



链滴

Java 服务器项目风格推荐 - 统一响应格式

作者: [suruns](#)

原文链接: <https://ld246.com/article/1559527743216>

来源网站: [链滴](#)

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



1.统一响应格式的作用

一方面，让自己的后台代码看起来更加整洁，另一方面，对其他端的对接也比较方便。有时候，你也会需要打印一下返回值，统一的响应格式让你也比较好查找问题。

2.定义

通常把统一响应格式定义为一个类，我一般定义为Result，由于返回值类型的不确定性，实现方式也多种，可以是Object，可以用泛型，也可以实现Map接口，我的方式是Object。

@Data

```
public class Result {  
  
    private Integer code;  
    private String message;  
    private Object data;  
  
    public Result(){}  
  
    public Result(int code, String msg) {  
        this.setCode(code);  
        this.setMessage(msg);  
    }  
  
    public Result(ResponseStatus carStatus) {  
        this.setCode(carStatus.getCode());  
        this.setMessage(carStatus.getMsg());  
    }  
  
    public Result(int code, String msg, Object data) {
```

```

        this.setCode(code);
        this.setMessage(msg);
        this.setData(data);
    }

    public Result(ResponseStatus carStatus, Object data) {
        this.setCode(carStatus.getCode());
        this.setMessage(carStatus.getMsg());
        this.setData(data);
    }

    public static Result of(int code, String msg){
        return new Result(code,msg);
    }
    public static Result of(ResponseStatus status){
        return new Result(status);
    }

    public static Result of(int status, String msg, Object data){
        return new Result(status,msg, data);
    }
    public static Result of(ResponseStatus carStatus, Object data){
        return new Result(carStatus, data);
    }

    public static Result ok(){
        return Result.of(ResponseStatus.OK);
    }
    public static Result ok(Object data){
        return Result.of(ResponseStatus.OK, data);
    }
    public static Result error(){
        return Result.of(ResponseStatus.INTERNAL_SERVER_ERROR);
    }
    public static Result error(String msg){
        return Result.of(ResponseStatus.INTERNAL_SERVER_ERROR.getCode(),msg);
    }
    public static Result badRequest(){
        return Result.of(ResponseStatus.BAD_REQUEST);
    }
    public static Result badRequest(String msg){
        return Result.of(ResponseStatus.BAD_REQUEST.getCode(),msg);
    }
}

```

一些常用的返回可以定义成枚举类型

```

public enum ResponseStatus {

    OK(200,"操作成功"),
    BAD_REQUEST(401,"请求无效"),
    INTERNAL_SERVER_ERROR(500,"服务异常");
}

```

```
private int code;
private String msg;

ResponseStatus(int code, String msg) {
    this.code = code;
    this.msg = msg;
}

public int getCode() {
    return code;
}

public String getMsg() {
    return msg;
}
}
```

3.使用

在写请求接口的时候，可以直接将Result作为返回值。

```
@GetMapping("list")
@ResponseBody
public Result list(@RequestParam Map<String, Object> params){
    return Result.ok(service.list(params));
}
```

响应结果为：

```
{
  "code": 1,
  "message": "操作成功",
  "data": []
}
```

以上就是我自己对统一响应格式的理解，若有不合适的地方，欢迎交流