



# 用starter实现Oauth2中资源服务的统一配置

原创

土味儿~

已于 2023-04-16 10:21:41 修改

阅读量761

收藏3

点赞数1

分类专栏:

# SpringSecurity OAuth2

# SpringBoot

文章标签:

java

spring boot



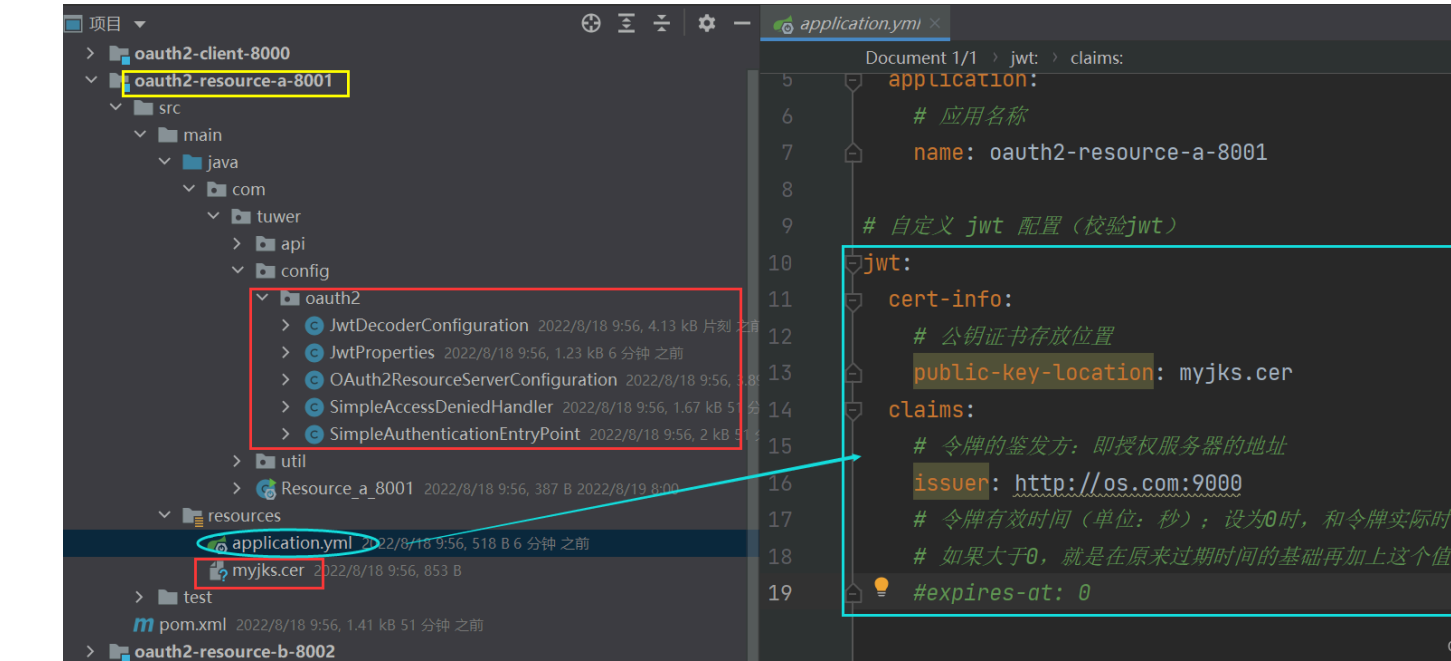
SpringBoot 同时被 2 个专栏收录

20 订阅 39 篇文章

一、前言

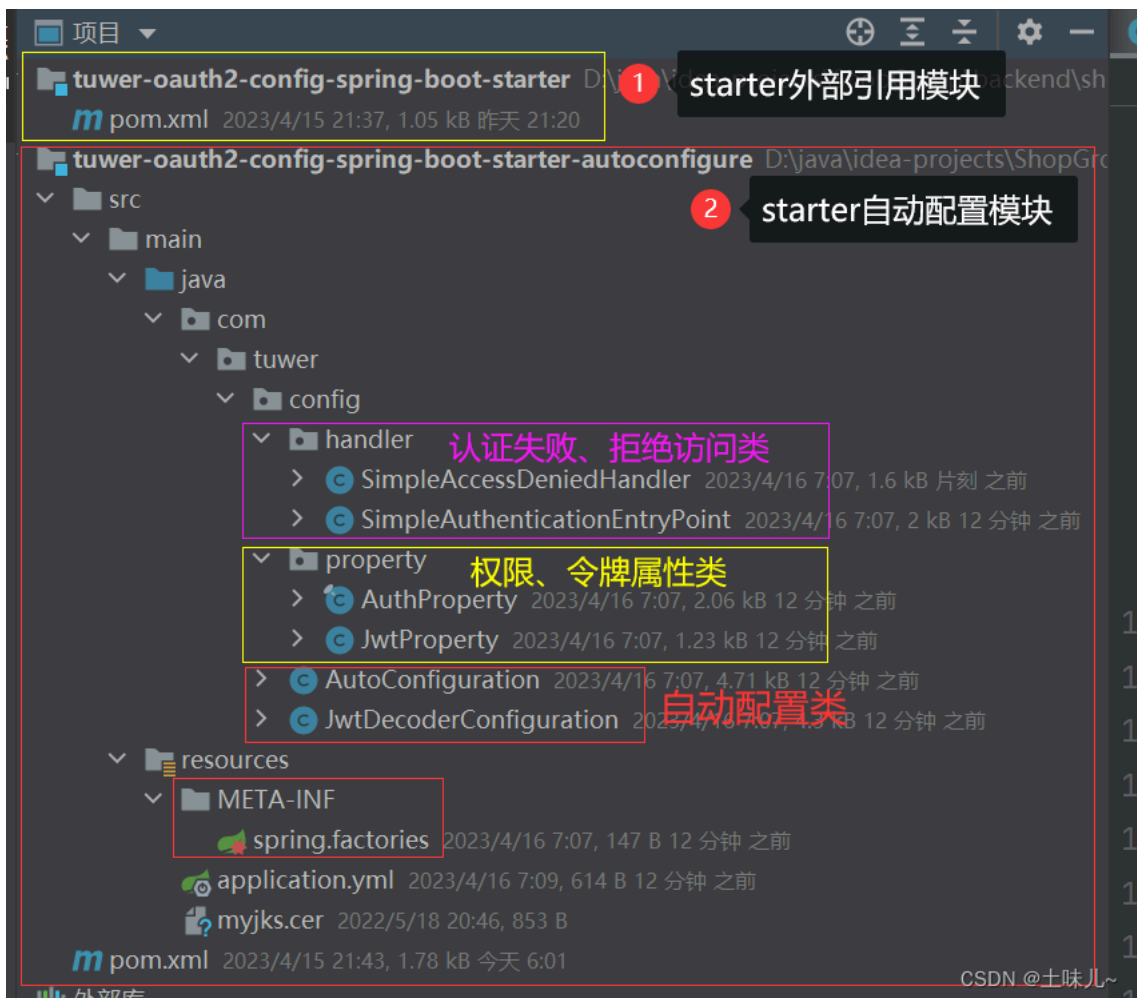
Oauth2中的资源服务Resource需要验证令牌，就要配置令牌的解码器JwtDecoder，认证服务器的公钥等等。如果有多个资源服务Resource，就要重复配置。把公共的配置信息抽取出来，制成starter，可以极大地简化操作。

- 未使用starter的原来配置





## 1、完整结构图



## 2、外部引用模块

- 名称: `tuwer-oauth2-config-spring-boot-starter`
- 普通的 maven 项目
- 资源服务中引入该模块的依赖即可
- 模块中只有一个pom.xml文件, 其余的都可删除

### • pom.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5     <modelVersion>4.0.0</modelVersion>
6
7     <groupId>com.tuwer</groupId>
8     <artifactId>tuwer-oauth2-config-spring-boot-starter</artifactId>
9     <version>1.0-SNAPSHOT</version>
10    <description>oauth2-config启动器</description>
11
```

## 3、自动配置模块

- 核心模块
- 名称: `tuwer-oauth2-config-spring-boot-starter-autoconfigure`



土味儿~

已关注

- spring boot 项目

- pom.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5     <parent>
6         <groupId>org.springframework.boot</groupId>
7         <artifactId>spring-boot-starter-parent</artifactId>
8         <version>2.6.7</version>
9     </parent>
10    <modelVersion>4.0.0</modelVersion>
11

