



# Spring Authorization Server入门 (八) Spring Boot引入Security OAuth2 Client对接认证服务

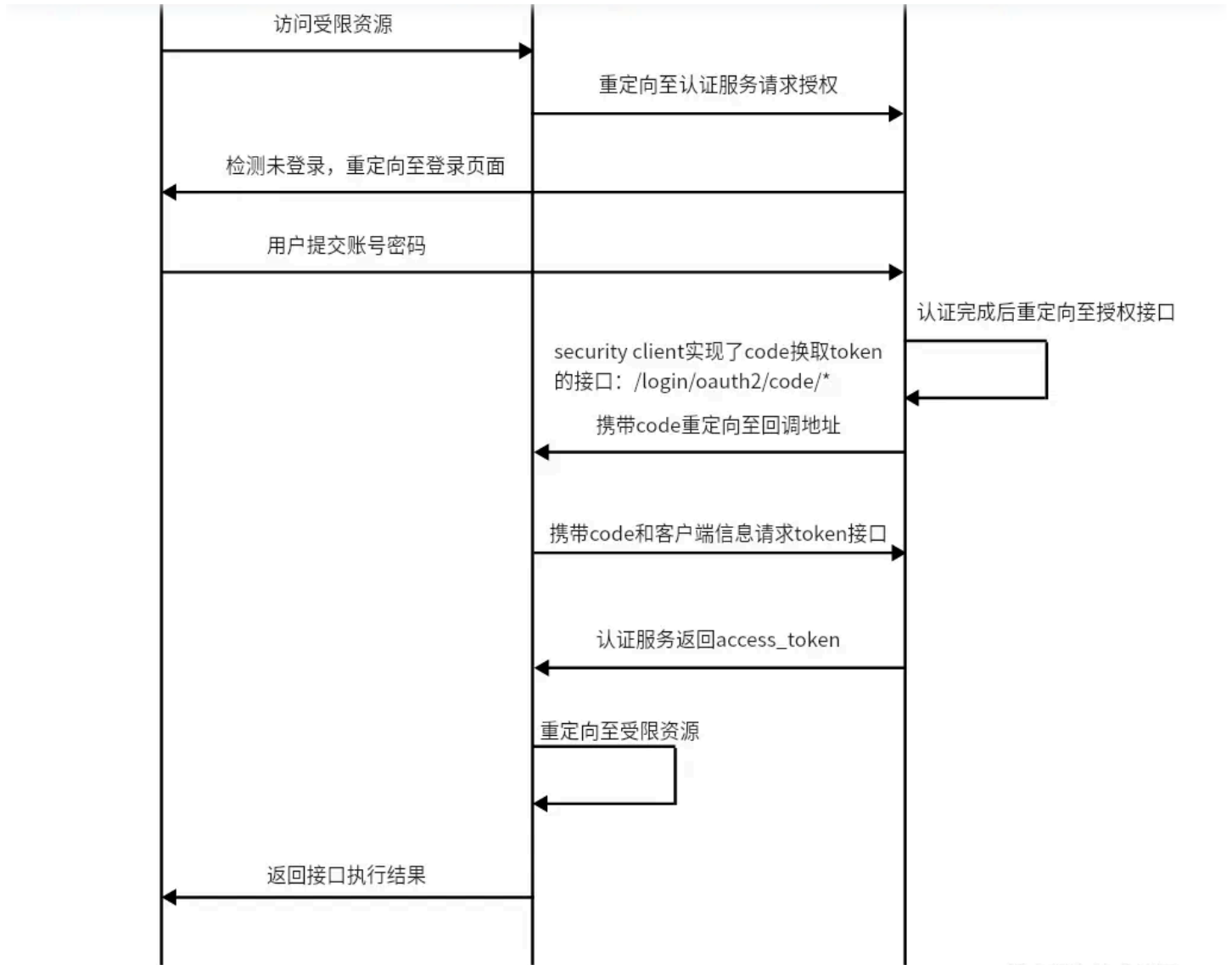
叹雪飞花 2023-06-12 2,753 阅读6分钟

关注

## 前言

在之前的文章中实现了一个认证服务，并且添加了一些自定义的内容，现在暂时没想到认证服务的新内容，本篇文章就先写一下客户端对接的吧。

## 流程说明



@稀土掘金技术社区

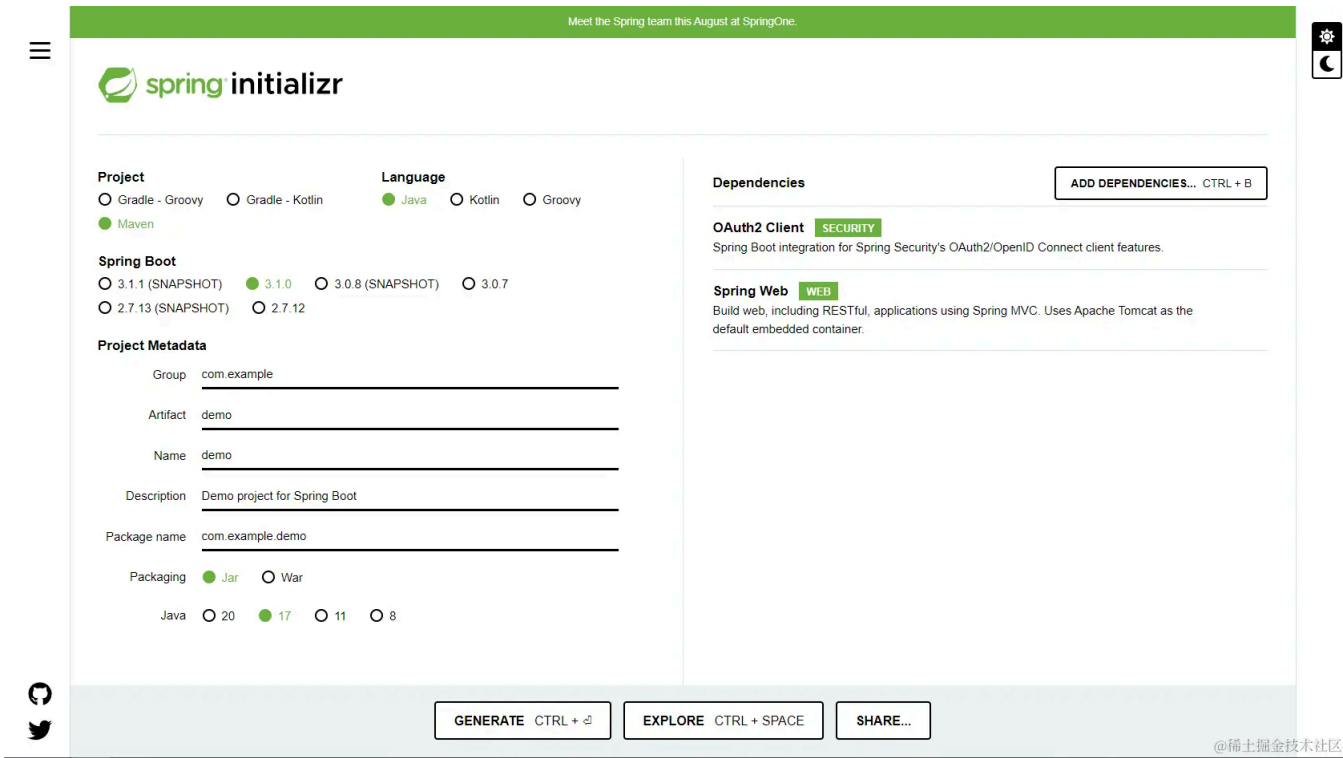
当用户通过客户端去访问一个受限的资源时，客户端会检测是否有登录信息，没有登录信息会重定向至认证服务器去请求授权，认证服务器会检测是否有登录信息(检查session)，检测到没有登录则重定向至登录页面返回给用户，用户输入账号密码后提交，认证服务器认证以后会重定向至授权接口，授权接口生成一个code之后携带code重定向至客户端配置的redirect\_uri，Security OAuth2 Client默认实现了一个处理回调的接口，会自动使用code获取token，地址为：/login/oauth2/code/\*，最后的\*要填配置客户端的registrationId，后边会提到；然后该接口请求认证服务去获取一个access\_token，用access\_token换取用户信息，框架会将token的信息存入session中，以后再发起请求时会从session中获取token。

## 使用SpringBoot创建一个oauth2客户端

客户端这里就是一个独立的项目了，跟之前的认证服务没有什么关联，读者可自选自己使用的Spring Boot版本，各版本的oauth2 client版本的对接大差不差，基本上差不多，可能实现会有所不同，但基本都一样的。在对接过程中我会放一些文档，读者可以去文档中找对应版本的文档去编写代码。

## 创建项目

使用idea或者在[Spring Initializr](#)创建一个SpringBoot项目；创建时引用oauth2-client和web依赖。



## pom.xml示例

▼

java 复制代码

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-i
3     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/mave
4     <modelVersion>4.0.0</modelVersion>
5     <parent>
6         <groupId>org.springframework.boot</groupId>
7         <artifactId>spring-boot-starter-parent</artifactId>
8         <version>3.1.0</version>
```

