本文目录:

一.InstrumentDriver介绍

二.测试环境的搭建

三.编写测试代码

四.编码及调试中的几个问题

五.如何与kelude集成？

下面开始逐一讲解：

一.InstrumentDriver介绍

InstrumentDriver是淘宝 Mobile自动化小组推出的iphone自动化测试工具，用java编写代码。顾名思义，就是用java代码驱动xcode的自动化录制工具instruments，当然中间是离不开js做“跳板”的。如果你js自以为很强悍的话，也可以直接用js编写相关代码，直接驱动instruments，最后说一下instruments是苹果官方推出的用于录放iphone应用的一款内嵌工具。

InstrumentDriver是CS架构。如果你已经checkout了框架代码至本地，根目录下有个JSLib文件夹，这里就是这个框架的客户端，CSRunner.js这个脚本会调用 TcpSocket.sh 这个文件，这个文件与server，java用例通信，请求用例步骤一步步往下执行，5566就是java用例开的服务端端口，大致原理就这样，要想继续搞清楚，请仔细研究CSRunner.js，TcpSocket.sh这个两个文件，不过我觉得暂时没必要，能立即解决问题才是王道。这里附上一篇介绍文章：<http://www.taobaotest.com/blogs/qa?bid=15425>，大家可以看一下,子雨同学是这个框架的author，有问题也可以直接向他请教。

二.测试环境的搭建：

该段篇幅相对较长，所以列出如下子目录：

1.前置条件

2.安装xcode

3.安装java

4.校验svn

5.Svn界面化访问

6.安装eclipse以及svn插件并获取被测试工程代码

7.新建或直接获取测试工程

8.Build a version

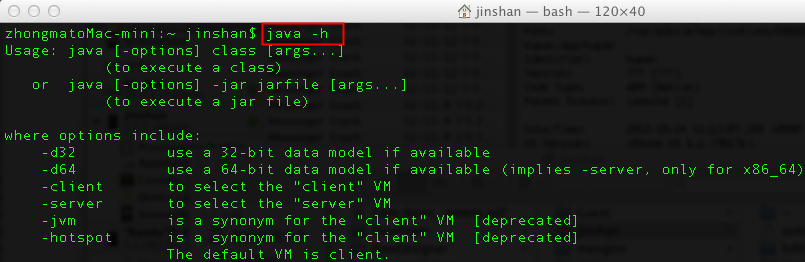
9.本地化设置

下面逐段讲解：

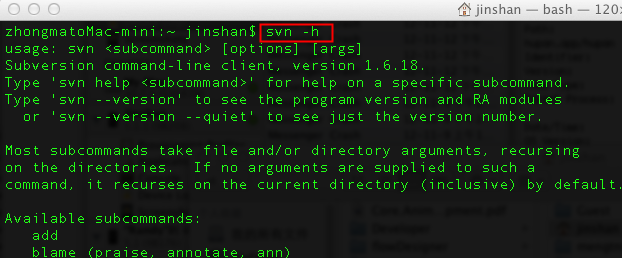
1.前置条件：你已经具备了一台mac或者mac mini，并已经安装了mac os 10.7.5或者更高的版本，或者你已经在pc上安装了同样的虚拟机。

2.安装xcode：直接安装最新的xcode即可，本框架已能兼容iOS6.0。

3.安装java：默认mac系统不会安装java。你在命令行终端直接输入java，并回车，即可进入java安装流程，安装好之后终端输入java –h，校验安装成功。



4.校验svn：mac 默认自带svn，可以在终端输入svn –h来校验。



用svn下载一个版本：

cd ~ //切换至相关目录

Svn co --username xxx –password xxx

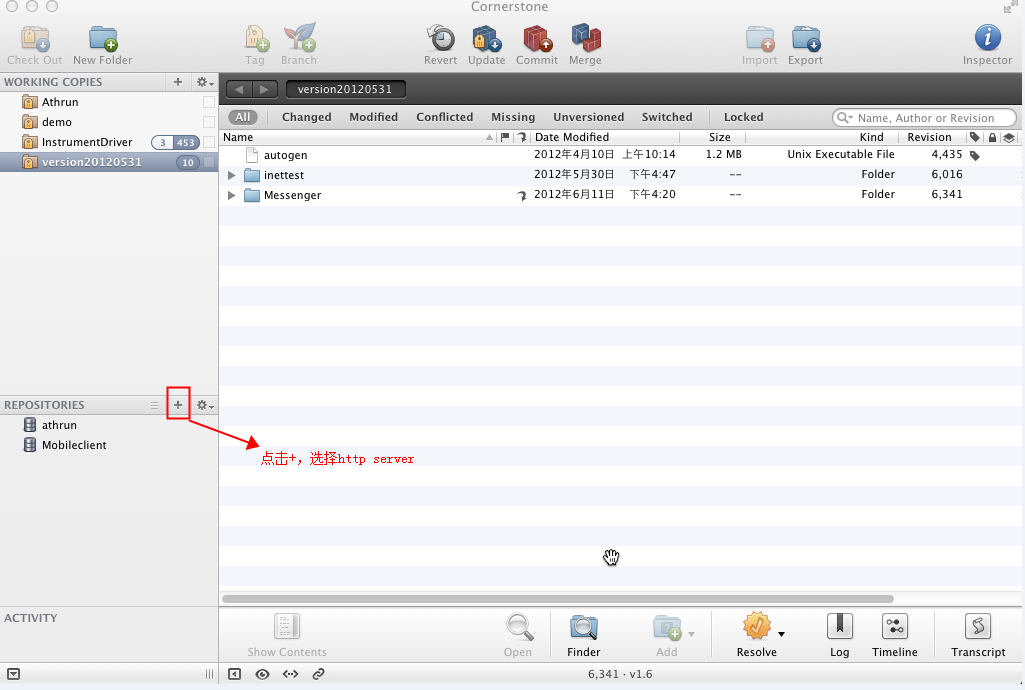
http://svn.app.taobao.net/repos/mobileclient/branches/iOS/version20121031

