



链滴

springboot+swagger 自动生成 restful 文档及在线调试

作者: [crick77](#)

原文链接: <https://ld246.com/article/1509205155949>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

spring-boot-swagger-starter

本文项目已发布到github，欢迎fork点赞。

<https://github.com/wangyuheng/spring-boot-swagger-starter>

已发布maven中央仓库，欢迎使用。

```
<dependency>
  <groupId>com.github.wangyuheng</groupId>
  <artifactId>swagger-starter</artifactId>
  <version>0.0.1</version>
</dependency>
```

swagger

swagger可以快速生成restful接口文档，并提供在线调试功能。通过springboot开发微服务时，swagger文档可以极大的提高协作效率。

springfox

[springfox](#) 将swagger整合到spring

定制starter

1. 定义swagger属性

用于文档页面显示及接口扫描路径，并使用@ConfigurationProperties读取属性

```
swagger:
  groupName: 分类(groupName)
  title: 标题(title)
  description: 介绍(description)
  termsOfServiceUrl: 服务URL(termsOfServiceUrl)
  version: 版本(version)
  contactName: 作者名(contactName)
  contactUrl: 作者主页(contactUrl)
  contactEmail: 作者邮箱(contactEmail)
  paths: /upload.*/,/category.*
  license:
  licenseUrl:
```

```
import org.springframework.boot.context.properties.ConfigurationProperties;
```

```
@ConfigurationProperties(prefix = "swagger")
public class SwaggerProperty {
```

```
    private String groupName;
    private String title;
    private String description;
    private String termsOfServiceUrl;
    private String version;
```

```
private String license;
private String licenseUrl;
//Contact
private String contactName;
private String contactUrl;
private String contactEmail;
private String[] paths;
}
```

2.设置swagger自动配置类

重点在于**@EnableSwagger2及声明Docket**

```
import com.google.common.base.Predicate;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.context.properties.EnableConfigurationProperties;
import org.springframework.context.annotation.Bean;
import pro.hemo.swagger.config.SwaggerProperty;
import springfox.documentation.builders.ApiInfoBuilder;
import springfox.documentation.service.ApiInfo;
import springfox.documentation.service.Contact;
import springfox.documentation.spi.DocumentationType;
import springfox.documentation.spring.web.plugins.Docket;
import springfox.documentation.swagger2.annotations.EnableSwagger2;

import static com.google.common.base.Predicates.or;
import static springfox.documentation.builders.PathSelectors.regex;

@EnableSwagger2
@EnableConfigurationProperties(SwaggerProperty.class)
public class SwaggerAutoApplication {

    @Autowired
    private SwaggerProperty swaggerProperty;

    @Bean
    public Docket docket() {
        checkValid(swaggerProperty);
        Predicate<String>[] selector = new Predicate[]{};
        if (null != swaggerProperty.getPaths() && swaggerProperty.getPaths().length > 0) {
            selector = new Predicate[swaggerProperty.getPaths().length];
            for (int i = 0; i < selector.length; i++) {
                selector[i] = regex(swaggerProperty.getPaths()[i]);
            }
        }

        return new Docket(DocumentationType.SWAGGER_2)
            .groupName(swaggerProperty.getGroupName())
            .apiInfo(apiInfo())
            .select()
            .paths(or(selector))
            .build();
    }
}
```

```

private void checkValid(SwaggerProperty swaggerProperty) {
}

private ApiInfo apiInfo() {
    return new ApiInfoBuilder()
        .title(swaggerProperty.getTitle())
        .description(swaggerProperty.getDescription())
        .termsOfServiceUrl(swaggerProperty.getTermsOfServiceUrl())
        .contact(new Contact(swaggerProperty.getContactName(), swaggerProperty.getContactUrl(), swaggerProperty.getContactEmail()))
        .license(swaggerProperty.getLicense())
        .licenseUrl(swaggerProperty.getLicenseUrl())
        .version(swaggerProperty.getVersion())
        .build();
}
}
}

```

3. 创建spring.factories

用于指定springboot自动配置文件路径，目录为src/main/resources/META-INF/spring.factories

```

org.springframework.boot.autoconfigure.EnableAutoConfiguration=\
pro.hemo.swagger.SwaggerAutoApplication

```

使用starter

1. 添加依赖

通过maven添加starter

```

<dependency>
  <groupId>pro.hemo</groupId>
  <artifactId>swagger-starter</artifactId>
  <version>0.0.1-SNAPSHOT</version>
</dependency>

