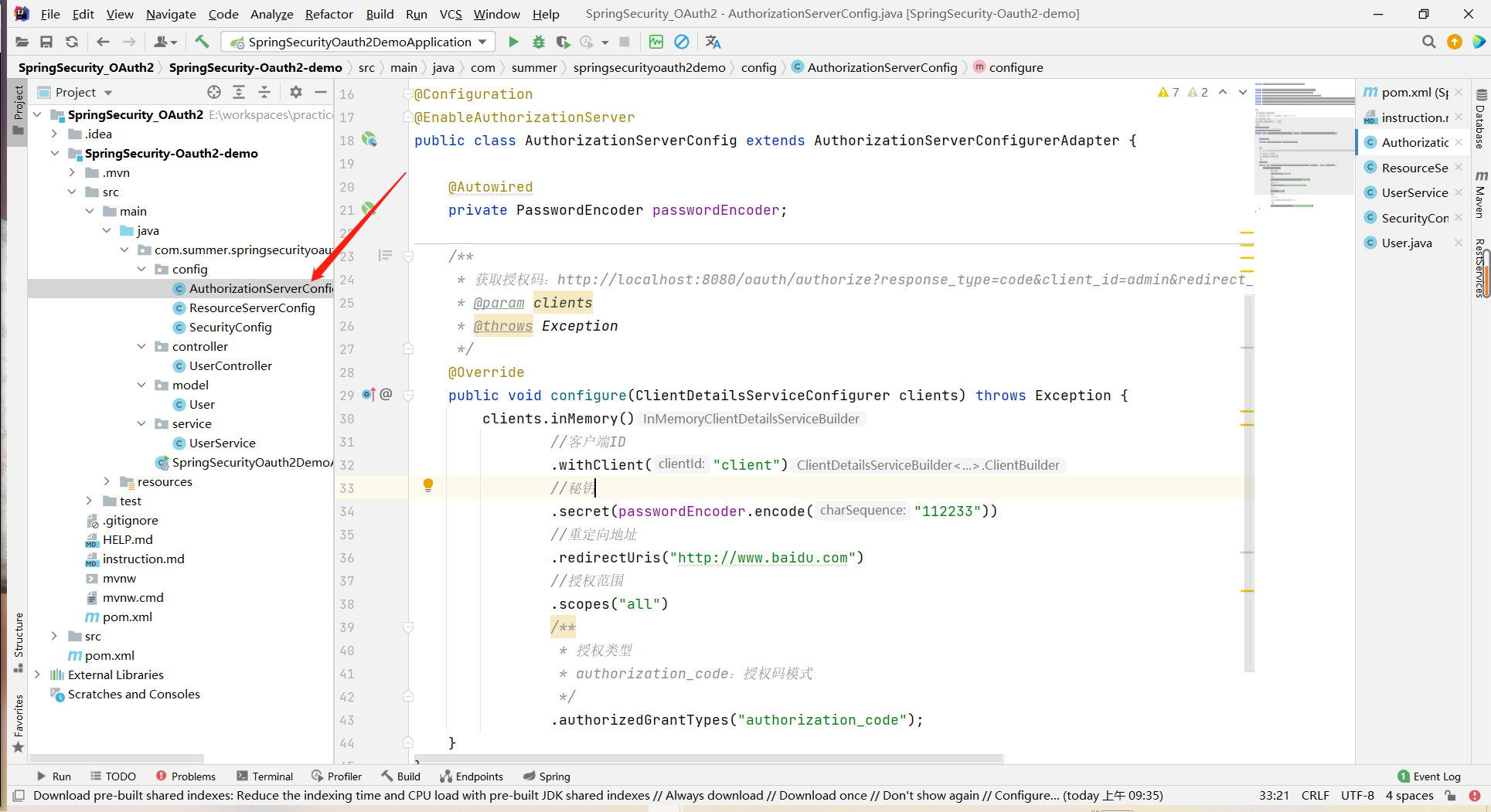
1. 接着我们去写一个自定义的User类，一定要实现SpringSecurity底层的UserDetails；



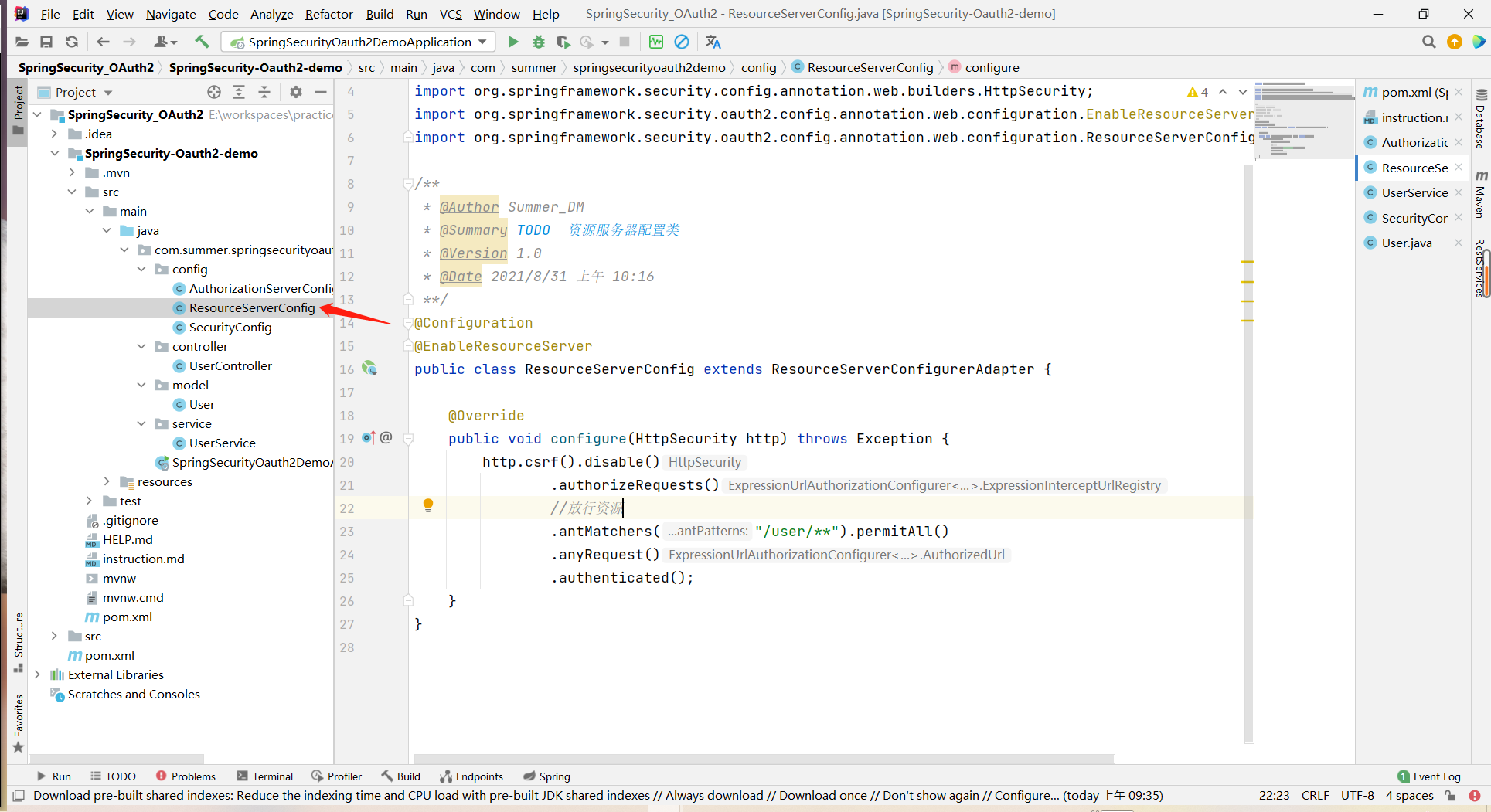
1. 这里需要写一个SpringSecurity的配置类，进行一些页面的放行和配置一些访问权限；



