

土味儿~

已关注

【oauth2 客户端模式】Spring Authorization Server + Resource + Client 资源服务间的相互访问

土味儿~ 已于 2023-04-16 08:54:31 修改 阅读量4.8k 收藏 17 点赞数 2

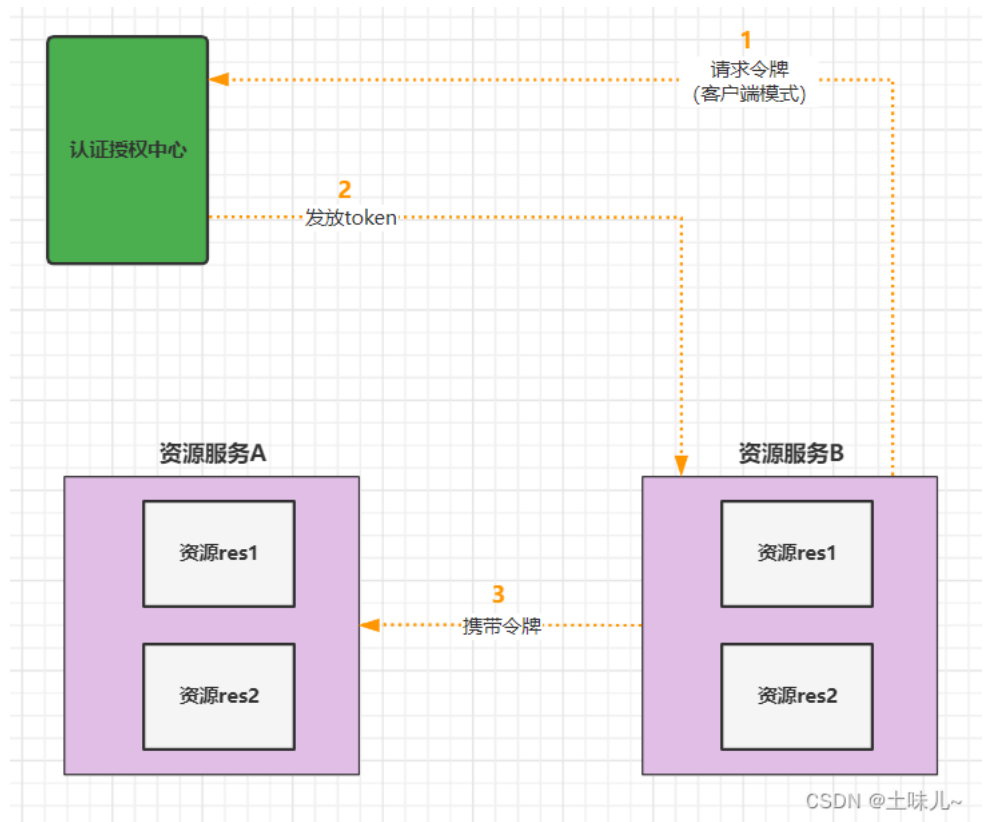
专栏：[# SpringSecurity OAuth2](#) [# SpringCloud](#) [# SpringBoot](#) 文章标签：[spring](#) [java](#) [http](#)

SpringBoot 同时被 3 个专栏收录

20 订阅

中介绍了项目的搭建，并实现了授权码模式的访问。在上一节的基础上，再来实现客户端模式。【图文详解】搭建 Spring Authorization Server + Resource + Client 资源服务间的相互访问是 授权码模式

服务(资源)间的访问是 客户端模式；客户端模式下，只需要提供注册客户端的ID和密钥，就可以向授权服务器申请令牌，授权服务器核实ID和密钥后，会直接发放令牌，无须再模块间的调用。



服务器中注册新客户端

请求资源的主体更加清晰，再注册一个客户端 `micro_service`，专门供资源服务器之间的相互调用。也可以用原来客户端 `my_client`，不过要在授权模式配置项 `client_credentials` 中配置。客户端模式直接返回token；不需要回调地址

又服务器的授权服务配置类 `AuthorizationServerConfiguration.java` 中添加

```
/**
 * 定义客户端（令牌申请方式：客户端模式）
 *
 * @param clientId 客户端ID
 * @return
 */
private RegisteredClient createRegisteredClient(final String clientId) {
    // JWT (Json Web Token) 的配置项: TTL、是否复用refreshToken等等
    TokenSettings tokenSettings = TokenSettings.builder()
        // 令牌存活时间: 1年
    }
```