```
12     <artifactId>authorization-client-example</artifactId>
13     <version>0.0.1-SNAPSHOT</version>
14     <name>authorization-client-example</name>
15     <description>authorization-client-example</description>
16     <properties>
17         <java.version>17</java.version>
18     </properties>
19     <dependencies>
20         <dependency>
21             <groupId>org.springframework.boot</groupId>
22             <artifactId>spring-boot-starter-oauth2-client</artifactId>
23         </dependency>
24         <dependency>
25             <groupId>org.springframework.boot</groupId>
26             <artifactId>spring-boot-starter-web</artifactId>
27         </dependency>
28
29         <dependency>
30             <groupId>org.springframework.boot</groupId>
31             <artifactId>spring-boot-starter-test</artifactId>
32             <scope>test</scope>
33         </dependency>
34     </dependencies>
35
36     <build>
37         <plugins>
38             <plugin>
39                 <groupId>org.springframework.boot</groupId>
40                 <artifactId>spring-boot-maven-plugin</artifactId>
41             </plugin>
42         </plugins>
43     </build>
44
45 </project>
46
```

## 配置application.yml，添加认证服务器信息和客户端信息



yaml 复制代码

```
1 server:
2   # 修改端口
3   port: 8000
```

```
7      oauth2:
8        client:
9          provider:
10           # 认证提供者, 自定义名称
11           custom-issuer:
12             # Token签发地址(认证服务地址)
13             issuer-uri: http://192.168.120.33:8080
14             # 获取用户信息的地址, 默认的/userinfo端点需要IdToken获取, 为避免麻烦自定义一个用户信息接口
15             user-info-uri: ${spring.security.oauth2.client.provider.custom-issuer.issuer-uri}/userinfo
16         registration:
17           # registration Id, 自定义
18         messaging-client-oidc:
19           # oauth认证提供者配置, 和上边配置的认证提供者关联起来
20           provider: custom-issuer
21           # 客户端名称, 自定义
22           client-name: message-client
23           # 客户端id, 从认证服务申请的客户端id
24           client-id: messaging-client
25           # 客户端密钥
26           client-secret: 123456
27           # 客户端认证方式
28           client-authentication-method: client_secret_basic
29           # 获取Token使用的授权流程
30           authorization-grant-type: authorization_code
31           # 回调地址, 这里设置为Spring Security Client默认实现使用code换取token的接口
32           redirect-uri: http://127.0.0.1:8000/login/oauth2/code/messaging-client-oidc
33           scope:
34             - message.read
35             - message.write
```

更多详细的客户端配置信息移步[官网文档](#)

文档中配置权限范围的字段是 `scopes` , 但是在代码中是 `scope` , 这里可能是文档中没更新吧, 大家在参阅文档时需要注意一下代码中具体的属性名是什么。

注意: 认证服务器和客户端在同一个机器上时不能使用同一个ip, 例如127.0.0.1, 在存储cookie时不会区分端口的, 比如127.0.0.1:8000和127.0.0.1:8080这两个, 他们的cookie是同一个的, 后者会覆盖前者; 如果配置认证服务的地址是127.0.0.1:8080然后通过127.0.0.1:8000去访问客户端则会在登录后出现 `[authorization_request_not_found]` 异常, 详见[spring-security issues 5946](#)

## 给认证服务添加一个用户接口

```
1 @ResponseBody
2 @GetMapping("/user")
3 public Map<String,Object> user(Principal principal) {
4     if (!(principal instanceof JwtAuthenticationToken token)) {
5         return Collections.emptyMap();
6     }
7     return token.getToken().getClaims();
8 }
```

## 测试

综上所述，一个很简单的客户端已经配置完成了。

## 编写一个测试接口

直接将认证服务的测试接口copy过来

```
1 package com.example.controller;
2
3 import org.springframework.security.access.prepost.PreAuthorize;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.RestController;
6
7 /**
8  * 测试接口
9  *
10  * @author vains
11  */
12 @RestController
13 public class TestController {
14
15     @GetMapping("/test01")
16     @PreAuthorize("hasAuthority('SCOPE_message.read')")
17     public String test01() {
18         return "test01";
19     }
20
21     @GetMapping("/test02")
```



```
25     }
26
27     @GetMapping("/app")
28     @PreAuthorize("hasAuthority('app')")
29     public String app() {
30         return "app";
31     }
32
33 }
```

## 开始测试

## 1. 启动认证服务

```

      _   _          _   _
     / \   \         / \   \
    ( )   ( )       ( )   ( )
   W  __) ( )__  W  __) ( )__
  _  ( )_  _  ( )_  _  ( )_  _
=====|_|=====|_|_/___/_/

:: Spring Boot ::                (v3.1.0)