用svn更新一个版本：

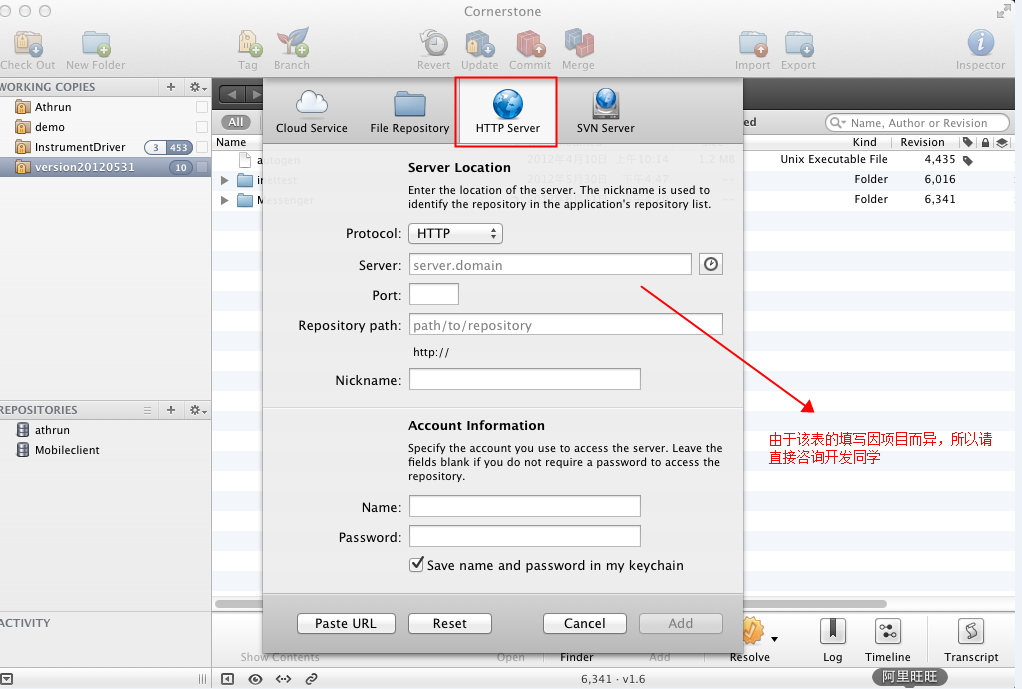
cd ~/version20121031

svn up

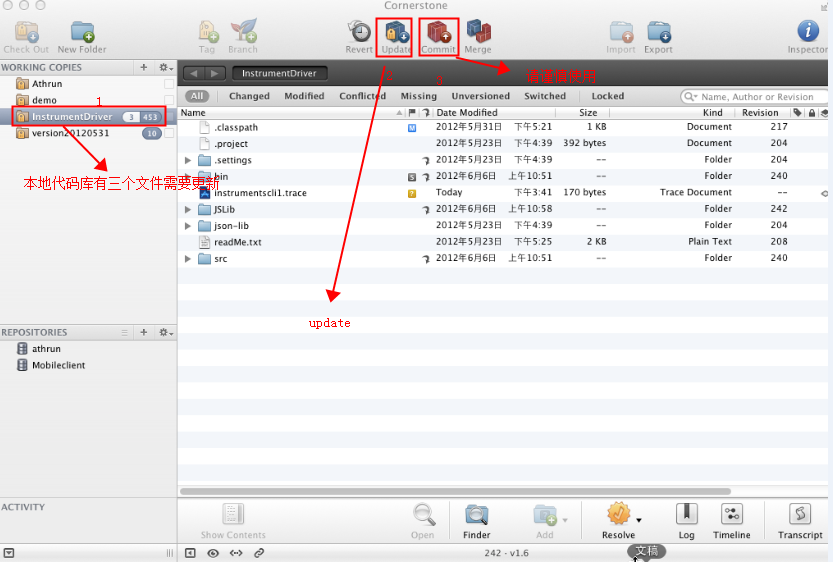
5.界面话的svn工具的使用借助于Cornerstone ，check out一个最新分支，如何使用该软件，参见下列截图