土味儿~ 已关注

注册方法：该方法仅注重功能，结构不够优雅，可以自行修改

```
/**
 * 注册客户端
 *
 * @param jdbcTemplate 操作数据库
 * @return 客户端仓库
 */
@Bean
public RegisteredClientRepository registeredClientRepository(JdbcTemplate jdbcTemplate) {
    // ----- 1、检查当前客户端是否已注册
    // 操作数据库对象
    JdbcRegisteredClientRepository registeredClientRepository = new JdbcRegisteredClientRepository(jdbcTemplate);
}
```

服务器之间访问

说明

服务器B 调用 资源服务器A 中的资源；

服务器B/res1 --> 服务A/res2；

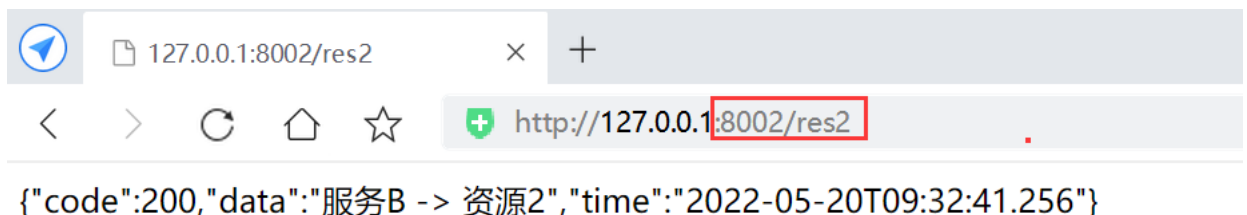
2 接口在前端用 my_client 是无法访问的；

```
HttpSecurity
// 对 /res1 的请求，需要 SCOPE_read 权限
authorizeRequests() ExpressionUrlAuthorizationConfigurer<...>.ExpressionInterceptUrlRegistry
.antMatchers( ...antPatterns: "/res1").hasAnyAuthority( ...authorities: "SCOPE_read", "SCOPE_write1")
.antMatchers( ...antPatterns: "/res2").hasAnyAuthority( ...authorities: "SCOPE_write1", "SCOPE_read")
// 其余请求都需要认证
.anyRequest().authenticated()
HttpSecurity
```