2023-06-12T16:36:14.850+08:00 INFO 92476 --- [main] c.e.AuthorizationExampleApplication : Starting AuthorizationExampleApplication using Java 17.0.5 with PID 92476 (D:\OtherFiles\lida\springboot3\AuthorizationExample\tomcat\bin\java.exe [jvmArgs=-Xmx1G -XX:+UseG1GC])
2023-06-12T16:36:14.865+08:00 INFO 92476 --- [main] c.e.AuthorizationExampleApplication : No active profile set, falling back to 1 default profile: "default"
2023-06-12T16:36:15.502+08:00 WARN 92476 --- [main] o.s.m.s.mapper.ClassPathMapperScanner : No MyBatis mapper was found in '[com.example]' package. Please check your configuration.
2023-06-12T16:36:15.861+08:00 INFO 92476 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2023-06-12T16:36:15.868+08:00 INFO 92476 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-06-12T16:36:15.868+08:00 INFO 92476 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.8]
2023-06-12T16:36:15.932+08:00 INFO 92476 --- [main] o.a.c.g.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2023-06-12T16:36:15.934+08:00 INFO 92476 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 987 ms
2023-06-12T16:36:16.225+08:00 INFO 92476 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2023-06-12T16:36:16.337+08:00 INFO 92476 --- [main] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Added connection com.mysql.cj.jdbc.ConnectionImpl@32a72c4
2023-06-12T16:36:16.338+08:00 INFO 92476 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2023-06-12T16:36:17.959+08:00 INFO 92476 --- [main] o.s.s.web.DefaultSecurityFilterChain : Will secure org.springframework.security.oauth2.server.authorization.config.annotation.web.c
2023-06-12T16:36:18.309+08:00 INFO 92476 --- [main] o.s.s.web.DefaultSecurityFilterChain : Will secure any request with [org.springframework.security.web.session.DisableEncodeUrlFilter

      _   _          _   _
     / \   \         / \   \
    ( )   ( )       ( )   ( )
   W  __) ( )__  W  __) ( )__
  _  ( )_  _  ( )_  _  ( )_  _
=====|_|=====|_|_/___/_/

3.5.3.1

2023-06-12T16:36:18.631+08:00 INFO 92476 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path '/'
2023-06-12T16:36:18.638+08:00 INFO 92476 --- [main] c.e.AuthorizationExampleApplication : Started AuthorizationExampleApplication in 4.267 seconds (process running for 5.173s)

```

## 2. 启动客户端服务

```

      ____ _
     / ___ \_/_ _/___\_____\__/\___\
    ( )_/___ \_/_ _/___\_____\__/\___\
     \___ \_/_ _/___\_____\__/\___\
      |___|_|_|_|_|_|_|_|_|_|_|_|_|_|
=====|=====|_/=/_/_/

:: Spring Boot ::                (v3.1.0)

