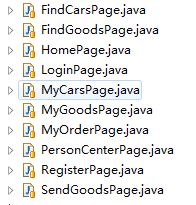
# 一、页面及元素声明规范

## 1.1、类的命名

在项目/com/incito/logistics/pages目录下存放的是测试中所需要的页面元素的java类，一个java类代表着一个页面，如图：



类的命名是这样规定的：类名分2部分，如LoginPage.java，其中的第一部分是：Login表示登录，Page代表页面，合起来就是“LoginPage=登录页面”，其他页面在新建时按照此规则来建立注意每个单词开头大写，保持统一。

## 1.2、页面中元素的声明



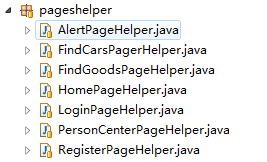
如图,声明之前先加上注释说明此常量(变量,有时候也会用到变量)是干什么的,比如图中的

注释说了是登录页面上的用户名输入框。声明为public（其他类可调用） 静态的 static(类名直接调用) final（常量）常量的命名规范为：

1. 全部大写
2. 常量名字意思明确，比如LP\_INPUT\_USERNAME ，LP表示LoginPage ,INPUT表示这个是一个输入框，USERNAME表示 是用户名的输入框 ，整体意思很明确 别人看你的代码页比较清晰

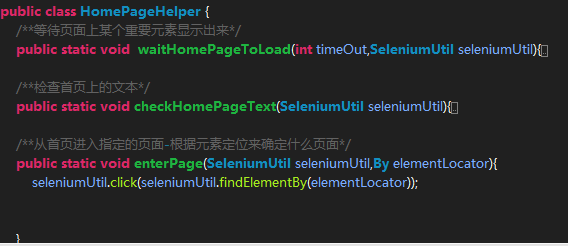
# 二、页面辅助类（XXPageHelper.java）

在项目/com/incito/logistics/pages/pageshelper目录下存放的都是页面帮助类，如图：



什么是页面帮助类？就是在这个页面上做的一个操作，都写在这个helper中。这样你在写用例的时候可以快速定位到你想要的方法在哪里存放，如果说该方法不存在 你可以将此方法在此页面帮助类中加上（千万不要忘记写注释）。

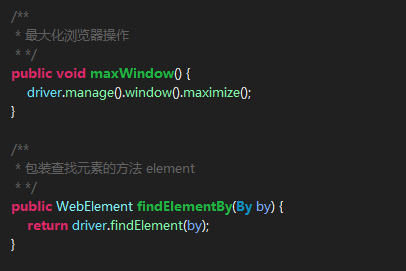
请看下图例子：



图中有三个方法，方法采用**public static**的声明方式，static直接类名调用比较方便，不用额外new对象去占用栈内存。方法名也一定要清晰明确，让人一眼看懂是干嘛的。比如waitHomePageToLoad这个方法看名字就知道 是等待主页加载完毕。每个方法都必须加上注释，这样便于维护代码，也让别人能看懂你的代码。

# 三、包装selenium方法以及通用方法类

在项目/com/incito/logistics/util目录下的SeleniumUtil.java文件，此java类包装了所有和项目有关的selenium操作以及一些通用的方法，目前来说肯定还不够，随着测试用例的增多，还会增加各种各样的selenium操作以及通用方法。注意：在前面说了一个helper类 存放的也有方法，但是那个方法只是针对某个页面使用的，而这里的方法是针对所有的页面都可以适用，比如：click方法。如图：



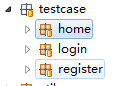
方法名也要清晰，别人看了知道什么意思。加上注释便于维护。

# 四、测试用例（类）的编写规范

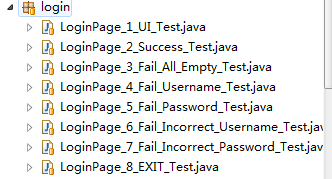
## 4.1、代码存放规则

测试用例存放在/com/incito/logistics/testcase目录下，在testcase目录下又按照功能模块建立对应的包名，如图：下面有 首页、登录、注册三个模块的用例，用例都存放在不同的模块下。

模块：



用例：

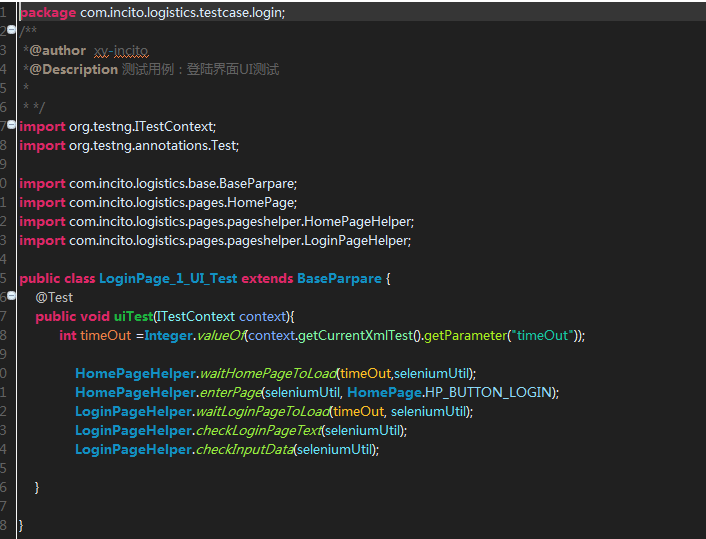


## 4.2、测试用例命名规范

如上图所示，如果你的用例是登录模块的那么命名方式是这样的：

首先说明是登录页面，即开头就是LoginPage，然后接上下划线作为分隔符\_，再接上用例序号（这里的序号是你当前写的第几个用例，而不是手动用例中的序号），然后接上用例的名字，比如UI，那么就是ui测试最后用Test结尾。每个部分之间都有一个分割符“\_”。

## 4.3、用例内容书写规范



如图，开头要加注 作者（author），我们这里规定为：xy-incito-yourname，xy代表襄阳，incito代表公司，yourname代表 你的姓名，比如我的写的就可以写成wy。

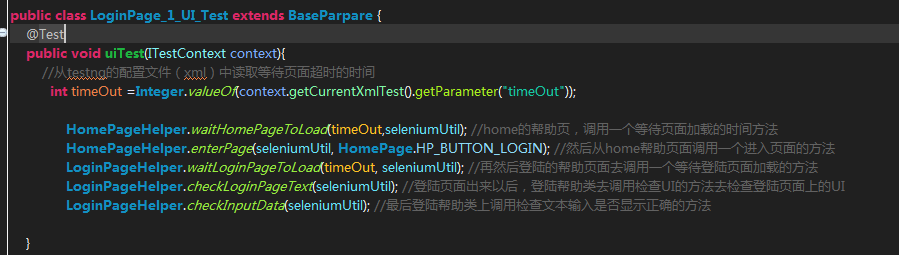
然后还有加描述（Description）来说明你这个用例具体是干嘛的。另外此类要继承BaseParpare类（这个类主要是做初始化：@beforeclass和结束浏览器的操作@after，具体可以查看此类）

其中的：



表示此用例中要用到的一个测试数据，timeOut这里表示等待页面加载的超时时间，这个值存放在testng的配置文件中（本项目在runAll.xml中），运行runAll.xml会自动加载此时间到用例中的变量处。

下面来解析下这个用例，直接看图有说明：



整个测试用例类很简洁，没有过多的代码。