资源服务器A 中资源的权限要求

/res2 需要有 write1 或 all 权限，用客户端模式申请的令牌，具有 all 权限

服务器B 无安全策略，可以直接访问



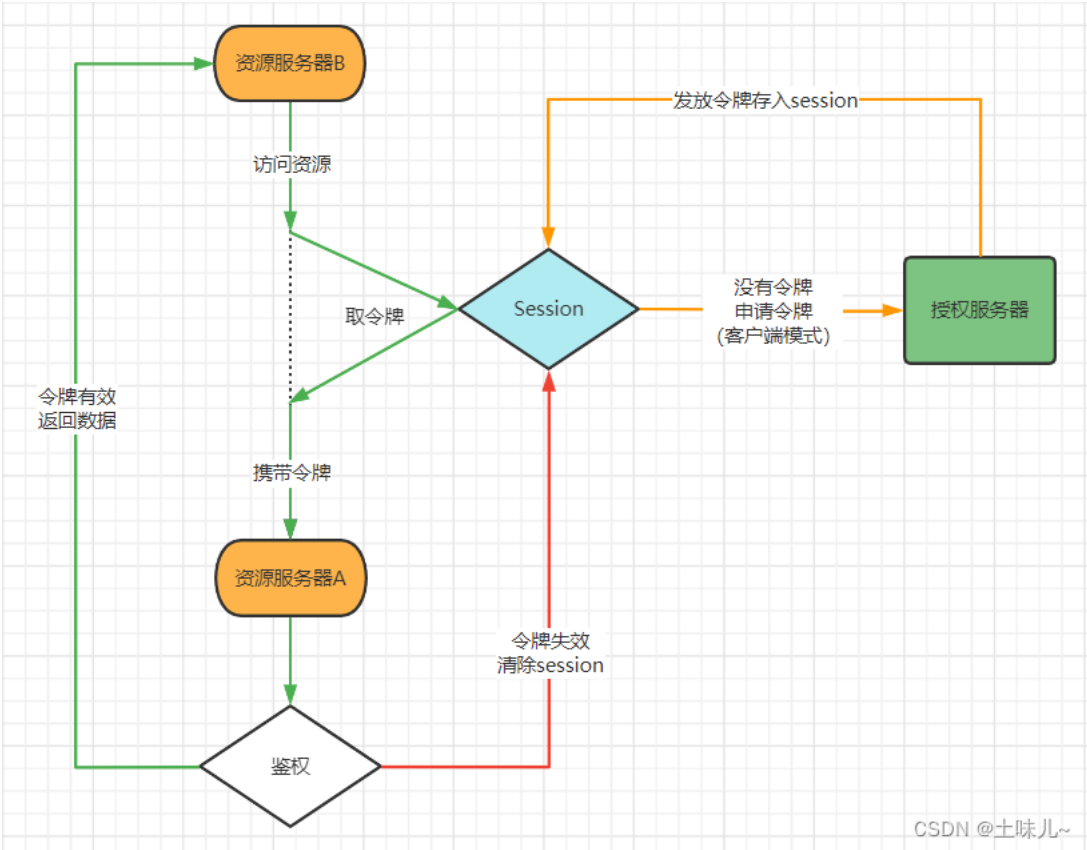
127.0.0.1:8002/res2

http://127.0.0.1:8002/res2

{"code":200,"data": "服务B -> 资源2", "time": "2022-05-20T09:32:41.256"}

CSDN @土味儿~

申请与使用 处理逻辑



资源服务器B

restTemplat

```
onfiguration(proxyBeanMethods = false)
blic class RestTemplateConfiguration {
    @Bean
    public RestTemplate oauth2ClientRestTemplate(RestTemplateBuilder restTemplateBuilder) {
        return restTemplateBuilder.build();
    }
}
```

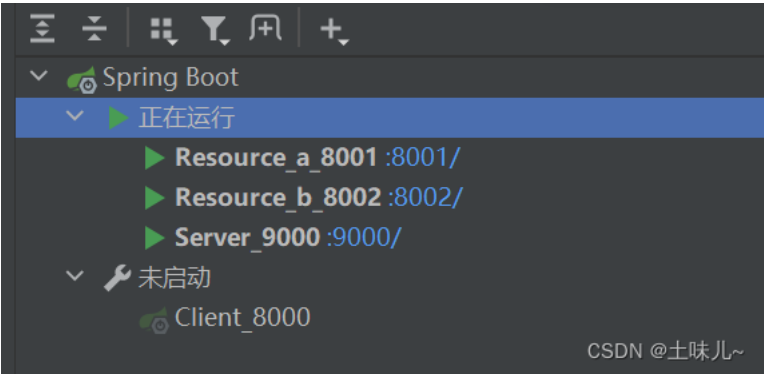
API接口类

```
estController
blic class ResourceController {
    @Autowired
    RestTemplate restTemplate;

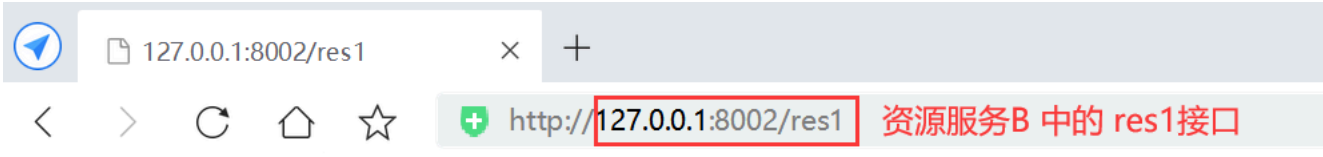
    @GetMapping("/res1")
    public String getRes1(HttpServletRequest request) {
        // 调用资源服务器A中的资源res2
        return getServer("http://127.0.0.1:8001/res2", request);
        //return JSON.toJSONString(new Result(200, "服务B -> 资源1"));
    }
}
```