2023-06-12T16:37:12.333+08:00 INFO 93872 --- [main] .e.AuthorizationClientExampleApplication : Starting AuthorizationClientExampleApplication using Java 17.0.5 with PID 93872 (G:\OtherFile
2023-06-12T16:37:12.335+08:00 INFO 93872 --- [main] .e.AuthorizationClientExampleApplication : No active profile set, falling back to 1 default profile: "default"
2023-06-12T16:37:13.044+08:00 INFO 93872 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8000 (http)
2023-06-12T16:37:13.052+08:00 INFO 93872 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-06-12T16:37:13.052+08:00 INFO 93872 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.8]
2023-06-12T16:37:13.118+08:00 INFO 93872 --- [main] o.s.c.o.C.[Tomcat].[/localhost]/[] : Initializing Spring embedded WebApplicationContext
2023-06-12T16:37:13.119+08:00 INFO 93872 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 735 ms
2023-06-12T16:37:13.609+08:00 INFO 93872 --- [main] o.s.s.web.DefaultSecurityFilterChain : Will secure any request with [org.springframework.security.web.session.DisableEncodeUrlFilter
2023-06-12T16:37:13.669+08:00 INFO 93872 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8000 (http) with context path ''
2023-06-12T16:37:13.675+08:00 INFO 93872 --- [main] .e.AuthorizationClientExampleApplication : Started AuthorizationClientExampleApplication in 1.657 seconds (process running for 2.495

```

### 3. 访问客户端的/app接口



机制说明

看到这里可能有些读者会比较疑惑，比如框架怎么知道认证服务器授权接口的地址？框架怎么知道认证服务获取token的接口？

项目在初始化时会根据配置的issuer-uri拼接url，即签发地址根目录/.well-known/openid-configuration，像项目中配置的就是<http://192.168.120.33:8080/.well-known/openid-configuration>；该接口会返回认证服务器的元信息，如下：



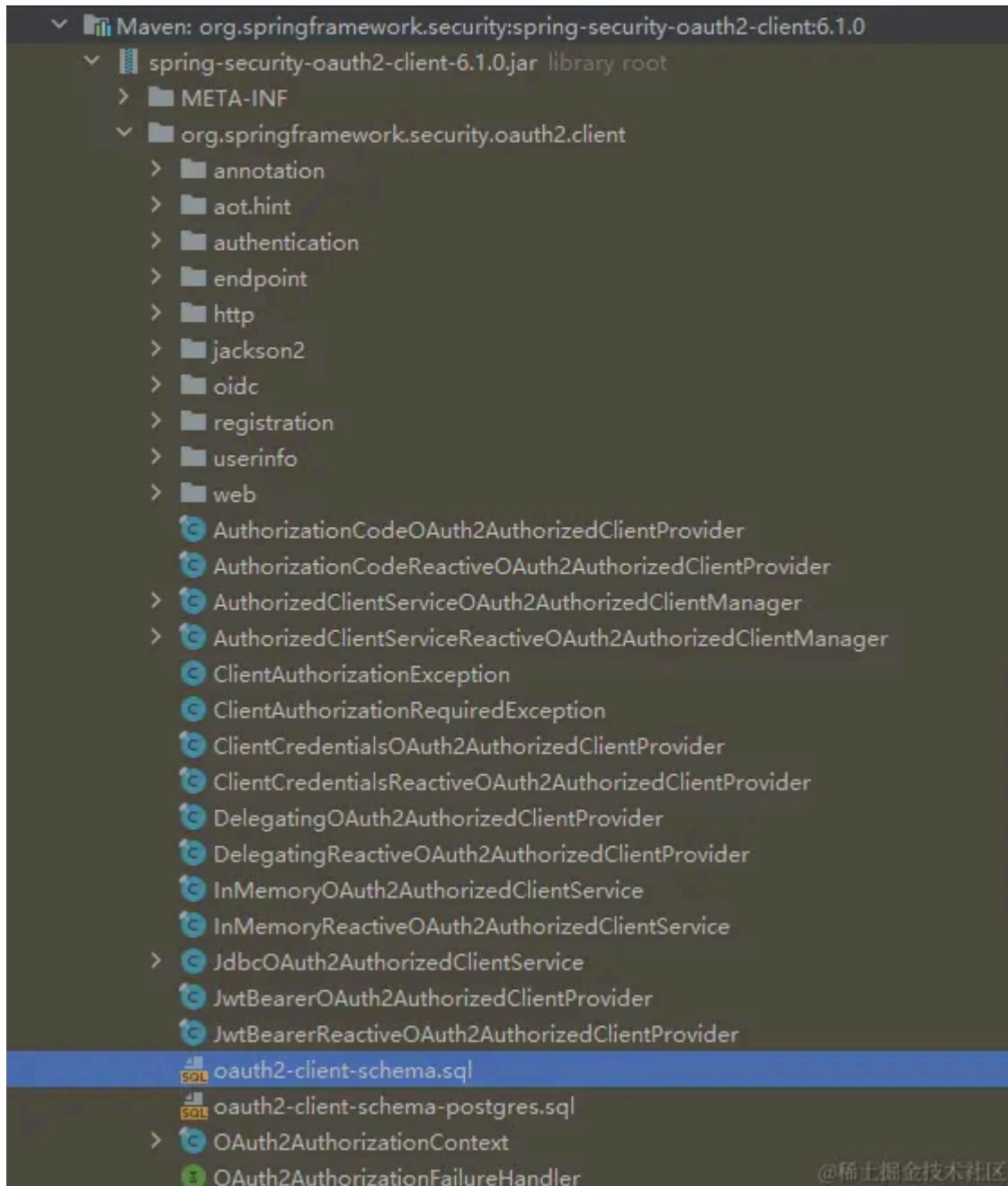
```
{
  "issuer": "http://192.168.120.33:8080",
  "authorization_endpoint": "http://192.168.120.33:8080/oauth2/authorize",
  "device_authorization_endpoint": "http://192.168.120.33:8080/oauth2/device authorization",
  "token_endpoint": "http://192.168.120.33:8080/oauth2/token",
  "token_endpoint_auth_methods_supported": [
    "client_secret_basic",
    "client_secret_post",
    "client_secret_jwt",
    "private_key_jwt"
  ],
