**Android自动化测试之Monkeyrunner从零开始**

   作者: qingchunjun

字体:  [小](javascript:doZoom('12');)  [中](javascript:doZoom('14');)  [大](javascript:doZoom('16');)  | [上一篇](http://www.51testing.com/batch.common.php?action=viewnews&op=up&itemid=229485&catid=96) [下一篇](http://www.51testing.com/batch.common.php?action=viewnews&op=down&itemid=229485&catid=96) | [打印](javascript:doPrint();)  | [我要投稿](http://bbs.51testing.com/thread-77515-1-1.html)  | 推荐标签： [软件测试](http://www.51testing.com/?action-tag-tagname-%C8%ED%BC%FE%B2%E2%CA%D4) [自动化测试](http://www.51testing.com/?action-tag-tagname-%D7%D4%B6%AF%BB%AF%B2%E2%CA%D4)

　　最近由于公司在组织一个Free CoDE的项目，也就是由大家自己选择研究方向来做一些自己感兴趣的研究。由于之前我学过一点点关于**[android](javascript:;" \t "_self)**的东西，并且目前android开发方兴未艾如火如荼，但**[自动化测试](javascript:;" \t "_self)**还基本上属于新兴领域，接触的人并不多，所以我给我们团队提出研究android自动化测试方面的项目，果然得到大家一致的赞同，大家的兴致都很高，呵呵。于是二话不说，我们开始看看现在android平台上现有的一些自动化测试工具。经过网上的一番搜索和**[学习](javascript:;" \t "_self)**， 我了解目前android SDK里自带的现成的测试工具有monkey 和 monkeyrunner两个。大家别看这俩兄弟名字相像，但其实是完完全全不同的两个工具，应用在不同的测试领域。总的来说，monkey主要应用在压 力和可靠性测试上，运行该命令可以随机地向目标程序发送各种模拟键盘事件流，并且可以自己定义发送的次数，以此观察被测应用程序的稳定性和可靠性，应用起 来也比较简单，记住那几个命令就行了。而monkeyrunner呢，相比之下会强大一些，它主要可应用于**[功能测试](javascript:;" \t "_self)**，回归测试，并且可以自定义测试扩展，灵活性较强，并且测试人员可以完全控制。

　 　好了，废话不多说，我们直接进入主题吧。之所以想把这个过程写下来，是由于经过我这段时间在网上的搜索和学习，发现关于monkeyrunner的教程 极少，几乎都是把android官方材料自己翻译了一下摆上去的，对于第一次接触的同学来说，可能还不太详细。我在这里就想以完全初学者的角度出来，真正 从0开始，记录我的学习过程。本文仅针对真正的菜鸟，大虾们请飘过哈，嘿嘿。

　　要使用Monkeyrunner，我们首先得干神马呢？当 然得准备使用环境。这个不外乎就是我们得事先安装好android的SDK，配置好开发环境(如Eclipse)，安装好Python(因为 Monkeyrunner其实是由Jython写的，直接调用android API来整的)。这些步骤网上都有非常详细的教程，教大家手把手地搭环境，我这里就不多废话了。

　　环境搭好后，接下来就开始我们的使用之 旅了。网上的monkeyrunner很多都直接从实例代码入手，写得很粗略，也没有把详细步骤说明下来，搞得我确实废了两三天才摸到头绪。所以我这里会 写得很详细，尽量帮助大家能够轻易地看懂。详细步骤如下(在模拟器上运行，由于不知道什么原因图片不能正常显示，所以我还是直接写命令吧)：

　　1. 用Elipse打开andorid的模拟器或者在CMD中用andorid命令打开模拟器，如下：

　　D:\Software\Android SDK2.3\android-sdk-windows\tools>emulator -avd AVD2.3.3

　　Comments: 定位到android SDK的路径下的tools目录，运行上面这个命令。“AVD2.3.3”是我的模拟器的名字，这个命令是每个人自己命名的，用的时候需改成实际的名字。

　 　2. 这个时候，如果一切正常的话，模拟器应该可以启动起来了。(运行monkeyrunner之前必须先运行相应的模拟器，不然monkeyrunner无法 连接设备。这点很多教程上都没有说，我也试了好多次才明白过来)。接着，我们打开另一个CMD窗口(前一个不要关)，还是定位到tools目录，输入命 令"monkeyrunner"，回车，将进入**[shell](javascript:;" \t "_self)**命令交互模式。

　　好了，接下来就可以导入monkeyrunner所要使用的模块了。当然是我们的"From...import..."

　　直接在shell命令中输入： from com.android.monkeyrunner import MonkeyRunner, MonkeyDevice  回车

　　OK,这步完成之后，我们就可以开始和模拟器“勾兑”了，哈哈，命令如下：

　　device = MonkeyRunner.waitForConnection()

　 　没有报错的话，就代表和模拟器勾兑成功，我们可以接着“骚扰”它了。接下来，我们可以在里面装点东西了。输入如下命令，并传入你的apk所在相对路径即 可。我这安装的是著名的ApiDemos。成功了会返回true。如果没有返回true，一般都是语法错误或者你传入的相对路径有问题，仔细检查下就没问 题了。

　　device.installPackage("../samples/android-10/ApiDemos/bin/Apidemos.apk")

　　接下来，我们就可以启动其中的任意activity了，只要传入package和activity名称即可。命令如下：

　　device.startActivity(component = "com.example.android.apis/com.example.android.apis.ApiDemos")

　　此时模拟器会自动打开ApiDemos这个应用程序的主页。

　　接下来我们还可以给模拟器发送更多的操作，如模拟任何按键时间和滚动等，并且可以截图。命令格式和上述命令类似，我就不再累述了。

　 　当然，我们也可以不这样麻烦得一行行地输入命令，可以将从“from...import...”开始的所有后面的语句都写到一个python文件里面， 例如 monkeyrunnerTest.py,然后我们再从命令行直接通过monkeyrunner运行它即可。比如，我们还是用上面的例子，语法如下：

　　D:\Software\Android2.3.3\android-sdk-windows\tools>monkeyrunner monkeyrunnerTest.py

　　接下来monkeyrunner会自动调用monkeyrunnerTest.py，并执行其中的语句，相当方便。

　　今天暂时先写到这里，今天主要的关键是明白了如何使得monkeyrunner和模拟器进行正确的连接和相关的操作，具体的monkeyrunner的方法和属性大家可以参考官方的文档，上面非常详细了，呵呵。