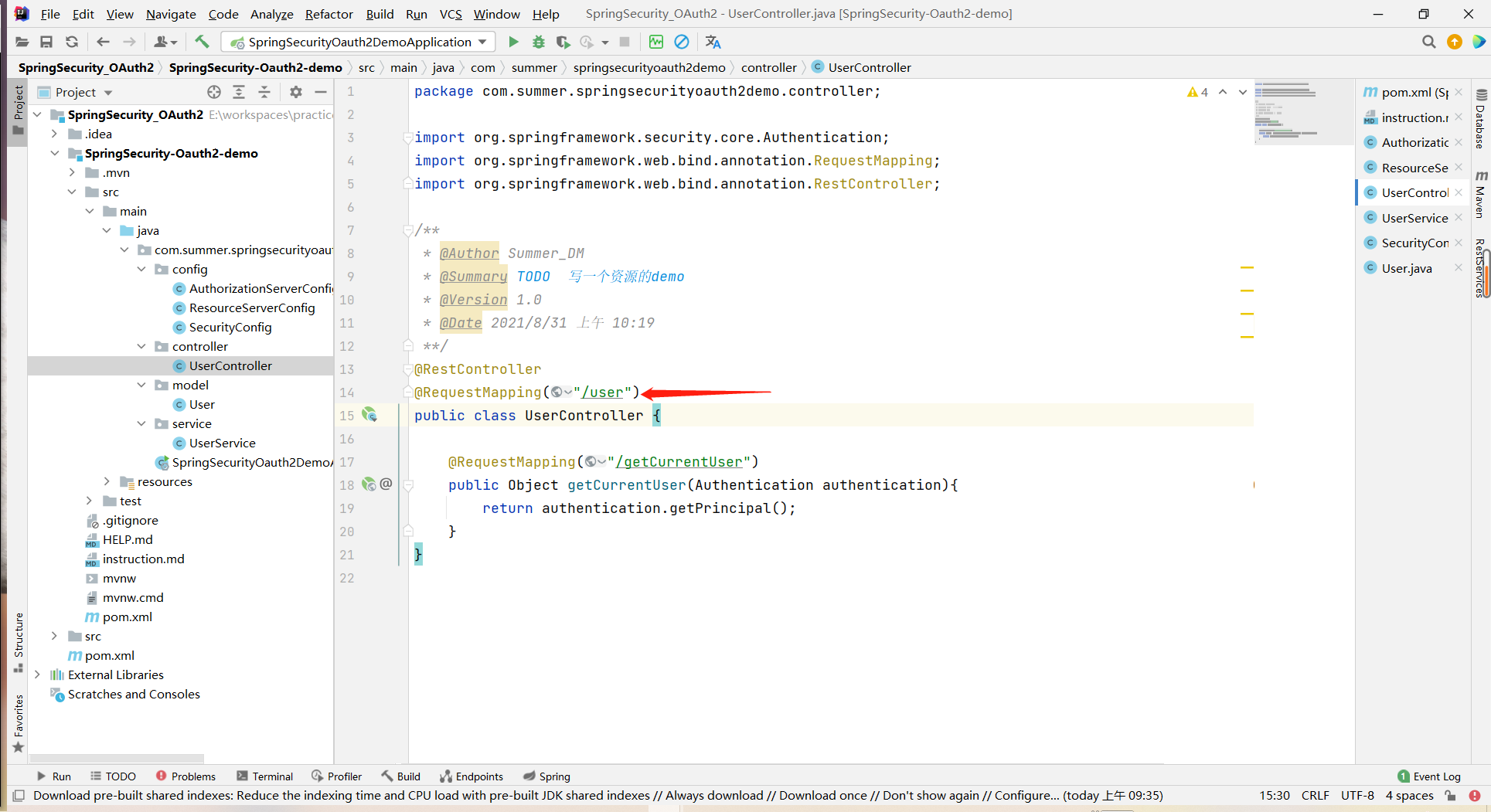
1. 再写一个oauth2的配置类，也就是授权服务器的配置，进行一些授权，配置授权类型（我们是授权码模式）之类的；



1. 还需要写一个资源服务器的配置类，主要是对我们需要访问的资源进行放行；



1. 最后我们需要写一个可以访问的资源controller；这里的访问路径一定是我们资源服务器中放行的路径。



1. 接下里进行调试，调试之前我们需要先去授权服务器获取一下授权码；

**获取授权码**

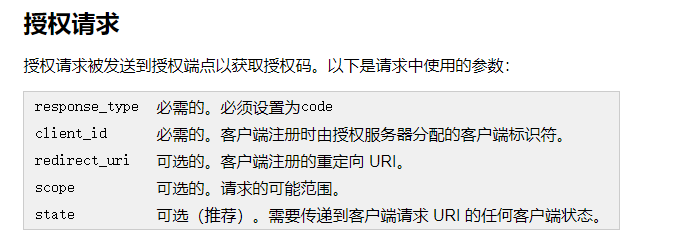
**[http://localhost:8080/oauth/authorize?response\_type=code&client\_id=client&redirect\_uri=http://www.baidu.com&scope=all](http://localhost:8080/oauth/authorize?response_type=code&client_id=admin&redirect_uri=http:/www.baidu.com&scope=all)**

