8.7-8.13

这周我们终于开始写了一点代码，mentor叫我们去做他项目的UT，然后叫我们去看下Jmockit的内容，这周就主要做了相关的东西。

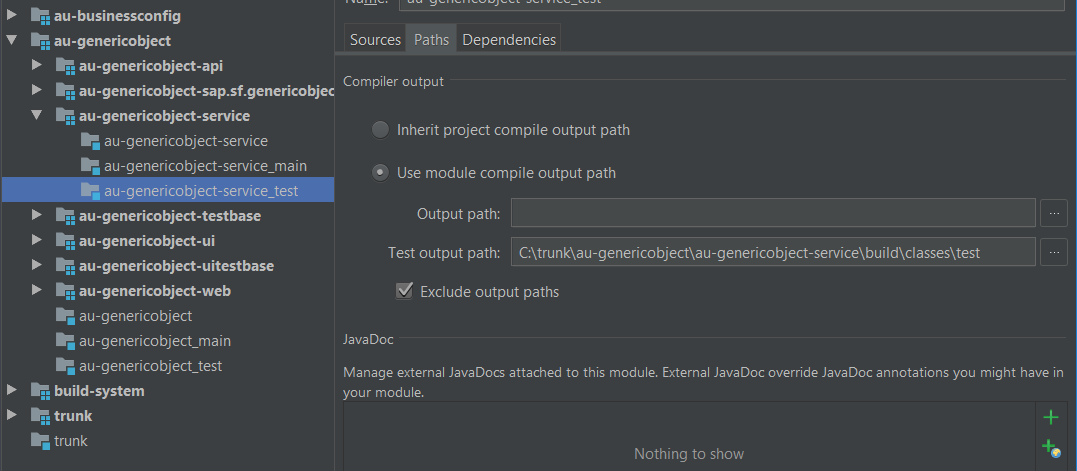
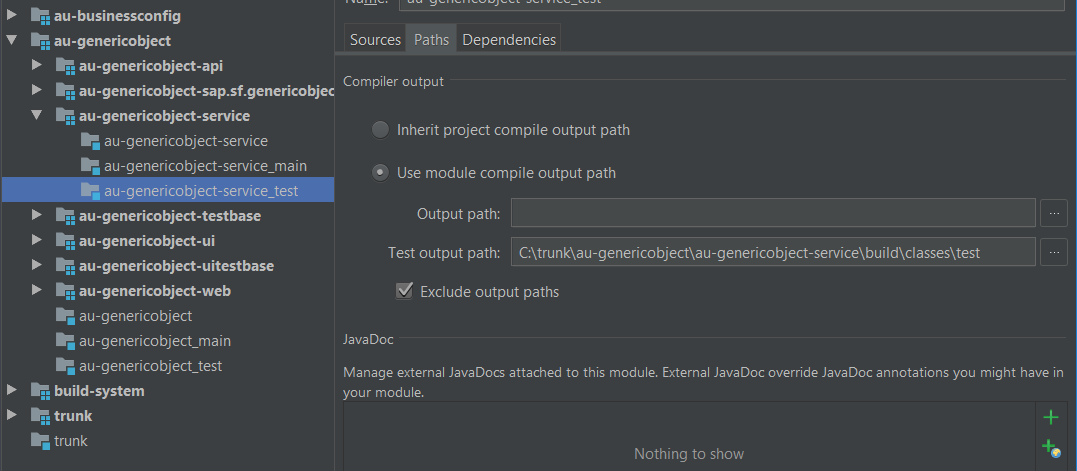
Jmockit的内容刚开始看还不算很简单，因为以前没有做UT的经验，更没有用过相关的框架，所以花了比较久的时间去jmockit官网熟悉这个，熟悉之后还是比较快速的完成了布置的任务。之后才被告知还要考虑代码覆盖率，于是在充分测试覆盖率的情况下又重新写了当前的代码，完成了百分之百的覆盖。

另外，这周还有好几次和我们PO Mike的skype会议，Mike说话比较快，用词又比较专业，所以我对于他讲的内容大概都不太懂，但还是坚持着听，几次会议的折磨之后，还是明白了一些内容，很感动。

按照惯例，把之前总结关于jmockit的坑也放在这里吧。

附件：

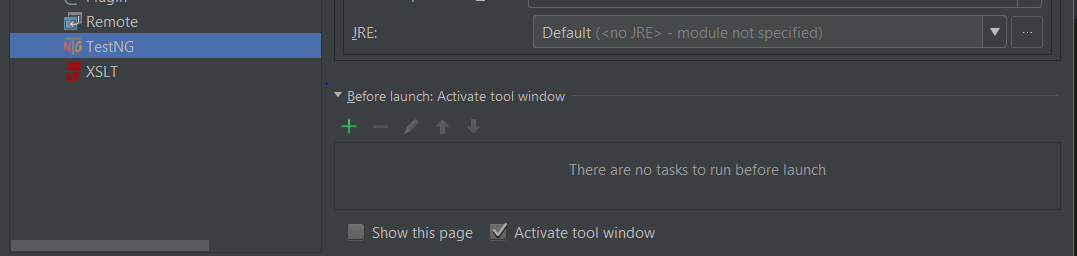
1. 在配jmockit的时候直接在项目的ivy.xml里添加就好，更新gradle会自动导入包，如果没信心可以去C:\Users\I332587\.gradle\caches\modules-2\files-2.1 确认一下有没有org.jmockit。
2. Jmockit run : 可以直接在ide里点击run，但是今天在idea里运行报了找不到某些类的错误，在根目录下gradle compileJava是可以过的，但是里面不行，最后发现需要在project structure对应module的path下观察那个output path，主要看这个目录存不存在以及里面是不是之前compile的结果，我的截图如下：



对应的路径里是有compille的结果的，所以没问题！这个时候update gradle或者compile这个file都不会报错的~

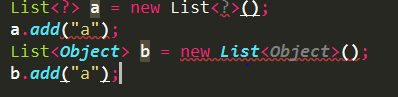
**疑问：我发现似乎每次build之后都需要重新配一下这个位置，而且到现在还不知道gradle是在哪里设定这个outpath的。。。**

3．还有个细节要注意，为了在跑测试的时候不需要每次都build整个项目，需要到run的edit configuration下



这个位置如果有compile就去掉~

4．泛型里List<?>和List<Object>的区别：第一个所有的对象可以取出来，但是都不能进去，第二个都可以存进去。



5．在具体用单元测试框架时选择testng而不是junit，为什么捏~

import ***org.testng.annotations.***Test;

@test

差点忘掉~~~一定要记得把test的东西放在test文件夹下，这个其实是在ivy.xml里的scoop配出来的，如果不放在test下是找不到的哦~

7. 运行单独的一个在gradle 的命令：

gradle :au-genericobject:au-genericobject-service:test --tests com.successfactors.genericobject.test.DRTMPurgeObjectFacadeImplTest

后面如果在加 .方法名 就可以只跑一个方法的test

8. Java运行时产生的匿名内部类叫 类名$n 的形式，有很多原因产生，其中一个可能是由switch，switch支持int和char，如果传入的是一个enum，编译器将生成一个switchMap的内部类，这个类主要将enum映射为数字，在进行switch。

9. 对于java中的可变长参数T…，如果直接传入null，编译不通过。这是因为传入null后java不理解是你不想传入参数还是传入一个null参数，对应产生null还是[null]这种，但是通过类型声明(String a=null,再传入a或者直接(String)null)，都可以通过。