```

- handler (拒绝访问、认证失败) 处理类

```

1 package com.tuwer.config.handler;
2
3 import com.fasterxml.jackson.databind.ObjectMapper;
4 import lombok.SneakyThrows;
5 import org.springframework.http.MediaType;
6 import org.springframework.security.access.AccessDeniedException;
7 import org.springframework.security.web.access.AccessDeniedHandler;
8
9 import javax.servlet.ServletException;
10 import javax.servlet.http.HttpServletRequest;
11 import javax.servlet.http.HttpServletResponse;

```

```

1 package com.tuwer.config.handler;
2
3 import com.fasterxml.jackson.databind.ObjectMapper;
4 import lombok.SneakyThrows;
5 import org.springframework.http.MediaType;
6 import org.springframework.security.core.AuthenticationException;
7 import org.springframework.security.oauth2.server.resource.InvalidBearerTokenException;
8 import org.springframework.security.web.AuthenticationEntryPoint;
9
10 import javax.servlet.ServletException;
11 import javax.servlet.http.HttpServletRequest;

```

- property (权限、令牌) 属性类

权限属性类: `AuthProperty` , 通过application.yml来配置权限, 避免在自动配置类中以硬编码的形式写入权限

```

package com.tuwer.config.property;

1 import lombok.Data;
2 import org.springframework.boot.context.properties.ConfigurationProperties;
3 import org.springframework.util.CollectionUtils;
4
5

