

* 学习目标

* 能够编写SpringBoot的starter

* 能够掌握SpringBoot集成Swagger

* 总复习一

* 回顾

* 能够编写SpringBoot的starter

```
1 * Excel工具类以starter封装，简化开发
2 * 构建springboot工程
3 * 添加依赖
4 <!--负责从properties文件中读取配置信息-->
5 <dependency>
6     <groupId>org.springframework.boot</groupId>
7     <artifactId>spring-boot-configuration-processor</artifactId>
8     <optional>true</optional>
9 </dependency>
10 <!--实现自动化配置-->
11 <dependency>
12     <groupId>org.springframework.boot</groupId>
13     <artifactId>spring-boot-autoconfigure</artifactId>
14 </dependency>
15 <dependency>
16     <groupId>org.apache.poi</groupId>
17     <artifactId>poi</artifactId>
18     <version>4.0.1</version>
19 </dependency>
20 <dependency>
21     <groupId>org.apache.poi</groupId>
22     <artifactId>poi-ooxml</artifactId>
23     <version>4.0.1</version>
24 </dependency>
25 <dependency>
26     <groupId>org.apache.poi</groupId>
```

```
27     <artifactId>poi-ooxml-schemas</artifactId>
28     <version>4.0.1</version>
29 </dependency>
30 <!-- 添加servlet依赖模块 -->
31 <dependency>
32     <groupId>javax.servlet</groupId>
33     <artifactId>javax.servlet-api</artifactId>
34     <scope>provided</scope>
35 </dependency>
36
37 * 复制Excel相关工具类
38 * 自动配置代码
39 @Configuration
40 @ConditionalOnClass(ExcelExportUtil.class)
41 @ConditionalOnProperty(
42     prefix = "excel", name = "auto", havingValue = "true", matchIfMissing = true
43 public class ExcelAutoConfiguration {
44     @Bean
45     public ExcelExportUtil excelExportUtil(){
46         ExcelExportUtil excelExportUtil=new ExcelExportUtil();
47         return excelExportUtil;
48     }
49     @Bean
50     public ExcelImportUtil excelImportUtil(){
51         ExcelImportUtil excelImportUtil=new ExcelImportUtil();
52         return excelImportUtil;
53     }
54 }
55
56 * 在resources目录新建META-INF
57 org.springframework.boot.autoconfigure.EnableAutoConfiguration=\
58 com.lg.excel.cfg.ExcelAutoConfiguration
59
60 * mvn -install
61 * 使用
62 <dependency>
63     <groupId>com.lg</groupId>
64     <artifactId>excel-spring-boot-starter</artifactId>
65     <version>0.0.1-SNAPSHOT</version>
66 </dependency>
```

```
67 * 配置
68 #excel:
69 # auto: false
```

* 能够掌握SpringBoot集成Swagger



```
1 * 案例
2 * 添加依赖
3 <dependency>
4   <groupId>io.springfox</groupId>
5   <artifactId>springfox-swagger2</artifactId>
6   <version>2.6.0</version>
7 </dependency>
8 <dependency>
9   <groupId>io.springfox</groupId>
10  <artifactId>springfox-swagger-ui</artifactId>
11  <version>2.6.0</version>
12 </dependency>
13 * 入口配置类
14 @Configuration
```

```

15 @EnableSwagger2
16 public class Swagger2Config {
17     @Value("${swagger.enabled}")
18     private boolean enableSwagger;
19
20     @Bean
21     public Docket createRestApi(){
22         return new Docket(DocumentationType.SWAGGER_2)
23             .apiInfo(apiInfo())
24             .enable(enableSwagger)
25             .select()
26             .apis(RequestHandlerSelectors.basePackage("com.lg.controller"))
27             .paths(PathSelectors.any())
28             .build();
29     }
30
31     private ApiInfo apiInfo(){
32         return new ApiInfoBuilder().title("亮哥教育Restful接口文档")
33             .description("接口说明")
34             .termsOfServiceUrl("http://localhost:8080/")
35             .contact(new Contact("亮哥", "http://localhost:8080/", "386777981@163.com"))
36             .version("1.0.0")
37             .build();
38     }
39 }
40 * yml文件配置
41 swagger:
42     enabled: true
43 * 接口注释
44 @RestController
45 @RequestMapping("/test1")
46 @Api(value = "RestFul测试接口",tags = "RestFulController",description = "测试接口")
47 public class RestFulController {
48
49     @PostMapping("/insert")
50     @ApiOperation(value = "添加用户",notes = "添加用户")
51     public String testPost(
52         @ApiParam(required = true,name = "user",value = "用户实体") User user){
53         System.out.println(user);
54         return "success";

