

安卓移动软件自动化测试

家 的 培 专 业 7 训 顾 翔



www.3testing.com Copyright@2005-2016

目录

啄木乌软件测试培训网

录

www.3testing.com

Copyright @2005-2016

E

- •移动软件测试技巧
- 安卓系统测试

▶Andriod 各种UI测试框架介绍 ≻UiAutomator UI自动化测试框架 ▶Android黑盒自动化测试Monkey/MonkeyRunner ▶Selenium/WebDriver的使用 ▶DDMS使用 ▶多平台测试 ▶星云测试 • APP测试讨论

目录

•移动软件测试技巧 •安卓系统测试 E ▶Andriod 各种UI测试框架介绍 ≻UiAutomator UI自动化测试框架 ▶Android黑盒自动化测试Monkey/MonkeyRunner ▶Selenium/WebDriver的使用 ▶DDMS使用 录 ▶多平台测试 ▶星云测试 • APP测试讨论 www.3testing.com Copyright @2005-2016





根据以下条件来确定要测试的设备

• 如果APP针对心率监测,指纹识别,近场通信(NFC),红外线操控选择



- 不同手机
 - 针对IOS设计的APP不测试安卓, 反之亦然
- 设备硬件 · 各种设备比较: www.phonearena.com
- 屏幕尺寸 · 4-5英寸单手操作, 6英寸双手操作



- 手机分辨率千差万别,处理标准16:9,还有15:9
- ratina 图片.在电子显示屏上享受到印刷级的分辨率, (http://Laike.baidu.com/picture/549889/7297520/0/aa18972bd40735f a690f28409e510fb30e24084f?fr=newalbum#aid=0&pic=aa18972bd40735 fa690f28409e510fb30e24084f)



网络切换





出错处理





多任务和意外情况的处理

1, APP正在处理事务,来了个电话,电话挂断,检查APP工

作是否正常?

2, 打开iphone时钟计数器APP, 切换到其他APP一分钟,

检查iphone时钟计数器APP工作是否正常?

3,同时打开QQ音乐与百度音乐,切换到QQ音乐,检查是 否播放QQ音乐;切换到百度音乐,检查是否播放百度音乐







关注用户体验-横竖屏幕测试 为什么还要测试横屏,大屏幕吸引力,但是某些APP只支持横屏,比如三国杀; 或某些APP只支持竖屏,比如微信。



测试试移动的特性 关注用户体验-规范与习惯 设计的APP不要与系统的规范与习惯冲突: 避免手势冲突(右向左滑呼出删除或烫多等): iOS菜单在下面;

啄木鸟软件测试培训网 、ついつの中国移动 🗢 🔅 下午6:47 **1** 89% 更新 已购项目 更新于 2016年4月7日 瑜伽瘦身减压音乐 打开 版本 2.40, 174 MB /Y-高颜值的手机直播平台 打开 版本 5.0.7, 89.3 MB **出发吧-旅行计划,自由行出境游自...** 版本 4.5.0,61.5 MB 新功能 ▼ 打开 **去哪儿睡-遇见你的酒店,旅游度假…** 版本 2.1.0, 16.7 MB 打开 淘宝 - 随时随地, 想淘就淘 版本 5.6.0, 121 MB

目录



•移动软件测试技巧

• 安卓系统测试

▶Andriod 各种UI测试框架介绍

≻UiAutomator UI自动化测试框架

▶Android黑盒自动化测试Monkey/MonkeyRunner

≻Selenium/WebDriver的使用

≻DDMS使用

▶多平台测试

▶星云测试

• APP测试讨论







Andriod应用测试现状:

- 1. 移动应用改变着我们的生活;
- Andriod手机数量全面超过10S系统手机;
- 3. Andriod在各个智能领域逐步扩展;
- 4. Andriod应用的增加,带来测试行业的发展;
- 5. 手工黑盒测试占据约六部分;
- 6. Andriod应用测试从业者逐年增加;

学习人群

- 1 有志从事软件测试行业的人员
- 2. 当前正在从事软件吃的是行业的人员
- 3. 想充电,不断学习的人群









- MonkeyRunner
- 1. 编写语言: Python
- 2. 运行环境: Python 环境, adb链接PC运行
- 3. 测试对象: UI测试
- 4. 测试限制:主要使用坐标,逻辑判断能力差
- MonkeyRunner代码风格示例

device=MonkeyRunner.waitForConnection()#链接手机设备

device.press('KEYCODE_HCivic MonkeyDevice.DOWN_AND_UP)#点击Home键





www.3testing.com

Andriod各种UI测试框架介绍

Instrumentation

编写语言: Python

});

运行环境: adb命令启动或者手机直接启动测试

测试对象:单个Activity测试,需与测试应用相同的签名

测试限制:主要用于白盒测试和UI测试

Instrumentation代码风格示例
 mActive.runOnUiThread(new Runnable(){
 public void run(){
 mSpinner.requestFocus();



- Robotioum
- 1. 编写语言: java 基于Instrumentation封装
- 2. 运行环境:与Instrumentation相同
- 3. 测试限制:与Instrumentation相同

Robotioum代码风格示例

Solo.clickOnMenuItem("Add note");

Solo.goBack();





UiAutomator诞生: 谷歌在Andtopd 4.1推出UiAutomator测试框架 UiAutomator作用 显而易见主要用于UI自动化测试 功能:模拟人对手机的操作;比如点齿,长按,滑动,下拉等操作

优点:

- 1. 编写快速
- 2. API简单易学
- 3. 无Active限值
- 4. 无需签名
- _5. 几乎可以模拟人的所有操作



- 1. 对权限控制不足
- 2. 无法像Instrumentation一样高权限操作应用
- 3. 很多andorid API无法利用





- 为啥要使用UiAutomator?
- 1. 编写灵活, 使用方便
- 2. 可快速学习
- 3. 限制少
- 4. 模拟目前90%以上的手工操作
- 5. 扩展性好





目录



•移动软件测试技巧

- 安卓系统测试
 - ▶Andriod 各种UI测试框架介绍

≻UiAutomator UI自动化测试框架

- ➢Android黑盒自动化测试Monkey, MonkeyRunner➢Selenium/WebDriver的使用
- ≻DDMS使用
- ▶多平台测试
- ▶星云测试
- APP测试讨论





录

UiAutomator UI自动化测试框架

- 环境准备
- 新建测试工程
- 编译和运行
- 运行命令与快速调试
- Uiautomator API详解



目



www.3testing.com

@2005-2016

UiAutomator UI自动化测试框架

环境编写

- •下载JDK,ADT等相关软件为运行UiAutomator
- 1. JDK:1.6以上
- 2. Eclipse
- 3. Android SDK

2,3 合为ADT (And: oid Development Tools-Android开发包)



配置编写环境(Java)

http://jingyan.baidu.com/article/48a42057cca4c3a9242504b8.html

在命令行运行

>javac

出现如图信息安装正确, 否则检查

前面安装

C:\WINDOWS\system32\cmd.exe ndows [Version 6.3.9600] rosoft Corporation. All rights reserved. kennu000>javac avac <options> <source files> here possible options include: Generate all debugging info Generate no debugging info -a:none -g:{lines,vars,source} Generate only some debugging info Generate no warnings -nowarn Output messages about what the compiler is doing -verbose Output source locations where deprecated APIs are -deprecation -classpath <path> Specify where to find user class files and annotati n processors Specify where to find user class files and annotat: -cp <path> n processors -sourcepath <path> Specify where to find input source files -bootclasspath <path> Override location of bootstrap class files -extdirs <dirs> Override location of installed extensions -endorseddirs <dirs> Override location of endorsed standards path Control whether annotation processing and/or compi -proc:{none,only} tion is done .1 Names of the Copyright @2005-2016



配置编写环境(Android)

http://jingyan.baidu.com/article/bea41d437a41b6b4c51be6c1.html;

打开命令框, 输入:

>adb ssh

检验,出现如图信息安装正确. 否则检

查前面安装

	:\WINDOWS\system32\cmd.exe	_ □	X
abb ppp <tty> [parameter Note you should not auto -ty> refers to the tty f [parameters] - Eg. defaul</tty>	s] - Run PPP over USB. matically start a PPP connection. for PPP stream. Eg. dev:/dev/omap_csmi_t troute debug dump local notty usepeerdne	ty1 S	^
db sync notes: adb sync [<localdir> can be interp</localdir>	<directory>] reted in several ways:</directory>		
- If <directory> is not odated.</directory>	specified, both /system and /data partic	tions will	be u
- If it is "system" or " is updated.	data", only the corresponding partition		۰.
environmental variables: ADB_TRACE of the following values	 Print debug information. A comma 1 or all, adb, sockets, packets, 	separated rwx, usb,	list sync
sysdeps, transport, jdwp ANDROID_SERIAL ty over this if given. ANDROID_LOG_TAGS oug tags are printed.	- The serial number to connect to. - When used with the logcat option	-s takes , only the	prior se de
:\Users\kenny000>			~
	Copyr	ight @20	05-20



配置编写环境(ANT环境)

http://jingyan.baidu.com/article/90808022c5eed8fd91c80f90.html;