```



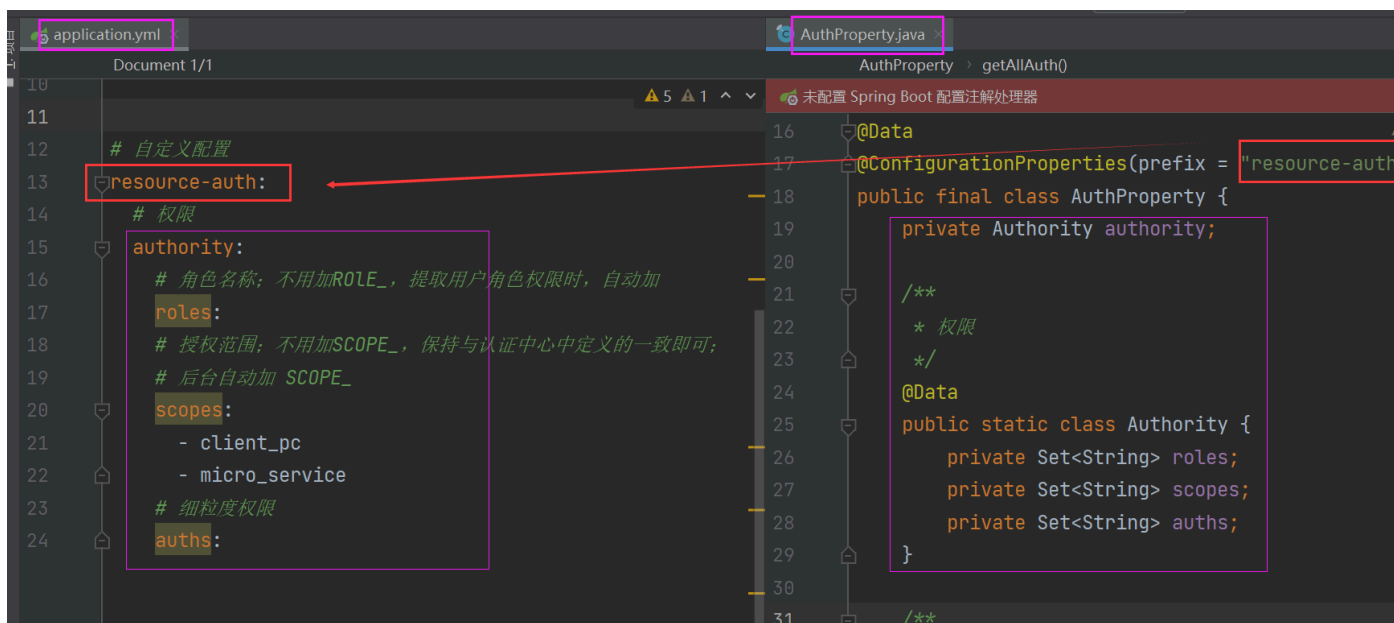
土味儿~

已关注

```

5
6
7 import java.util.Set;
8
9 /**
10  * <p>权限属性类</p>
11

```



```

1 package com.tuwer.config.property;
2
3 import lombok.Data;
4 import org.springframework.boot.context.properties.ConfigurationProperties;
5
6 /**
7  * <p>属性配置类</p>
8  *
9  * @author 土味儿
10  * Date 2022/5/11
11  * @version 1.0

```

```

1 package com.tuwer.config;
2
3 import com.nimbusds.jose.jwk.RSAKey;
4 import com.tuwer.config.property.JwtProperty;
5 import lombok.SneakyThrows;
6 import org.springframework.beans.factory.annotation.Qualifier;
7 import org.springframework.boot.context.properties.EnableConfigurationProperties;
8 import org.springframework.context.annotation.Bean;
9 import org.springframework.context.annotation.Configuration;
10 import org.springframework.context.annotation.Primary;
11 import org.springframework.core.io.ClassPathResource;

```



#### • 主自动配置类：安全权限等

```

1 package com.tuwer.config;
2
3 import com.tuwer.config.handler.SimpleAccessDeniedHandler;
4 import com.tuwer.config.handler.SimpleAuthenticationEntryPoint;
5 import com.tuwer.config.property.AuthProperty;
6 import org.springframework.boot.autoconfigure.condition.ConditionalOnBean;
7 import org.springframework.boot.context.properties.EnableConfigurationProperties;
8 import org.springframework.context.annotation.Bean;
9 import org.springframework.context.annotation.Configuration;
10 import org.springframework.security.config.annotation.web.builders.HttpSecurity;
11 import org.springframework.security.config.annotation.web.configuration.WebSecurityCustomizer;