式

erver、resource，无须启动client



访问resource_b



成功访问到了 资源服务A 中的 资源2

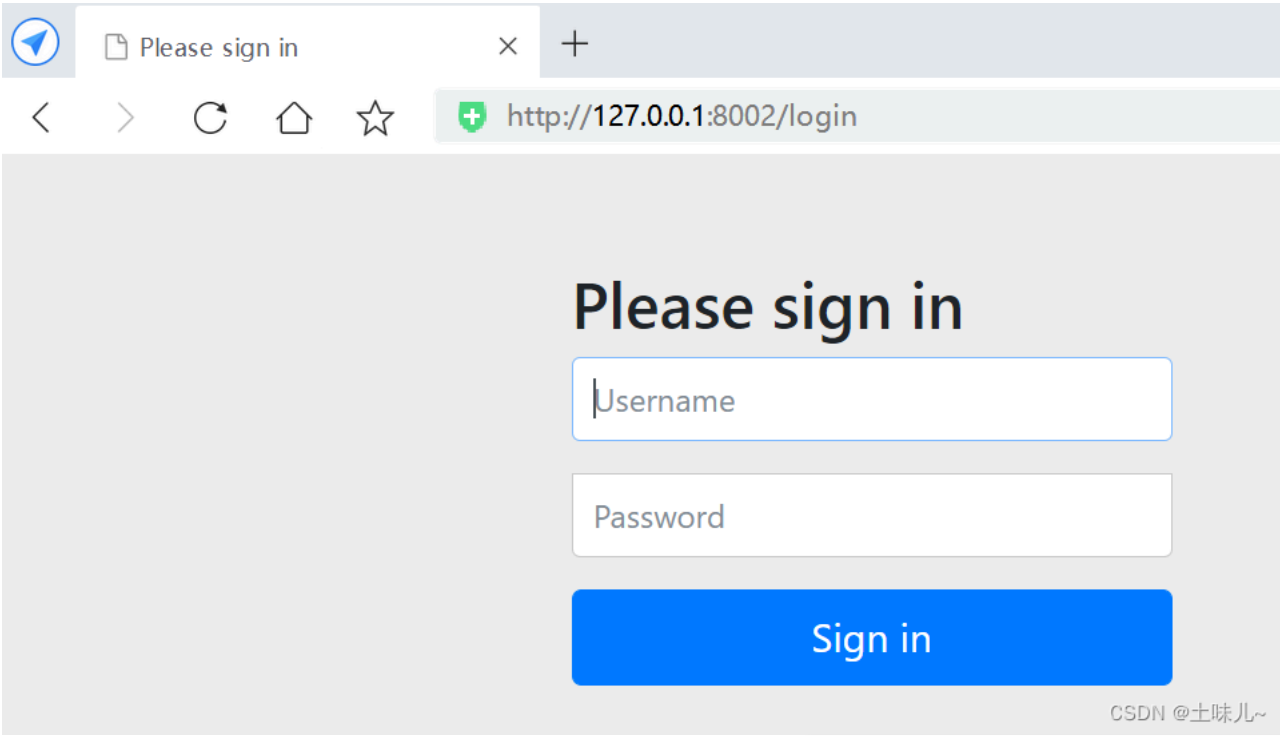
CSDN @土味.

服务器B添加安全策略

依赖

```
-- 资源服务器 -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-oauth2-resource-server</artifactId>
</dependency>
```