  "jwks_uri": "http://192.168.120.33:8080/oauth2/jwks",
  "userinfo_endpoint": "http://192.168.120.33:8080/userinfo",
  "end_session_endpoint": "http://192.168.120.33:8080/connect/logout",
  "response_types_supported": [
    "code"
  ],
  "grant_types_supported": [
    "authorization_code",
    "client_credentials",
    "refresh_token",
    "urn:ietf:params:oauth:grant-type:device_code"
  ],
  "revocation_endpoint": "http://192.168.120.33:8080/oauth2/revoke",
  "revocation_endpoint_auth_methods_supported": [
    "client_secret_basic",
    "client_secret_post",
    "client_secret_jwt",
    "private_key_jwt"
  ],
  "introspection_endpoint": "http://192.168.120.33:8080/oauth2/introspect",
  "introspection_endpoint_auth_methods_supported": [
    "client_secret_basic",
    "client_secret_post",
    "client_secret_jwt",
    "private_key_jwt"
  ],
  "subject_types_supported": [
    "public"
  ],
  "id_token_signing_alg_values_supported": [
    "RS256"
  ],
  "scopes_supported": [
    "openid"
  ]
}
```

@稀土掘金技术社区

所以通过该接口就可以很清楚的知道各授权端点的请求路径。

默认是基于内存的，基本上来说服务注册的客户端不会很多，基于内存也就够用了；框架也提供了基于db的方式，需要自己注入一个JdbcOAuth2AuthorizedClientService；让客户端认证信息持久化。[文档](#)

sql位置



## 客户端解释

客户端在oauth2角色解释中是第三方的一个应用，一般会配合资源服务一起使用



token，然后获取token中的权限，目前token中的权限只有scope的权限，并且不太好自定义，所以就需要通过资源服务器配置去更好的解析token。

2. 在分布式项目中：

- 在网关中添加客户端依赖，检查用户认证信息，由网关代理的微服务添加资源服务依赖，解析网关通过令牌中继的方式携带的access\_token；各个微服务添加自己的授权校验。
- 在网关中集成客户端依赖，同时集成资源服务依赖，由网关检查用户的认证和授权信息；各个微服务不用添加任何的认证与授权相关的处理，可以直接访问；这种方式需要屏蔽各微服务其它ip的访问，只能由网关代理访问。

总结

本篇文章以较少的代码集成了Security OAuth2 Client，体验到了springboot最开始说的 约定大于配置 的好处，框架添加了大量的默认配置，只需更改必须修改的自定义部分即可，本次的代码部分只有更改yml和编写一个测试接口，其它的重定向至认证服务和获取token的配置都已经默认实现了。

下篇会记录一下资源客户端的集成。

代码已提交至Gitee: [gitee.com/vains-Sofia...](https://gitee.com/vains-Sofia)

标签： Spring Boot    Spring    话题： 我的技术写作成长之路

本文收录于以下专栏

1 / 2



Spring Authorization Server  
Spring Authorization Server系列文章  
178 订阅 · 25 篇文章

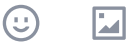
专栏目录

订阅

上一篇 Spring Authorization Server...    下一篇 Spring Authorization Server...



平等表达，友善交流



0 / 1000 ? 发送

最热 最新



用户2518816878179

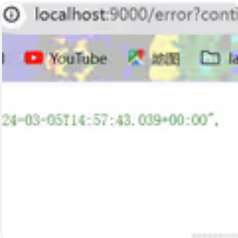
大佬这是我的客户端配置 请问一下为什么 输入密码账号之后 页面显示重定向次数过多