在命令行中运行

>ant检验

出现如图信息安装正确,否则检 查前面安装







环境检查

- 1. 打开ADT自带的Eclipse;
- 2. 打开菜单Windows->属性->Android
- 3. 如图配置SDK路径



_		⊡ ₹	ebtest.java 🛿 🏼 🧾	mywebtest.java			
	0			Preferences] ×
	type filter ter.	An	droid			⇔ ▼	
	▶ Gener	An	droid Preferences				
	Android	SD	K Location: C:\AE)T∖sdk		Brow	vse
	Help	No	te: The list of SDK	Targets below is	only reloaded once y	ou hit 'App	ly' or '(
	Install/Update	T	arget Name	Vendor		Platform	API .
	⊳ Java	A	ndroid 4.4.2	Android Ope	en Source Project	4.4.2	19
	▹ Kun/Debug ▷ Team						
	▷ XML						
		Sta	andard Android p	latform 4.4.2			
Г					Restore Defaults	Ap	ply
	?				ОК	Cano	el



配置编写环境

练习: 下载并且配置JDK, AND和Am

时间: 30分钟





录

www.3testing.com

Copyright @2005-2016

目

UiAutomator UI自动化测试框架

- 环境准备
- 新建测试工程
- 编译和运行
- 运行命令与快速调试
- Uiautomator API 洋解





新建测试工程

- 1. 打开Eclipse
- 2. 新建一个Java工程,包
- 3. 增加build path
- 4. 新建测试类,继承UiAutomatorTestCase
- 5. 编写用例,方法名公须为test开头
- 6. 编译与运行







建立一个Java Project

Select a wizard Create a Java project				
Wizards:				
type filter text				
Java Project Java Project from Exist	ing + nt Buildfile			
🕸 Plug-in Project				
> 🗁 General				
→ Eclipse Modeling Fran	work	~		
3	< Back Next > Fin	ish Cancel		
3	< Back Next > Fin	ish Cancel	A A A A A A A A A A A A A A A A A A A	ing con



新建测试工程

为Project起个名字,并且选择存放路径

Project name <mark>: Demo1</mark>					
Use default location					
Location: C:\Myproject\Dem	101		Browse		
JRE					
Use an execution enviro	nment . RE: J. vaSE-	1.7	~		



www.3testing.com

Copyright @2005-2016

新建测试工程

确定Project信息



	New Java Project	
ava Settings		
Define the Java build settings.		
Source Projects libraries	Order and coort	
<mark>않</mark> 않 않 ▲		a 🛱 🛱 🖗 🕐
🛯 😂 myAndroidProject		
> 🕮 src		
Details		
Sector is source folder: use the	nis if you want to add a new so	ource folder to your project.
♣ Li <u>ik a ditional source</u> : use this additional source folder.	if you have a folder in the file s	system that should be used as
Add project 'myAndroidProject the root of packages and sourc used for building.	<u>' to build path</u> : Add the projec e files. Entries on the build pat	ct to the build path if the project is th are visible to the compiler and
Allow output folders for source for	olders	
Default output folder:		
mvAndroidProiect/bin		Browse



新建测试工程

为Project建立一个包

Image: Source Relactor Navigate Search Pro Image: Source Relactor Rel	Window Help 2 • x ⊕ @ • A • @ 2 • x ⊕ @ • A • @ 0 New Java Package	
 a a beno1 b most src b most most state and the state of the state of	Java Package Create a new Java package. Creates folders corresponding to packages. Source folder: Demo1/src Name: com.jikexueyuan Create package-info.java	
	Finish Cancel 0 i ams E concel 0 concel Cancel	ion Type Www.3testing.com Copyright @2005-2016



新建测试工程

建立临时文件目录,命名为libs

Source Refactor Navigate Sea	arch Project Run	Window Help	Java - Eclipse
Explorer %	0 • 9 <u>6</u> • × #	8 @ • <i>*</i> • @	ĝ.•ĝ.•© ¢
New Go Into Open in New Window Open Type Hierarchy Show In Copy Copy Qualified Name Paste Delete Build Path Source Refactor Import Export	F4 Alt+Shift+W + Ctrl+C Ctrl+V Delete Alt+Shift+S + Alt+Shift+T +	 Java Project Android Application Project Package Class Interface Enum Annotation Source Folder Java Working Set Folder File Untitled Text File JUnit Test Case 	a Project
Refresh Close Project Assign Working Sets Run As Debug As Profile As Team Compare With Restore from Local History.	F5	Citer	Ctrl+N arstion Resource

New Folder		_ 🗆 ×
ol%er		
Create a new folder resource.		
iter o select the parent folder:		
nyAndroidProject/src		
1 ~ \$		
😂 myAndroidProject		
🗁 .settings		
🗁 bin		
🗁 src		
older name: libs		
44 Advanced H		
Use default location		
 Polder is not located in the file system (virtual Folder) C Liele to elemente la setier (Liele d Folder) 		
	Browse	Variables
Choose file system: default \vee		
Resource Filters		
		C 1
?	Finish	Cancel
?	Finish	Cancel
) • • • • • • • • • • • • • • • • • • •	Finish	



新建测试工程

寻找android.jar文件,拷贝这个目录下两个文件android.jar,和 uiautomator.jar

👢 🕨 TI	his PC → Windows (C:) → <mark>adt-bundle-wind</mark>	dows-x86_64-20140321	latform and via 19	
	Name	Date modified	Туре	Size
	🐌 data	2015/6/8 18: 0	File folder	
5	🐌 skins	2015/6/8 18:20	Hin folder	
ces	🐌 templates	2015/6/8 8:20	File folder	
	🛓 android	2014/.1/21 12:11	Executable Jar File	21,330 KB
	build.prop	2111/2/21 12:12	PROP File	2 KB
	framework.aidl	20.4/2/2112:12	AIDL File	3 KB
	sdk.properties	2014/2/21 12:11	PROPERTIES File	1 KB
5	source.properties	20 4/2/21 12:11	PROPERTIES File	1 KB
s	🛓 uiautomator	2014/2/21 12:11	Executable Jar File	10 KB
4				





新建测试工程

粘贴到libs下



🗏 Package Explorer 🛛		⊑⊈¦ ⊜ ▽ ⊓		
🔺 😂 myAndroidProject				
🕮 src				
🗦 🛋 JRE System Lib. a.v [Java]	SE-1.7]			
🖌 🗁 libs			S	
🗎 androioljar			1.55	
📄 uiautomator.jar				
		6	203	www.3testing.com Copyright @2005-2016



新建测试工程

将这两个文件build path,然后进入了Referenced Libraries下

				Ja	va - E
Edit Sour	ce Re	efactor Navigate Search P V • 👔 🕸 • 🔘 •	roject Run Windo	w Help • AP • 🙆 👘	- ŞI
'ackage Explo Demo1	orer X om.jike system ndroid.	kueyuan Library [JavaSE-1.7]			
🏂 ui	au	New			
		Open Show In	F3 Alt+Shift+W •		
	@@@ ₩	Copy Copy Qualified Name Paste Delete	Ctrl+C Ctrl+V Delete		
		Build Path	· ·	Add to Build Path	1
		Refactor	Alt+Shift+T +		•/.
		Import Export			
	60	Refresh Assign Working Sets	F5		
		Run As	1	Javadoc 🚊 Deslaration	
		Profile As	· · ·		F
		Team			
		Compare With			
		Replace With	100		

File Edit Source Refactor Navigate	Search Project Ru
🔁 🛨 🔛 😨 🔊 🦘 🖓 🕶 🚇 🕶 🖷	S 🕶 😂 🔗 🕶
Package Explorer 23	
🔺 😂 myAndroidProject	
🕮 SI -	
JRE System Library [JavaSE-1.7]	
🔺 🛋 Referenced Libraries	
> 🔤 android.jar	
🖓 🔤 uiautomator.jar	
🖌 🗁 libs	
🚔 android.jar	
🗟 uiautomator.jar	
	Copyright @2005-2016



在这个包下建立类 (class)

新建测试工程






www.3testing.com

Copyright @2005-2016

新建测试工程

为Class起个名字,点【Browse...】

Type name is dis letter	couraged. By convention, Java ty	pe names usually start v. th and	unpercase
Source folder:	myAndroidProject/src		Browse
Package:	com.jerry.demo		Browse
Enclosing type:			Browse
Superclass: Interfaces:	java.lang.Object		Add
	EAN N		Remove
Which method stul	 would you like to create? public static void main(String) Constructors from superclass nherited abstract methods] args)	



www.3testing.com

Copyright @2005-2016

新建测试工程

继承UiAutomatorTestCase类

	Choose a type:	
kage:	uia Browse	
Enclosing ty	Matching items: Browse.	
ne: difiers:	UIAutomatorTestCase - com.android.uiautomator.testrunner	
erclass:	, 'to, "ea	
arfaces:	com android uiautomator testrunnroidProject/libs/u_vutomator.iar	
	Remove	9
ich method	OK Cancel	
ich method	? OK Cancel	Remove