```



#### • spring.factories

指明自动配置类的地址，在 `resources` 目录下编写一个自己的 `META-INF\spring.factories`；有两个自动配置类，中间用逗号分开

#### 注意点：

如果同一个组中有多个starter，自动配置类名称不要相同；如果相同，将只有一个配置类生效，其余的将失效。

如：

- starterA中：com.tuwer.config.AutoConfiguration
- starterB中：就不要再使用 com.tuwer.config.AutoConfiguration 名称，可以改为 com.tuwer.config.AutoConfigurationB

```

1 org.springframework.boot.autoconfigure.EnableAutoConfiguration=\
2   com.tuwer.config.JwtDecoderConfiguration,\
3   com.tuwer.config.AutoConfiguration

```

#### • application.yml

令牌、权限的配置可以放在引用starter的资源服务中；如果每个资源服务的配置都一样，可以放在starter中

```

# 自定义 jwt 配置 (校验jwt)
jwt:
1 cert-info:
2   # 公钥证书存放位置
3   public-key-location: myjks.cer
4 claims:
5   # 令牌的签发方: 即授权服务器的地址
6   issuer: http://os.com:9000
7
8

```



已关注

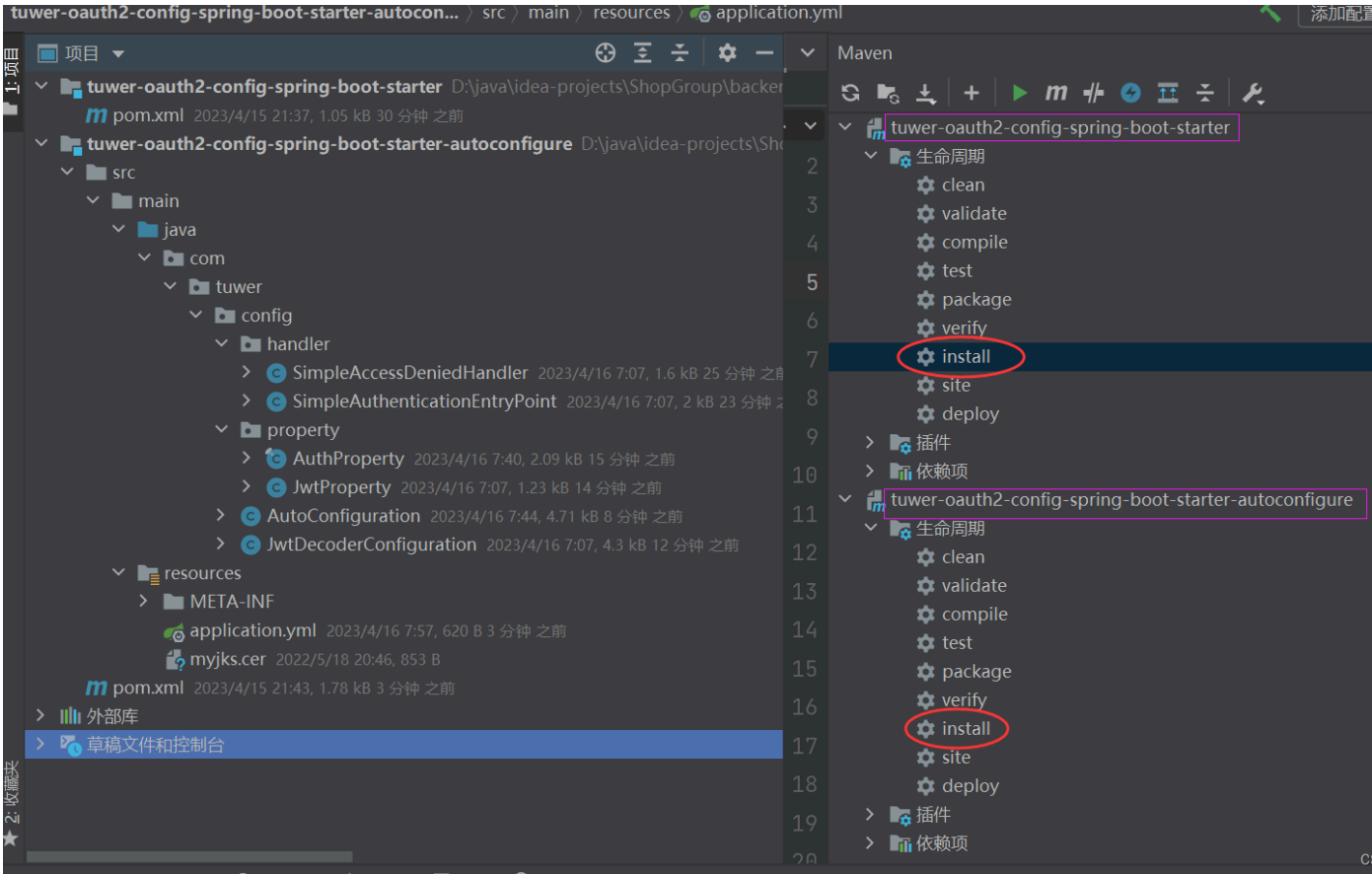
```
9 |
10 |
11 | # 自定义权限配置
