

写代码的三点感悟

周桢堡

提纲

- 偷懒
- 优雅
- 靠谱

偷懒

- 用别人的轮子
- 造好用的轮子

用别人的轮子

- Guava
 - Base
 - Collection
 - Strings
- OrikaMapper
- JodaTime
- Lombok

造好用的轮子

- `JsonBody`
- `Config`
- `ShortDate`

JsonBody(普通对象)

```
@RequestMapping("object")
@JsonBody
public User object() {
    return new User("alice");
}
```

```
{
  "status": 0,
  "message": null,
  "data": { "name": "alice" }
}
```

JsonBody(异常处理)

```
@RequestMapping("exception")
@JsonBody
public Object exception() {
    throw new IllegalArgumentException("Oops.");
}

{
    "status": -1, // 异常默认为-1
    "message": "Oops",
    "data": null
}
```

JsonBody(JSONP回调)

```
@RequestMapping("callback")
    @JsonBody(callback="cb") // 默认回调函数为
callback
    public String callback() {
        return "bob";
    }
```

GET /callback?cb=hello

```
hello({"status": 0, "data": "bob"})
```


Config(加载资源配置文件)

- 自带reload机制

```
Config config = Config.get("config.properties");
```

```
String name = config.getProperty("name");  
long size = config.getLong("size", 60000);  
boolean online = config.getBoolean("online",  
false);
```

```
Map<String, String> all = config.getAll(); // 只包含有  
值项
```

ShortDate

```
new ShortDate(20160501); // 2016年5月1日  
ShortDate.today();    //  
ShortDate.valueOf("2016-05-01");  
ShortDate.valueOf(new Date());
```

提纲

- 偷懶
- 优雅
- 靠谱

- 代码规范
 - 1. 代码组织规范
 - 2. 注释规范
 - 3. 最佳实践
 - 4. 安全规范
 - 5. 代码格式化

- 重构

提纲

- 偷懶
- 优雅
- 靠谱

- 代码规范
 - 1. 代码组织规范
 - 2. 注释规范
 - 3. 最佳实践
 - 4. 安全规范
 - 5. 格式化代码

- 重构 (看代码)

提纲

- 偷懶
- 优雅
- 靠谱

- 单元测试
- 精准日志
- 监控报警

单元测试/接口测试

- 补充qunit截图

精准日志

- 使用logback, 不要使用System.out
- 打印的日志要能唯一标示出代码运行的业务; 对于接口, 打印出接口的输入和输出
- 对于捕获的异常信息, 需要打印出完整堆栈
- 日志代码不要影响正常业务(尤其注意NPE)
- 注意性能, 使用占位符, 而不是要使用字符串相加

监控报警

- 监控原则
- 监控种类
- 报警

监控原则

- 功能/性能相关的时间和次数
- 资源使用情况
- 系统间的边界

监控种类

- 业务监控
 - a. JVM (线程数, gc情况)
 - b. 接口调用(次数, 执行时间, 成功/失败/异常统计)
 - c. 业务状况(下单量, 支付量等)
 - d. 缓存(命中率).

- 系统监控(TOMCAT外)
 - a. 服务器(CPU, IO,内存,磁盘)
 - b. 网络(连接数, 流量)
 - c. 数据库(耗时, 慢查询)

- 合理设置报警
 - 报警每一个核心指标
 - 把报警当做一个故障

Q && A