新建测试工程

完成后,基本代码框架生成





新建测试工程

练习:新建一个测试工程

时间: 15分钟





UiAutomator UI自动化测试框架

环境准备	囙	
新建测试工程	·	
编译和运行		
运行命令与快速调试		录
Uiautomator API 详解	23 W	ww.3testing.con Copyright @2005-201

编译和运行







编译和运行

🕖 myTest.java 🛛

package com.jerry.demo;

import com.android.uiautomator.testrunner.UiAutomatorTestCase; import com.android.uiautomator.core.UiDevice;

```
public class myTest extends UiAutomatorTestCase {
    public void testDemo()
```

UiDevice.getInstance().pressHome();

建立测试代码,根据.unit3的规则,函数名必须以"test"开头

英





点文件->属性,获得java文件project位置

编译和运行

0	Properties for myTest.java		
	Resource	⇔ ▼ ⇔ ▼ ▼	
英 ource	Path: /Demo1/src/com/jerry/myTest.java		
└────────────────────────────────────	Settin Type: File (Java Source File)		and the second
	Location: <u>C:\Myproject\Demo1\</u> src\com\jerry\myTest.java		
	Size: 273 bytes		
	Last modified: 2015年6月9日卜午5:51:32		
	Attributes:		
	Read only		
	✓ Archive		
	• Default (inherited from co. tainer: GB^{ν})		
	O Other: GBK		
K	>	Restore Defaults Apply	
?		OK Cancel	
	(F)	0	Copyright @2005-2016



编译和运行

打开命令行,进入类路径







编译和运行

C:\Myproject\Demo1>android create uitest-project -n myTest -t 1 -p C:\Myproject\ Demo1 Added file C:\Myproject\Demo1\build.xml

:\Myproject\Demo1>



- 4 😂 Demo1
 - 🔺 🗁 src
 - 🔺 🖶 com.jerry
 - 👌 🗾 myTest.java
 - > March JRE System Library [jdk*
 - A Seferenced Libraries
 - 🤉 👼 android.jar
 - 🤉 🗟 uiautomator.jar
 - 🔺 🗁 libs
 - 🛓 android.jar
 - 🗟 uiautomator.jar
 - 🔊 build.xml
 - local.properties
 project.properties

创建build.xml文件:

>android create uitest-project -n myTest -t 1 -p C:\Myproject\Demo1 获得三个文件



-t选啥通过id来获得: >android list -n 类名 -p 类所在的路径





打开新产生的build.xml中的default="help"->default="build"

编译和运行





编译和运行

通过ant命令进行编译: >ant-buildfile build.xml





编译和运行

在Eclipse启动模拟器

File Edit Refactor Source Navigate Search Project Run Window Help

ď

- 69

📑 🖾 🔽 🔻

📑 🔻 🔛 🗟 📥

🛱 Package Explorer 🛛

Demo

Bis > UiAutomatorDebug [UiAutomatorHelper master]

D myTest.).vo ? D UiAutomatorHelper. java
package com.jerry;

- 🕶 🗣 🞯 🕆 🎥 🔄 💉 🖗 – 🍄 🌛 😜 🗐 🕤 🖢 🖛

pinport android.os.Bundle;

whic class myTest extends HiAuto



编译和运行



点Tools菜单

		Davi	Chatur
	API	Kev.	Status
Android SDK Toc.'s		22	Installed
Android SDi Platfort - tools		19	🗟 Installed
🖉 🖌 Android รมไห้ ชินเปิ่ง-tools		19	🗟 Installed
🖌 🔄 🔁 And oid 4.4.2 (API 19)			
SDK Fiatform	19	3	🗟 Installed
🔲 🗉 ८ हि। 🖉 EABI v7a System Image	19	2	🗟 Installed
Extras			
C P Idroid Support Library		19	🗟 Installed

www.3testing.com Copyright@2005-2016

编译和运行



			4	Android SDK Manag	er	
ackages Tools						
DK Path: C:\adt-bundle-windows-x86_64-2014	0321\s	ch.	ň	And	roid Virtual Dovico Manager	_ 🗆 X
ackages				Anu		
Name AP	Rev	. Status	Android Virtual D	evices Device Definitions		
Tools			List of existing A	id Virtual Devices loca	ated at C:\Users`, ^k enny000,, ndroid,, vd	
Android SDK Tools	22	🗟 Installed	AVD Name	Target Name	r'attorm API Lev C U/ABI	New
Android SDK Platform-tools	19	🗟 Installed	✓ Jerry	Android 4.4.2	4. ' 2 ' 9 ARM (armeabi-v7a)	r h
🖉 🖌 Android SDK Build-tools	19	🗟 Installed				Edit
🗖 🖳 Android 4.4.2 (API 19)						Delete
🖷 SDK Platform 19	3	Installed				Repair
ARM EABI v7a System Image 19	2	Installed				Repair
Extras		-				Details
Android Support Library	19	🖗 Installed				Start
						Refresh
			✓ A valid Androi	d Virtua' Devica. 🗟 A repa	airable Android Virtual Device.	
			× And Did V	tual Device that failed to	load. Click 'Details' to see the error.	





编译和运行

点击new

Create r	new Android Virtual Device (AVE) 🔺				
AVD Name:	Tom					
Device:	3.2" HVGA slider (ADP1) (320 × 480: 1.)	dpi) 💙				
Target:	Android 4.4.2 - API Level 19					
CPU/ABI:	ARM (armeabi-v7a)					
Keyboard:	🗹 Hardware keyboard present					
Skin:	No skin	~				
Front Camera:	None	\checkmark				
Back Camera:	Emulate	~				
Memory Options:	RAN: 512 VM Heap: 16					
Internal Storage:	200	MiB 🖌				
SD Clad:						
	• Size:	MiB 🖌				
	O File:	Browse				

Emulation Options: Snapshot Use Host GPU

Override the existing AVD with the same name



编译和运行



www.3testing.com

Copyright @2005-2016

X

?

Launch Options

选择,	点击Start
	•

List of existing A	ndroid Virtual Devices locate	ed at C:\Users\ken	nv000\.and	droid\avd	Stale display to real	size
AVD Name	Target Name	Platform	API Lev	CPU/ABI	New.	
 ✓ Jerry 	Android 4.4.2	4.4.2	19	ARM (armeabi-v7a)	Edit	
					Screen Size (in): 5.4	1
					Monitor dpi: 14	4
					Details Scale: def	ault
				X	Refresh Wipe user data	
	id Virtual Device. 🗟 A repair	rable Android Virtu	al Device.		Launch from snapsh	ot
✓ A valid AndrcX An Android V	irtual Device that failed to lo	oad. Click 'Details' t	o see the r	51.		
 ✓ A valid Andrc X An Android V 	irtual Device that failed to lo	oad. Click 'Details' t	o see the r		Save to snapshot	
A valid Andrc X An Android V	irtual Device that failed to lo	oad. Click 'Details' t	o see the r		Save to snapshot	







编译和运行

- 命令总结:
- cd C:\Myproject\Demo
- android create uitest-project -n Demo -t 1 -p C:\Myproject\Demo
- ant -buildfile build.xml
- adb push C:\Myproject\Demo\bin\Demo.jar /data/local/tmp
- adb shell uiautomator runtest Demo.jar -c com.jerry.myTest #mytest stesting.com





编译和运行

- 练习:编译和运行一个工程
- 时间: 20分钟





录

UiAutomator UI自动化测试框架

- 环境准备
- 新建测试工程
- 编译和运行
- 运行命令与快速调试
- Uiautomator API详解





目



■ 运行命令说明

	adb shell uiautomate	or runtest <jars> -c <classes> [options]</classes></jars>	
子命令	操作参数	横木	
	<jars></jars>	指定运行的 jar包文件名心子路径/duta/local/tmp/下	
		测试一个类下的所有用例语式:package_name.class_name	
	-c <classes></classes>	测试特定的某个方法:package_name.class_name#method_name	
runtest		可指定多个-c	
		如果不指定,则运行整个jar包下的所有用例	
	nohup	指定此参数,可断开PC进行运行测试用例	
	-e <name> <value-< td=""><td>传入一个键值对到测试程序中,可用于一些需要变化的参数的传入</td></value-<></name>	传入一个键值对到测试程序中,可用于一些需要变化的参数的传入	





修改代码:

package com.jerry; import android.os.RemoteException; import com.android.uiautomator.testrunner.UiAutomatorTestCase; import com.android.uiautomator.core.UiDevice;

```
public class myTest extends UiAutomatorTestCase {
    public void testDemo()
    {
      UiDevice.getInstance().pressHome(); // 核Home 純
    }
```

```
public void testMenu(){
UiDevice.getInstance().pressMenu(); // 衣菜单會
1
```

public void testRecent() throws Remote Sxception{ UiDevice.getInstance().press RecentApps(); //执行最近运行的程序



>adb shell uiautomator runtest Demo.jar -c com.jerry.myTest # testDemo -c com.jerry.myTest # testRecent -c com.jerry.myTest # testMenu