```
spring:
  cloud:
    auth:
      server:
        uri: http://127.0.0.1:8080/oauth2/authorize
        client-id: client-id
        client-secret: client-secret
        grant-type: authorization_code
        redirect-uri: http://127.0.0.1:8080/oauth2/authorize/callback
        scope: openid
        client-name: client-name
        client-authentication-method: client_secret_basic
        client-authentication-uri: http://auth-server:8080
```

13天前 点赞 3



用户2518816... : 重新来了一次 登录完成之后出现这种错误如何解决呀



12天前 点赞 回复



叹雪飞花 作者 回复 用户2518816... : 可能原因如图



12天前 点赞 回复



查看全部 3 条回复



Josie421

server，登录后,返回客户端 `http://127.0.0.1:8082/login?error`。页面显示 `invalid token response`。




2月前

 点赞

 8

...

 叹雪飞花 作者


: 在客户端获取token的地方断点看一下是不是响应错误了，应该是在OAuth2LoginAuthenticationFilter中处理的

2月前

 点赞

 回复

...

 Josie421 回复 叹雪飞花 作者

: 谢谢回复！  
client的controller的JwtAuthenticationToken 不能被resolve，所以我map了 / app来进行测试。是不是和这有关系？  
JwtAuthenticationToken 应该从哪个library获取呢？



2月前

 点赞

 回复

...

 Josie421 回复 叹雪飞花 作者


: 抱歉! / user应该在auth server，突然意识到放错了位置!!

2月前

 点赞

 回复

...

 叹雪飞花 作者 回复 Josie421

: 是的，在认证服务中；JwtAuthenticationToken 是Spring Authorization Server提供的，所以在client中无法引用 😊


2月前

 1

 回复

...

请问如果想要在auth server添加新的验证方法, auth server通过外部一个endpoint来验证用户 (post方法传username和password到一个外部终端, 返回用户信息) 改如何整合到目前项目中呢, 是通过重写UserDetailsService里面的loadUserByUsername方法吗?

2月前  点赞  回复 ...



叹雪飞花 作者 回复 Josie421 : 是的, 现在的用户信息是从数据库获取的, 你也可以改造一下调用一个接口调用外部系统获取用户信息

2月前  1  回复 ...



Josie421 回复 叹雪飞花 作者 : 请问大佬如何改造client server使其可以前后端分离呢? 前端vuejs后端java

2月前  点赞  回复 ...



叹雪飞花 作者 回复 Josie421 : 我写有一篇授权码模式的前后端分离, 你可以参考下;

还有一个是单页面应用使用PKCE模式获取token的文章, 你也可以看下。

认证服务的登录页面分离后客户端对接时就是使用分离的前端页面, 这里的登录页面可以自由指定的;

《Spring Authorization Server入门 (十二) 实现授权码模式使用前后端分离的登录页面》: [juejin.cn](https://juejin.cn)

《Spring Authorization Server入门 (十八) Vue项目使用PKCE模式对接认证服务》: [juejin.cn](https://juejin.cn)

[收起](#)

2月前  点赞  回复 ...



用户463560802877

大佬你好, Authorization Server端按照之前的文章一步步走下来, 集成oauth2-client的时候, 死活不会回调到client里面来. 配置按照demo来的.

配置在附件图片中.

这个问题如何排查呢?





2月前 点赞 9

...



用户4635608... : 这是认证服务中的client配置

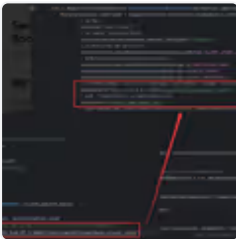


2月前 点赞 回复

...



叹雪飞花 作者 : authorization server中客户端信息的回调地址需要再添加一个回调地址，也就是客户端服务中配置的回调地址：http://1922.168.57.1:8000/...，如图



2月前 点赞 回复

...

查看全部 9 条回复



路修

大佬和你请教个问题，是关于 Spring Authorization Server 方面的。当认证授权服务器配置SSL+域名时， Security OAuth2 Client对接认证授权服务器 issuer-uri:https://domain 启动报错 Unable to resolve Configuration with the provided Issuer of "https://domain"。当认证授权服务器端点配置是 http://127.0.0.1:8080 时， Security OAuth2 Client对接认证授权服务器 issuer-uri:http://127.0.0.1:8080 启动正常。我看看了源码，应该是认证授权服务器配置SSL+域名时，客户端获取签发地...  
[展开](#)

4月前 点赞 评论

...



只有光头才能变得更强大 java开发  
大佬能出一期集成gateway网关的吗？



6月前 1 回复

...



听鹈深处 LV.2

登录成功后，重定向到/app这些接口是没有带上token是吧？请问该如何处理呢？

6月前 点赞 3

...



叹雪飞花 作者：对于客户端来说已经通过token获取到认证信息了，请求时不需要token了，如果是gateway网关，塔提供令牌中继(TokenRelay)的功能，代理其它资源服务时会携带access token

客户端好像也暴露了获取token的方式，不过我不太确定有没有，需要去文档中找一下

6月前 点赞 回复

...



听鹈深处 回复 叹雪飞花 作者：感谢博主的回复！是这样，我把client和resource整到一起，没带上token的话resource的接口就校验不了权限403了。我打开方式是否有问题

6月前 点赞 回复

...

查看全部 3 条回复



i吧啦啦啦 LV.2 吧啦啦啦 @啊吧啊吧  
client 授权登录后 如何退出呢？

7月前 点赞 1

...



叹雪飞花 作者：Spring Security 提供了退出的端点：/logout