```

- 公钥

把认证中心的公钥文件 myjks.cer 放到resources目录下

4、install

把starter安装install到本地maven仓库中



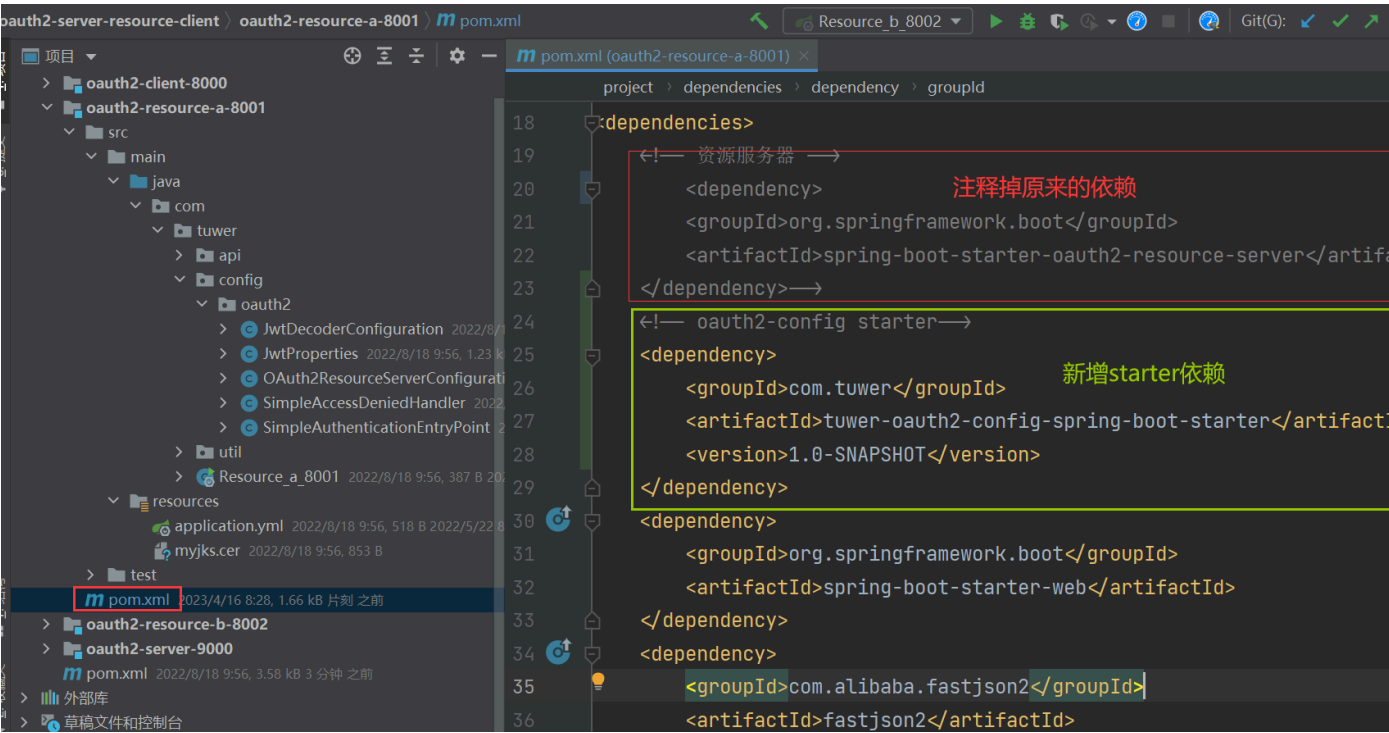
« apache-maven-3.8.1 » maven-repo » com » tuwer				在 tuwer 中搜索	
名称	修改日期	类型			
java_web	2021/12/9 18:28	文件夹			
mylog-spring-boot-starter	2022/1/26 19:59	文件夹			
mylog-spring-boot-starter-autoconfigure	2022/1/26 19:29	文件夹			
sg-common-base	2023/3/3 16:01	文件夹			
sg-common-security	2023/3/3 16:01	文件夹			
tuwer-mybatis-plus-spring-boot-starter	2023/4/15 21:31	文件夹			
tuwer-mybatis-plus-spring-boot-starter-autoconfigure	2023/4/15 21:31	文件夹			
tuwer-oauth2-config-spring-boot-starter	2023/4/16 8:06	文件夹			
tuwer-oauth2-config-spring-boot-starter-autoconfigure	2023/4/16 8:05	文件夹			
tuwer-spring-boot-starter	2022/1/26 20:49	文件夹			
tuwer-spring-boot-starter-autoconfigure	2022/1/26 20:49	文件夹			