运行命令与快速调试

依次执行com.jerry.myTest # testDemo , com.jerry.myTest # testRecent, com.jerry.myTest # testMenu





运行命令与快速调试

package com.jerry;

import android.os.Bundle;

```
import android.os.RemoteException;
```

import

```
com.android.uiautomator.testrunner.UiAutomatorTestCase;
import com.android.uiautomator.core.UiDevice;
```

```
public class myTest extends UiAutomatorTestCase {
    public void testDemo()
```

```
UiDevice.getInstance().pressHome();
```

sleep(2000);

Bundle bundle=getParams(); / 这系统中获得参数 String phone=bundle.getString('phone"); System.out.println("PHONE: "+phone); public void tostMenu(){ UiDevice getInstance().pressMenu(); sleep(2000);

public void testRecent() throws
RemoteException{
 UiDevice.getInstance().pressRecentApps();
 sleep(2000);





编译后执行

>adb shell uiautomator runtest Demo.jar -c com.jerry.myTest # testDemo





进入: https://github.com





寻找: uiautomatorhelper





找到如图"fan2597/UiautorHelper",进入





建议使用Copy,不要使用Download

🕝 5 commits	₽ 1 branch		© 0	releases	ଙ୍କି 0 contributors
Branch: master - New pull request	New file	Find Ne	HTTPS -	https://github.com/fan25	🔁 🔛 Download ZIP
fanhq result info				Latest co	mmit 4c9954b on 14 Jun 2015
UiAutomatorDebug			result inf	ö	10 months ago
				0203	www.3testing.com Copyright @2005-2016



Wi	Git Repository Exploring - Demo/src/com/ierry/myTest iava - Eclipse Vi O Clone Git Repository – 🗆 🗙	È
r add a	Source Git Repository ick Acc Enter the location of the source repository. ick Acc Location ick Acc URI: https://github.com/fan2597/y/Automator/eliper.git Local File Host: github.com github.com Repository path: /fan2597/UiAutomatorHelper.git on.at Connection Port: ing Authentication ing ing User: Password: ing	<pre>mess I av. Cit.epoc.vyExploring h.er.UiA.tom/torTestCase; i.be.vice; hr:"estC.se {e(); ("phone"); hone);</pre>
itory <u>I Git</u>	Store in Secure Store aging (2) < Back Next. Fit ish Cancel T T	▲Git Reflog D 载 完 毕 www.3testing.com Copyright @2005-2016



www.3testing.com

Copyright @2005-2016

运行命令与快速调试

Import到已有的Project中





type filter text to filter unselected projects			Select All	
✓ 😂 UiAutomatorDebug (C:\Users\kenny000\git\UiAutomatorHelper\UiAutomau				
✓				
Add project to working sets	;			
Working sets:		~	Select	



确定是否导入成功

ackage Explorer 🛛 🕒 🛱 🕼 🍸	🖥 🗋 myTest. java 🛿	Ē	1 🗐 Task
≥ Demo	<pre>package com.jerry;</pre>	^	
4 🕮 src			Find
🖌 🖶 com.jerry	eimport android.os.Bundle;		
> 🛽 mvTest.iava	import android usetterstor textrummer HiAutomatorTestCase:		
→ JRE System Library [JavaSE-1.7]	import com.android.uia tomator. estrumer.orAutomatorrestcase,		
	<pre>public class myTes: e.tepur UiAutomatorTestCase {</pre>		
	public void tes. Demo()		(i) c
			~ Con
local.properties	UID(v).ce.getIns.ance().pressHome();		<u>Con</u>
project.properties	Sindle undle=getParams():		or <u>c</u>
> UiAutomatorDebug [UiAutomatorHelper mas	string phone=bundle.getString("phone");		
› 講 src	Systeout.println("PHONE: "+phone);		E- Outil
IRE System Library [JavaSE-1.7]			+ (
Referenced Libraries			4 1 1
> 🕞 libs	e proli void testMenu() {		
🖪 build.xml	oj evice.getinstance().pressmenu();		
🗟 local.properties	Sieep(2000),		1
🗟 project.properties			1
🗟 runlog.log	<pre>public void testRecent() throws RemoteException{</pre>		· · · · ·
	<pre>UiDevice.getInstance().pressRecentApps();</pre>		⊿ © r
	sleep(2000);	~	•
			1

ww.3testing.com Copyright@2005-2016



查看UiAutomatorHelper.java,它把创建build.xm?,修改build.xml,编译,传输,执行通过程序自动化起来了。

聞 Package Explorer ⊠

- 🔺 😂 Demo
- 🔺 🕮 src
- 🔺 🖶 com.jerry
- 🔅 🗵 myTest.java
- ➡ JRE System Library [JavaSE-1.7]
- ➡ Referenced Libraries
- > 🗁 libs
- 🖻 build.xml
- local.properties
- project.properties
- ▲ 😹 > UiAutomatorDebug [UiAutomatorHelper master]
 - 4 🗿 src
 - 4 📠 com.jikexueyuan.demo
 - 🗉 🛺 Demo1.java
 - 🛛 🖪 UiAutomatorHelper.java
 - ➡ JRE System Library [JavaSE-1.7]
 - ➡ Referenced Libraries
 - 려 libs
 - 🗟 build.xml
 - 🗟 local.properties
 - project.properties
 - 🗟 runlog.log

```
// 1--判断是否有build
public boolean isBuild() {
    File buildFile = paw File("build.xml");
    if (buildFile.exists()) {
        return true;
    // 创建build x. 7
    execCmd('cm'/c android create uitest-prc
           + android id + " -p " + workspace
    return false ·
 / 创 thuild.xml
exec(Ld('cmd /c android create uitest-prc
           + android id + " -p " + "\""+work
// 2---修改build
public void modfileBuild() {
    StringBuffer stringBuffer = new StringBuf
    try {
        File file = new File("build.xml");
        if (file.isFile() && file.exists())
            InnutStreamReader read = new Innu
```

🔝 Problems @ Javadoc 🗟 Declaration 📮 Console 🛛

🗋 myTest. java 🔄 UiAutomatorHelper. java 🛛



运行命令与快速调试



🖻 🔹 💱 🔽 🗖 🗏 Package Explorer 🛛 🔺 😂 Demo 🔺 🗁 src 🔺 🖶 com.jerry D myTest.java CODY UiAutomatorHelper.java → ■ JRE System Library [JavaSE-1.7] Referenced Libraries > 🗁 libs d build.xml local.properties project.properties ▲ 😹 > UiAutomatorDebug [UiAutomatorHelper master] 🔺 舚 src 🔺 📠 com.jikexueyuan.demo > 🛺 Demo1.java › Ila UiAutomatorHelper.java. → ■ JRE System Library [JavaSE-1.7] > al Referenced Libraries > 🕞 libs 💁 build.xml local.properties project.properties 🗟 runlog.log

把UiAutomatorHelper.java拷贝到我的包下




运行命令与快速调试

加入一个main函数



程序在Eclipse中运行





运行命令与快速调试

练习:配置,自动化运行本章节提供的代码

package com.jerry;

import android.os.Bundle;

import android.os.RemoteException;

import com.android.uiautomator.testrunner.UiAutomatorlies:Case;

import com.android.uiautomator.core.UiDevice;

```
public class myTest extends UiAutomatorTestCas@ {
    public void testDemo()
    {
      UiDevice.getInstance().pressHome();
      sleep(2000);
      Bundle bundle=getParams();
      String phone=bundle.getString("phone");
      System.out.println("PHONE: "+phone);
    }
}
```





运行命令与快速调试

```
public void testMenu(){
  UiDevice.getInstance().pressMenu();
  sleep(2000);
```

```
}
```

public void testRecent() throws RemoteException{
 UiDevice.getInstance().pressRecentApps();
 sleep(2000);

```
}
```

```
public static void main(String[] args){
String jarName="Demo";
String testClass="com.jerry.myTest";
String testName="testMenu";
String androidID="1";
new UiAutomatorHelper(jarName,testClass,testName,androidID);
```



录

www.3testing.com

Copyright @2005-2016

目

UiAutomator UI自动化测试框架

- 环境准备
- 新建测试工程
- 编译和运行
- 运行命令与快速调试
- Uiautomator API 洋解



UiDevice 类介绍
1.UiDevice 代表设备状态
2.UiDevice 为单例模式
获取UiDevice实例的方式:
1) UiDevice.getInstance()

2) getUiDevice()

注意: 第二种方式获取UiDevice实例, 当含有该实例的类被别的类调用时会报空指针错误。 package com.uiautomatortest;

import com.android.uiautomator.testrunner.UiAutomatorTestCase;

```
public class TestGetUiDevice1 extends 'J'ActomatorTestCase {
public void press(){
```

```
getUiDevice().pressMenu();
getUiDevice().pressHome();
```



package com.uiautomatortest; import android.os.Bundle; import android.os.RemoteException; import com android uiautomator core UiD

import com.android.uiautomator.core.UiDevice;
import

com.android.uiautomator.testrunner.UiAutoma

public class TestGetUiDevice2 extends UiAutomatorTestCase {

public void testDevice(){

TestGetUiDevice device1=new GetUiDevice();

```
device1.press():
public static void main(String[] args){
    String jarName, testClass, testName,
androidId;
    jarName="DemoTest";
    testClass="com.uiautomatortest.Test";
    testName="testDevice";
    androidId="1";
    new UiAutomatorHelper(jarName,
testClass, testName, androidId);
```



