

SpringDoc基础配置和集成OAuth2登录认证教程

叹雪飞花 2023-11-20 👁 363 ⌚ 阅读4分钟

关注

本期内容

1. 学会通过注解和Java代码的方式添加SpringDoc配置。
2. 在swagger-ui提供的页面上提供OAuth2登录认证，在集成Security的情况下便捷获取access_token并在请求时按照OAuth2规范携带。

为什么集成OAuth2登录认证？

现在大部分教程是在swagger-ui页面添加一个请求头，使用时先去获取一个token，然后再将获取的token填充至页面的token输入框内，如果是通过某个接口自己生成token的方式使用这种配置没什么太大问题，但是如果是通过OAuth2登录获取token时就比较麻烦，要填充token type和access token，所以在swagger-ui提供的页面中提供OAuth2登录的入口，OAuth2登录自动配置，一步到位。

配置方式说明

SpringDoc提供了两种方式来添加配置：

1. 注解
2. Java类的方式

注解

配置方式如下代码所示，详细内容请看代码中的注释

java 复制代码

```
1 package com.example.config;
2
3 import io.swagger.v3.oas.annotations.OpenAPIDefinition;
4 import io.swagger.v3.oas.annotations.enums.SecuritySchemeType;
5 import io.swagger.v3.oas.annotations.info.Info;
6 import io.swagger.v3.oas.annotations.info.License;
7 import io.swagger.v3.oas.annotations.security.OAuthFlow;
8 import io.swagger.v3.oas.annotations.security.OAuthFlows;
9 import io.swagger.v3.oas.annotations.security.OAuthScope;
10 import io.swagger.v3.oas.annotations.security.SecurityRequirement;
11 import io.swagger.v3.oas.annotations.security.SecurityScheme;
12 import io.swagger.v3.oas.annotations.security.SecuritySchemes;
13 import org.springframework.context.annotation.Configuration;
14
15 /**
16  * Spring Doc OpenApi 注解配置
17  *
18  * @author vains
19  */
20 @Configuration
21 @OpenAPIDefinition(
22     info = @Info(
23         // 标题
24         title = "${custom.info.title}",
25         // 版本
26         version = "${custom.info.version}",
27         // 描述
28         description = "${custom.info.description}",
29         // 首页
30         termsOfService = "${custom.info.termsOfService}",
31         // license
32         license = @License(
33             name = "${custom.license.name}",
34             // license 地址
35             url = "http://127.0.0.1:8080/example/test01"
36         )
37     ),
38     // 这里的名字是引用下边 @SecurityScheme 注解中指定的名字，指定后发起请求时会在请求头中按照OAuth2
39     security = @SecurityRequirement(name = "${custom.security.name}")
40 )
41 @SecuritySchemes({@SecurityScheme(
42     // 指定 SecurityScheme 的名称(OpenAPIDefinition注解中的security属性中会引用该名称)
```



```
46      // 设置认证流程
47      flows = @OAuthFlows(
48          // 设置授权码模式
49          authorizationCode = @OAuthFlow(
50              // 获取token地址
51              tokenUrl = "${custom.security.token-url}",
52              // 授权申请地址
53              authorizationUrl = "${custom.security.authorization-url}",
54              // oauth2的申请的scope(需要在OAuth2客户端中存在)
55              scopes = {
56                  @OAuthScope(name = "openid", description = "OpenId登录"),
57                  @OAuthScope(name = "profile", description = "获取用户信息"),
58                  @OAuthScope(name = "message.read", description = "读"),
59                  @OAuthScope(name = "message.write", description = "写")
60              }
61          )
62      )
63  )))
64  public class SpringDocAnnoConfig {
65  }
66
```

yml中的配置



yaml 复制代码

```
1  custom:
2    info:
3      title: example-api
4      version: 0.0.1
5      description: 这是一个使用SpringDoc生成的在线文档.
6      terms-of-service: http://127.0.0.1:8080/example/test01
7    license:
8      name: Apache 2.0
9    security:
10     name: Authenticate
11     token-url: http://kwqqr48rgo.cdhttp.cn/oauth2/token
12     authorization-url: http://kwqqr48rgo.cdhttp.cn/oauth2/authorize
13
```