点击后的结果见下图



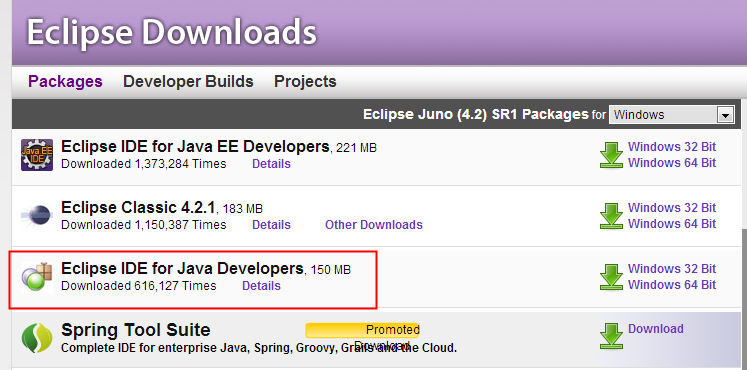
测试框架代码有更新是 如何Update svn代码，参见截图



填写好后点击Add，就可以查看svn上的代码了，再选择自己想要的分支右键checkout到本地。完成了这一步后，我就当你能够而且已经把开发的某个分支下到本地了,也就是被测试工程！

6.安装eclipse以及svn插件:

a.访问<http://www.eclipse.org/downloads/>下载eclipse，如图，选择mac 32位



b.安装svn 插件，在线安装地址：<http://subclipse.tigris.org/update_1.8.x>

c. 此版本的eclipse默认自带maven插件（如果没有，在线安装地址：

<http://m2eclipse.sonatype.org/sites/m2e>），我们只需要在mac本地配置一下即可。具体配置：家目录新建.m2隐藏目录，向框架作者获取配置文件并放在该目录中！

