**Swagger在线文档集成**

# 说明

版权归属：鑫学猿公司

微信：youhouduanyanfa

Java开发工程师转行辅导体系,保障拿到java工程师offer

图形用户界面, 文本, 应用程序

描述已自动生成

**背景**

在前后端分离开发中，为了减少与其它团队的沟通成本，一般都会构建一份 **RESTful API** 文档来描述所有的接口信息。但传统的方式有许多弊端，不仅编写文档工作量巨大，而且维护不方便，测试也不方便（需要借助第三方工具，如 **Postman** 来测试）

为解决这些问题，我们可以使用 **Swagger 2** 来构建在线接口文档，把代码和文档融为一体，减少人工维护文档的成本

**目标**

学习在Springboot项目中集成Swagger在线文档

**内容**

**安装配置**

1. 首先编辑项目的 **pom.xml** 文件，添加 **Swagger 2** 相关依赖

|  |
| --- |
| 1. swagger和spring版本兼容问题需要注意，2.9.x的swagger对应spring的版本需要是2.6.x以下，3.0的wagger对应spring版本为2.6.5+ 2. 因为这里引入的 **swagger ui** 是 **2.7** 以上的版本，所以还需要引入 **guava**，否则会因为 **guava** 兼容性问题造成项目启动报错 |

|  |
| --- |
| XML <dependency>  <groupId>io.springfox</groupId>  <artifactId>springfox-swagger2</artifactId>  <version>2.9.2</version> </dependency> <dependency>  <groupId>io.springfox</groupId>  <artifactId>springfox-swagger-ui</artifactId>  <version>2.9.2</version> </dependency> <dependency>  <groupId>com.google.guava</groupId>  <artifactId>guava</artifactId>  <version>20.0</version> </dependency> |

1. 创建 **Swagger 2** 的配置类，代码如下

|  |
| --- |
| **代码说明**   * 首先通过 **@EnableSwagger2** 注解开启了 **Swagger 2**，然后最主要的是配置一个 **Docket** * 通过 **apis** 方法配置要扫描的 **controller** 位置，通过 **paths** 方法配置路径 * 在 **apiInfo** 中构建文档的基本信息，例如描述、联系人信息、版本、标题等 |

|  |
| --- |
| Java @Configuration @EnableSwagger2 public class SwaggerConfig {   @Bean  public Docket createRestApi() {  return new Docket(DocumentationType.SWAGGER\_2).apiInfo(apiInfo()).select()  .apis(RequestHandlerSelectors.basePackage("com.xyy.web"))  .paths(PathSelectors.any())  .build();  }   private ApiInfo apiInfo() {  return new ApiInfoBuilder()  .title("multi-mod系统-接口数据文档")  .description("multi-mod接口文档")  .version("V1.0")  .contact(new Contact("xyy","IT乌托邦",""))  .build();  } } |

**使用示例**

1. 首先我们在 **Controller** 上添加相关的 **@Api** 注解

*（1）****@Api*** *注解标注在类上用来描述整个* ***Controller*** *信息。  
（2）****@ApiOperation*** *注解标注在方法上，用来描述一个方法的基本信息。  
（3）****@ApiImplicitParam*** *注解标注在方法上，用来描述方法的参数。其中* ***paramType*** *是指方法参数的类型，有如下可选值：*

* ***path****：参数获取方式为* ***@PathVariable***
* ***query****：参数获取方式为* ***@RequestParam***
* ***header****：参数获取方式为* ***@RequestHeader***
* ***body***
* ***form***

*（4）如果有多个参数，可以将多个参数的* ***@ApiImplicitParam*** *注解放到* ***@ApiImplicitParams*** *中。  
（5）****@ApiResponse*** *是对响应结果的描述。****code*** *表示响应码，****message*** *为相应的描述信息。如果有多个* ***@ApiResponse****，则放在一个* ***@ApiResponses*** *中。  
（6）****@ApiIgnore*** *注解表示不对某个接口生成文档。*

|  |
| --- |
| Java @RestController @Api(tags = "用户数据接口") public class UserController {   @ApiOperation(value = "查询用户", notes = "根据姓名查询用户")  @GetMapping("/user/{name}")  public String queryUserByName(@PathVariable String name) {  return "查找的用户为：" + name;  } } |