**说明：**

**Oauth2官网链接：可以查询**

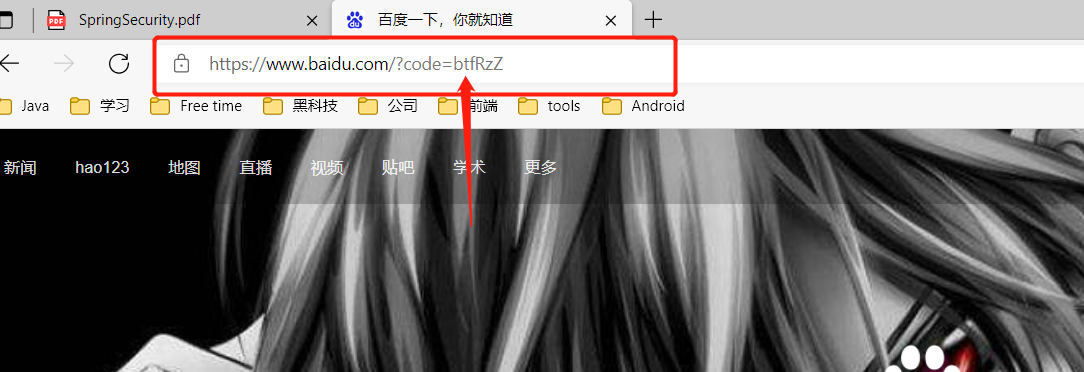


**<http://tutorials.jenkov.com/oauth2/authorization-code-request-response.html>**



下面是访问完，出现的页面：oauth2自己的页面。我们允许一下。



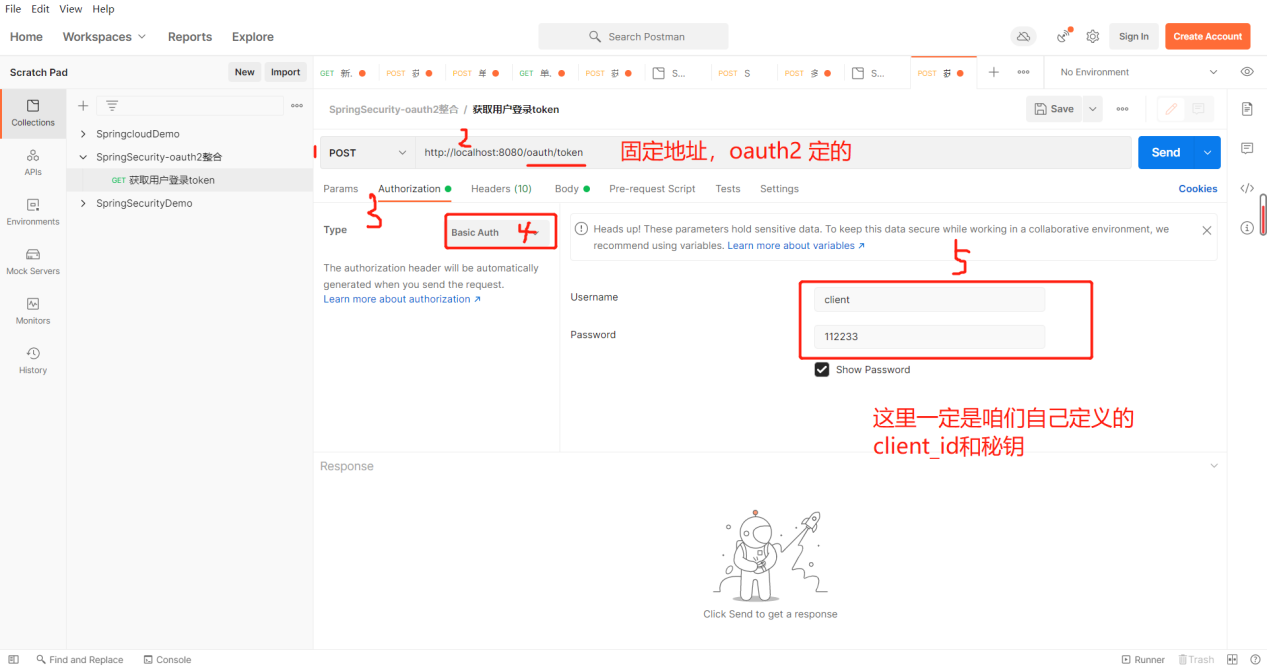


可以看到，允许之后，就跳转到了我们配置的百度页面；

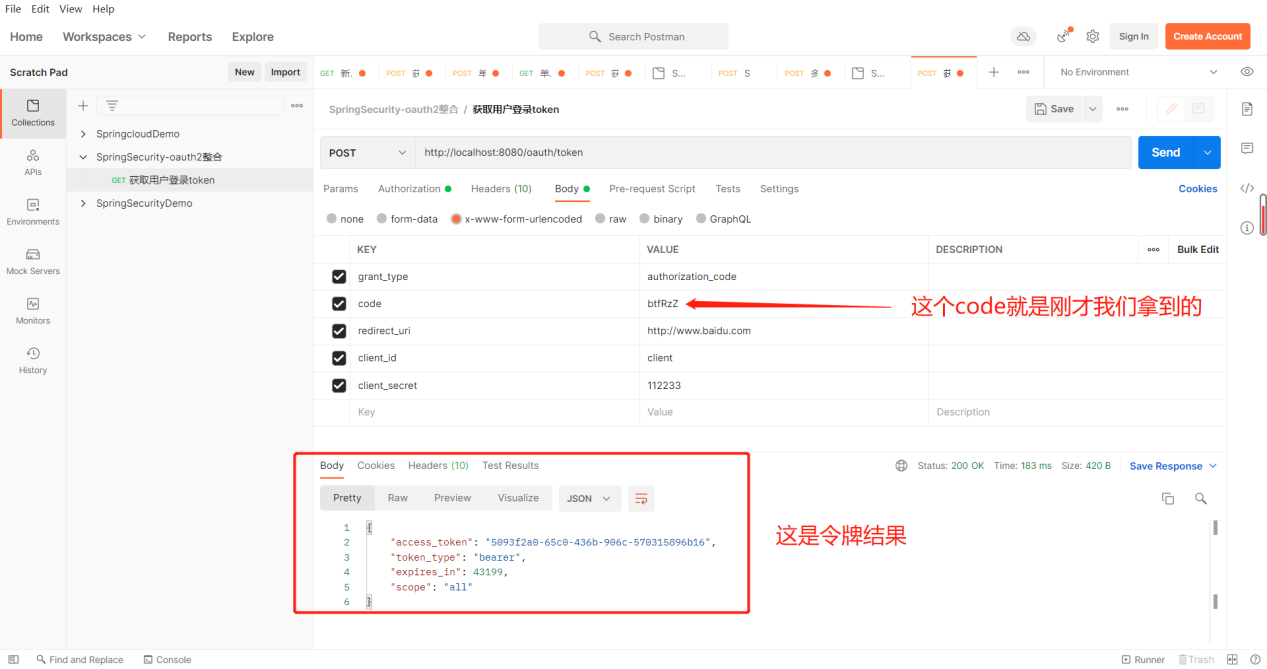
这里需要注意上面的url，后面跟着的code就是我们要的授权码；