7.新建或直接获取测试工程：如果你项目组内第一个从事自动化的同事，你需要新建maven工程，将其上传至svn，详情参考：

<http://code.taobao.org/p/athrun/wiki/instrumentDriver/kelude/>

如之前已建立测试工程，只需要svn导入eclipse即可，执行一下clean命令，完成maven依赖的下载以及工程的编译工作

8. 前七步都完成后，可以考虑build一个版本供自动化测试使用了。

xcode里编译版本有很多中命令，分别适用于不同的目的。本segment只讲一个命令，profile。

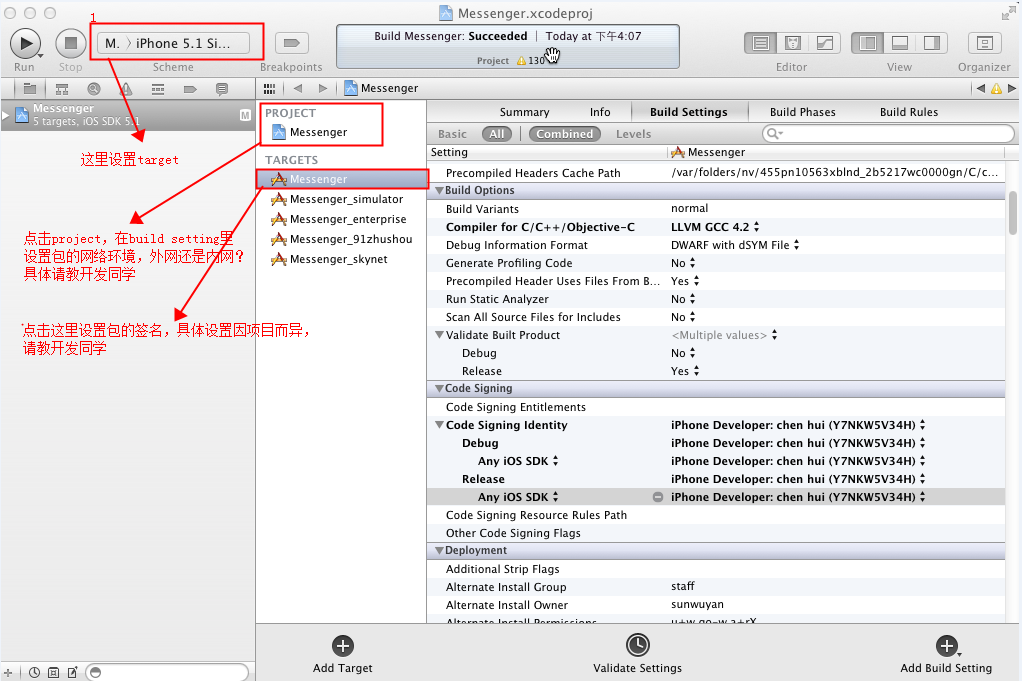
8.1).编译前的设置。假设已在xcode中打开最新分支，你需要设置三项：

a.设置target，应设置为“当前工程名>iphone 5.X simulator”；

b.设置打包的网络环境是内网，还是外网，这个不同项目设置不同，这里不予多说，建议到时直接问开发同学

c.设置包的签名，这个也因项目而异，这里只给出掌上旺信的设置方法，参见截图，其余项目比如湖畔可以询问开发同学

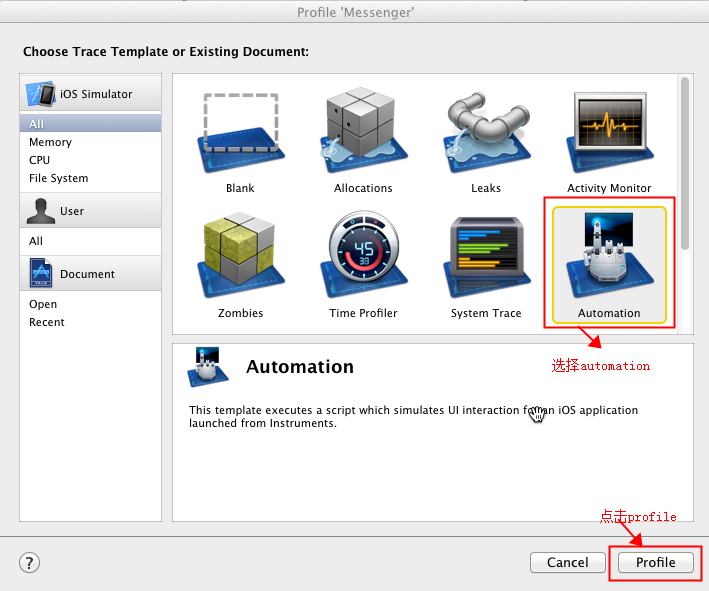