类TestGetUiDevice2里调用类TestGetUiDevice1中包含getUiDevice()的press方法时报空指针错误:java.lang.NullPointerException

如果将TestGetUiDevice1.java中的getUiDevice()换成UiDevice.getInstance()则可以正常运行,修改后的TestGetUiDevice1.java如下:

li

i!

package com.uiautomatortest;

import com.android.uiautomator.core.UiDevice; import

com.android.uiautomator.testrunner.UiAutomator TestCase;

public class TestGetUiDevice1 extends UiAutomatorTestCase {

public void press(){

getUiDevice().pressMenu(); getUiDevice().pressHome();

UiDevice.getInstance().pressMenu(); UiDevice.getInstance().pressHome();





因此目前都是采用UiDevice.getInstance()方式获取UiDevice实例。 3.UiDevice功能

1) 获取设备信息:屏幕分辨率,旋转状态,亮灭屏状态等
 2) 操作:按键,坐标操作,滑动,拖拽,灭屏唤醒屏幕,截图等
 3) 监听功能

按键与KEYCODE使用 一、手机常见按键: 1.HOME 主屏幕键 2.MENU 菜单键 3.BACK 返回键 4.VOLUME_UP音量加键

- 5 VOLUME_DOWN音量减键 5. RecentApps 最近使用APP
- 7. POWER 电源键
- 8. Dpad 上下左右键



二、按键API说明:

返回值	方法名	描述
boolean	pressBace()	模拟短弦返回back键
boolean	pressDPadCenter()	模拟轨迹动中点按键
boolean	pressDPadDown()	模拟轨迹球向下按键
boolean	pressDPadLeft()	模拟轧迹球向左按键
boolean	pressDPadRight()	袄拟轨迹球向右按键
boolean	pressDPadUp()	模拟轨迹球向上按键
boolean	pressDelete()	模拟短按删除delete按键
boolean	pressEnter()	模拟短按回车键
boolean	pressHome()	模拟短按home键
boolean	pressKeyCode(int keyCode, int metaState)	模拟短按键盘代码keyCode
boolean	pressKeyCode(int keyCaste)	模拟短按键盘代码keyCode
boolean	pressMenu()	模拟短按menu键
boolean	pressRecentApost	模拟短按最近使用程序
boolean	pressSearch()	模拟短按搜索键



Uiautomator API详解

ſ

举例:	public void testMenu(){
package com.uiautomatortest;	UiDevice.getInstance().pressMenu();
import android.os.Bundle;	sleep(2000);
import android.os.RemoteException;	ξ
import com.android.uiautomator.core.UiDevice;	
import	public void testRecent() throws
com.android.uiautomator.testrunner.UiAເປລາກator Test	RemoteException{
Case;	
public class Test extends UiAutomator TestCase {	UiDevice.getInstance().pressRecentAp
public void testHome(){	ps();
UiDevice.getInstance().pressHome();	sleep(2000);
sleep(2000);	}
1	1

ſ

www.3testing.com Copyright@2005-2016



- 三、KEYCODE 键盘映射码:
 - 1) KeyEvent 按键事件
 - **2) META KEY**

辅助功能键: ALT、SHIFT、CAPS_LOCK

列	激活状态	metaState
base	META_key未被激活	0
caps	Shift或Caps Lock被激活	1
fn	Alt被激活	2
caps_fn	Alt、Shitt或Caps Lock同时被激活	3



public void testKeyCode(){

UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_A);//小写a UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_G);//小写b UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_C);//小写c

UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_A,1);//大写A UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_B,1);//大写B UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_C,1);//大写C



}





在Eclipse中添加 DDMS





eclipse中如何查看一个android模拟器的内部文件

- eclipse中如何查看一个android模拟器的内部文件,有时要在其中添加一个文件夹或是什
- 么的,要手动的做这件事,而不能够用代码去完成时,就要用这个方法了。
- 1. 首先, 打开一个安卓模拟器。
- 2. 这个时候,点击eclipse中DDMS图标,打开android模拟器的控件台。
- 3. 在这里,你就能够看到这个模拟器了,同时它的一些状态,你也是可以看到的。
- 4. 下面就点击右侧file explore就可以看到其中的文件系统了,在这里面,你就可以看到你要的文件了。

获取坐标与坐标点击

一、坐标相关的知识:

1) 手机屏幕坐标: 左六角开始到右下角结束

- 2) DP: 设备独立 像素, 例如320像素显示到640像素上要拉伸一倍



二、坐标相关:API

返回值	方法名	描述
boolean	click(int x, int y)	使用坐际点击屏幕
int	getDisplayHeight()	获取屏幕高度
Point	getDisplaySizeDP()	获取显示尺寸返回显示大小(设备独立像素)
		屏幕旋转返回的显示大小调整
int	getDisplayWidth()	获取屏幕宽度

三、UiAutomator Viewer获取屏幕快照

进入android SDK的tools目录下找到uiautomatorviewer.bat,双击打开这个工具,就可以定用了



- package com.uiautomatortest;
- import android.graphics.Point;
- import android.os.Bundle;
- import android.os.RemoteException;
- import android.view.KeyEvent;
- import com.android.uiautomator.core.UiDevice;
 import
- com.android.uiautomator.testrunner.UiAutonatorTest Case;
- public class Test extends UiAutomatorTestCase { public void testClick(){

//get the display height and width
int h=UiDevice.getInstance().getDisplayHeight();
int w=UiDevice.getInstance().getDisplayWidth();
Point p=UiDevice.getInstance().getDisplaySizeDp();

System.out.println("The display width is: "+w); System.out.println("The display height is: "+h); System.out.println(p); //click whe clock UDevice.getInstance().click(159,223); }





拖拽与滑动

一、概念介绍:

- 1) 拖拽:将组建从一个坐标移动到另一个坐标
- 2) 移动:从一二坐标点移动到另一个坐标点
- 3) 步长:从一点滑动到另一点使用的时间

二、拖拽与滑动的相关API:

返回值	方法名	描述
boolean	drag(int startX, int startY, int endX, int endY, int steps)	拖动对象从一个坐标拖 动到另一个坐标
boolean	swipe(Point[] segments, int segmentSteps)	再点阵列中滑动,5ms 一步
boolean	swipe(int startX, int startY, int endX, int endY, int steps)	通过坐标滑动屏幕
		Copyright @2005-2016



三、API应用举例:

package com.uiautomatortest;

import android.graphics.Point; import android.os.Bundle; import android.os.RemoteException; import android.view.KeyEvent;

import com.android.uiautomator.core.UiDevice; import com.android.uiautomator.testrunner UiAstomatorTestCase;

public class Test extends UiAutomator?ectCase {

public void testDragAndSwipe()

//`[64,577][128,640]
 int startX, startY, endX, endY, steps;
 startX=(128-64)/2+64;
 startY=(640-577)/2+577;
 endX=startX;
 endY=startY-200;
 steps=100;
 UiDevice.getInstance().drag(startX, startY,
endX, endY, steps); //step次数

Point p1=new Point(); Point p2=new Point(); Point p3=new Point(); Point p4=new Point(); p1.x=78;p1.y=30; p2.x=235;p2.y=309; p3.x=224;p3.y=414; p4.x=76;p4.y=409;



```
Point[] ps={p1,p2,p3,p4,p1};
UiDevice.getInstance().swipe(ps, 50);
```

```
//(278,374),(69,373)
int startX=278;
int startY=374;
int endX=69;
int endY=373;
int steps=100;
UiDevice.getInstance().swipe(start\, startY, endX, endY, steps);
```







屏幕旋转

- 一、屏幕旋转相关知识:
- 1) 旋转方向: 0度, 90度(向左转), 180度, 270度(向右转)
- 2) 重力感应器: 重力感应器是旋转所依靠的
- 3) 固定位置:指将屏幕方向固定在0度,90度或者180度等
- 4) 物理旋转:物理旋转与重力感应器关联在一块,关闭物理旋转就是关闭了重力感应器,反之亦然) 二、旋转屏幕相关API:

返回值	方法名	描述
void	setOrientationLeft()	通过禁用传感器,然后模拟设备向左转,并且固定位置
void	setOrientationNatural()	通过禁用传感器, 然后模拟设备转到其自然默认的方向,并且固定位置
void	setOrientationRight()	通过禁用伐感器,然后模拟设备向右转,并且固定位置
void	unfreezeRotation()	重新启动传感器和允许物理旋转
boolean	isNaturalOrientation()	检测设各是否处于默认旋转状态
int	getDisplayRotation()	返回当前的显示旋转,0度,90度,180度,270度值分别为:0、1、2、3
void	freezeRotation()	漆用传感器和冻结装置物理旋转在其当前旋转状态



package com.UiAutomator;

import java.io.File;

import android.os.Bundle;

import android.os.RemoteException;

import android.view.KeyEvent;

import com.android.uiautomator.core.UiDevice;
import

com.android.uiautomator.testrunner.UiAutomatoricestCase;

public class Test1 extends UiAutomatorTestCase {

```
public void testOrientation() throws RemoteException{
    int r=UiDevice.getInstance().getDisplayRotation();
    if(r==0){
```

```
System.out.println("r="+r);
UiDevice.getInstance().: <a href="https://www.system.out.println">www.system.out.println("r="+r);</a>
```

```
if(r==1){
   UiDevice.getInstance().setOrientationNatural();
   cleep(1000);
   UiDevice.getInstance().setOrientationLeft();
:f(r==2){
   UiDevice.getInstance().setOrientationNatural();
   sleep(1000);
   UiDevice.getInstance().setOrientationLeft();
if(r==3){
   UiDevice.getInstance().setOrientationNatural();
```



UiSelector对象

这个对象可以理解为一种条件对象,描述的是一种条件,经常配合UiObject使用,可以得到 某个(某些)符合条件的控件对象。

focusable(boolean val)	与checked类似
index(index)	用当前对象在父对象集中的索引作为游述
packageName(String name)	用包名作为条件描述
selected(val)	描述一种选择关系
text(text)	最为常用的一种关系,用控件上的文本即可找到当前控件,需要注意,所有使用text属性找到的控件. 必须是英文的。也就是说,不支持通过中文查找控件!
textContains(text)	与text类似
textStartsWith(text)	与text类似
checked(boolean val)	描述一种check状态为val的关系。
className(className)	描述一针类名为className的对象关系
clickable(boolean val)	与checked类似,描述clickable状态为val的关系
description(desc)	不解释
descriptionContains(desc)	与description类似



UiSelector wx=new UiSelector().textStratWith("微"); UiObject obj = new UiObject (wx); Obj.click();

```
构建一个以'微'开头的UiSelector
```

UiSelector wx=new UiSelector().textContains("微"); UiObject obj = new UiObject (wx); Obj.click ();

```
构建一个有'微'的UiSelector
```





UiSelector wx=new UiSelector().className("android.widget.TextView").text("微信"); UiObject obj = new UiObject(wx); Obj.click();

构建一个class属性为android.widget.TextView_text属性为微信的UiSelector

UiSelector x=new UiSelector().fousable(true).className("android.widget.CheckBox"); UiObject obj = new UiObject(x); Obj.click();

构建一个聚焦的className为androndwidget.CheckBox聚焦的控件







UiObject对象

这个对象可以理解为控件的对象。 一般一个UiObject对象可以通过一下形式得到: UiObject mltem = new UiObject(new UiSelector().text("English")); 也就是配合一个UiSelector就可以得到一个控件。

click()	点击控件
clickAndWaitForNewWindow()	点击某个控件,并等待窗口刷新
longClick()	长按
clearTextField()	清除:江瓜, 主要针对编辑框
getChildCount()	这个方学可以看出,其实UiObject也可以是一个控件的集合
getPackageName()	谷到控件的包名
getSelector()	得到当前控件的选择条件
getText()	得到控件上的Text
isCheckable();isChecked();isClickable() isLongClickable();isScrollટંઠ\લ્{} isSelected()	判断是否具备某个属性



UiObject对象

- 1, UiObject cancelButton = new UiObject(new UiSelector().text("Cancel")); UiObject okButton = new UiObject(new UiSelector().text("OK"));
- 2, UiObject clock = new UiObject(new UiSelector().description("Clock")); clock.clickAndWaitForNewWindow();

3, UiObject settingsValidation = new UiObject(new UiSelector().packageName("com.android.deskclock")); // 如果不存在则出错疑示 assertTrue("Unable to detect Clock", settingsValidation.exists());





点QQ

```
UiSelector qq=new UiSelector().text("QQ");
UiObject obj = new UiObject(qq);
obj.click();
```

长按QQ

```
UiSelector qq=new UiSelector().text("QQ"),
UiObject obj = new UiObject(qq);
obj.longClick();
```

把QQ移动到【560,600】坐标处,40为step

```
UiSelector qq=new UiSelector(j.text("QQ");
UiObject obj = new UiObject(qq);
obj.dragTo(560,600,40);
```



- 交换QQ与计算器的位置
- UiSelector qq=new UiSelector().text("QQ"); UiObject obj = new UiObject(qq); UiSelector calc=new UiSelector().text("计算器"); UiObject obj1 = new UiObject(calc); obj.drapTo(obj1,40);
- 向左划屏,按着第3个(0开始,所以为2)APP左划
- UiSelector View=new UiSelector().className("android.view.View").instance(2); UiObject obj = new UiObject(View);
- Obj.SwipeLeft(10);







判断对象是否存在

UiSelector View=new UiSelector().className("androidiview/.View").instance(2); UiObject obj = new UiObject(View); If (obj.exists ())

```
System.out.println("Object exists!");
```

}else{

```
System.out.println("Object is nut exist!");
```





模拟短信输入

- UiSelector edit=new UiSelector().className("android.view.EditText").instance(0);
- UiObject obj = new UiObject(edit);
- obj.setText("13681732596");
- UiSelector edit1=new UiSelector().className("android.view.EditText").instance(1);
- UiObject obj1 = new UiObject(edit1);
- obj1.setText("hi");
- obj1.clearTextField();
- obj1.<mark>setText</mark>("hello");
- 判断QQ APP坐标位置
- UiSelector qq=new UiSelector().text("QQ");
- UiObject obj = new UiObject(qq);
- String rect = obj.getBounds().toString();
- assertEquals("Rect(500,534-710,720",rect);



获得屏幕下方的文本信息

UiSelector View=new UiSelector().className("android_view.View").instance(4); UiObject obj = new UiObject(View); int count = obj.getChildCount(); for(int j=0;j<count;j++){ UiObject child = obj.getChild(new_viiCelector().index(j)); System.out.println(child.getText'_));





UiCollection对象

- 这个对象可以理解为一个对象的集合。因为UiSelector描述后得到的有可能是多个满足条件的控件集合,因此可以用杂生成UiCollection:
- UiCollection mUiCollection = new UiCollection (new
- UiSelector().text("Settings"));

getChild(selector)

从集合中再次通过UiSelector选择一个UiObject对象

getChildByDescription(childPattern,
text)

从一个匹配模式中再次以text为条件选择UiObject







Uiautomator如何支持中文	
-------------------	--

选择java文件,右击选择"属性"

0	Properties for		
type filter text	Resource		
Run Debug Settin	Path: /Uiautomatortest/src/com/jerry/m Type: File (Java Source File) Location: C:\Myproject\Uiautomatortest\src Size: 3,938 bytes Last modified: 2016年4月5日下午3:17:33 Attributes: Read only ✓ Archive Derived Text file encoding Other: Other: UTF-8		
< >			
?			
	and the second second second		



练习

- 演示: 15分钟
- 打开安卓时钟, 然后切换到计时器
- 练习: 30分钟
- 安装我提供的Demo5.apk
- >adb install c:\Demo5.apk
- 编写测试代码验证
- 打开主菜单
- 找到Demo5
- 向左转设备
- 点第一张图片, 切浇到第二张图片
- 点第二张图片, 切换到第一张图片
- 返回主页面



目录



•移动软件测试技巧		
•安卓系统测试	FI	
▶Andriod 各种UI测试框架介绍	E	
▶UiAutomator UI自动化测试框架		
▶Android黑盒自动化测试Monke;/MonkeyRunner		
>Selenium/WebDriver的使用		
▶DDMS使用		录
▶多平台测试		
▶星云测试		
·APP测试讨论	223	vww.3testing.com
		Copyright @2005-2016

Android黑盒自动化测试Monkey/MonkeyRunner

啄木鸟软件测试培训网




Monkey是Android中的一个命令行工具,可以运行在模拟器里或实际设备中。 它向系统发送伪随机的用户事件流(如按键输入、触摸屏输入、手势输入等), 实现对正在开发的应用程序进行压力测试。Monkey测试是一种为了测试软 件的稳定性、健壮性的快速有效的方法。

Monkey的特征

- 1、测试的对象仅为应用程序包,有一定的局限性。
- 2、 Monkey测试使用汽事件流数据流是随机的,不能进行自定义。
- 3、可对Monkey Text的对象,事件数量,类型,频率等进行设置。



Monkey介绍

基本语法如下

> adb shell monkey [options]

如果不指定options, Monkey将以无反馈摸式启动,并把事件任意发送到安装在目标环境中的全部包。下面是一个更为典型的命令行示例,它启动指定的应用程序,并向其发送500个伪随机事件:

> adb shell monkey -p your package.name -v 500







检查模拟器中有啥包

- 1、 通过eclipse启动一个Android的emulator
- 2、 在命令行中输入: >adb devices查看设备连接情况

C:\Documents and Settings\Administrator>adb devices

list of devices attached

emulator-5554 device

3、 在有设备连接的前提下,在命令行中输入: aclb shell 进入shell界面

C:\Documents and Settings\Administrator>adb_hell

#

4、 查看data/data文件夹下的应用程序包。注:我们能测试的应用程序包都在这个目录下面 C:\Documents and Settings\Administrator>adb shell

ls data/data

ls data/data

com.google.android.btrouter com.android.providers.telephony

the second second second second second



www.3testing.com

Copyright @2005-2016

Monkey介绍

:\Myproject\LaunchSettings>adb shell oot@generic:/ # ls data/data ls data/data com.android.backupconfirm com.android.browser com.android.calculator2 com.android.calendar com.android.camera com.android.certinstaller com.android.contacts com.android.customlocale2 com.android.defcontainer com.android.deskclock com.android.development com.android.development_settings com.android.dialer com.android.documentsui com.android.dreams.basic com.android.email com.android.emulator.connectivity.test com.android.emulator.gps.test com.android.exchange com.android.externalstorage com.android.fallback com.android.galleru

在我的虚拟几种有

- com.android.camera
- com.android.email
- com.android.clock
- com.android.coiculator2

等应用程序已

运行命令

>adb fiell monkey -p com.android.calculator2 100,随即向 com android.calculator2发出100个按键消息,你会发现没有请求,因 为monkey默认是在安静模式下运行的。要查看详细信息用-v (普通 信息)或者-v-v (详细信息)

C:\Myproject\LaunchSettings)add shell monkey - p com.android.calculator2 100

:\Myproject\LaunchSettings

Monkey介绍



www.3testing.com

Copyright @2005-2016





参数	基本功能	举例 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-p	参数-p用于约束限制, 用此参数指定一个或多 个包(Package,即 APP)。 指定包之后,Monkey将 只允许系统启动指定的 APP。如果不指定包, Monkey将允许系统启动 设备中的所有APP。	 指定一个包: adb shell monkey -p com.htc.Weather 100 说明: com.htc.Weather为包名, 100是事件计数(即让 Monkey程序模拟100次随机用户事件)。 指定多个包: adb shell monkey -p com.htc.Weather -p com.htc.pdfreader -p com.htc.photo.widgets 100 不指定包: adb shell monkey 100 说明: Monkey随机启动APP并发送100个随机事件。 夹室看设备中所有的包,在CMD窗口中执行以下命令: >adb shell #cd data/data #ls



参数	基本功能	举例 人 人 人	
-v	用于指定反馈信	・日志级别 Level O	
	息级别(信息级	示例 adb shell monkey -p com.htc.Weather -v 100	
	别就是日志的详	说明 缺省值, 仅提供启动程示、测试完成和最终结果等少量	信
	细程度),总共	息	
	分3个级别:level	日志级别 Level 1	
	0-2	示例 adb shell monkey -p com.htc.Weather -v -v 100	
		说明 提供较为详细的日志,包括每个发送到Activity的事件作	言
		日志级别 Lavel 2	
		示例 act hell monkey -p com.htc.Weather -v -v -v 100	
		说明 最详细的日志,包括了测试中选中/未选中的Activity	信
		息	





参数	基本功能	举例
·S	用于指定伪随机数生 成器的seed值,如果 seed相同,则两次 Monkey测试所产生的 事件序列也相同的。	 Monkey测试1: adb shell monkey -p com.htc.Weather -s 10 100 Monkey测试2: adb shell monkey -p com.htc.Weather -s 10 100 两次测试的效果是相同的,因为模拟的用户操作序列 (每次操作按照一定的先后顺序所组成的一系列操作, 即一个序列)是一样的。操作序列虽然是随机生成的, 怎是只要我们指定了相同的Seed值,就可以保证两次测 试产生的随机操作序列是完全相同的,所以这个操作序 对伪随机的.
throttle 〈毫秒〉	用于指定用户操作 (即事件)间的时延, 单位是毫秒	 adb shell monkey -p com.htc.Weather –throttle 3000 100



参数	基本功能	举例
ignore-crashes	用于指定当应用程序崩溃时 (Force & Close错误), Monkey是否停止运行。如果使 用此参数,即使应用程序崩溃 , Monkey依然会发送事件,直 到事件计数完成。	 示例1: aob shell monkey -p com.htc.Weather igncre-crathes 1000 测试过程中即使Weather程序崩溃, Monkey依 然会继续发送事件直到事件数目达到1000为止; 示例2: adb shell monkey -p com.htc.Weather 1000 测试过程中,如果Weather程序崩溃, Monkey将 会停止运行。
ignore- timeouts	用于指定当应用程序发生ANR (Application No Responding) 错误时, Monkey龙否停止运行。 如果使用此参数,即使应用程 序发生ANIR错误, Monkey依然 会发送事件,直到事件计数完成。	示例同上



参数	基本功能	举例
ignore- security- exceptions	用于指定当应用程序发生许可错误时(如证书许可,网络许可等),Monkey是否停止运行。如果读用此参数,即使应用程序发生许可错误,Monkey依然会发送事件,直到事件计数完成。	示例同上
kill-process- after-error	用于指定当应用程序发生错误时,是否停止其运行。如果指定 此参数,当应用程序发生错误时,应用程序停止运行并保持在 当前状态(注意:应页程序仅是静止在发生错误时的状态,系 统并不会结束该应用程序的进程)。	示例同上
monitor- native-crashes	用于指定是否监视并报告应用程序发生崩溃的本地代码。	示例同上





参数	基本功能	举例
pct- {+事件类别} {+ 事件类别百分比}	用于指定每种类别事件的数百百分比 (在Monkey事件序列中, 该类事件数目 占总事件数目的百分比)	详情见下列具体参数
pct-touch {+百分比}	调整触摸事件的百分比(缸摸事件是一个 down-up事件, 它发生在屏幕上的某单一 位置)	adb shell monkey -p com.htc.Weatherpct- touch 10 1000
pct-motion {+百分比}	调整动作率件的百分比(动作事件由屏幕 上某处的一个down事件、一系列的伪随 机事件和一个up事件组成)	adb shell monkey -p com.htc.Weatherpct- motion 20 1000
pct-trackball {+百分比	<u>试整轨迹事件的百分比(轨迹事件由一个</u> 或几个随机的移动组成,有时还伴随有 点击)	adb shell monkey -p com.htc.Weatherpct- trackball 30 1000





参数	基本功能	举例
pct-nav {+百分比}	调整"基本"导航事件的百分论(导航事件由来自方向输入设备 Up/Down/Left/Right组成)	adb shell monkey -p com.htc.Weather pct-nav 40 1000
pct-majornav {+百分比}	调整"主要"导频率件的百分比(这些导航事件通常引发图形界面中的动作,如: 5-way键盘的平间按键、回退按键、菜单按键)	adb shell monkey -p com.htc.Weather pct-majornav 50 1000
pct-syskeys {+百分比}	调整"系纸"按键事件的百分比(这些按 键 适常被保留,由系统使用,如Home、 Back、Start Call、End Call 及音量控制键)	adb shell monkey -p com.htc.Weather pct-syskeys 60 1000
pct-appswitch {+百分比}	调整启动Activity的百分比。在随机间隔 二, Monkey将执行一个startActivity()调 用,作为最大程度覆盖包中全部Activity 的一种方法	adb shell monkey -p com.htc.Weather pct-appswitch 70 1000



Monkey介绍

参数	基本功能	举领
-pct-anyevent {+百分比}	调整其它类型事件的百分比。它包罗了所有其 它类型的事件,如:按 键、其它不常用的设备 按钮、等等	 · 將定单个类型事件的百分比: adb shell monkey -p com.htc.Weather - ·pct-anyevent 100 1000 · 指定多个类型事件的百分比: adb shell monkey -p com.htc.Weather - -pct-anyevent 50pct-appswitch 50 1000 注意: 各事件类型的百分比总数不能 超过100%



- 利用Monkey进行性能能测试 利用Monkey命令,我们可以根据输出信息查看,安卓APP软件是否存在性能 问题,为了看起来方便,我们利用>重定向方法。 比如:
- adb shell monkey -p com.xy.android.junit -s 500 --ignore-crashes --ignoretimeouts --monitor-native-crashes -v -v 10000 > E:\monkey_log\java_monkey_log.txt
- 然后分析E:\monkey log\java monkey log.txt, 比如:
- ...
- //CRUSH: com.myapppiction.mygame(pid 557) //short Mse: andriod.util.AndroidRuntimeException



演示Monkey的使用:

10分钟

练习Monkey的使用: 15分钟





Monkey脚本运行,必须在由SIM卡的设备上运行,编辑文件scripts.mks

Start of Script

type= user

count= 49

speed= 1.0

start data

LaunchActivity(com.android.clock, com.android.calculator2)

3120021258

DispatchPress(KEYCODE_3)

UserWait(200)

DispatchPress(KEYCODE_1)

UserWait(200)

DispatchPress(KEYCODE_3)

UserWait(200)

DispatchPress(KEYCODE_5)