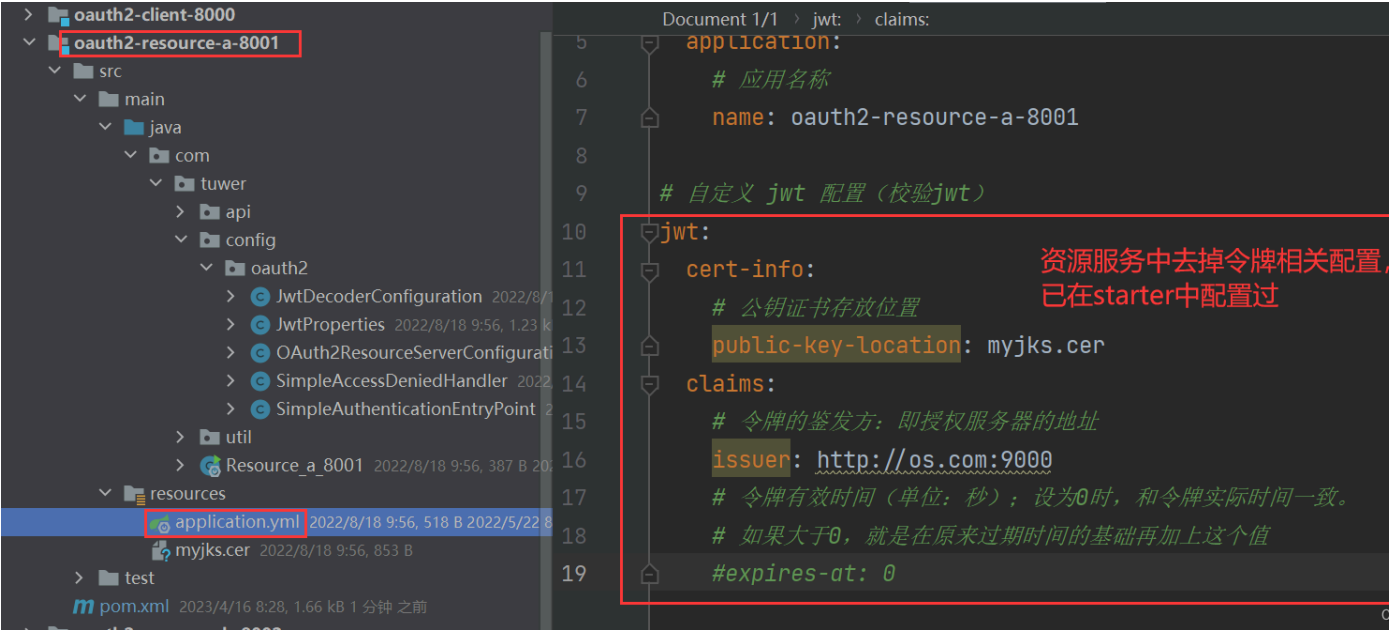
### 三、使用starter

#### 1、引入starter依赖

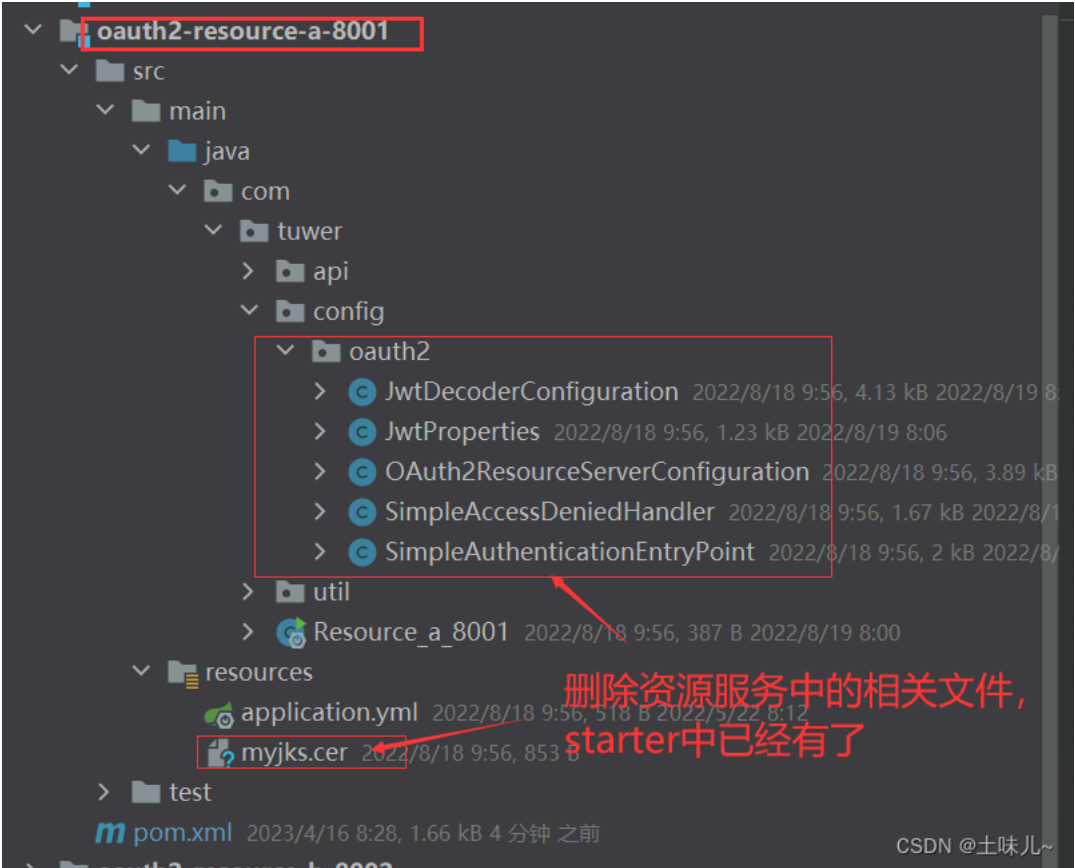
在资源服务中引入 `tuwer-oauth2-config-spring-boot-starter`



