黑客派

spring boot security oauth2 构建简单安全的 restful api

作者：lizhongyue248

原文链接：https://hacpai.com/article/1562330859635

来源网站：黑客派

<img src="https://static.hacpai.com/images/img-loading.svg" alt="u2420595868169895353fm26gp0jpg" data-src="https://resources.echocow.cn//file/2019/01/7345016a48f34bc3b3279be1c988b6c_u2420595868169895 2353fm26gp0.jpg">
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></in>
<script>
    (adsbygoogle = window.adsbygoogle || []).push({});
</script>
前段时间写了一篇博客，spring boot restful API 从零到一完整实践 ，通过上篇文章构建了两个版本的 ESTFul API，这篇博客，则是要在这基础上面，添加一个安全措施，我选择的是 oauth2 和 jwt 进行护我们的 API，通过 Spring security oauth2 进行一步一步的配置我们的安全 API 接口服务。
<blockquote>
 博客地址：https://echocow.cn
 项目地址：github-security-one 分支
</blockquote>
<h2 id="Oauth2">Oauth2</h2>
OAuth 2.0 关注客户端开发者的简易性。要么通过组织在资源拥有者和 HTTP 服务商之间的被批的交互动作代表用户，要么允许第三方应用代表用户获得访问的权限。同时为 Web 应用，桌面应用手机，和起居室设备提供专门的认证流程。百度百科。这篇文章同上一篇一样，将会详细记录如何使用 Spring ecurity oauth2 进行构建一个安全 RESTFul API ，当然，oauth2 的概念和核心我不再赘述，具体可看文末的参考链接。在这之前，我们需要先做一番准备。
<h2 id="这篇文章能够带给你什么">这篇文章能够带给你什么</h2>
使用 Spring security oauth2 autoconfigure 自动配置一个简单的 oauth2 认证
<h2 id="在这之前">在这之前</h2>
你需要拥有一个已经能够成功构建起来的 Spring boot 的项目，它能够正常启动与访问访问，这里我们使用上一篇 spring boot restful API 从零到一完整实践 构建的 RESTFul API 作为基础。如果你要快速体验，你只需要建立一个拥有 Helloworld API 接口的新项目即可。然后你需要加入以下依赖（radle）
<pre><code class="highlight-chroma">// 提供 spring security 支持
implementation('org.springframework.boot:spring-boot-starter-security')
// 提供 oauth2 自动化配置
implementation("org.springframework.security.oauth.boot:spring-security-oauth2-autoconfigre:${springBootVersion}")
</code></pre>
<img src="https://static.hacpai.com/images/img-loading.svg" alt="在这之前" data-src="https://resources.echocow.cn//file/2019/01/49c932992ee34adebd7b3c3e624d92a2_image.png">
同时你必须具备如下知识：

 oauth2 中什么是 授权服务器
 oauth2 中什么是 资源服务器

oauth2 中的四种授权模式（我们会使用到 授权码 模式和 密码 模式）
Spring security 的部分知识
jpa 的使用

以上概念本文不再提起，如有疑惑可选择去文末的参考链接自行选择性学习。
<h2 id="请求-url">请求 url</h2>
虽然上面说明了需要知道的东西，但是我在这里还是需要对我们需要使用到的 url 进行一个简单明，但是参数我不再详说。

/oauth/authorize GET 授权码模式获取授权码
/oauth/token POST 获取 token、刷新 token
/oauth/check_token POST 检测 token

<h2 id="自动配置">自动配置</h2>
Spring boot 之所以能够如此受欢迎，最大的原因莫过于他提供的模板配置以及自动配置，我们至不需要写什么代码，只需要见得修改一下配置文件即可构建一个基于内存的简单的安全服务，所以重要的，需要先配置一个 授权服务器，通过它下发令牌
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins>
<script>
    (adsbygoogle = window.adsbygoogle || []).push({});
</script>
<h2 id="使用默认配置">使用默认配置</h2>
1、 添加注解：你需要为我们启动类添加一个启动的注解

2、 我们通过一个注解就已经完成一个安全的授权的创建，运行查看输出日志。

3、 携带生成的 client id 进行访问，这里 /api 是我自己添加项目路径，如果没有添加亲直接访问 /oauth/authorize
<pre><code class="highlight-chroma">http://localhost:8080/api/oauth/authorize?response_type=code&client_id=d7003bdc-981c-4745-9eb4-673028b4c4e0&redirect_uri=http://example.com&scope=all
</code></pre>

4、 访问报错，这是因为我们没有配置 Spring security 造成的，所以需要回去配置一下，使用认配置即可

5、 再次运行，跳转登录界面

6、 用户名 <code>user</code>，密码为刚才生成的 随机密码，登录


7、 修改配置文件。注册一下回调地址

8、 重启，使用新生成的 client id ， 重新访问以及登录

9、 选择 Approve 同意授权，获取到授权码

10、 使用授权码请求 token

这就是使用他的自动配置的认证过程，接下来我们通过配置文件进行控制他的自动配置
<h2 id="自定义配置">自定义配置</h2>
1、 修改 Spring boot 的一些默认配置

2、 配置两个用户

3、 现在我们拥有两个用户了，就可以去使用密码模式获取 token 了。

<h2 id="测试访问">测试访问</h2>
1、 直接访问我们已有的资源

2、 携带上一步获取的 token 访问

发现还是失败，为什么呢？因为我么并没有开启 资源服务器 他没办法进行验证，所以我们接下就是开启一个资源服务器
<h2 id="开启-资源服务器">开启 资源服务器</h2>
同样，一个注解即可

重新获取 token 后，携带 token 访问

<h2 id="解析-token">解析 token</h2>
1、 尝试解析 token

2、 403，我们需要配置以支持 token 解析。

3、 重启后获取 token 再次解析

<h2 id="刷新-token">刷新 token</h2>
<blockquote>
遗憾的是，如果只是用配置文件，是不能够 刷新 token 的，至少我没有成功。
</blockquote>

发现不行，查看原因

可惜通过尝试各种办法都不行（在不增加类的情况下）解决办法参见

stackoverflow：spring-security-oauth2 2.0.7 refresh token UserDetailsService Configuration - UserDetailsService is required
segmentfault: spring security auth2 之 refresh token

使用他的配置文件，我们不需要写任何代码，就完成了一个简单的内存认证，甚至我们可以直接过修改 userDetailsService 完成用户的认证，不过也发现了，他使用配置文件的方式功能很有限，局性很强，不能够刷新 token 是一个痛点啊，所以我们更期望于手动配置。
<h2 id="参考链接">参考链接</h2>

<a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fwww.ruanyifeng.com%

Fblog%2F2014%2F05%2Foauth_2_0.html" target="_blank" rel="nofollow ugc">理解 OAuth 2.0 阮一峰
Oauth2 授权
spring oauth2 auto config