注意：b，c的设置都是在“build settings” tab下里设置的



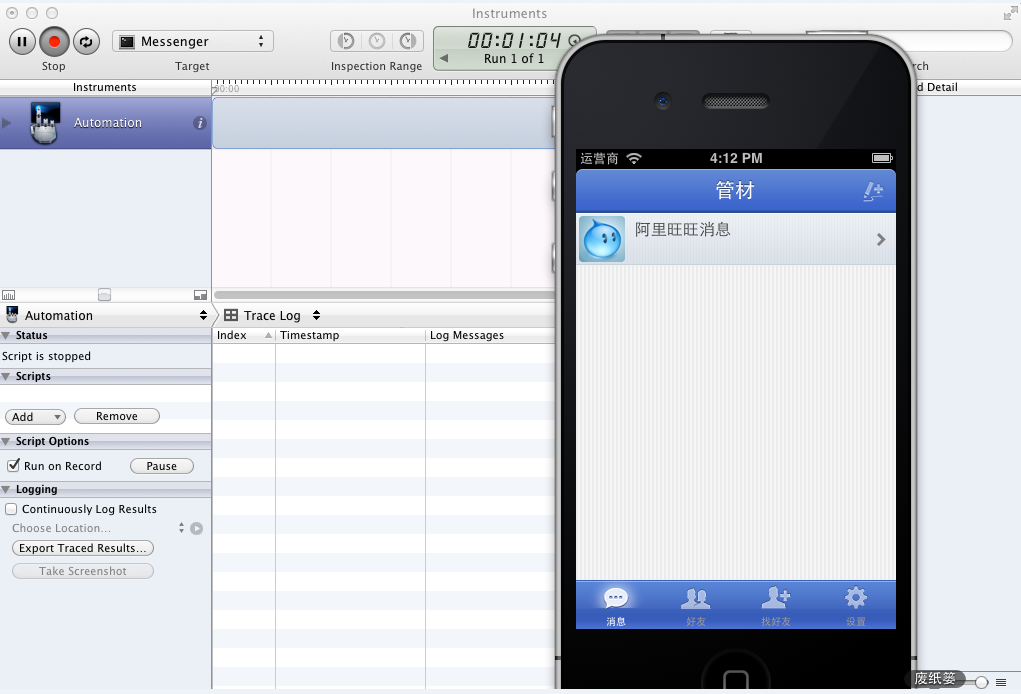
8.2).execute profile

A．一切设置ok后，顶部菜单栏，选择product->profile

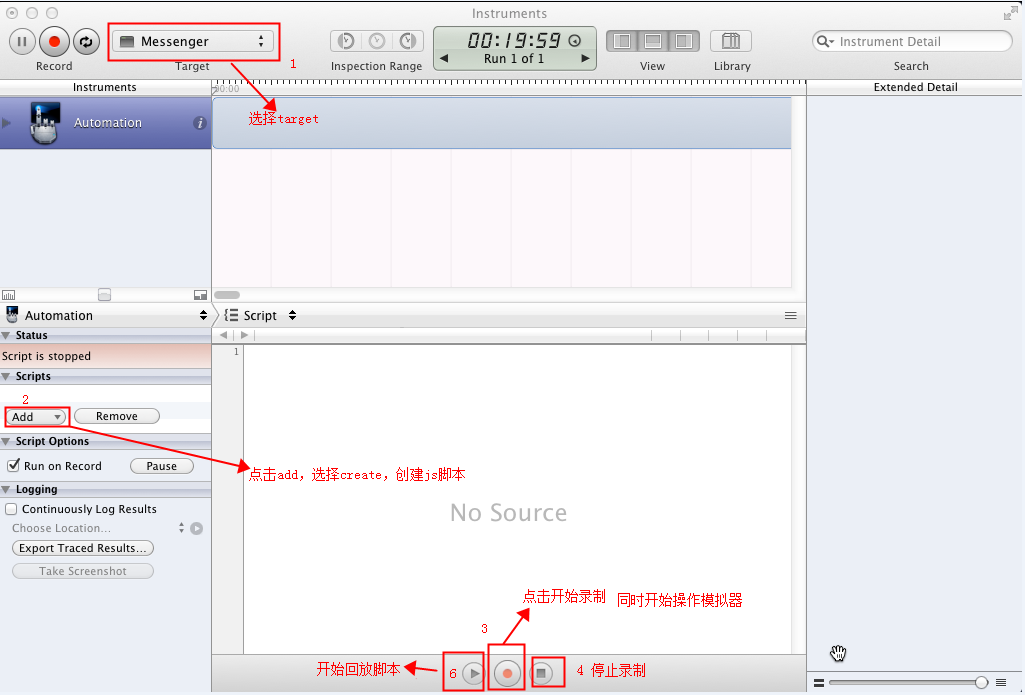
B．如果编译无错，有错请开发同学排错，就可以启动instruments了



C.点击profile后，开始启动instruments 及模拟器



D.这里可以开启iphone自动录放功能的尝试，但这不是重点，有兴趣可以看一下这一步

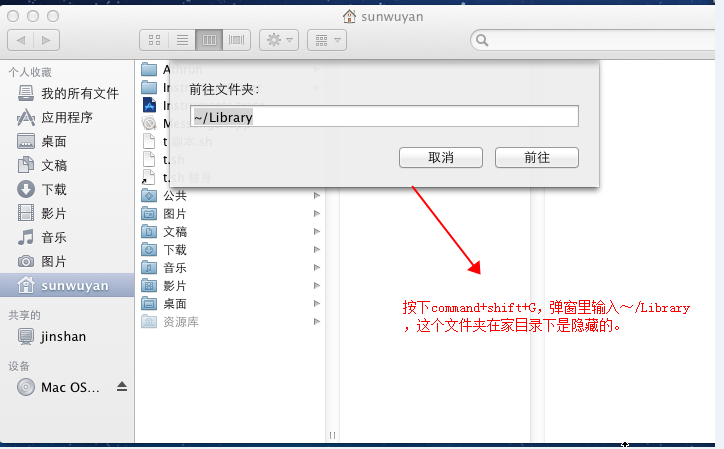


录制后的脚本可以在代码区域右键导出，以便日后回放。

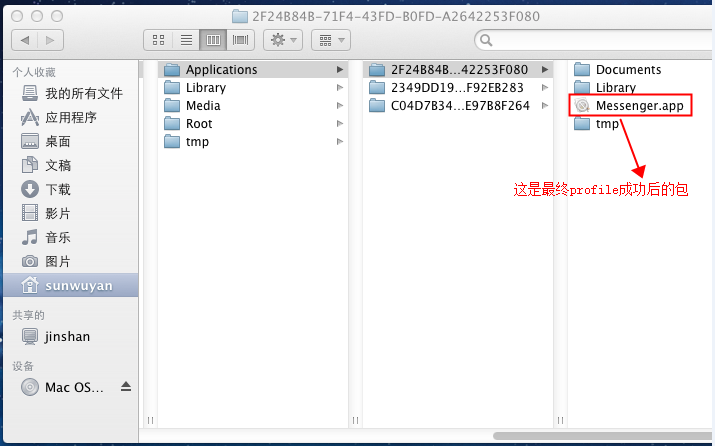
E.如果你打算使用java版的自动化框架，上一步可以帮助你探测元素，但一般情况下我们不需要直接使用instruments，所以现在请退出instruments跟模拟器。到此为止，profile一个版本算是成功了。

