



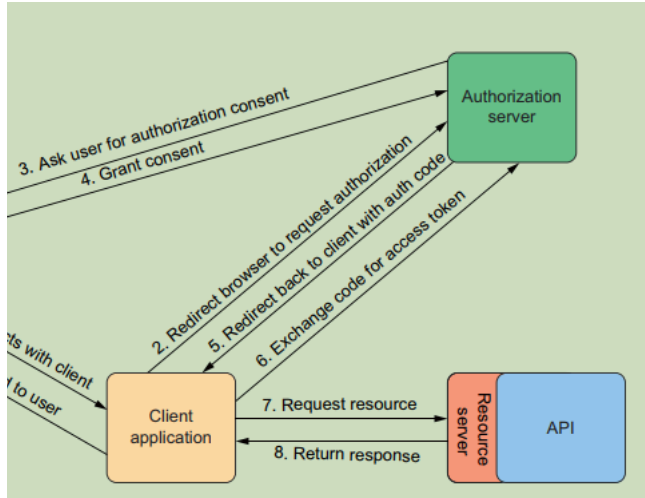
2024-04-30 21:05:05 修改 阅读量5.4k 收藏6 点赞数

spring java spring boot oauth2

内容已被华为云开发者联盟社区收录

## 授权码模式 (Authorization Code)

是用户凭证获取token 后来获取资源的访问权限。其交互步骤如下图：



些功能希望从资源服务器获取数据

求到授权服务器要求授权，在重定向之前客户端程序会给授权服务器传递一个参数作为回调地址

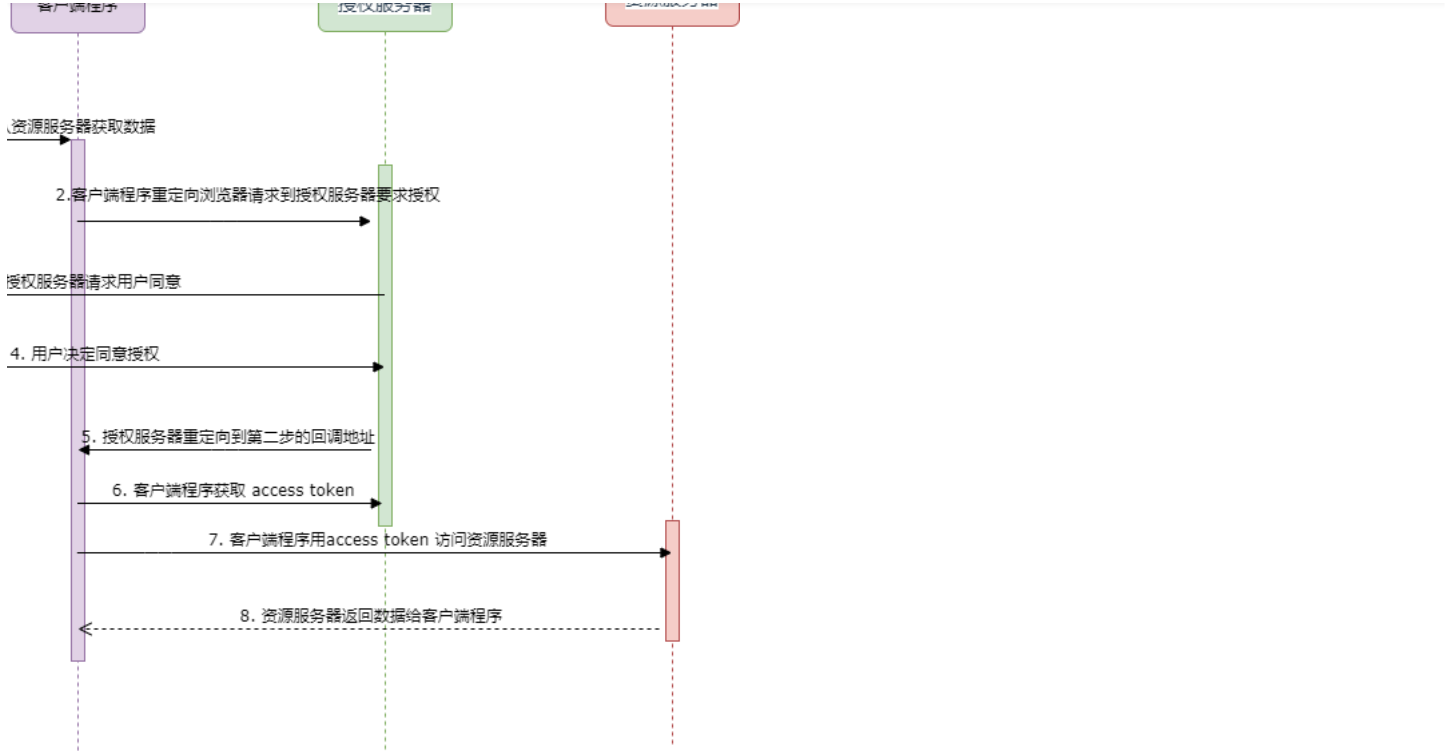
这个步骤一般需要用户先登录，如果已经登录则可能弹出一个交互页面请求用户同意授权

的回调地址，并且在URL 后附带一个Authorization Code，这个Authorization Code 是明文传递给客户端程序的，所以不安全

Code 和 认证的密钥获取 access token

访问资源服务器（也就是实际的业务接口）

端程序



一个spring 专门处理授权服务的项目，我们构建授权服务器就是基于它。我们先一步一步操作完成后再看整个流程。

2 master]

```

:Config.java
{
/a
iva

```

h-server

```


}

coding="UTF-8">
/maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http
0</modelVersion>

pringframework.boot</groupId>
ring-boot-starter-parent</artifactId>
</version>
/> <!-- Lookup parent from repository -->

groupId>
-auth-server</artifactId>

```



xuejianxinokok

关注

Authorization-server这个依赖是最重要的

下  
引

o/spring-boot/docs spring文档  
io/spring-boot/docs/2.6.3/reference/html/application-properties.html#application-properties

```
na h? dev"
```

□

据库中，所以先建立用户模型

```
r")  
ements UserDetails {  
  
al long serialVersionUID = 1L;  
  
ategy = GenerationType.AUTO)
```


≡

```
epository extends CrudRepository<User, Long> {  
  
String username);
```

配置

```
onfig {  
  
n defaultSecurityFilterChain(HttpSecurity http) throws Exception {  
off
```

置了<iframe>标签的“同源策略”，解决了上面出现的问题  
tps://blog.csdn.net/weixin\_43981241/article/details/108240756  
ders().frameOptions().sameOrigin()  