手动测试，已经无法直接访问；需要通过客户端去访问



改造资源服务器B

土味儿~ 已关注

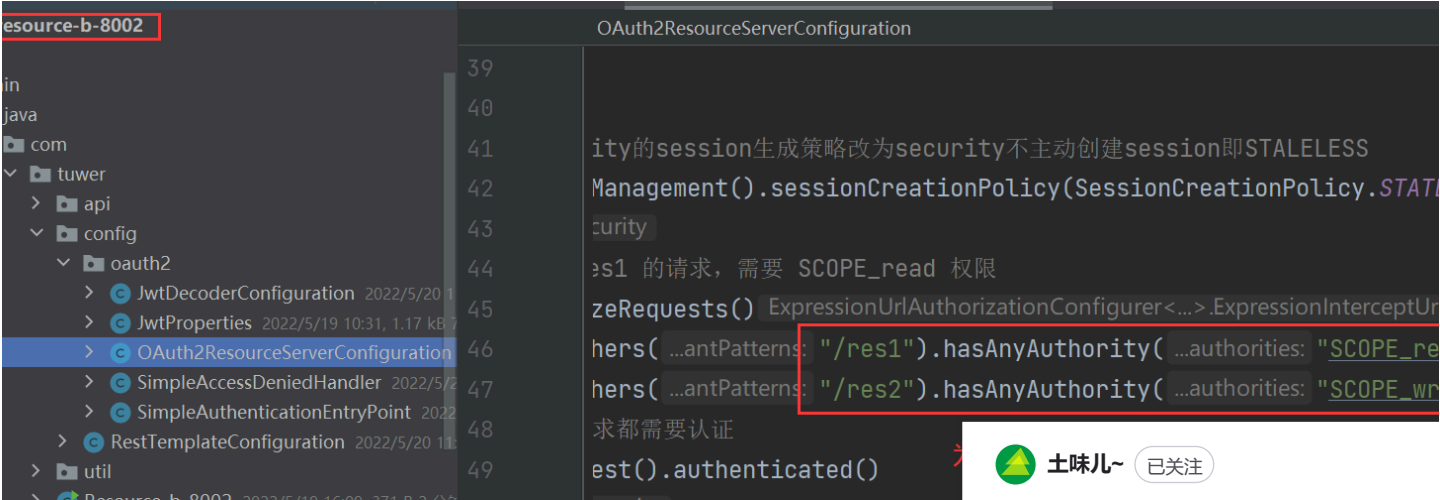
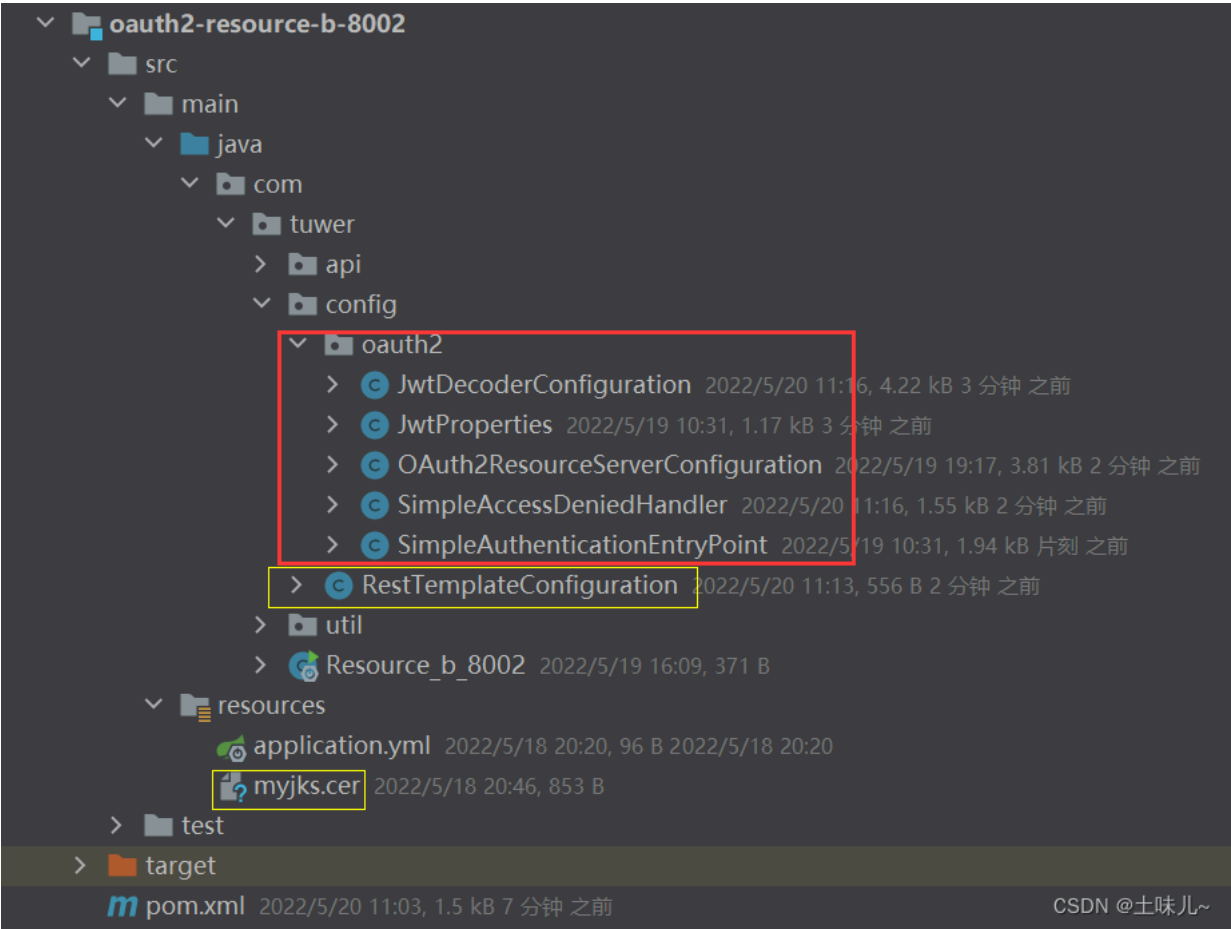
源服务器A 中配置策略到 资源服务器B 中来