#### 2、application.yml



#### 3、删除资源服务中原文件



文章知识点与官方知识档案匹配，可进一步学习相关知识

Java技能树 首页 概览 149710 人正在系统学习中

云服务器ECS：云服务器直降 最高93%降幅

>>了解详情

...Security OAuth2-资源模块配置\_resourceServerConfig extends ResourceServerConfigurerAdapter并添加相关注解: @Configuration @EnableResourceServer:资源服务器 @EnableGlobalMethodSecurity

Oauth2 resource server入门配置\_oauth2ResourceServer  
Oauth2 resource server入门配置 这篇博客介绍了OAuth2.0资源服务器的配置过程,包括pom.xml和application.yml的设置,是OAuth2.0 Authorization Server配置的后续部分

Spring Boot2.0 OAuth2 服务器和客户端配置及理 weixin\_28718769的  
一、应用场景 为了理解OAuth的适用场合,让我举一个假设的例子。有一个“云冲印”的网站,可以将用户储存在Google的照片,冲印出来。用户为了使用该服务,必须让

Spring Security OAuth2 Redis 资源服务器配置 OceanSky的i  
1.资源服务器相关依赖 <dependency> <groupId>org.springframework.boot</groupId> <artifactId>spring-boot-starter-security</artifactId> </de...

...OAuth2.0 统一认证、资源分离的配置,用于分布式架构、模块化开发的...  
之前用 SpringSecurity 做了一个基于 RBAC 的权限管理系统的基础配置,所以对 SpringSecurity 算是比较了解了,于是 OAuth2.0的实现,也想用 SpringSecurity 的来做,心想

oauth2资源服务配置及源码解析\_missing tokeninfo and userinfo...  
首先,oauth2会检查头部或查询参数是否携带access\_token,没有则使用本地安全配置投票查看是否可以访问对应资源,如果存在则从授权服务指定地址获取认证信息,然后检

Spring Security OAuth2 JWT资源服务器配置 OceanSky的i  
1.POM相关依赖 <dependency> <groupId>org.springframework.boot</groupId> <artifactId>spring-boot-starter-security</artifactId> </depe...

oauth2-resource-server授权配置介绍 言午玉  
也就是授权 (token中存放用户名、授权信息), 接着资源服务器的token处理过程,可以直接请求授权服务器,由授权服务器验证;也可以在授权服务器确定固定的公钥和

OAuth2资源访问控制 OAuth2.0认证服务器配置和资源服务器配置(OAuth2....  
第二步:配置资源服务 创建一个资源类继承org.springframework.security.oauth2.config.annotation.web.configuration.ResourceServerConfigurerAdapter类。代码如下: pac

...2.0自定义资源服务核心配置——ResourceServer\_resourceServerConfig...  
实现ResourceServerConfigurerAdapter类重写方法自定义资源配置 上一篇写了授权服务器的配置,在Spring Sec

OAuth + Security - 2 - 资源服务器配置

详解Springboot Oauth2 Server搭建Oauth2认证服务

Spring Boot Oauth2 Server 是一个基于 Spring Boot 和 Spring Security 的 OAuth2 认证服务实现，它可以帮助开发者构建安全的 RESTful API 和 Web 应用。OAuth2 是一个广泛使用的开放标准，用于授权和认证。它允许第三方应用在不拥有用户密码的情况下访问用户数据。

springcloud-oauth2:本项目基于spring-cloud-starter-oauth2构建的认证中心和资源服务器的微服务项目，项目不仅仅简单的演示，项目的出发点在于实战应用。，仅需要稍作调整就可以部署到生产环境。

spring-cloud-security: Spring中实现的分布式应用程序的安全性问题

通过OAuth2，客户端应用程序可以获取访问令牌，用以访问受保护的资源，而无需知道用户的原始凭证。这增强了系统的安全性，因为敏感的用户信息不会在每个请求中传输。

resource-management:使用Spring Boot应用程序进行资源管理

本文将深入探讨如何使用Spring Boot来实现应用程序中的资源管理。首先，Spring Boot的核心是Spring框架，它引入了自动配置和“起步运行”（Starter）概念，大大简化了Spring应用的开发。