1. 接下来需要我们使用postman去请求访问令牌；因为他是一个post请求；

按照12345去写。



下面再去写入参：



**参数说明：**

**grant\_type ：**授权类型，填写authorization\_code，表示授权码模式

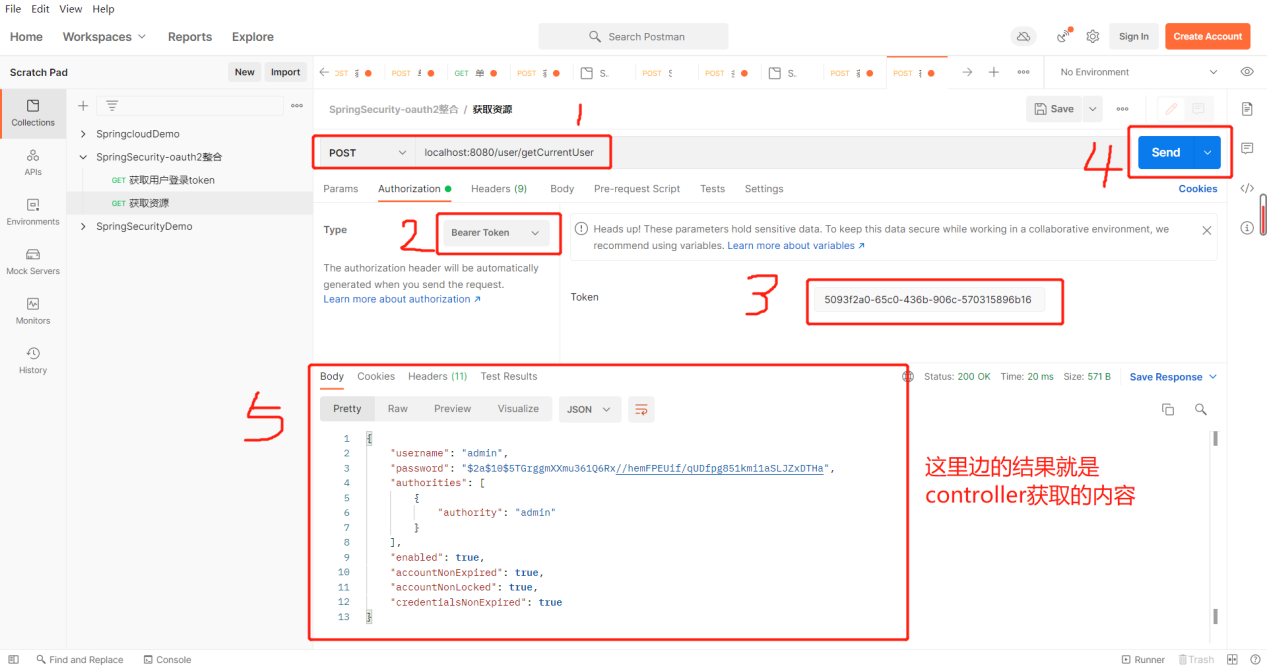
**code** ：授权码，就是刚刚获取的授权码，注意：授权码只使用一次就无效了，需要重新申请。

**client\_id :**客户端标识

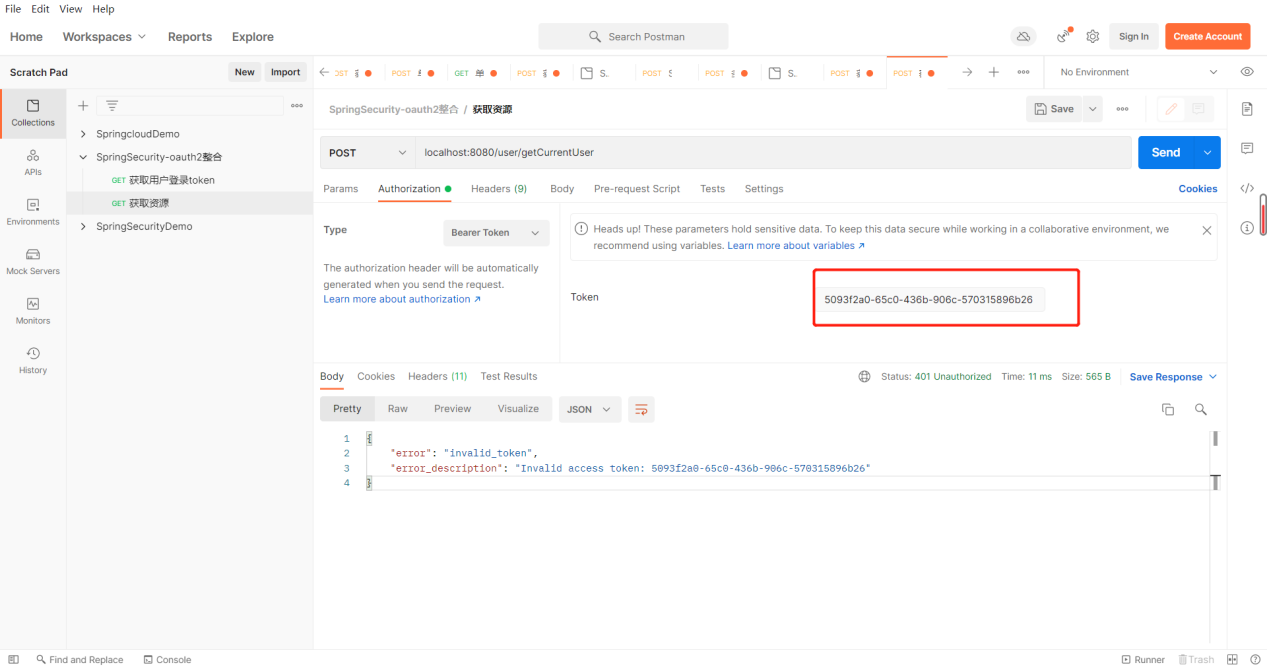
**redirect\_uri** ：申请授权码时的跳转url，一定和申请授权码时用的redirect\_uri一致。