er公钥文件

tion.yml中添加jwt配置

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

oauth2 配置包; 如下图



端client访问测试

iven 的 clean 清理项目
server、resource (a和b)、client



客户端

⌕ Title

× +

< > ↺ 🏠 ☆

+ http://127.0.0.1:8000/

登录用户: {"name":"admin","authorities":["ROLE_USER","SCOPE_read","SCOPE_write"]}

退出

这里登录的是 用户admin

- [服务A —— 资源1](#)
- [服务A —— 资源2](#)
- [服务B —— 资源1](#)
- [服务B —— 资源2](#)

CSDN @土味儿~

资源服务A

⌕ 127.0.0.1:8000/server/a/res1

× +

< > ↺ 🏠 ☆

+ http://127.0.0.1:8000/server/a/res1

{"code":200,"data":"服务A -> 资源1","time":"2022-05-20T11:31:11.711"}

访问成功；请求主体是：my_client

CSDN @土味儿~

⌕ 127.0.0.1:8000/server/a/res2

× +

< > ↺ 🏠 ☆

+ http://127.0.0.1:8000/server/a/res2

{"msg":"拒绝访问","uri":"/res2"}

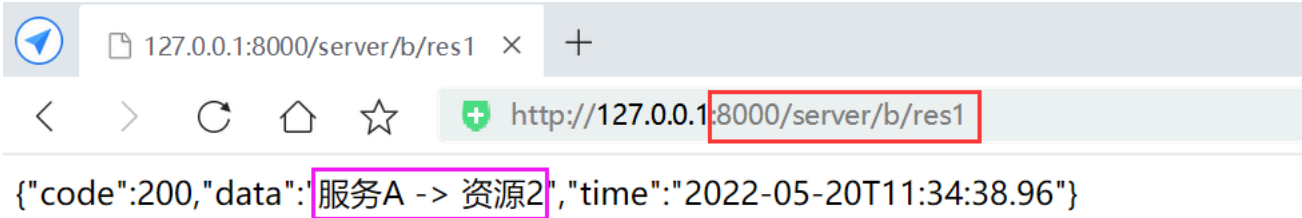
访问失败；请求主体是：my_client，没有访问 res2 的权限

CSDN @土味儿~

资源服务B

土味儿~

已关注



访问成功；请求主体是：my_client，具有 资源服务B 中 res1接口 的访问权限

资源服务B 中的 res1接口 又调用了 A 中的 res2，这里的请求主体是：micro_server，它具体 all 权限，可以访问

my_client --> 服务B/res1 --> 服务A/res2
my_client 间接实现了对 服务A/res2 的访问

CSDN @土味儿~



访问失败；请求主体是：my_client，没有权限

CSDN @土味儿~

要 资源服务器A 调用 B 中资源；可以把 B 中的实现逻辑复制过去就行。
把资源服务器中的公共部分抽离出来，制成starter...

16：用starter实现Oauth2中资源服务的统一配置

ECS：云服务器直降 最高93%降幅

...