oauth2全套（授权服务器+资源服务器+客户端独立版）

独立的授权服务器，资源服务器，客户端单点登录系统。采用OAuth2进行token认证，模拟用户信息进行登录。

OAuth2：资源服务器

OAuth2：资源服务器

Leon\_Jinhai\_Sun的博客

Spring OAuth2 Resource Server 配置 最新发布

Spring OAuth2 Resource Server 是 Spring Security 中的一个模块，用于保护资源服务器上的API资源，确保只有持有合法访问令牌（access token）的客户端才能访问受保护的资源。

semicolon\_hello的博客

OAuth2资源服务器搭建

OAuth2资源服务器搭建 接下来我们搭建一个资源服务器。大家网上看到的例子，资源服务器大多都是和授权服务器放在一起的，如果项目比较小的话，这样做是没问题的。

Curtisjia的博客

配置OAuth2认证服务器和资源服务器，实现基于令牌的身份验证和授权

配置OAuth2认证服务器和资源服务器是实现基于令牌的身份验证和授权的关键步骤。创建一个WebSecurityConfigurerAdapter的子类，并覆盖configure(HttpSecurity http)方法。

qiuyehuanguan的博客

SpringSecurity(二十) ---OAuth2:实现资源服务器（上）资源服务器搭建以及直接调用授权服务器模式

本章将讨论如何使用Spring Security实现一个资源服务器，资源服务器是管理用户资源的组件。另外，学习本章有个前提，需要先把前面搭建授权服务器的相关文章先给阅读一遍。

学习不止境的博客

【SpringSecurity】十六、OAuth2.0授权服务器、资源服务器的配置（理论部分）

授权服务：负责校验接入的客户端、登录的用户账户是否合法，以及颁发token.资源服务：校验token，返回资源信息

llg\_的博客

十六 OAuth2.0实战--搭建资源服务器

OAuth2.0 资源服务器

专搞技术的小强

Spring Security Oauth2资源服务器Starter配置

首先开下starter的结构： | |--oauth2 | |--auth-center-provider 认证中心 | |--auth-provider-api 共用 | |--auth-spring-boot-autoconfig auto | |--autoconfig

zhuwei\_clark的博客

oauth 资源服务器配置

要配置 OAuth 资源服务器，你需要进行以下步骤： 1. 获取 OAuth 2.0 服务器的访问令牌，这将于访问受保护的资源。这通常涉及到使用你的客户端凭据（客户端ID和客户端秘密）向授权服务器请求令牌。



土味儿~  
码龄3年 暂无认证

229  
原创

6729  
周排名

38万+  
总排名

36万+  
访问

  
等级

2909  
积分

1万+  
粉丝

435  
获赞

194  
评论

2109  
收藏





私信

已关注

AI圈早知道，每日最新动态  
了解全球AI新鲜事！

立即参与

大额流量券免费送

发布一篇就可获得！




去查看

搜博主文章

热门文章

- 哈夫曼树、哈夫曼编码与压缩比 22635
- UML 建模步骤 用例图 类图 对象图 包图 顺序图/时序图 状态图 活动图 协作图 19865
- 图解PV操作 19754
- 【图文详解】搭建 Spring Authorization Server + Resource + Client 完整Demo 14578
- 图解模块间7种耦合关系 11795

分类专栏

	Ubuntu	4篇
	ElasticSearch	6篇
	前端	7篇
	Bootstrap	5篇
	Spring	5篇
	SpringCloud	25篇

最新评论

- 【图文详解】搭建 Spring Authorization ...  
笔墨画卷: 你这个是springcloud项目吗？前后端分离下的实现方案吗？不太理解这’ ...
- 【图文详解】搭建 Spring Authorization ...  
emapjan: 感谢作者，我在一个节点卡主了，登录成功获取用户信息的时候，Autl ...
- 【图文详解】搭建 Spring Authorization ...  
锦瑟-华年: 访问资源的时候只带了token并没有refreshtoken，确定可以自动刷新to ...
- Java NIO网络编程深入剖析【尚硅谷】  
南风Candy: FileChannelDemo1 这个demo中 为什么没有写入的操作 还要进行读写 ...
- 图解模块间7种耦合关系  
m0\_70295566: 大佬讲的太清晰了，能再讲讲内聚吗

最新文章

- SpringBoot中java操作Excel【EasyExcel】
- Nacos的安装与使用（SpringCloud）
- 限流与令牌桶

 土味儿~

已关注

2023年 19篇      2022年 126篇  
2021年 84篇



目录

- 一、前言
- 二、制作starter
  - 1、完整结构图
  - 2、外部引用模块
  - 3、自动配置模块
  - 4、install
- 三、使用starter
  - 1、引入starter依赖
  - 2、application.yml
  - 3、删除资源服务中原文件