8.3).查看并拷贝build成功后的包。

a.找包。第一次build成功后，你未必能找到最终的包，我不保证所有版本的xcode生成的包都跟我说的是一个位置，如果按照下面截图找不到包，建议去询问开发同学。



然后依次选择一系列文件夹Application Support🡪iPhone Simulator🡪5.1🡪Applications,



下面是我机子上的profile成功之后app的位置

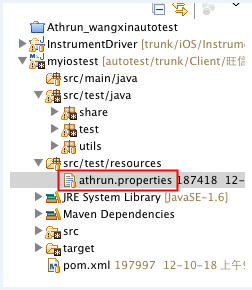
/Users/sunwuyan/Library/Application Support/iPhone Simulator/5.1/Applications/2F24B84B-71F4-43FD-B0FD-A2642253F080，这个是因人而异的，我不能保证此方法是通用的，如果你没找到，还得请教于开发同学，这里再说一遍，最好将其拷贝至家目录下。

b.拷贝包至家目录。因为这种命令生成的包，路径很深，调instruments时有可能会出错，建议采用本文第五大段补充的第四点，将其拷贝到家目录下。

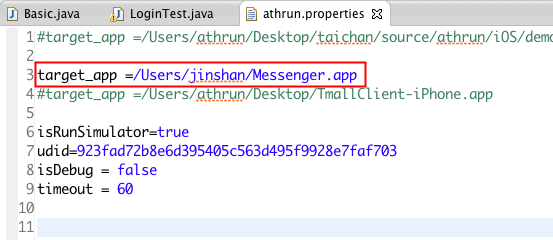
好的，至此打包完成！

9.本地化设置

编好包之后，需要告知InstrumentDriver 目标应用是哪一个？即被测试的app的路径。在测试工程的目录结构里，如下图：



打开，athrun.properties文件，修改其中的target\_app,为你上面第8.3点最后操作所在的路径，如是家目录，不要使用“~”来代替，框架不能识别，如图：



至此自动化测试环境已全部搭建好！

三.编写测试代码

(一).该java文件可以位于工程src文件夹下自己新建的一个包下，tc的代码，参见截图



(二).常用api介绍,详细可以阅读UIAElement这个类的源码，这里只介绍几个常用的接口。

|  |
| --- |
| 如何获取框架源码：  C:\Program Files (x86)\AliWangWang\profiles\cntaobao近山\images\fa\fa858f75fa853b6714a7c1c09d73c48e.png  框架源码svn地址：<http://code.taobao.org/svn/athrun/trunk/iOS/InstrumentDriver> |

1.一旦继承了InstrumentDriverTestCase这个类，那么，变量target，app,win就可以直接使用了，分别代表target对象，当前应用，与当前窗口。

2.UIAElement#findElementByText(String text,Class<T> elementType),"#"表示是实例方法，该方法返回一个包含text文本的T类型的元素，比如

win.findElementByText("消息",UIAButton.class).tap()，返回一个文本为“消息”的按钮,tap()是单击这个按钮的意思，类似于iphone上的“摸一下”；还可以在找到的元素的基础上再找另外的元素，相当于在一个子集下寻找一个元素，比如win.navigationBar().findElementByText("取消",UIAButton.class);这行代码将会在当前窗口的顶部导航栏中查找一个按钮。

3.UIAElement#printElementTree(),比如win.printElementTree(),会打印出当前窗口的所有元素的树结构，这对于确认和定位元素很有帮助。

4.延时处理。页面上需要停留时可以使用target.delay(int sec)，这个对于等待后端响应有帮助。

5.弹窗的捕获与处理，有两种弹出，一是alert弹窗，一是actionsheet。

5.1.捕获alert弹窗，可以使用app.alert()捕获，相应的,如果写成

app.alert().printElementTree(),就可以打印出该弹窗的内部元素结构，从而便于进一步的处理

5.2.捕获actionsheet弹窗，actionsheet是由窗口底部弹起的，可以使用app.actionSheet()捕获,其余的处理同于alert弹窗

6.滚动。代码target.scrollUp(),可以向下滚动半屏高度。

7.校验。虽然框架的查找方法在找不到控件时会自动让tc失败，但如果你想强制tc失败，或者是其他的校验，可以使用junit的Assert类的相关方法来达到目的，比如Assert.fail(),可以直接让tc失败，Assert.assertEqual(String msg,String expected,String actual)可以用来校验字符串相等，详情请参考Assert类。

8.随意点击的实现,直接看图吧！

C:\Program Files (x86)\AliWangWang\profiles\cntaobao近山\images\67\67c9a86d58c668dd7ba6bf4813f13b10.png

(三).碰到无法侦查的元素怎么办？

用profile命令启动instruments，然后尝试录制一下相关的操作，在生成的js代码中查找线索。

具体流程，参考<iOS自动化之环境搭建篇>的第七步第2小点execute profile部分的截图。

另外，有时间的话请多读读框架的源码

(四).如何更好的做好iOS自动化测试。请参考：

<https://developer.apple.com/library/ios/#documentation/DeveloperTools/Reference/UIAutomationRef/_index.html>，常读这里的代码将会使你的iOS自动化测试技术不断走向深处！iOSMonkey的测试技术也来自于此！