7月前 点赞 回复

...



i吧啦啦啦 LV.2 吧啦啦啦 @啊吧啊吧  
client 直接配置 @PreAuthorize注解不生效，我这边授权登录后所有接口都能访问没走@PreAuthorize鉴权

7月前 点赞 1

...



叹雪飞花 作者：详见本系列的第九篇文章：[juejin.cn](https://juejin.cn)



 稀土掘金 [首页](#) ▾

探索稀土掘金 🔍





7月前     点赞     回复    ...



i啦啦啦啦  啦啦啦啦 @啊吧啊吧

/oauth2/consent用户授权后跳转到 /oauth2/authorize 页面404 是哪个地方有问题呀

7月前     点赞     8    ...



叹雪飞花  : 要断点确认一下授权确认提交的请求是否到达接口(POS方式请求的/oauth2/authorize), 到达以后确认一下请求是不是POST方式, 有没有缺什么参数

7月前     点赞     回复    ...



i啦啦啦啦 回复 叹雪飞花  : 我是通过域名和网关转发的, 我认证服务器在/ocean-oauth 下, 其中 model.addAttribute("requestURI", "/oauth2/authorize"); 那么这个路径是/ocean-oauth/oauth2/authorize 是吗

7月前     点赞     回复    ...



i啦啦啦啦 回复 i啦啦啦啦 : 还有一点[authorization\_request\_not\_found] 这个异常, 我是用域名也会出现这个异常吗, 必须要另一台电脑启动客户端 程序吗?

7月前     点赞     回复    ...



叹雪飞花  回复 i啦啦啦啦 : 是的

7月前     点赞     回复    ...



叹雪飞花  回复 i啦啦啦啦 : 这个如果是两个子域名不会出现这个问题

7月前     点赞     回复    ...



i啦啦啦啦 回复 叹雪飞花  : 6啊 我设置了子域名 就可以了。😓搞了两三天了, 我还以为我其他配置有问题

7月前     点赞     回复    ...



i啦啦啦啦 回复 叹雪飞花  : 是的, 可以了

7月前     点赞     回复    ...

目录

收起 ^

前言

流程说明

使用SpringBoot创建一个oauth2客户端

环境介绍

创建项目

pom.xml示例

配置application.yml，添加认证服务器信息和客户端信息

给认证服务添加一个用户接口

测试

编写一个测试接口

开始测试

机制说明

客户端注册信息

客户端解释

总结

相关推荐

Spring Authorization Server入门 (四) 自定义设备码授权

2.0k阅读 · 4点赞

Spring Authorization Server入门 (五) 自定义异常响应配置

1.8k阅读 · 4点赞

Spring Authorization Server入门 (六) 自定义JWT中包含的内容与资源服务jwt解析器

2.6k阅读 · 7点赞

Spring Authorization Server常见问题解答(FAQ)

896阅读 · 1点赞

Spring Authorization Server入门 (十五) 分离授权确认与设备码校验页面

精选内容

MySQL性能优化盲区（高并发情况下，事务内的数据先更新还是先查询？）

小松聊PHP进阶 · 109阅读 · 1点赞

Java基本语法之程序流程控制

小明爱吃火锅 · 80阅读 · 0点赞

不知道RAID/SAN/NAS的小可爱来看看这个吧！

JoyT · 78阅读 · 0点赞

MySQL的JOIN到底是怎么玩的

码上遇见你 · 185阅读 · 1点赞

美团大规模KV存储挑战与架构实践

美团技术团队 · 565阅读 · 5点赞

为你推荐

Spring Authorization Server入门 (二) Spring Boot整合Spring Authorization Server

叹雪飞花

9月前

 6.7k

 31


 76

Java

Spring Authorization Server入门 (十二) 实现授权码模式使用前后端分离的登录页面

叹雪飞花

8月前

 4.7k

 24

 65

后端

Spring ...

Spring

Spring Authorization Server入门 (十) 添加短信验证码方式登录

叹雪飞花

9月前

 3.4k

 20

 9

Spring

Spring ...

Spring Authorization Server入门 (十六) Spring Cloud Gateway对接认证服务

叹雪飞花

6月前

 2.5k

 17

 44

Spring ...

Spring ...

安全

Spring Authorization Server入门 (十三) 实现联合身份认证，集成Github与Gitee的OAu...

叹雪飞花

8月前

 2.3k

 13

 51

Spring

Spring ...

安全

Spring Authorization Server入门 (十一) 自定义grant\_type(短信认证登录)获取token

叹雪飞花

9月前

 2.4k

 15

 39

Spring

Spring ...

安全

Spring Authorization Server入门 (七) 登录添加图形验证码

叹雪飞花

9月前

 2.8k

 18

 2

Spring ...

Spring Authorization Server入门 (九) Spring Boot引入Resource Server对接认证服务

叹雪飞花 9月前  1.8k  13  8

Spring Spring ...

Spring Authorization Server优化篇：添加Redis缓存支持和统一响应类

叹雪飞花 8月前  1.7k  6  12

Spring Spring ... 安全

Spring Authorization Server入门 (十五) 分离授权确认与设备码校验页面

叹雪飞花 7月前  1.6k  14  8

Spring ... Spring Vue.js

Spring Authorization Server入门 (十九) 基于Redis的Token、客户端信息和授权确认信...

叹雪飞花 4月前  1.2k  8  19

Spring ... 后端 Redis

Spring Authorization Server入门 (二十) 实现二维码扫码登录

叹雪飞花 1月前  891  16  6

Spring ... Spring Java

Spring Authorization Server入门 (十七) Vue项目使用授权码模式对接认证服务

叹雪飞花 6月前  751  9  12

Vue.js 安全 Spring ...

Spring Cloud Gateway集成SpringDoc，集中管理微服务API

叹雪飞花 3月前  733  7  评论

Spring ... Spring ... Java