```

因为没有发布到仓库，需要先将starter项目发布到本地仓库

在SwaggerStarter目录执行

```

mvn clean install

```

2. 编写Restful接口

Spring可以很方便的编写Restful接口，可以添加**@Api**等注解，用于生成文档。注解在io.swagger.annotations包下，后续会介绍常用注解。

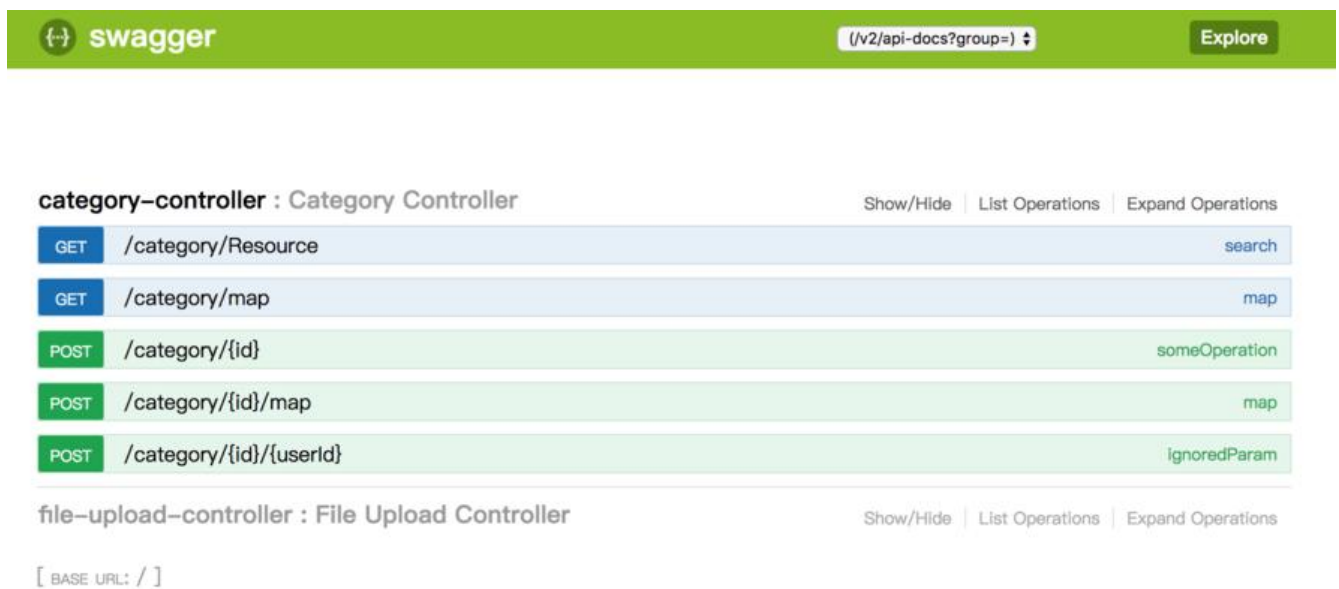
3. 添加配置文件

通过path指定接口扫描目录

```
swagger:
  groupName:
  title:
  description:
  termsOfServiceUrl:
  version:
  contactName:
  contactUrl:
  contactEmail:
  paths: /upload.*, /category.*
  license:
  licenseUrl:
```

4. 启动项目

启动springboot项目，访问<http://localhost:8080/swagger-ui.html> 可以看到接口列表。



The screenshot shows the Swagger UI interface. At the top, there is a green header with the 'swagger' logo, a dropdown menu showing '(v2/api-docs?group=)', and an 'Explore' button. Below the header, the 'category-controller : Category Controller' section is expanded, showing a list of API endpoints:

Method	Endpoint	Operation
GET	/category/Resource	search
GET	/category/map	map
POST	/category/{id}	someOperation
POST	/category/{id}/map	map
POST	/category/{id}/{userId}	ignoredParam

Below this, the 'file-upload-controller : File Upload Controller' section is visible but collapsed. At the bottom left, there is a '[BASE URL: /]' link.

这里有几点需要注意

1. swagger-ui.html 为自动生成映射
2. 注意静态资源不要被拦截
3. swagger-ui.html为页面，接口以json格式通过加载并渲染。json地址为<http://localhost:8080/v2/api-docs?group>