(五).框架目前的不足

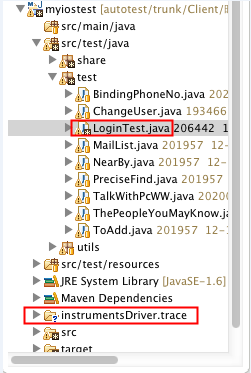
本套框架很容易入门，但随后可能就会发现有些问题比较棘手，比如：

1.头疼的socket超时问题

2.暂未提供某些比较重要的iphone的界面操作接口，比如，开关wifi，滑动删除等等但这些并不影响我们对常规功能实现自动化。

四．编码及调试中的几个问题

1.测试case类名需以Test结尾，不然kelude远程调用不成功！如图，



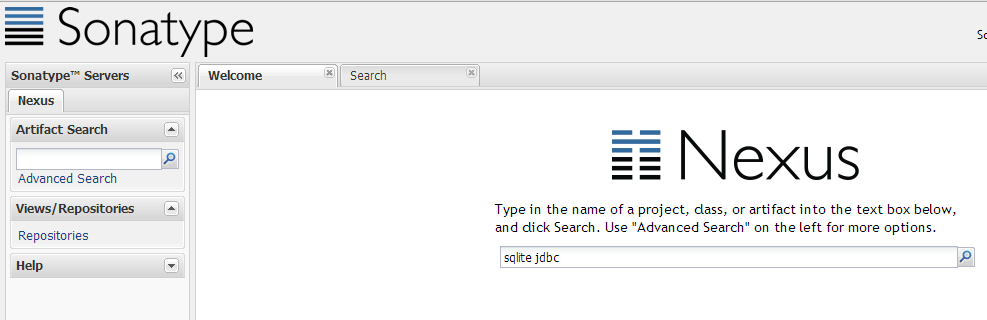
2.将本地测试工程上传svn时 不要上传截图中的另一个文件夹”instrumentDriver.trace”,否则kelude远程调用socket超时的可能性很大。

3.日常调试时偶会出现socket超时问题，只需杀掉模拟器进程重新运行tc即可！

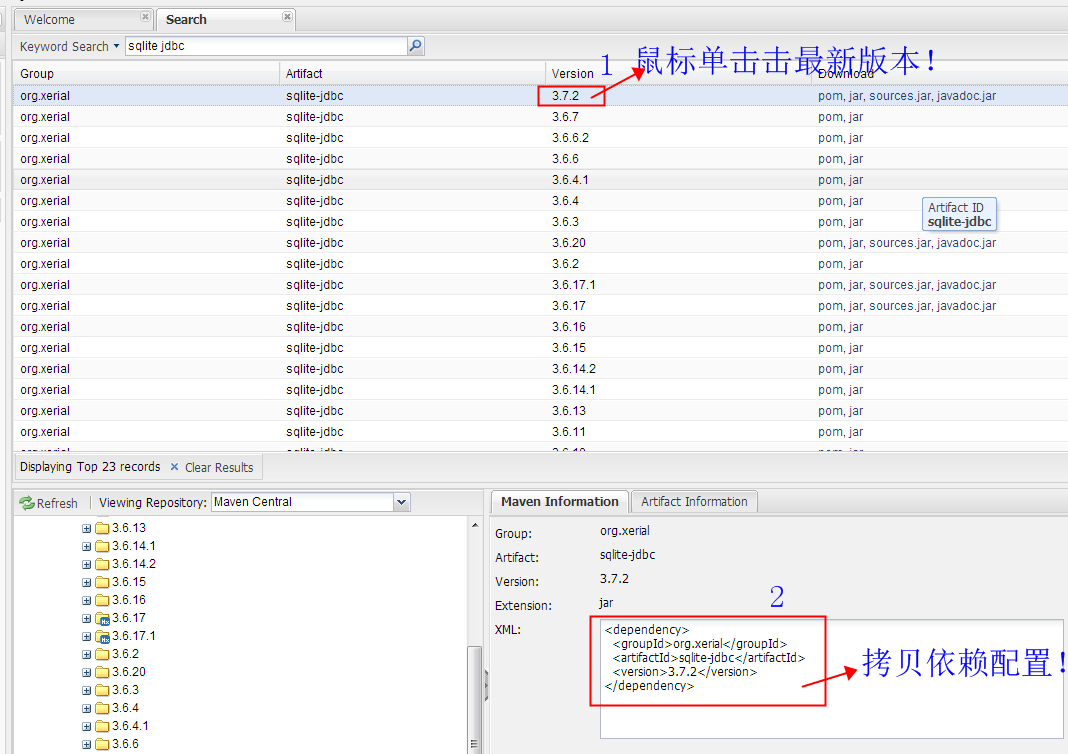
4.载入新的依赖。在编写测试代码时，maven工程往往需要使用一些新的依赖，本文档提供一些相对简单的方法来实现。

比如我想加入java语言的sqlite的jdbc驱动可以如下：

a.访问<http://mvnrepo.taobao.ali.com:8081/nexus/index.html#welcome>，并输入“sqlite jdbc”搜索，如图：