注意：要想请求时携带登录获取的 `access_token`，`@OpenAPIDefinition` 注解中必须指定 `security` 属性，并且 `name` 值要和 `@SecurityScheme` 注解中的 `name` 属性一致。

还有一件事，使用注解的方式可以直接从yml中获取yml配置文件中的值，当然也可以直接使用具体的常量值(e.g. `@License` 注解中的url属性)。

还有一件事，如果该配置类上只有 `@OpenAPIDefinition` 或 `@SecurityScheme` 注解，并且该配置类中没有任何实现，则该配置类会在本机编译时消失，请添加 `@Configuration` 注解避免这种情况。

官网原文: *If you give `@OpenAPIDefinition` or `@SecurityScheme` to a class that has no implementation, that class will disappear when you natively compile. To avoid this, give the class a `@Configuration`.*

还有一件事...

Java代码配置

这里的注意事项和上边注解配置是一样的，不过由注解转为了Java类。

配置方式如下代码所示，详细内容请看代码中的注释

```
1 package com.example.config;
2
3 import io.swagger.v3.oas.models.Components;
4 import io.swagger.v3.oas.models.OpenAPI;
5 import io.swagger.v3.oas.models.info.Info;
6 import io.swagger.v3.oas.models.info.License;
7 import io.swagger.v3.oas.models.security.OAuthFlow;
8 import io.swagger.v3.oas.models.security.OAuthFlows;
9 import io.swagger.v3.oas.models.security.Scopes;
10 import io.swagger.v3.oas.models.security.SecurityRequirement;
11 import io.swagger.v3.oas.models.security.SecurityScheme;
12 import org.springframework.context.annotation.Bean;
13 import org.springframework.context.annotation.Configuration;
14
15 /**
16  * Spring Doc OpenApi Java代码配置
17  *
18  * @author vains
19  */
20 @Configuration
21 public class SpringDocBeanConfig {
```