**scope :**授权范围。

1. 接下里就可以获取我们的资源了，继续测试：



假设我们修改一下token，改成错误的，看看返回结果：



报错了，显示非法的令牌。至此授权码获取的方法就结束了。

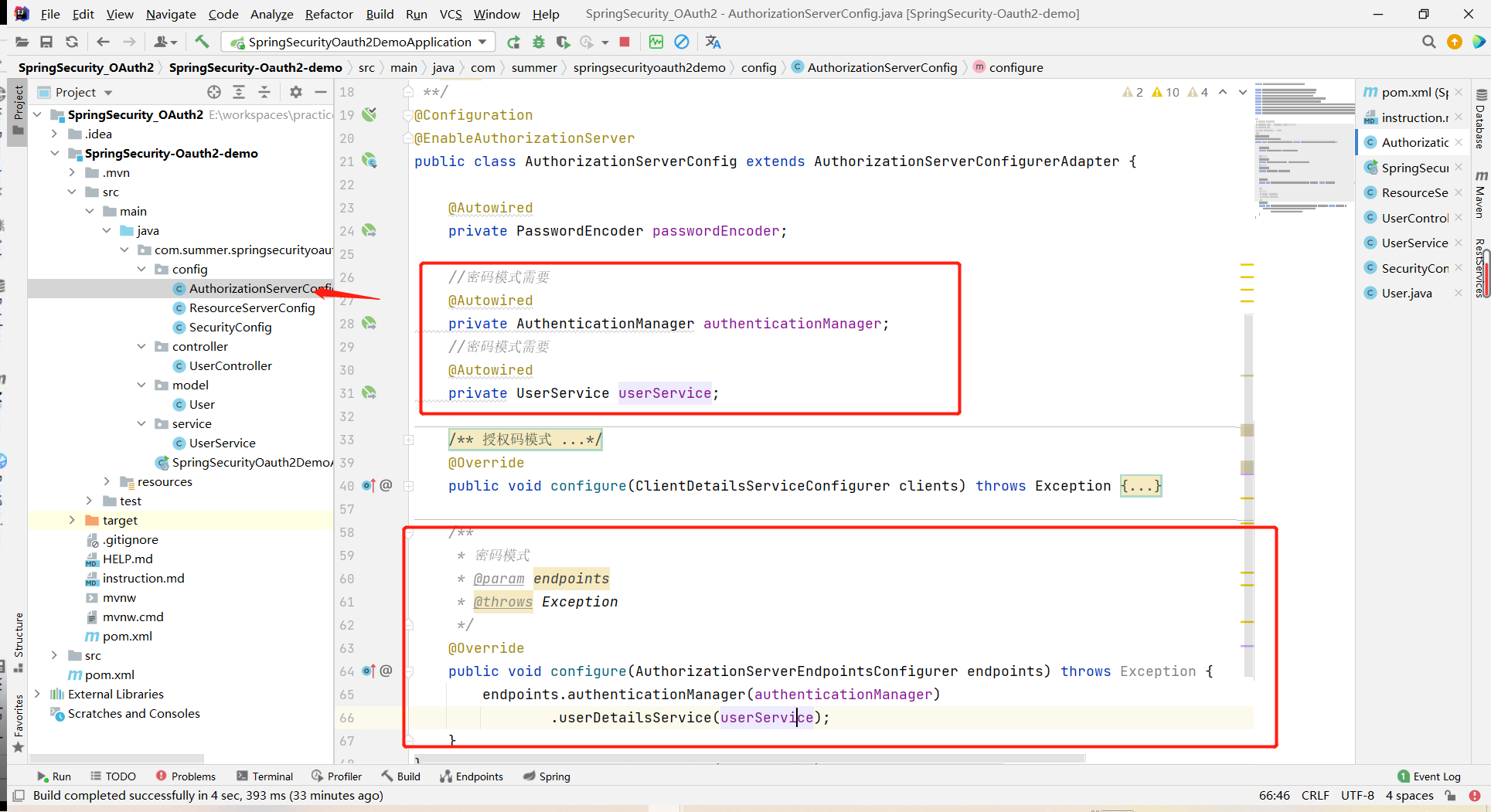
1. 下面在写一下密码模式；直接在原来的代码基础上修改一下。

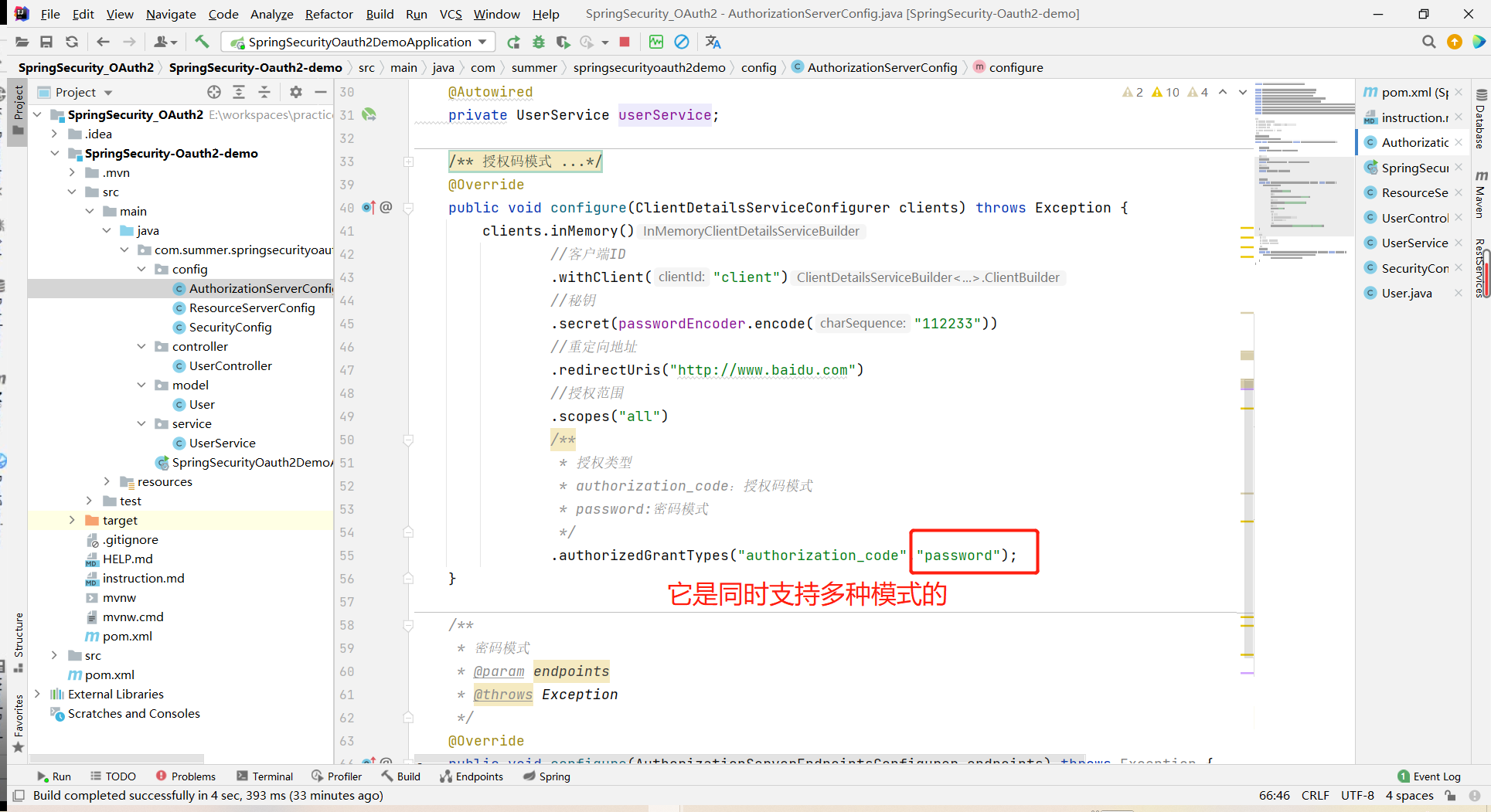
需要修改的点：

①、在SecurityConfig中新加一个注入的Bean;

