

酒店预订系统

测试回顾总结记录



南京大学软院学院

Leftovers

2016-12

**Leftovers团队测试回顾总结会议记录**

## 一、会议时间：2016-12-18

## 二、会议地点：Music Bar

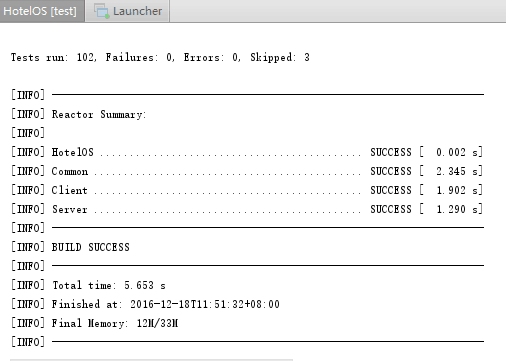
## 三、与会人员：Leftovers全体成员

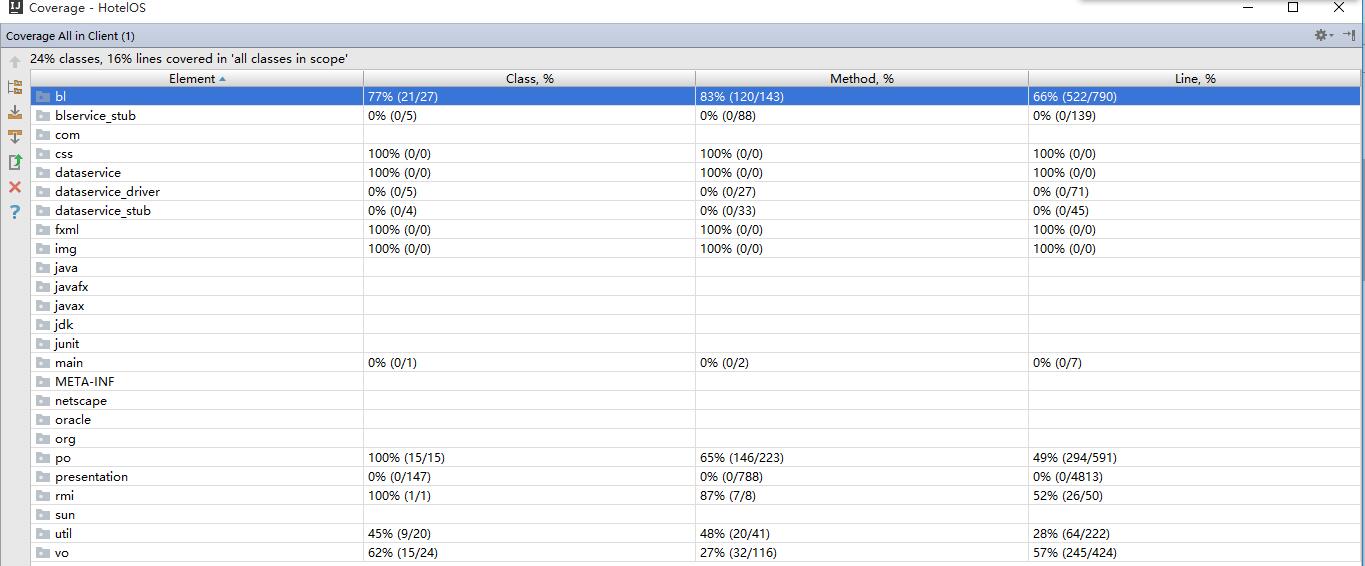
## 四、会议内容：对酒店预订系统的测试回顾

### 1. 测试情况

#### 1.1单元测试

截图





详情见《Test Report》

#### 1.2系统测试，功能测试

详情见《测试总结》

有效性分析：主要使用手工测试，辅助和和测试和白盒测试，比较有效。

## 五、总结：对酒店预订系统的测试回顾

### 1. 教训

#### 1.1任务分工不科学

描述：没有按照功能分工，导致测试互相关联的部分不能顺利串起ui，bl，server部分。

弥补手段：合理分工，最好按照功能分工，是每一位成员都能够独立测试自己的部分。

#### 1.2 mock，stub存储位置错误

描述：没有正确理解mock和stub的意义，导致在实际写bl和controller的时候删除了mock和stub，在之后的集成测试中，出现了大量的错误。

弥补手段：将mock和stub单独放置，便于统一修改。

#### 1.3 controller接口写的过大

描述：没有详细分开各个接口的职责，导致测试过少，实际方法太过庞大，代码显得非常臃肿，重复代码偏多。

弥补手段：在需求和设计阶段详细定义各个接口，接口的定义要经过小组全体的讨论，是之后的工作可以顺利进行。

#### 1.4 po，vo没有事先定好

描述：在bl和ui的对于vo,po层的数据要求不统一，导致修改过于频繁，是测试的方法频繁报错。

弥补手段：vo和po的定义要是想商量好，对于修改，尽量不要涉及po。Vo要有无参数的构造方法。

#### 1.5接口不稳定：

在测试驱动阶段写了一部分测试，但是由于接口不稳定，接口又拆又改，导致测试驱动写的代码浪费了

#### 1.6测试环境搭建不完备

描述：在初期，没有重视测试环境（如数据库环境）的搭建，使得在很长一段时间自动化测试无法正常进行。

弥补手段：在初期重视测试环境的搭建。

#### 1.7时间分配不太合理

描述：任务没有平均分配到所有时间里，在deadline前工作量太大，导致测试的质量不是很高。比如构造阶段最后的作业只留一点时间用于打包，结果打包出现了问题，导致提交的客户端的jar包无法打开。

弥补手段：大家好好分配好时间，多做计划，做好风险预估与预防。

#### 1.8 开会时容易偏离主题

描述：开会讨论时容易专注某个点而忘记了讨论的既定目标，或是讨论着讨论着就偏移到别的地方，导致讨论的效率、质量提高不起来。

弥补手段：在讨论过程中应时刻提醒讨论的目的还有要达成的讨论结果，大家一起注意一下，含糊不清的问题可以先有一个默认的答案，记起来放在一边。

#### 1.9 UI层设计做得不好

关于UI层设计，只能说自己too young too simple，人机交互阶段做的文档形同虚设，当时并没有很好的理解原型图的概念，很多工作都是等到构造阶段才重新规划，花费了太多的时间，设计没有提前做好的另外一个弊端就是维护起来太麻烦了，类似一些重写组件的类可以最大化的归并在一起。

### 2. 经验

1. 使用google，stackoverflow解决了很多技术上的问题，如javafx，数据库操作，数据处理等。

2. 数据库操作等代码可以多写泛型的、较为抽象的方法，可以提高代码质量。

3. ui设计提前做好，图标等素材要统一并备份。图标可在阿里icon上找，配色可在flatuicolors等网站上借鉴。

4. 不必畏惧bug，遇到bug不应该捶胸顿足，多利用断点查找错误，如果是技术上的难题直接StackOverflow（ps：很多bug都是因为简单地复制了一两行代码）

5. 做好方法的防御，因为方法会在各种意想不到的情况下被调用。

6. 委托式编程