java 复制代码

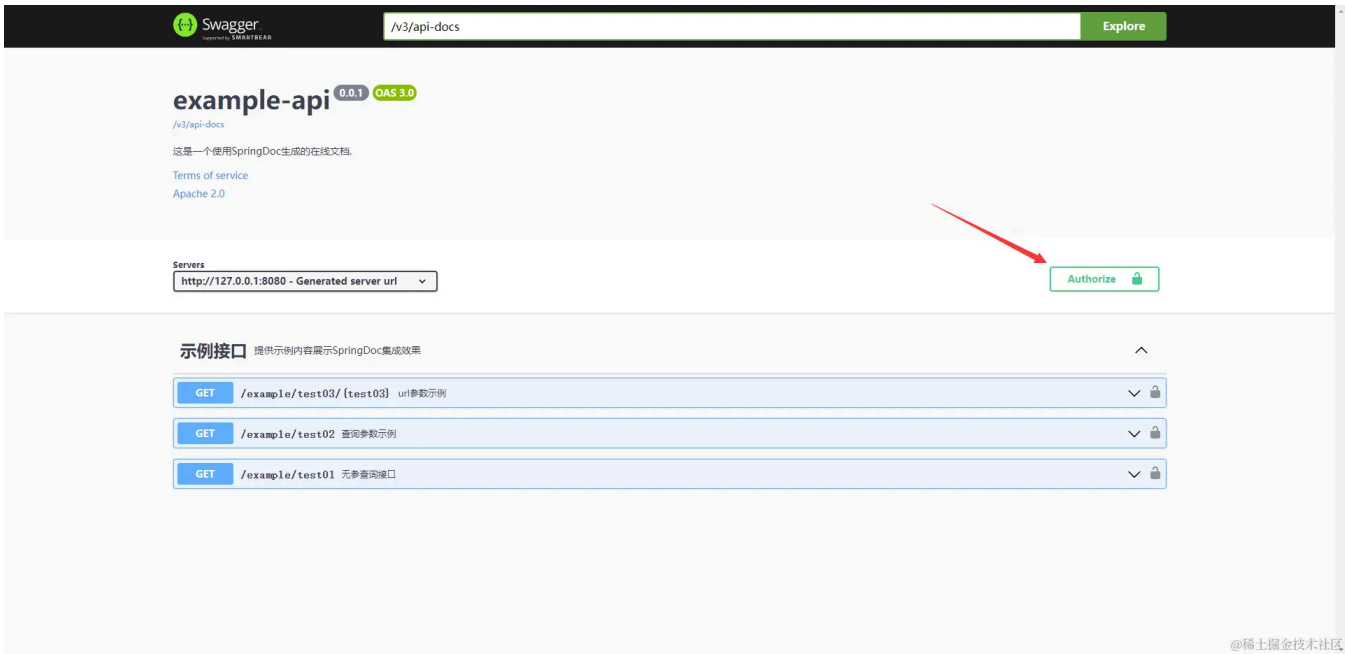
```
25 // 基础信息
26 Info info = new Info()
27     .title("example-api-java")
28     .version("0.0.1")
29     .description("这是一个使用SpringDoc生成的在线文档.")
30     .termsOfService("http://127.0.0.1:8080/example/test01")
31     .license(new License().name("Apache 2.0").url("http://127.0.0.1:8080/example/tes
32
33 // 组件
34 Components components = new Components();
35 // 安全认证组件
36 SecurityScheme securityScheme = new SecurityScheme();
37
38 // 创建一个oauth认证流程
39 OAuthFlows oAuthFlows = new OAuthFlows();
40 // 设置OAuth2流程中认证服务的基本信息
41 OAuthFlow oAuthFlow = new OAuthFlow()
42     // 授权申请地址
43     .authorizationUrl("http://kwqqr48rgo.cdhttp.cn/oauth2/authorize")
44     // 获取token地址
45     .tokenUrl("http://kwqqr48rgo.cdhttp.cn/oauth2/token")
46     .scopes(new Scopes()
47         .addString("openid", "OpenId登录")
48         .addString("profile", "获取用户信息")
49         .addString("message.read", "读")
50         .addString("message.write", "写")
51     );
52 // 使用授权码模式
53 oAuthFlows.authorizationCode(oAuthFlow);
54
55 // OAuth2流程
56 securityScheme.flows(oAuthFlows)
57     .type(SecurityScheme.Type.OAUTH2);
58
59 // 安全认证名
60 String securityName = "Authenticate";
61 // 将认证配置加入组件中
62 components.addSecuritySchemes(securityName, securityScheme);
63
64 SecurityRequirement securityRequirement = new SecurityRequirement();
65 // 将安全认证和swagger-ui关联起来
66 securityRequirement.addList(securityName);
67 return new OpenAPI()
68     // 基础描述信息
69     .info(info)
70     // 添加OAuth2认证流程组件
```

```
74     }  
75  
76   }  
77
```

