

一、统一返回数据格式

项目中我们会将响应封装成json返回，一般我们会将所有接口的数据格式统一，使前端(iOS, Android, Web)对数据的操作更一致、轻松。

一般情况下，统一返回数据格式没有固定的格式，只要能描述清楚返回的数据状态以及要返回的具体数据就可以。但是一般会包含状态码、返回消息、数据这几部分内容

例如，我们的系统要求返回的基本数据格式如下：

列表：

```
1 {
2   "success": true,
3   "code": 20000,
4   "message": "成功",
5   "data": {
6     "items": [
7       {
8         "id": "1",
9         "name": "刘德华",
10        "intro": "毕业于师范大学数学系，热爱教育事业，执教数学思维6年有余"
11      }
12    ]
13  }
14 }
```

分页：

```
1 {
2   "success": true,
3   "code": 20000,
4   "message": "成功",
5   "data": {
6     "total": 17,
7     "rows": [
8       {
9         "id": "1",
10        "name": "刘德华",
```

```
11         "intro": "毕业于师范大学数学系，热爱教育事业，执教数学思维6年有余"
12     }
13 ]
14 }
15 }
```

没有返回数据：

```
1 {
2     "success": true,
3     "code": 20000,
4     "message": "成功",
5     "data": {}
6 }
```

失败：

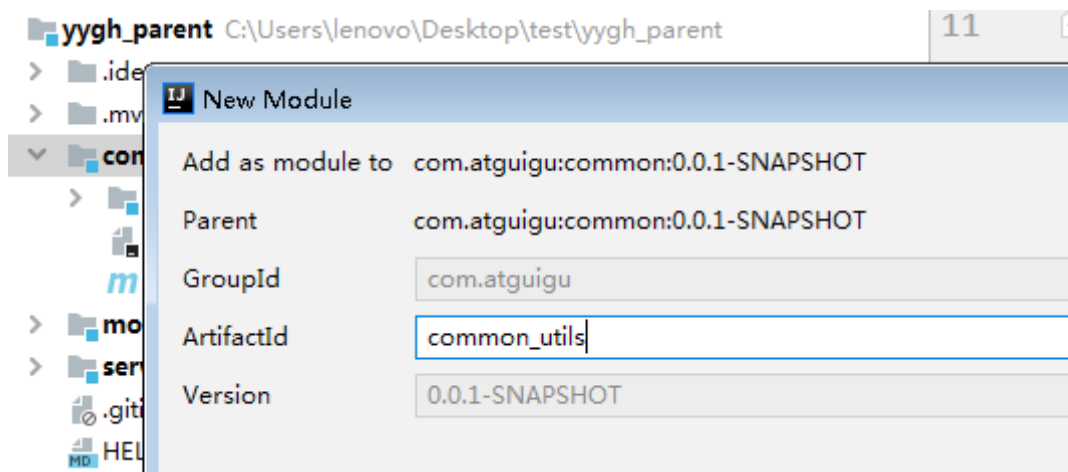
```
1 {
2     "success": false,
3     "code": 20001,
4     "message": "失败",
5     "data": {}
6 }
```

因此，我们定义统一结果

```
1 {
2     "success": 布尔，//响应是否成功
3     "code": 数字，//响应码
4     "message": 字符串，//返回消息
5     "data": HashMap //返回数据，放在键值对中
6 }
```

二、创建统一结果返回类

1、在common模块下创建子模块common_utils



2、创建接口定义返回码

创建包com.atguigu.yygh.common，创建接口 ResultCode.java

```
1 public interface ResultCode {  
2     public static Integer SUCCESS = 20000;  
3  
4     public static Integer ERROR = 20001;  
5 }  
6
```

4、创建结果类

创建类 R.java

```
1 @Data  
2 public class R {  
3     @ApiModelProperty(value = "是否成功")  
4     private Boolean success;  
5  
6     @ApiModelProperty(value = "返回码")  
7     private Integer code;  
8  
9     @ApiModelProperty(value = "返回消息")  
10    private String message;  
11
```

```
12 @ApiModelProperty(value = "返回数据")
13 private Map<String, Object> data = new HashMap<String, Object>();
14
15 private R(){}
16
17 public static R ok(){
18     R r = new R();
19     r.setSuccess(true);
20     r.setCode(ResultCode.SUCCESS);
21     r.setMessage("成功");
22     return r;
23 }
24
25 public static R error(){
26     R r = new R();
27     r.setSuccess(false);
28     r.setCode(ResultCode.ERROR);
29     r.setMessage("失败");
30     return r;
31 }
32
33 public R success(Boolean success){
34     this.setSuccess(success);
35     return this;
36 }
37
38 public R message(String message){
39     this.setMessage(message);
40     return this;
41 }
42
43 public R code(Integer code){
44     this.setCode(code);
45     return this;
46 }
47
48 public R data(String key, Object value){
49     this.data.put(key, value);
50     return this;
51 }
52
53 public R data(Map<String, Object> map){
54     this.setData(map);
```

```
55         return this;
56     }
57 }
```

二、统一返回结果使用

1、在service模块中添加依赖

```
1 <dependency>
2     <groupId>com.atguigu</groupId>
3     <artifactId>common_utils</artifactId>
4     <version>0.0.1-SNAPSHOT</version>
5 </dependency>
```

2、修改Controller中的返回结果

```
1 //查询所有医院设置
2 @ApiOperation(value = "医院设置列表")
3 @GetMapping("findAll")
4 public R findAll() {
5     List<HospitalSet> list = hospitalSetService.list();
6     return R.ok().data("list",list);
7 }
8
9 @ApiOperation(value = "医院设置删除")
10 @DeleteMapping("{id}")
11 public R removeById(@ApiParam(name = "id", value = "讲师ID", required = true) @PathVariable
12     hospitalSetService.removeById(id);
13     return R.ok();
14 }
```

