

# 基于SpringBoot+knife4j实现后端API开发

[https://doc.xiaominfo.com/knife4j/documentation/get\\_start.html](https://doc.xiaominfo.com/knife4j/documentation/get_start.html)

## 创建SpringBoot项目-apidemo

### Selected Dependencies

#### 开发工具

Lombok ×

Spring Configuration Processor ×

#### Web

Spring Web ×

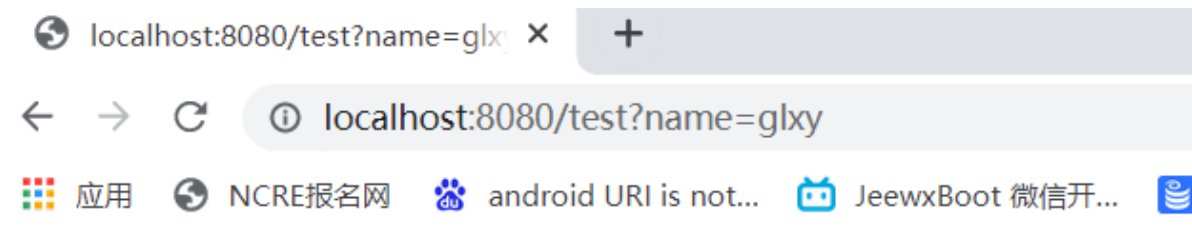
## 创建RESTful接口服务

```
package com.glxxy.apidemo.controller;

import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class ApiController {
    //http://localhost:8080/test
    @RequestMapping(value = "/test", method =
{RequestMethod.GET, RequestMethod.POST})
    public String test( String name) {
        return "hello,"+name;
    }
}
```

## 测试项目能否正常运行



# hello,glxy

## 在maven项目的pom.xml中引入Knife4j的依赖包

```
<dependencies>
    <!-- knife4j -->
    <dependency>
        <groupId>com.github.xiaoymin</groupId>
        <artifactId>knife4j-spring-boot-starter</artifactId>
        <version>2.0.9</version>
    </dependency>
    <!-- Knife4j API文档 -->
    <dependency>
        <groupId>com.github.xiaoymin</groupId>
        <artifactId>knife4j-spring-ui</artifactId>
        <version>2.0.9</version>
    </dependency>
</dependencies>
```

## 配置knife4j

```
#knife4j配置  https://doc.xiaominfo.com/knife4j/documentation/enhance.html
knife4j.enable=true
#开启生产环境屏蔽
knife4j.production=false
# 是否开启一个默认的跨域配置,该功能配合自定义Host使用,默认值为false
knife4j.cors=true
```

```
package com.glxy.apidemo.config;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import springfox.documentation.builders.ApiInfoBuilder;
import springfox.documentation.builders.PathSelectors;
import springfox.documentation.builders.RequestHandlerSelectors;
import springfox.documentation.spi.DocumentationType;
import springfox.documentation.spring.web.plugins.Docket;
import springfox.documentation.swagger2.annotations.EnableSwagger2WebMvc;

@Configuration
@EnableSwagger2WebMvc
public class Knife4jConfiguration {

    @Bean(value = "defaultApi2")
```

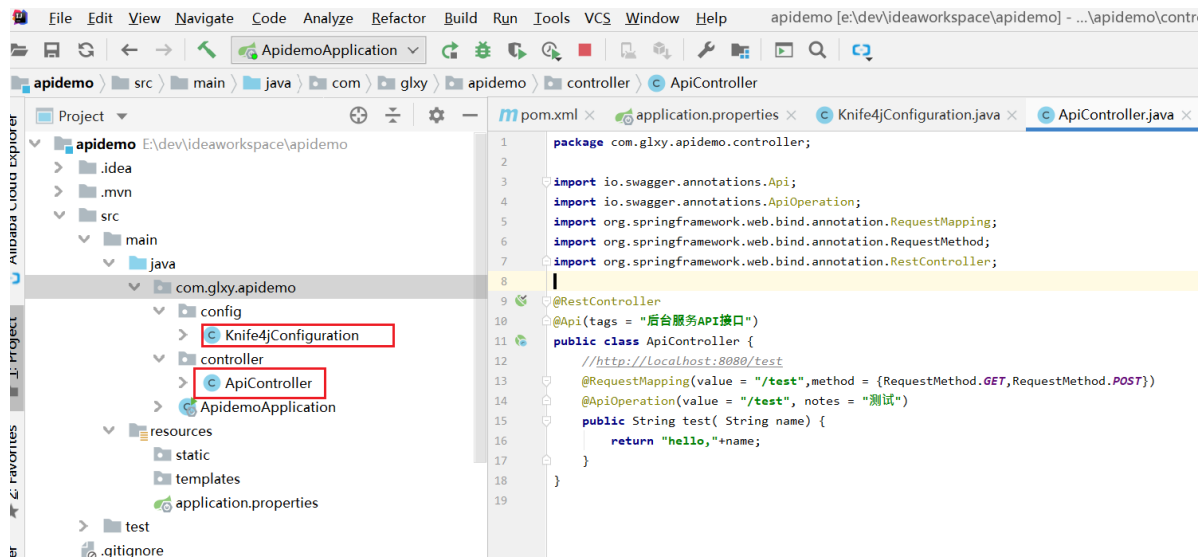
```

public Docket defaultApi2() {
    Docket docket=new Docket(DocumentationType.SWAGGER_2)
        .apiInfo(new ApiInfoBuilder()
            .title("后端RESTful APIs")
            .description("# 后端RESTful APIs")
            .termsOfServiceUrl("http://www.xx.com/")
            .contact("xx@qq.com")
            .version("1.0")
            .build())

    //分组名称
    .groupName("2.x版本")
    .select()
    //这里指定扫描包路径
    .apis(RequestHandlerSelectors.basePackage("com.glxxy.apidemo"))
    .paths(PathSelectors.any())
    .build();
    return docket;
}
}