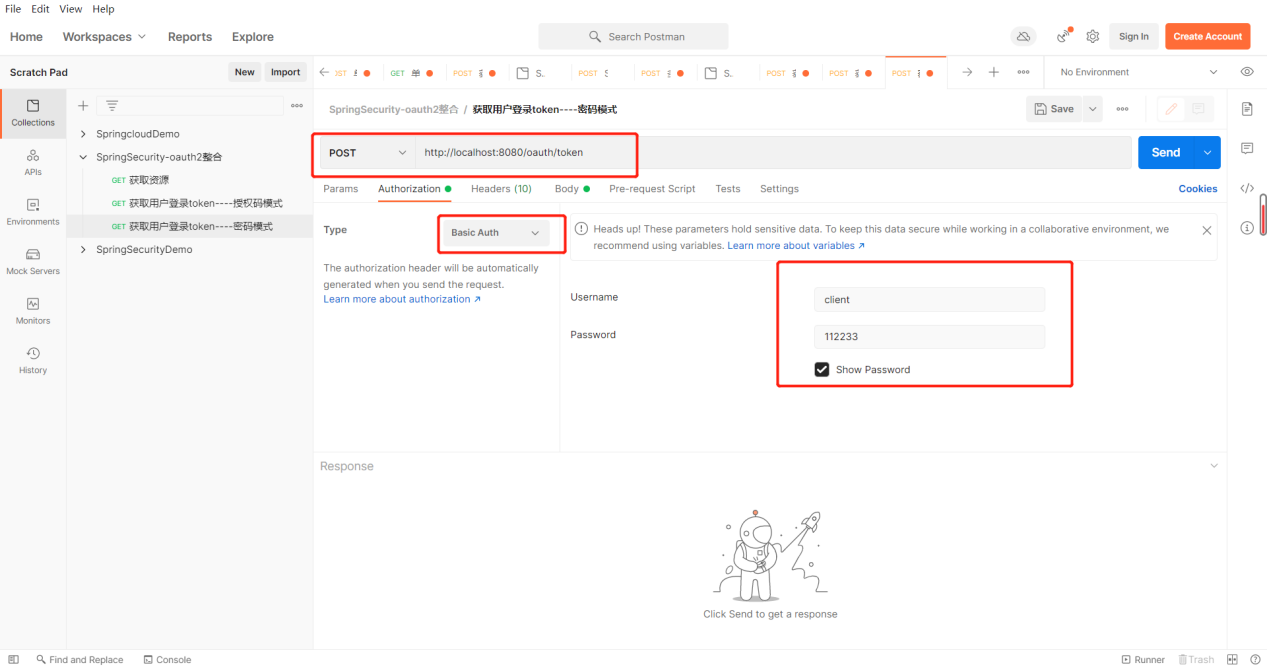


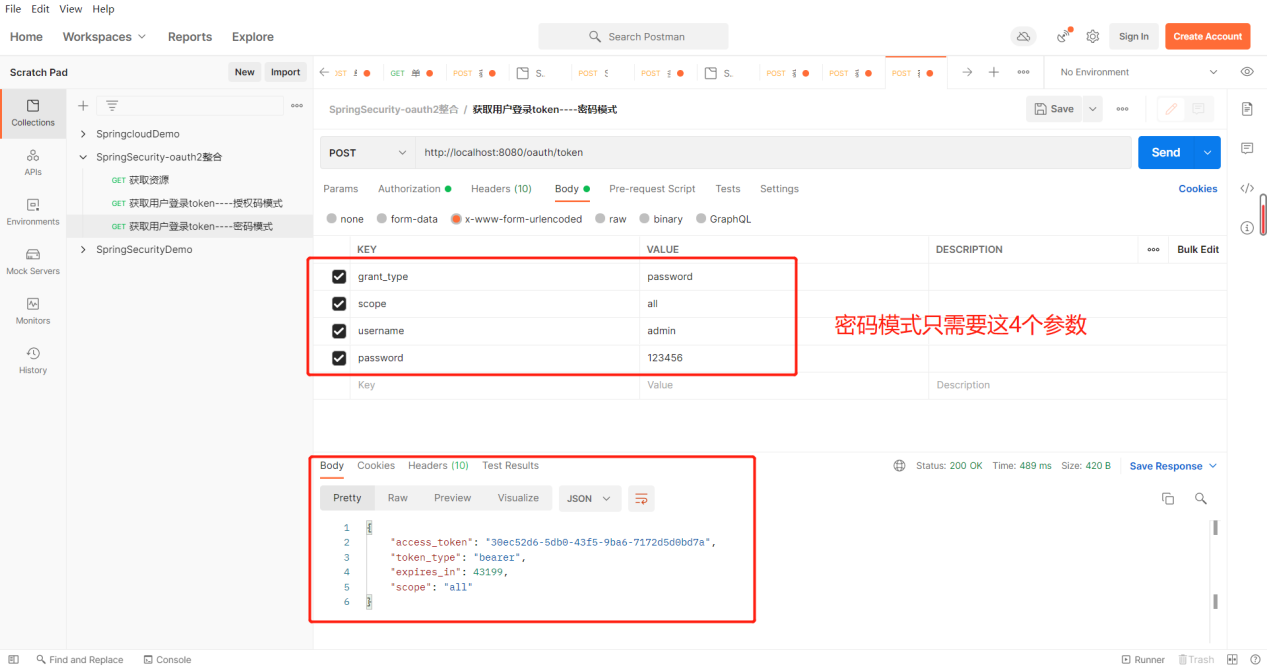
②、在授权服务器配置中新加一些配置：



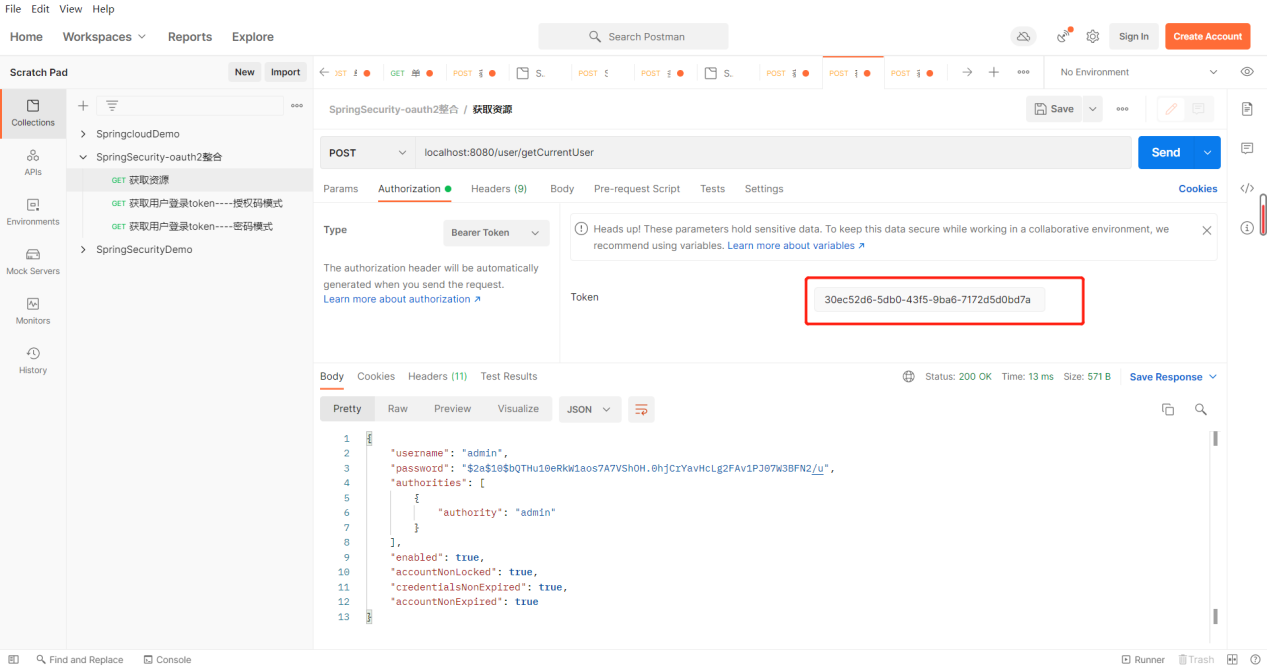


1. 重启一下，测试结果；我们只需要在客户端输入用户名密码，然后



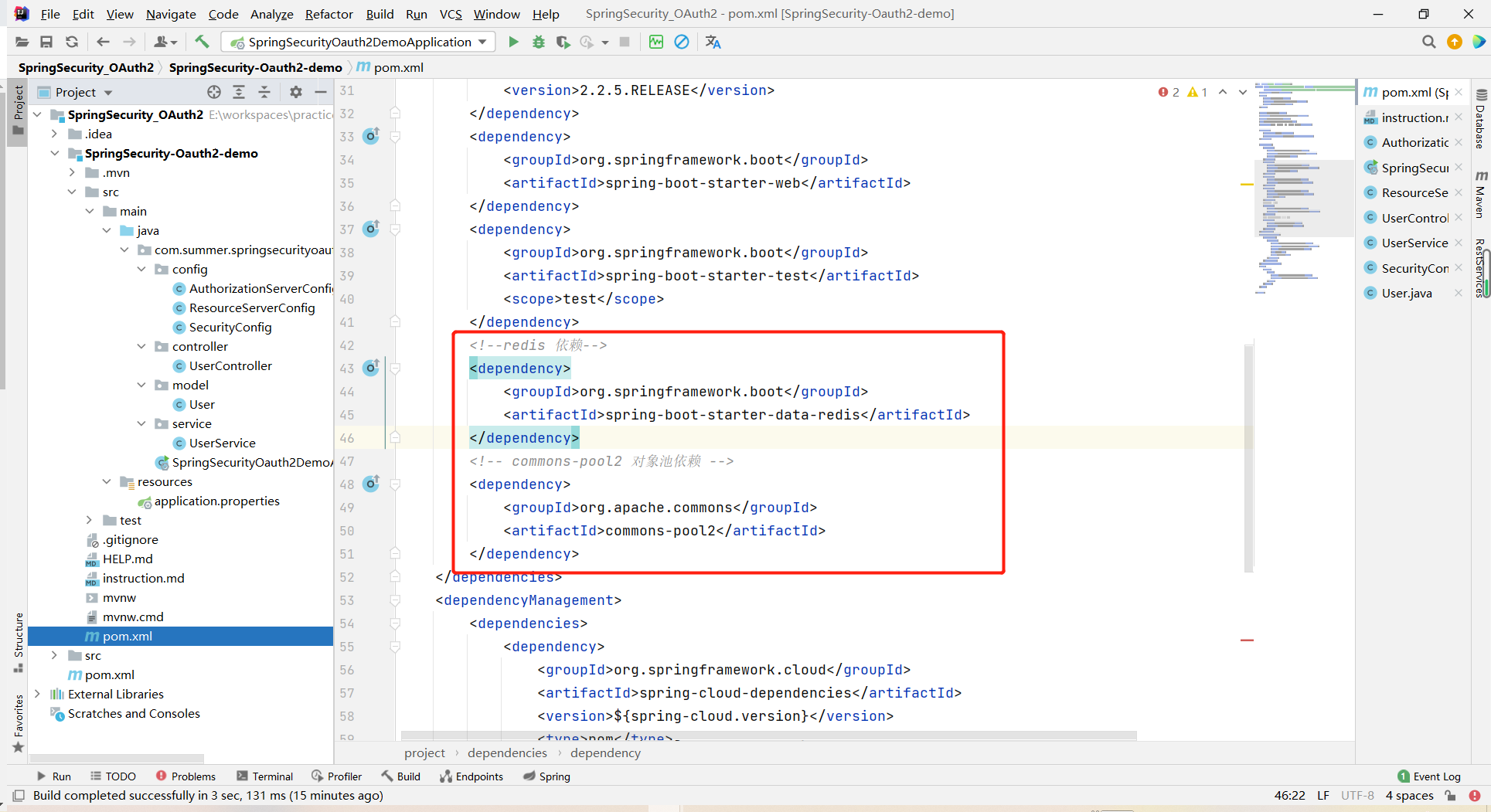