效果预览

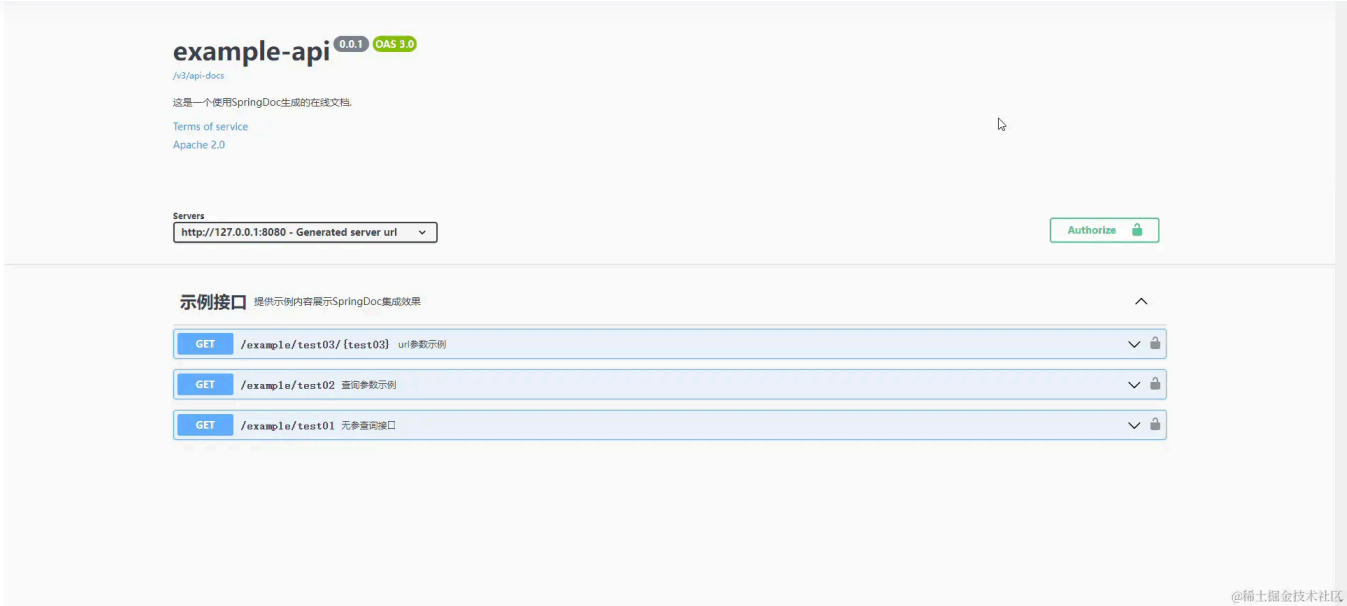
启动项目后打开提供的在线文档地址：

<http://127.0.0.1:8080/swagger-ui/index.html>



右侧多了一个 **Authorize** 按钮。

操作演示流程如下



其它基础配置

关于[springdoc-openapi-core-properties](#)和[swagger-ui properties](#)在官网中有详细的介绍了，就不展开说明了，读者可以对照文档在yml中添加自己的相关配置。

附录

1. [SpringDoc官网](#)
2. 代码仓库: [Gitee](#)、[Github](#)

标签：

后端


Spring Boot

Java

话题：

每天一个知识点

本文收录于以下专栏



SpringDoc

专栏目录

适用于SpringBoot3.x版本的在线文档工具——SpringDoc框架。

14 订阅 · 4 篇文章

订阅

上一篇

SpringBoot3.x最简集成Sprin...

下一篇

SpringDoc枚举字段处理与Sp...



抢首评，友善交流



0 / 1000



发送

暂无评论数据

目录

收起 ^

- 本期内容
- 为什么集成OAuth2登录认证?
- 配置方式说明
- 注解
 - 注解配置类
 - yml中的配置
- Java代码配置
- 效果预览
- 其它基础配置
- 附录

相关推荐

 Spring Authorization Server (1) 认证、授权、oauth2概念和流程初步介绍



368阅读 · 1点赞

 Spring Authorization Server 文章说明和目录

554阅读 · 2点赞

 Spring Authorization Server (2) 授权服务、资源服务、客户端核心配置讲解

557阅读 · 3点赞

【安全篇】Spring Boot 整合 Spring Authorization Server

3.5k阅读 · 1点赞

精选内容

利用Python实现网络运维自动化：实战示例

海拥 · 192阅读 · 0点赞

Python异步任务实现指南：探索高效编程的多种途径

有追求的开发者 · 196阅读 · 1点赞

docker入门原来如此简单！详细图文！

石小石Orz · 228阅读 · 2点赞

VSCode 的 Remote Development 扩展与 Docker 扩展使用

写bug写bug · 135阅读 · 0点赞

精通Java Socket编程：深入剖析系统调用与TCP状态管理

码拉松 · 224阅读 · 0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

 6.8k

 31

 86

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

 4.8k

 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 (八) Spring Boot引入Security OAuth2 Client对接认...

叹雪飞花

9月前

 2.8k

 13

 43

Spring ...

Spring



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 👁 2.5k 👍 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 👁 2.9k 👍 18 💬 4 Spring ...

SpringBoot3.x最简集成SpringDoc-OpenApi

叹雪飞花 4月前 👁 2.3k 👍 17 💬 评论 后端 Spring ... Java

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 👁 1.9k 👍 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 👁 1.3k 👍 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 2月前 👁 936 👍 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 👁 762 👍 9 💬 12 Vue.js 安全 Spring ...