( )  
得h? console而不去做csrf检查

 xuejianxinokok

关注

都会重定向到登录页面

```
anMethods = false)
tionServerConfig {

    HEST_PRECEDENCE)
    terChain authorizationServerSecurityFilterChain(HttpSecurity http) throws Exception {
    ationServerConfiguration.applyDefaultSecurity(http);
    rmLogin(Customizer.withDefaults()).build();
}
```

对从内存种加载，正式环境需要从数据库种加载



Application.java中加载一些数据

```
rApplication {

    main(String[] args) {
    ion.run(AuthServerApplication.class, args);
}
```


数据


```
Runner dataLoader(UserRepository repo, PasswordEncoder encoder) {
    {
}
```





1, 首页代码请参看上面的源码连接


tt:9000

内网119

码云

数学

MN

xuejianxinokok

关注





访问 localhost 的请求遭

您未获授权，无法查看此网页。

HTTP ERROR 403

重新加载

oauth2/authorize?response\_type=code&client\_id=taco-admin-client&redirect\_uri=http://127.0.0.1:9090/login/oauth2/code/taco-admin-client&scope

户为 a1

nt required

wants to access your account **a1**

ns are requested by the above app.  
e and consent if you approve.

deleteIngredients

writeIngredients

Commit Consent

Cancel

对外接口

```
er.java
```

```
ory.java
```

```
ra
```

了以下内容

```
rg.springframework.boot</groupId>
d>spring-boot-starter-oauth2-resource-server</artifactId>
```

```
onfig {
```

```
n defaultSecurityFilterChain(HttpSecurity http) throws Exception {
off
```

置了<iframe>标签的“同源策略”，解决了上面出现的问题

[tps://blog.csdn.net/weixin\\_43981241/article/details/108240756](https://blog.csdn.net/weixin_43981241/article/details/108240756)

```
ders().frameOptions().sameOrigin()
```

```
()
```