1. 接下里和刚才一样，使用token去访问资源：

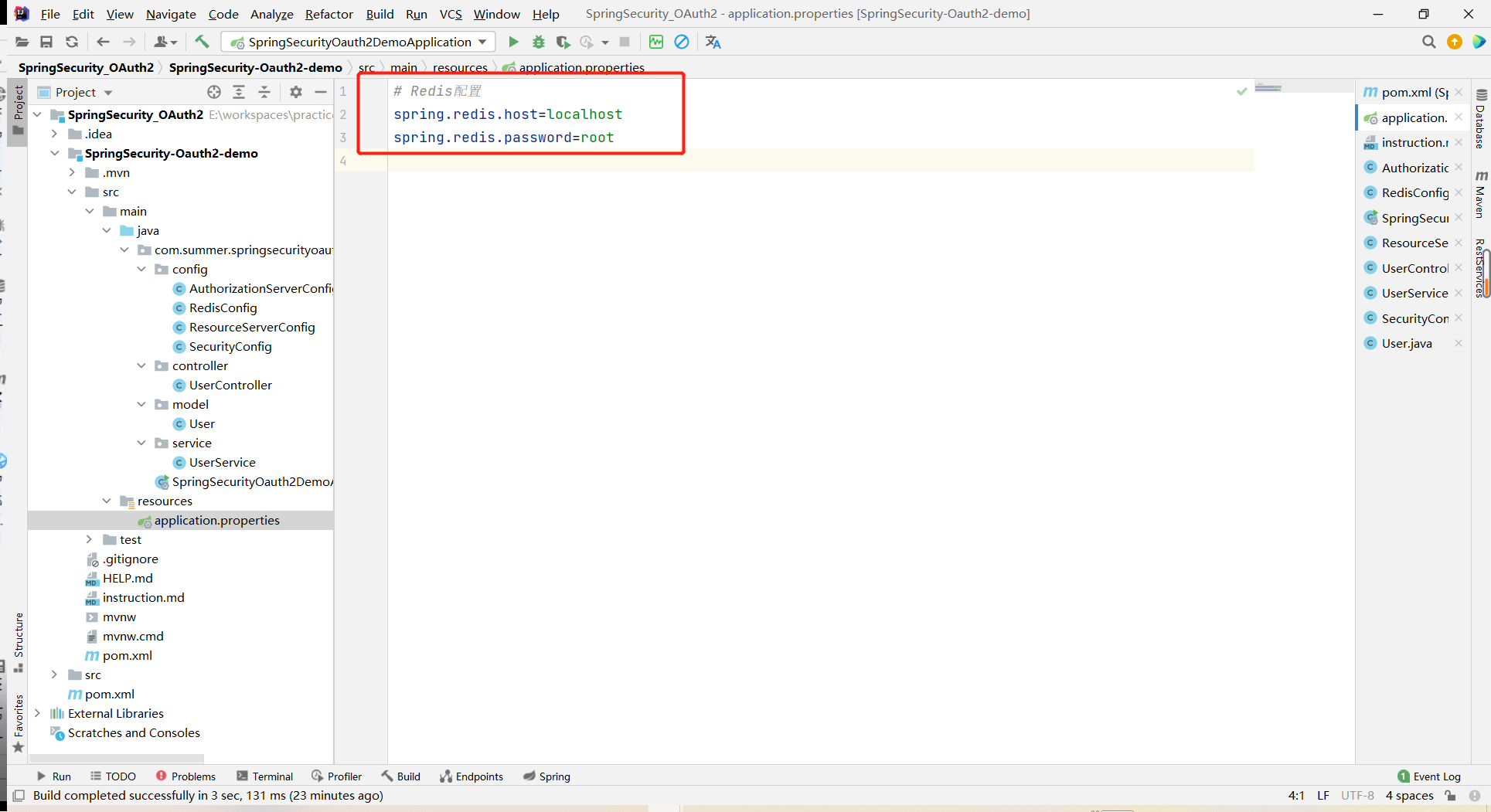


**可以看出，密码模式相比较授权码模式，要简单的多。**

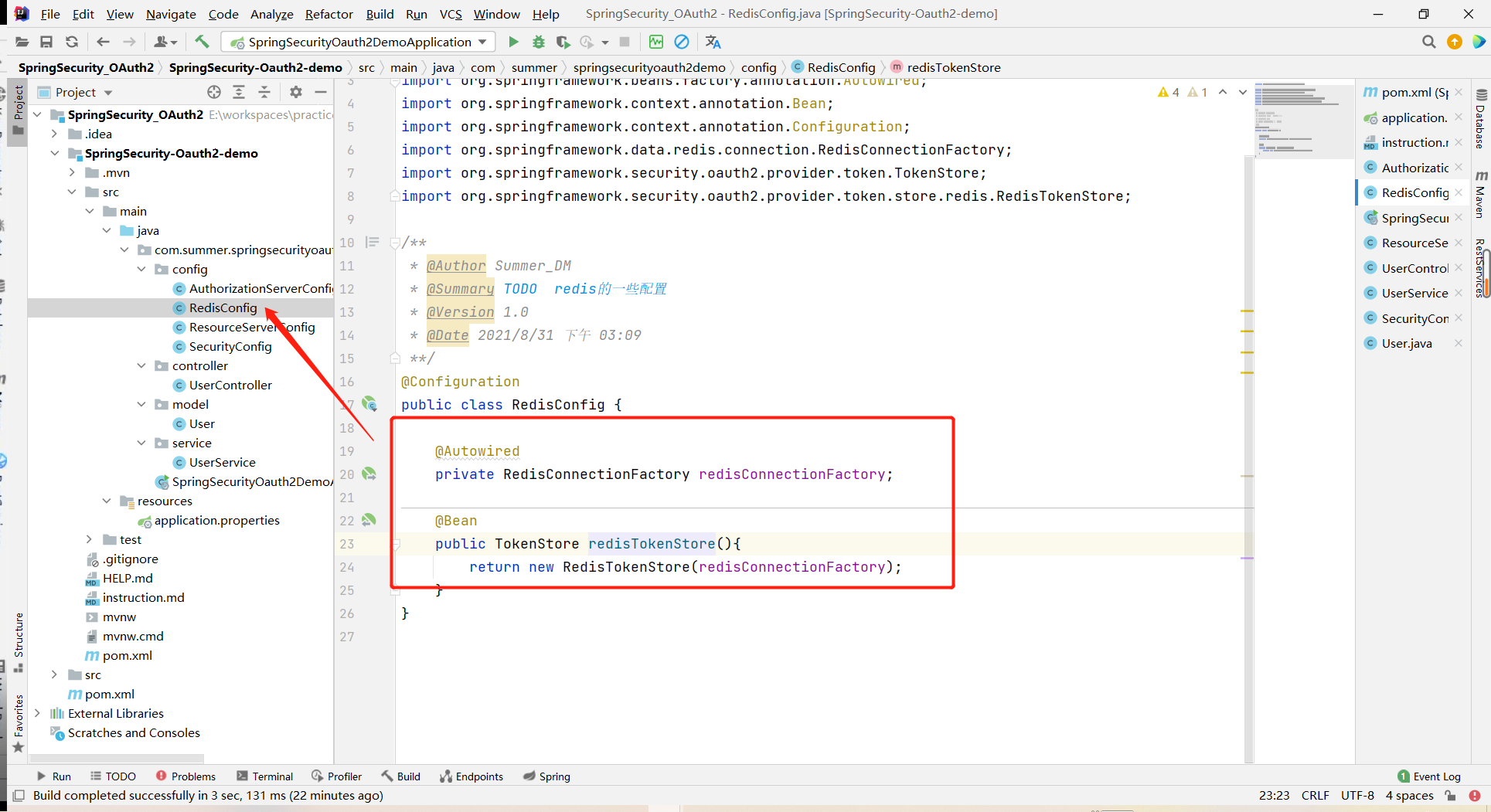
1. 接下来需要讲一下redis去存储token，生产环境一般都是这样。
2. 第一步需要引入redis的依赖