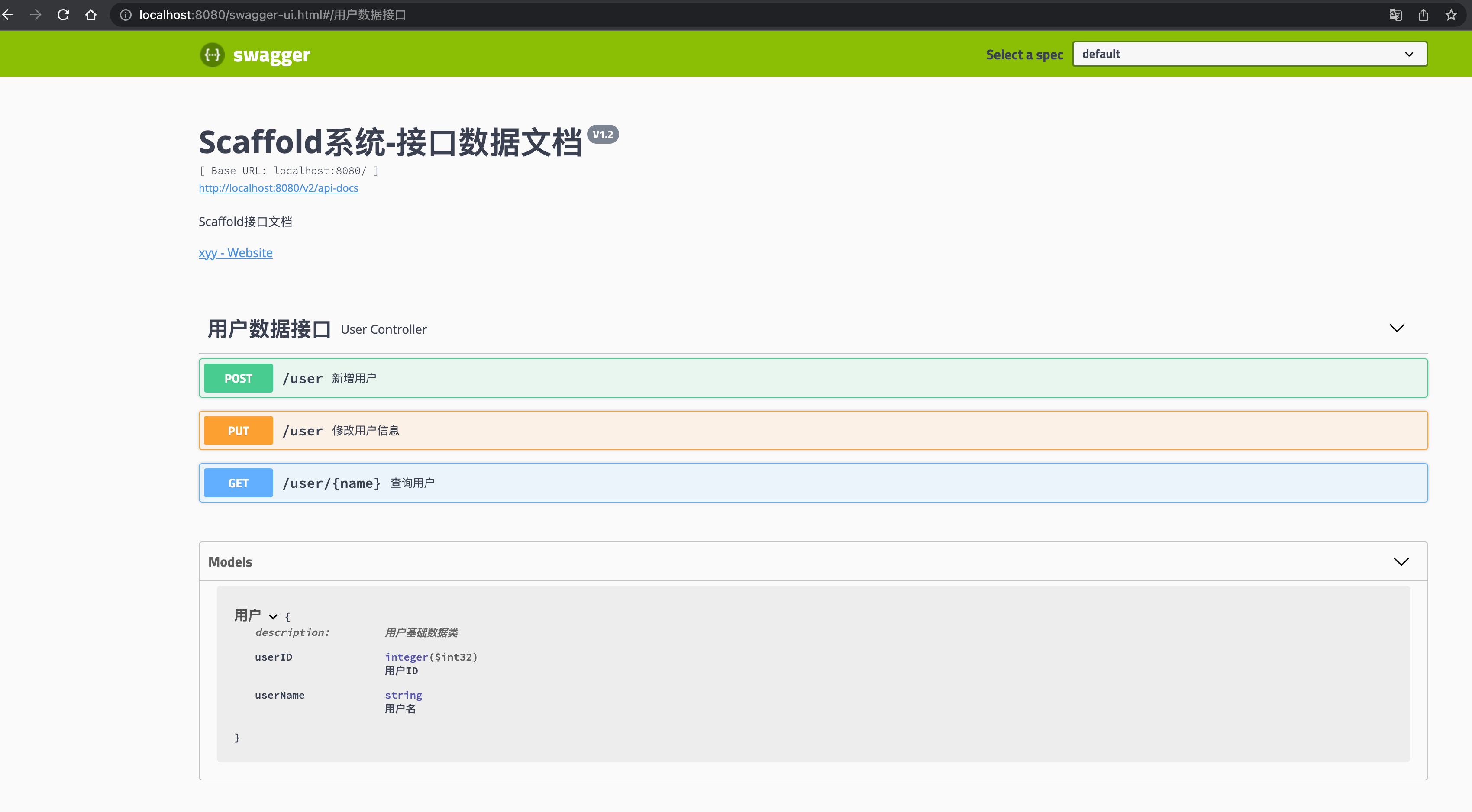
1. 对模型对象也添加相关的注解

|  |
| --- |
| 由于在上面 **Controller** 里的 **updateUser** 方法中，使用 **@RequestBody** 注解来接收数据，此时可以通过 **@ApiModel** 注解和 **@ApiModelProperty** 注解配置 **User** 对象的描述信息 |

|  |
| --- |
| Java @Getter @Setter @ToString @ApiModel(value = "用户", description = "用户基础数据类") public class UserDTO {  @ApiModelProperty(value = "用户名")  private String userName;  @ApiModelProperty(value = "用户ID")  private Integer userID; } |

**查看接口文档**

1. 启动 **Spring Boot** 项目，在浏览器中输入 <http://localhost:8080/swagger-ui.html> 即可看到接口文档



1. 点击任一接口，可以进行测试

|  |  |
| --- | --- |
|  |  |

**课件代码**