得在 console 而不去做 ref 检查



```
.and()
```

```
//需要拦截的API
```

```
.authorizeRequests()
```

```
.antMatchers(HttpMethod.POST, "/api/ingredients")
```

```
.hasAuthority("SCOPE_writeIngredients")
```

```
.antMatchers(HttpMethod.DELETE, "/api/ingredients")
```

```
.hasAuthority("SCOPE_deleteIngredients")
```

```
.and()
```

```
//开启 ResourceServer
```

```
.oauth2ResourceServer(oauth2 -> oauth2.jwt())
```

```
.build();
```

```
ter: on
```





na.h2.dev"



2.resourceserver.jwt .jwk-set-uri指向了 认证服务器http://localhost:9000/oauth2/jwks

naster]

a

:Controller.java

iva

ice.java

rg.springframework.boot</groupId>

d>spring-boot-starter-oauth2-client</artifactId>

的

\_0[]

\_0[]



xuejianxinokok

关注

```
onfig {

n defaultSecurityFilterChain(HttpSecurity http) throws Exception {
Requests(authorizeRequests -> authorizeRequests.anyRequest().authenticated())
2Login(oauth2Login -> oauth2Login.loginPage("/oauth2/authorization/taco-admin-client"))
2Client(withDefaults()));
ild();
}
```

客户端通过 rest api 访问资源服务器的具体实现类

```
dientService implements IngredientService {

restTemplate;

tService() {

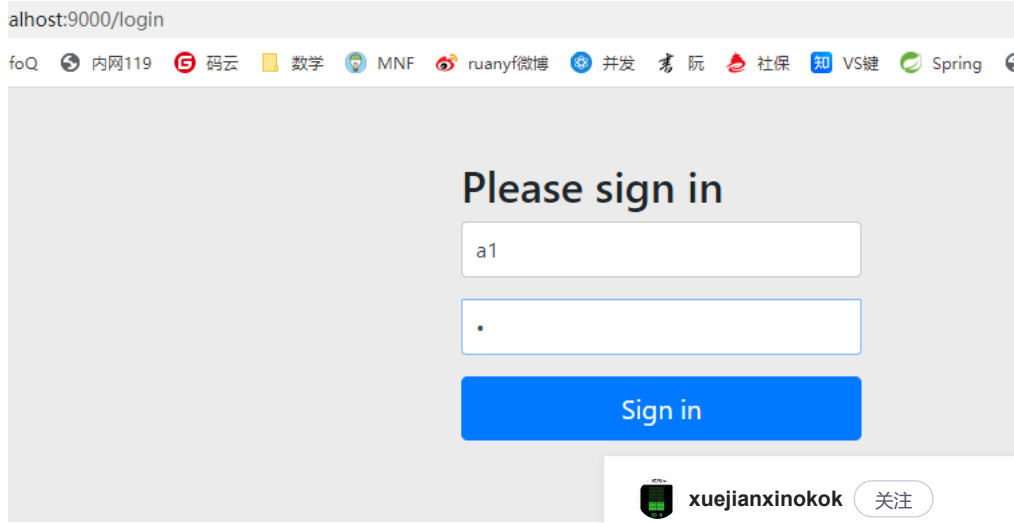
tService(String accessToken) {
= new RestTemplate();
null }
}
```

来一个全流程的测试

h-server  
server  
op

ith2-client-app 程序 在浏览器中输入 http://localhost:9090/ 访问

第6步的页面，由于我们没有登录 被拦截后重定向到认证服务器 页面,如下图



# Consent required

taco-admin-client wants to access your account **a1**

The following permissions are requested by the above app.  
Please review these and consent if you approve.

- ☒ deleteIngredients
- ☒ writeIngredients

Submit Consent

Cancel



## Taco Cloud Admin

- 原材料管理

客户端可以访问资源服务器提供的接口



## nin: Ingredients




应用中需要访问 oauth2-res-server 的一些接口，

私有信息所以需要我们授权，在访问接口时会重定向到oauth2-auth-server 要求授权，

ient-app 会获取一个token

回数据

 xuejianxinokok

关注

人正在系统学习中

模式的实现

授权码模式的实现，文中通过示例代码介绍的非常详细，对大家的学习或者工作具有一定的参考学习价值，需要的朋友们下面随着小编来一起学习学习吧

深入authorization\_code授权码模式完整实现

daxian-zhu/online\_edu 项目是最新的，文章可能不是最新的 这里先简单的介绍项目，不然有的地方可能我表达不清楚容易造成误解 online\_edu | |----online\_edu\_com 公共模块

Code(授权码)相对更安全\_id4授权码模 ...