搜索结果页面：



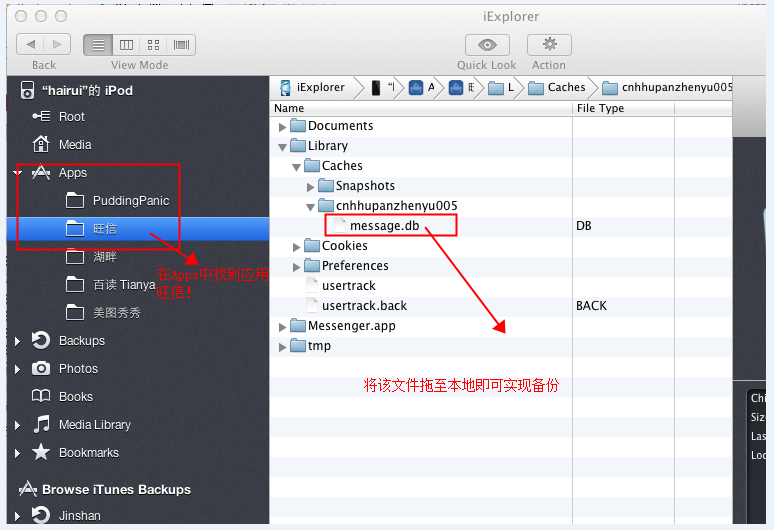
b.在测试工程里打开maven POM文件的xml格式：



c. clean一下测试工程，下载最新的依赖到本地仓库！

5.客户端数据库的备份与访问。

5.1数据库的备份。在一个客户端版本使用了一段时间后，发现需要重装版本因为各种原因，但当前安装上又有很多“珍贵”数据舍不得抹掉，这时就需要对数据库进行备份。可以采用iExplorer以及SQLiteManager这两个工具来实现目的。读者可以去web上下载这些工具的dmg文件安装到本地。具体操作不做详述，其实很简单，可以参见以下截图：



如果本地已经下载安装了SQLiteManager，直接双击该db文件即可打开该数据库！

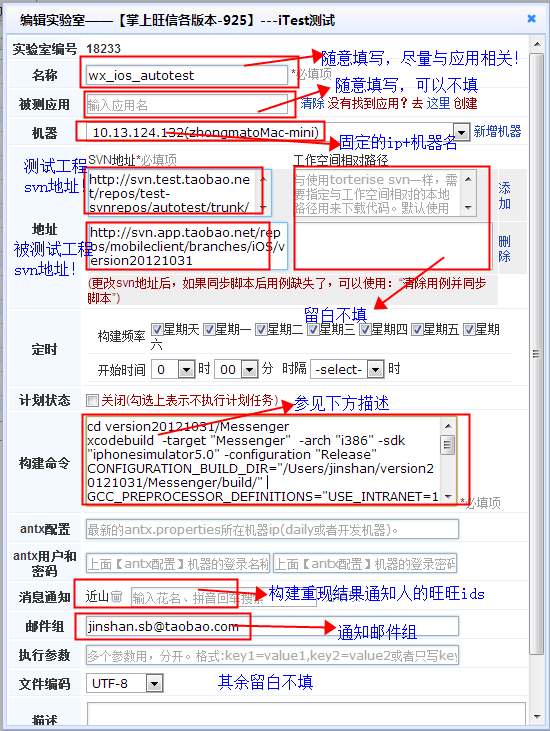
5.2数据库的访问。这里的访问是指访问运行时的模拟器的数据库，目前仍无办法在运行时访问真机的数据库。其实方法很简单，模拟器是安装在本地mac的，那用户登录后生成的数据库文件也是存于本地的，只需要找到该文件，再在测试代码中加入对数据库的操作代码，就可以实现测试工程运行时的客户端数据库数据操作。下面显示按照本文2.8.3节打包方式生成的数据库文件的路径：

/Users/jinshan/Library/Application Support/iPhone Simulator/6.0/Applications/C81F334F-CAC9-4784-8BEF-3E232D25F7C9/Library/Caches/cnhhupan一只展翅待飞的牛虻/message.db

五.如何与kelude集成？

1.可以先参考<http://code.taobao.org/p/athrun/wiki/instrumentDriver/kelude/>

2.详细配置解说：



3.关于构建命令

3.1构建命令实际上是在我们自己的测试机器上的某个目录下执行的，比如这个目录在我的测试机器上是”/Users/jinshan/hifit/workspace/wx\_ios\_autotest18233”,其中的18233是我的测试实验室的id。通过上面配置的两个svn地址，分别下载测试代码与开发代码到测试机器！

如下图！构建命令主要是在开发代码工程文件夹里执行的，在这里是version20121031。



3.2构建命令的核心是xcodebuild命令，该命令的详细使用方法可以请教google！构建命令将会帮助我们编译一个app包，然后调用maven命令也运行测试工程（myiostest）中的测试用例（带有Test后缀的类中的@测试方法）！