笑看夕阳红 热评 拿到token有什么用？也不能访问

- 配置详解,百度java面试难度_registeredclientrepository...
steredClientRepository registeredClientRepository(JdbcTemplate jdbcTemplate) { return new JdbcRegisteredClientRepository(jdbcTemplate); } 别忘记调用save(RegisteredClient)
- 核心组件和内置 Filter_registeredclientrepository...
licRegisteredClientRepositoryregisteredClientRepository(JdbcTemplatejdbcTemplate){RegisteredClientregisteredClient=RegisteredClient.withId(UUID.randomUUID()).toString()
- horization-server 公共客户端方式获取授权码和Token的流程
orization-server 公共客户端 (public clients)方式获取授权码和Token的流程
- ion server && client && resource 使用1
en Authorization) 是一个关于授权 (authorization) 的开放网络标准，允许用户授权第三方应用访问他们存储在另外的服务提供者上的信息，而不需要将用户名和密码提供给第
- uth2 授权服务器配置详解_oauth2配置资源服务
了,那自然需要操作该表的JDBC服务接口了,这个接口为RegisteredClientRepository。我们需要声明一个实现为Spring Data JPA 这里选择基于 JPA 的实现 @Repository public class RegisteredClientRepository implements RegisteredClientRepository {
- urity6+oauth2 实现数据库配置客户端_spring security6 oauth2...



土味儿~

已关注

teredClientRepository; } 数据库脚本(样例为sqlserver,其它脚本需自行调整) CREATE TABLE oauth2_registered_client (id nvarchar(100) NOT NULL, client_id nvarchar(100) NC

实践 Spring Authorization Server 搭建授权服务器 + Resource + Client

实践 Spring Authorization Server 搭建授权服务器 + Resource + Client

【搭建 Spring Authorization oauth2-server-resource-client-gateway-eureka 完整Demo

1.1、概述 本项目是在前面章节的基础上，进行的升级改造。增加了注册中心、网关，更加贴近于实际需求。 在进行本节之前，请先搭建前面项目：【图文详解】搭建 Spring

服务Spring Security OAuth 2实战_spring-boot-starter-oauth...

ClientRepository:主要用于管理客户端 JwksSource:用于签名访问令牌 KeyPair: 启动时生成的带有密钥的KeyPair实例,用于创建上面的JwksSource JwtDecoder:JwtDecoder的一

字存储改为数据库存储和redis存储(主要是授权服务端的用户信...

teredClientRepository A1.修改用户登录信息的持久化为MySQL 思考一下,这个用户登录信息持久化,只有pigDaoAuthenticationProvider类会涉及到,他需要调用this.userDetailsS

OAuth2的三种模式的实践 (授权码模式, 密码模式, 客户端模式)

纪大了，写一篇文章后简直就是老眼昏花了，OK，言归正传，上篇文章写了怎么部署springOAuth2以及个人对这方面知识的简单理解，传送门在这里https://blog.csdn.net/qq_2

（授权服务器+资源服务器+客户端独立版）

中，它实现了OAuth2的四个主要授权类型：授权码（Authorization Code）、密码（Resource Owner Password Credentials）、刷新令牌（Refresh Token）和客户端凭据（Cli

ults()和registeredclientreposito

RegisteredClientRepository; import org.springframework.security.oauth2.server.authorization.client.RegisteredClient; import org.springframework.security.oauth2.server.authori

认证方式之 client_secret_jwt

tion_type:固定值urn:ietf:params:oauth:client-assertion-type:jwt-bearer client_assertion:client生成的jwt 环境准备 授权服务器 同样的,基于快速搭建一个授权服务器文章中的示例

curity+OAuth2 精讲，打造企业级认证与授权

色包括资源所有者（Resource Owner）、客户端（Client）、授权服务器（Authorization Server）和资源服务器（Resource Server）。通过授权码流程、密码模式、客户端凭

nt:使用Spring实现基本的OAuth2客户端

个主要角色：资源所有者（Resource Owner）、资源服务器（Resource Server）、客户端（Client）和授权服务器（Authorization Server）。 2. **Spring Security OAuth2**:

gisteredClient_jdbcregisteredclientrepository

RegisteredClientRepository,它和InMemoryRegisteredClientRepository只能二选一,所以需要注释掉后者。在咱们自己的配置类OAuth2AuthorizeSecurityConfig中: @Beanpublicl

curity OAuth2 授权码模式的实现

urity OAuth2 授权码模式的基本结构主要由三个部分组成：authorization-code-authorization-server、authorization-code-resource-server 和 authorization-code-client。authoriza

urity-oauth2-authorization-server.zip

ng-security-oauth2-authorization-server”将带你深入理解如何利用Spring Security OAuth2构建一个授权服务器，以保护你的API并提供安全的访问控制。 OAuth2是一种开放标准

OAuth2.0 - springsecurity + spring-authorization-server —客户端凭证模式

OAuth2.0 - springsecurity + spring-authorization-server —客户端凭证模式

【搭建 Spring Authorization Server + Resource + Client 完整Demo

Demo，有认证端，有资源端，有客户端；采用当前最新的技术。非常感谢 码农小胖哥，仔细研读了他的很多文章。本项目中的很多逻辑和代码都源自于他。如果想深入学习O

curity OAuth2-授权码模式(Finchley版本) 热门推荐

模式原理解析(来自理解OAuth 2.0) 授权码模式（authorization code）是功能最完整、流程最严密的授权模式。它的特点就是通过客户端的后台服务器，与“服务提供

curity OAuth专题学习-密码模式及客户端模式实例

g.csdn.net/icarusliu/article/details/87911093一文中，介绍了OAuth的一些背景知识；本文将编写一个简单的示例，演示授权模式中的密码模式及客户端模式如何实现。本示例

thorization Server 自定义授权模式

ng Authorization Server 授权模式 - 密码模式（password）

OAuth 2.0授权模式及应用场景

授权码模式和客户端凭证模式。 OAuth和OAuth 2.0都是为了解决第三方应用程序访问资源服务器（Resource Server）中的用户资源的问题。但是它们之间存在一些重要的区别

curity 6.x 系列【31】授权服务器篇之OAuth2AuthorizationService

中，我们完成了授权码、客户端、刷新令牌模式获取访问令牌案例，授权码作为最安全，流程最复杂的授权模式，我们需要深入学习和了解。

urity-oauth2-authorization-server 如何用

urity-oauth2-authorization-server是一个基于Spring Security的OAuth 2.0授权服务器，用于为客户端提供安全的访问资源的授权服务。您可以按照以下步骤使用该库： 1. 在您的S

关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心

家长监护 网络110报警服务 中国互联网举报中心 Chrome商店下载 账号管理规范 版权与免责声明 版权申诉 出版物许可证 营业执照

©1999-2024北京创新乐知网络技术有限公司



土味儿~

已关注

土味儿~

3年 暂无认证

29

38万+

36万+

姓名

总排名

访问

等级

5+

435

194

2109

粉丝

获赞

评论

收藏

已关注

道，每日最新动态

新鲜事！

量券免费送

就可获得！

夫曼编码与压缩比

22635

聚 用例图 类图 对象图 包图 顺

状态图 活动图 协作图

19754

搭建 Spring Authorization

source + Client 完整Demo

种耦合关系

11795

itu	4篇
icSearch	6篇
	7篇
Bootstrap	5篇
ig	5篇
SpringCloud	25篇
▼	

- 搭建 Spring Authorization ...
这个是springcloud项目吗？前
实现方案吗？不太理解这’ ...
- 搭建 Spring Authorization ...
射作者，我在一个节点卡主
获取用户信息的时候，Aut ...
- 搭建 Spring Authorization ...

土味儿~ 已关注

访问资源的时候只带了token并
oken, 确定可以自动刷新to ...

各编程深入剖析【尚硅谷】
FileChannelDemo1 这个demo
与写入的操作 还要进行读写 ...

种耦合关系
i6: 大佬讲的太清晰了, 能再讲

java操作Excel【EasyExcel】
与使用 (SpringCloud)

第 2022年 126篇



- 器中注册新客户端
- 器之间访问
- 说明
- 申请与使用 处理逻辑
- 与资源服务器B
- 式
- 服务器B添加安全策略
- 改造资源服务器B
- 端client访问测试