```

```
55     }
56     @GetMapping("/get/{id}")
57     @ApiImplicitParam(name = "id",value = "用户id",
58         required = true,dataType = "Integer",paramType = "path")
59     @ApiOperation(value = "查询用户",notes = "查询用户")
60     public String testGet(@PathVariable("id") Integer id){
61         System.out.println("Get:"+id);
62         return "success";
63     }
64
65     @PutMapping("/update/{id}")
66     @ApiOperation(value = "更新用户",notes = "更新用户")
67     @ApiImplicitParam(name = "id",value = "用户id",
68         required = true,dataType = "Integer",paramType = "path")
69     public String testPut(@PathVariable("id") Integer id,User user){
70         System.out.println(id+": "+user);
71         return "success";
72     }
73     @DeleteMapping("/delete/{id}")
74     @ApiOperation(value = "删除用户",notes = "删除用户")
75     @ApiImplicitParam(name = "id",value = "用户id",
76         required = true,dataType = "Integer",paramType = "path")
77     public String testDelete(@PathVariable("id") Integer id){
78         System.out.println("Delete:"+id);
79         return "success";
80     }
81 }
82
83 * 启动应用
84 * 访问: http://localhost:8088/lg/swagger-ui.html
```

亮哥教育Restful接口文档

接口说明

Created by 亮哥
See more at <http://localhost:8080/>
[Contact the developer](#)

employee-controller : Employee Controller	Show/Hide	List Operations	Expand Operations
goods-controller : Goods Controller	Show/Hide	List Operations	Expand Operations
hello-controller : Hello Controller	Show/Hide	List Operations	Expand Operations
index-controller : Index Controller	Show/Hide	List Operations	Expand Operations
RestFulController : 测试接口	Show/Hide	List Operations	Expand Operations
DELETE /test1/delete/{id}			删除用户
GET /test1/get/{id}			查询用户
POST /test1/insert			添加用户
PUT /test1/update/{id}			更新用户
sys-log-controller : Sys Log Controller	Show/Hide	List Operations	Expand Operations
test-controller : Test Controller	Show/Hide	List Operations	Expand Operations
thymeleaf-controller : Thymeleaf Controller	Show/Hide	List Operations	Expand Operations
transfer-employee-controller : Transfer Employee Controller	Show/Hide	List Operations	Expand Operations

[BASE URL: /lq , API VERSION: 1.0.0]

RestFulController : 测试接口

Show/Hide | List Operations | Expand Operations

DELETE /test1/delete/{id} 删除用户

Implementation Notes

删除用户

Response Class (Status 200)

string

Response Content Type

/ ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	1	用户id	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
204	No Content		
401	Unauthorized		
403	Forbidden		

[Try it out!](#) [Hide Response](#)

Curl

```
curl -X DELETE --header 'Accept: text/plain' 'http://localhost:8088/lq/test1/delete/1'
```

Request URL

```
http://localhost:8088/lq/test1/delete/1
```

Request Headers

```
{  "Accept": "*/*"}
```

Response Body

```
success
```

Response Code

```
200
```

* 总复习一