```

## 项目结构如下



## 利用注解配置生成接口api在线技术文档

示例代码：

```

@Api(tags = "首页模块")
@RestController
public class IndexController {

    @ApiImplicitParam(name = "name",value = "姓名",required = true)
    @ApiOperation(value = "向客人问好")
    @GetMapping("/sayHi")
    public ResponseEntity<String> sayHi(@RequestParam(value = "name")String
name){
        return ResponseEntity.ok("Hi:"+name);
    }
}

```

```
@RestController
@Api(tags = "后台服务API接口")
public class ApiController {
    //http://localhost:8080/test
    @RequestMapping(value = "/test",method =
{RequestMethod.GET,RequestMethod.POST})
    @ApiOperation(value = "/test", notes = "测试")
    public String test( String name) {
        return "hello,"+name;
    }
}
```

启动Spring Boot工程，在浏览器中访问: <http://localhost:8080/doc.html>



## 增强功能

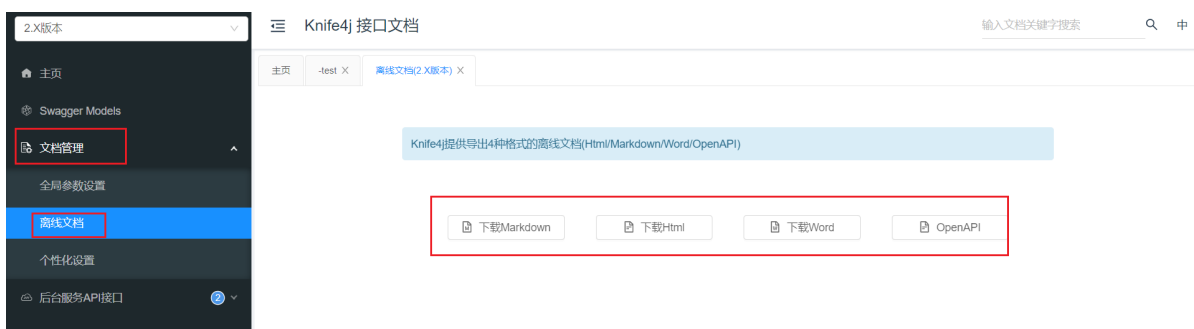
增强功能需要通过配置yml配置文件开启增强,自2.0.6开始，需配置：

```
knife4j.enable=true
```

如接口排序规则需使用 knife4j 提供的增强注解 @ApiOperationSupport

```
@ApiOperationSupport(order = 33)
```

## 导出离线文档



# Slf4j的使用

添加lombok依赖

```
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <optional>true</optional>
</dependency>
```

在类定义前使用@Slf4j注解，在方法中使用log.xxx()方法输出日志信息;

```
4  import io.swagger.annotations.Api;
5  import io.swagger.annotations.ApiOperation;
6  import lombok.extern.slf4j.Slf4j;
7  import org.springframework.web.bind.annotation.RequestMapping;
8  import org.springframework.web.bind.annotation.RequestMethod;
9  import org.springframework.web.bind.annotation.RestController;
10
11  @RestController
12  @Api(tags = "后台服务API接口") //添加接口模块描述
13  @Slf4j
14  public class ApiController {
15      //http://localhost:8080/test
16      @RequestMapping(value = "/test", method = {RequestMethod.GET, RequestMethod.POST})
17      @ApiOperation(value = "/test", notes = "测试") //添加接口描述
18      @ApiOperationSupport(author = "2695341@qq.com") //添加接口作者
19      public String test( String name) {
20          log.info(name);
21          return "hello,"+name;
22      }
23  }
```

在配置文件中可配置日志输出级别

```
#输出日志级别
logging.level.com.glx.apidemo.controller=info
```