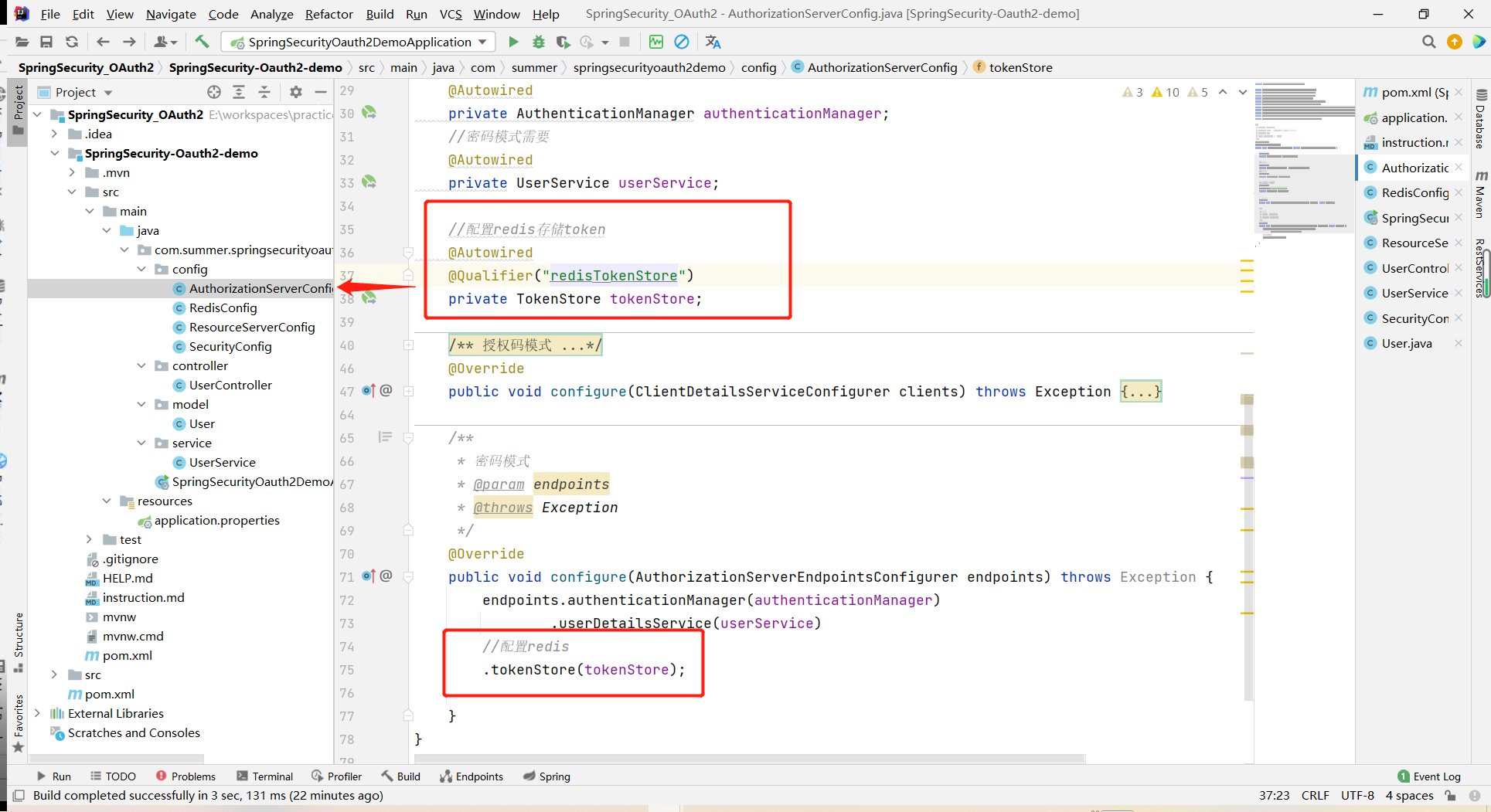
1. 需要在application.yml中增加一些redis的配置；



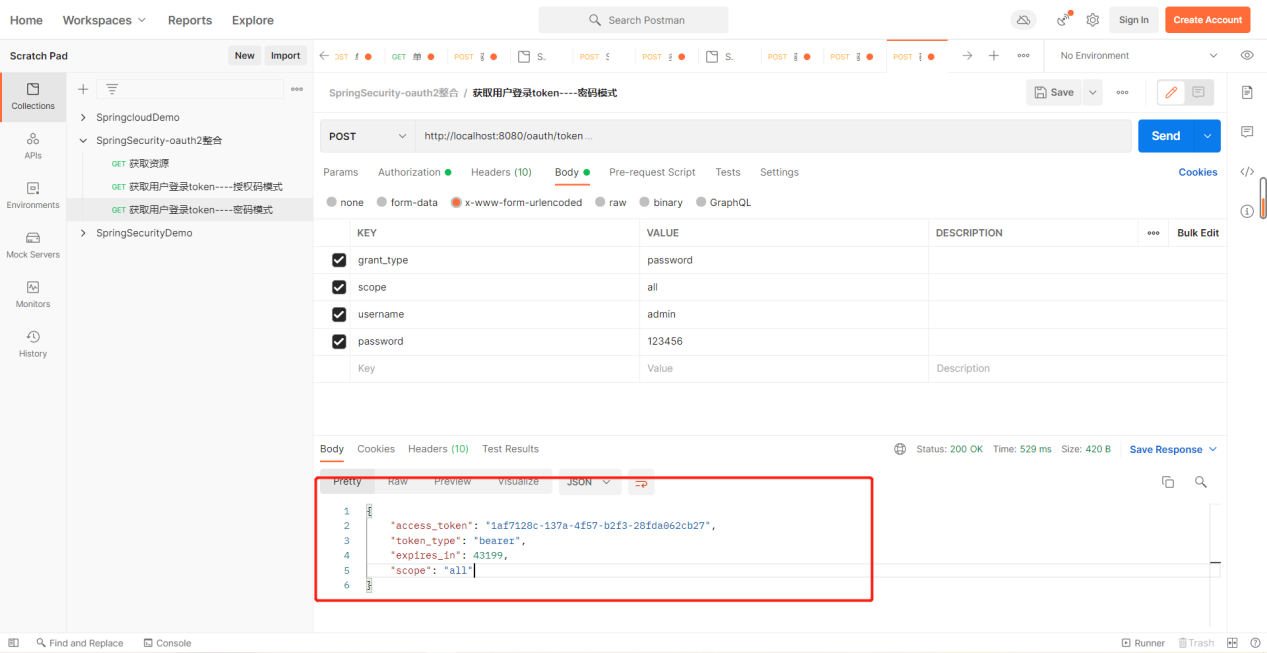
1. 需要新加一个RedisConfig类，来注入一个redis的配置类；



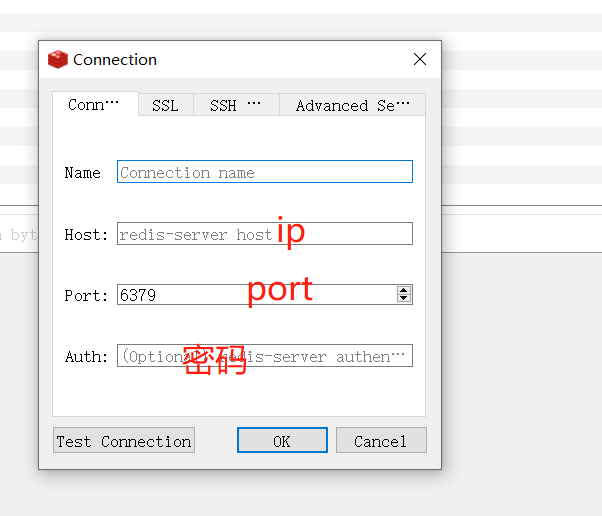
1. 最后需要在授权服务器添加redis存储token的配置；

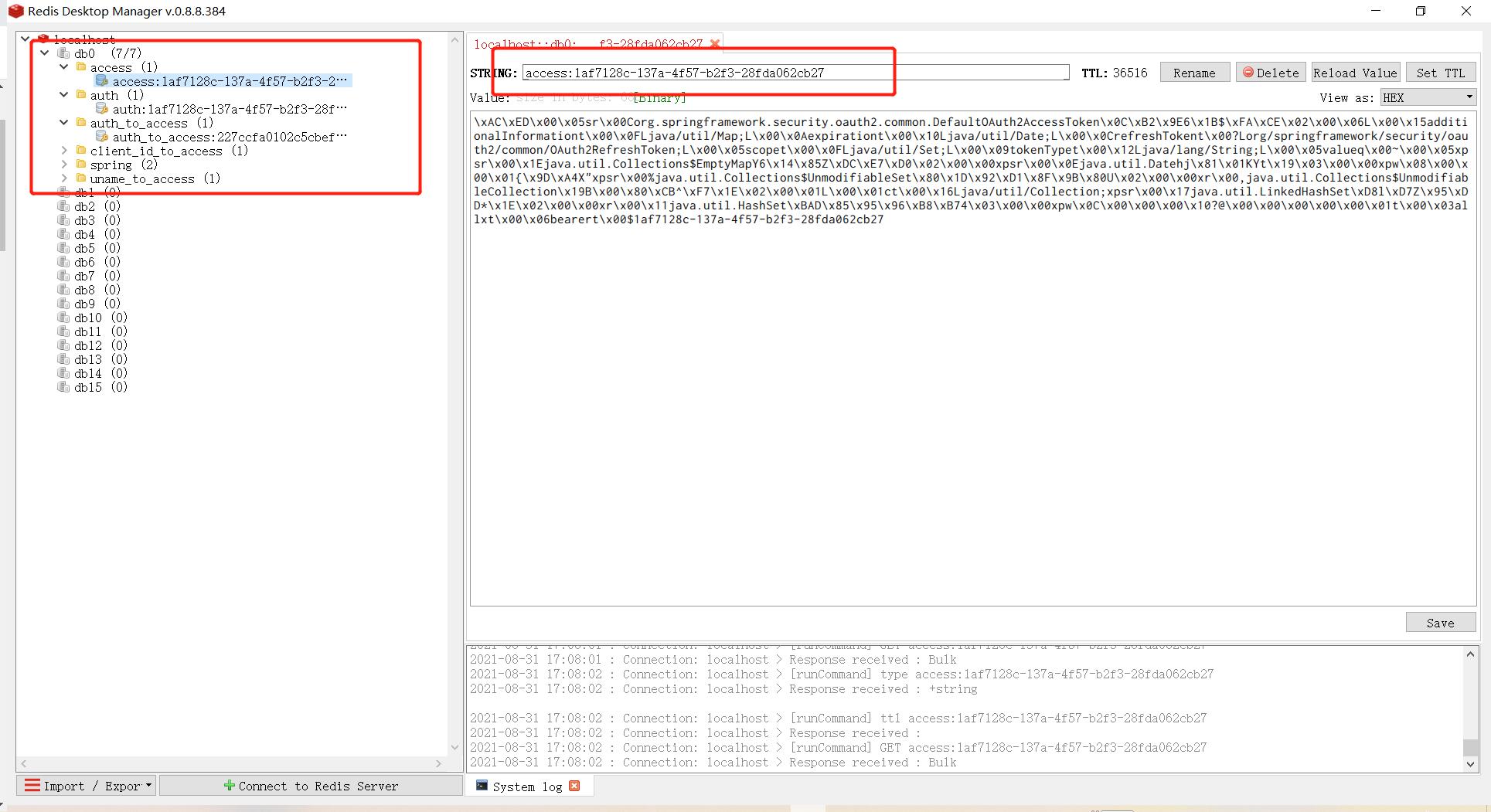


1. 最后重启服务，去postman测试一下，这里还是使用密码模式：



**这里应该再去redis中看一下是否存储token数据；**





**可以看到redis已经存储了我们的token。**用户名和客户端id；