后返回的code,这个code有时效性,而且只能使用一次。通过上面的流程,Authorization Code(授权码)模式其实更适合有后台的客户端,比如MVC程序,接收code和通过code获取ToI

式(authorization\_code)访问\_grant\_type...

\_token 在上一步中拿到的code,通过POSTMAN发送post请求,grant\_type为authorization\_code,请求地址如下: http://localhost:8080/oauth/token? grant\_type=authorization\_code

发布

zation protocol)，这里的授权是以委派代理（delegation）的方式。可以这样理解，OAuth 2.0提供一种协议交互框架，让某个应用能够以安全地方式获取到用户的委派书，这

tion\_code)过程

器/oauth/authorize 端点：（只用于"implicit", "authorization\_code"） GET: http://127.0.0.1:8080/oauth/authorize? client\_id=client&response\_type=code&redirect\_uri=http://w

le 模式(一)\_spring oauth2 code-CSDN...

以说明 OAuth2 Authorization\_code 模式是 OAuth2 认证中最完善的认证方式。其完整的认证模式如下: (A)用户访问客户端,客户端将用户引导向认证服务器。

=authorization\_code

orization\_code。code:必须参数,通过上面第一步所获得的Authorization Code。 client\_id:必须参数,应用的API Key。 client\_secret:必须参数,应用的Secret Key。

ngBoot集成Spring Security、OAuth2实现authorization code授权码模式

Boot项目集成Spring Security、OAuth2实现资源访问授权认证。支持client credentials、password、authorization code认证模式。项目默认最为复杂的authorization code授权

redirect\_uri, code 例如微信网页授权获取用户信息过程 1. 让用户明白所做的操作并请求认证 当牵涉到OAuth认证时，首先应最好能更进一步的让用户知道该操作到底会发生什

horization\_code)模式+jwt 实现微服 ...

tion\_code 模式 password模式 如果你是看我这个文章来的 我们只需要 修改以下内容,如果你是新来了,可以去我的gitHub看这篇文章的源码 源码地址 1.3:修改 WebSecurityConf

式(authorization\_code)

—— 授权码模式 (Authorization Code)

个，可能说的不是很准确，说下大体流程，我们在授权服务器上配置了这个回调地址，授权服务器在用户授权成功后，返回授权码的地址就是它，另外我们后续申请token时，也

uth2授权码模式的认证 (一)

auth2授权码模式的认证，花了一些时间，踩了一些坑，经验整理分享给大家。

众号「Java大师」，专注于分享Java领域干货文章http://www.javaman.cn/sb2/oauth-code 授权码（authorization code）方式，指的是第三方应用先申请一个授权码，然后再用该

模式实现登录验证授权

实现登录验证授权

登陆的方法

授权码登陆的方法，本文给大家介绍的非常详细，具有一定的参考借鉴价值,需要的朋友可以参考下

授权码模式的文档，希望可以帮助学习者

壹

. 配置部署

本概念，请参考 本文的上半部分: Debezium connector for MySQL 基本概念 5. 部署 要部署 Debezium MySQL 连接器，您需要安装 Debezium MySQL 连接器插件程序，配置连

本问题 com.microsoft.sqlserver.jdbc.SQLServerException: The driver could not establish a secure connection to SQL Server by using Secure Sockets Layer (SSL) encryption




 xuejianxinokok

关注

1信  
架，它提供了便捷的配置和开发方式。OAuth2是一种授权协议，用于保护Web应用的资源。而微信是中国最大的社交媒体平台之一，也提供了OAuth2的接口用于用户授权。在

“相关推荐”对你有帮助么？

-  非常没帮助
-  没帮助
-  一般
-  有帮助
-  非常有帮助

[关于我们](#) [招贤纳士](#) [商务合作](#) [寻求报道](#)  400-660-0108  [kefu@csdn.net](mailto: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

 等级

922

收藏







747

0728

0

15篇

18篇


2篇

2篇

15篇

19篇

下载...

 xuejianxinokok

关注

历

看 ...

【CS

看 ...

【CS

看 ...

户...

【CS

看 ...



烈推荐

好用?

|

,

,

,

不

可


3...



xuejianxinokok

关注

√...



xuejianxinokok

关注