UserWait(200) DispatchPress(KEYCODE_0; UserWait(200) DispatchPress(KEYCOUE_2) UserWait(20C) DispatchPress(KEYCODE_1) UserWait(200) DispatchPress(KEYCODE_2) UserWait(200) **DispatchPress(KEYCODE_5)** UserWait(200) uspatchPress(KEYCODE_8) **UserWait(200)** # Pin 12345

DispatchPress(KEYCODE_DPAD_DOWN) UserWait(250) **DispatchPress(KEYCODE_1)** UserWait(200) **DispatchPress(KEYCODE_2)** UserWait(200) **DispatchPress(KEYCODE_3)** UserWait(200) **DispatchPress(KEYCODE_4)** UserWait(200) DispatchPress(KEYCODE_5) UserWait(200) **#** Down and enter DispatchPress(KEYCODE_DPAD_DOWN) UserWait(250) DispatchPress(KEYCODE_ENTER)



Monkey介绍

push scripts.mks到安卓设备/sdcard/中

>adb shell push scripts/sdcard/

#运行

>adb shell monkey -f /so'ard/scripts.mks





Monkey命令列表

DispatchKey(downTime,eventTime,action,code,repeat_meteState,device,scancode)用于发送 案件消息

@param long downTime //键最初被按下的时间

@param long eventTime //事件发生的时间

@param int action // (ACTION_DOWN=0, ACTION_UP=1, ACTION_MULTIPLE=2)

@param int code //键值,比如KEYCODE_DPAD_DOWN(20)

@param int repeat//

@param int metaState//当前按下的meta键的标识

```
@param int device//事件发生的设备id
```

```
@param int scancude//
```



Monkey介绍

DispatchPointer(downTime, eventTime, action, x, y, pressure, size, metaState, xPrecision, yPrecision, device, edgeFlags) 向一个单定位置发出一个手势消息 @param long downTime //键最初被按下的时间 @param long eventTime //事件发生的时间 @param int action // (ACTION_DOWN=0, ACTION_MOVE=1, ACTION_UP=2, ACTION CANCEL=3) @param float x //x坐标 @param float y //y坐标 @param float pressure //当前事件的压力 范围0-1 @param float size //触摸的近似值, 范围0-1 @param int metaState //当前校下的meta键的标识 @param float xPrecision //x坐标精确值 @param float yPrecision //y 公标精确值 @param int device //事件来源,范围0-x, 0表示不来自物理设备 @param int sedgeFlags //



Monkey介绍

DispatchTrackball(downTime, eventTime, action, x, y, pressure, size, metaState,

xPrecision, yPrecision, device, edgeFlags) 用于向一个指定位置发单个跟踪消息

- Tap(x,y,duration) 模拟一次手指单击
- DispatchPress(String key_name)向shell命令输入字符串
- DispatchFlip(boolean keyboardOpen) 打开或关闭软键盘
- LaunchActivity(String pkg_name, String cl_name, long alarmTime) 启动任意一个活动 (界面)
- UserWait(long sleeptime) 要休眠的时间,以毫秒为单位
- LongPress() 模拟模议敲击键盘事件







Monkey介绍

- RoateScreen (rotationDegree, presist) 发送屏幕旋转事件
- LaunchInstrumentation 运行一个仪表盘测试程序
- LongPress 长按两秒
- PowerLog(power_log_type,test_case_status) 漠拟电池量信息
- RunCmd(cmd) 在设备上运行shell 脚本
- ProfileWait() 等待5秒钟
- PressAndHold (x,y,pressDuration) 模拟一个长按事件,时间可指定
- Drag(xstart, ystart, xend, yend, stepCount) 模拟一个拖拽



Monkey介绍



•Pinchzoom(pt1xstart,pt1ystart,pt1end,pt1end,pt2xstart,pt2ystart,pt2xend,pt2yend,stepco

- unt) 模拟缩放手势
- •StartCaptureFramerate() 开始获取帧率要求viewancestor.profile_redering=true
- •EndCaptureFramerate(input) 帧率
- •StartCaptureAPPFramerate(app) 开始获取指定应用的帧率
- •EndCaptureAPPFramerate(app,input) 结束获取指定应用的帧率





Android黑盒自动化测试Monkey/MonkeyRunner

啄木鸟软件测试

录

www.3testing.com

Copyright @2005-2016

E

Monkey介绍

MonkeyRunner介绍



Monkey, MonkeyRunner区别





Monkey, MonkeyRunner区别





Monkeyrunne采用Pythone语言编写 但可以用monkey_recorder.py,monkey_playback.py录制与回放

在windows下运行Pythone必须先运行福件











Copyright @2005-2016

MonkeyRunner





按Home键

from com.android.monkeyrunner import MonkeyRunner,MonkeyDevice,MinkeyImage device =MonkeyRunner.waitForConnection() device.press('KEYCODE_HOME',MonkeyDevice.DOWN_AND_UP)

菜单键: KEYCODE_MENU 挂视键: KEYCODE_ENDCALL

- HOME键: KEYCODE_HOME 音量+键: KEYCODE_VOLUE_UP
- 返回键: KEYCODE_BACK 音量-键: KEYCODE_VOLUE_DOWN
- 搜索键: KEYCODE_SEARCH 电源键: KEYCODE_POWER
- 呼叫键: KEYCODE_CALL
- 照相机键: KEYCODE_CAMERA



设备唤醒

from com.android.monkeyrunner import MonkeyRunner,MonkeyDevice,MinkeyImage device =MonkeyRunner.waitForConnection{} device.wake()

按菜单键

from com.android.monkeyrunner import MonkeyRunner,MonkeyDevice,MinkeyImage device =MonkeyRunner.waitForConnection() device.press('KEYCODE_IMENU',MonkeyDevice.DOWN_AND_UP)



输入内容

from com.android.monkeyrunner import
MonkeyRunner,MonkeyDevice,MinkeyImage
device =MonkeyRunner.waitForConnection()
device.type('Monkeyrunner')

控制多个设备

from com.android.monkeyrunner import

MonkeyRunner,MonkeyDevice,MinkeyImage

device1 = MonkeyRunner.waitForConnection(5,'b4726a2d')

device2 = MonkeyRunner.waitForConnection(5,'5dfadsf32scda')

device1.press('KEYCODE_HOME',MonkeyDevice.DOWN_AND_UP) device2.type('Monkeyrunner')



对比截屏与已知图片

from com.android.monkeyrunner import MonkeyRunner,MonkeyDevice,MinkeyImage device = MonkeyRunner.waitForConnection(5,'b8886a88d') result = device.takeSnapshot() result.writeToFile('D:\\result.png', ong') pic2=monkeyrunner.loadImageFromFile('D:\\picture2.png') if (result.sameAs(pic2,0.9) print("Success") else print("Failure")





MonkeyRunner 单击操作

from com.android.monkeyrunner import MonkeyRunner,MonkeyDevice,MinkeyImage device =MonkeyRunner.waitForConnection(5,'b3886a88d') device.touch(200,300,'DOWN_AND_UP);

安装APK包

from com.android.monkeyrunner.import MonkeyRunner,MonkeyDevice,MinkeyImage device =MonkeyRunner.waitForConnection(5,'b8886a88d') device.installPackage("f:\\simpleapp.apk")







卸载APK

from com.android.monkeyrunner import MonkeyRunner,MonkeyDevice,MinkeyImage device =MonkeyRunner.waitForConnection(5,'b8886a88d') device.removePackage("f:\\simpleapp.apk")

启动Activity

from com.android.monkeyrunner import MonkeyRunner,MonkeyDevice,MinkeyImage device =MonkeyRunner.waitForConnection(5,'b8886a88d') device.startActivity(component='simple.app/simple.app.SimpleAppActivity')







功

- 练习:使用monkey_recorder.py 录制一段脚本 10分钟
- 由于monkeyruner在windows下不稳定,所以练习可能不成



目录



录

www.3testing.com

Copyright @2005-2016

E

•移动软件测试技巧 •安卓系统测试 ▶Andriod 各种UI测试框架介绍 ▶UiAutomator UI自动化测试框架 ▶Android黑盒自动化测试Monkey/MonkeyRunner ▶Selenium/WebDriver的使周 ▶DDMS使用 ▶多平台测试 ▶星云测试

• APP测试讨论


Selenium/WebDriver的使用







- **1. Setup Android emulator**
 - a. Download the Android SDK
 - http://developer.android.com/sdk/index.htmi
- Note that there is an emulator bug on Gingerbread((2.3.x) that might cause WebDriver to crash. My testing is on ice Cream Sandwich (4.0.x)
- **b. Install Android SDK:**
 - http://developer.android.com/sdk/installing.html
- c. Start Android SDK Manager (SDK Manager.exe)
- d. Select and install Package online
- e. Start AVD Manager.exe
- f. Create an emulator



Add JARs... Add External JARs... Add Variable... Add Library... Add Class Folder... Add External Class Folder...

> Edit... Remove

Migrate JAR File..

Cancel

OK

com -2016

😓 🔻 🖨 🔻 🤜

Properties for WebTest

安装

- •到网上下载, 解压selenium-2.33.0后存在本地。
- ·在Eclipse上建立Project、Package、Class
- · 右击Project, 在弹出菜单中点属性。
- ・进入JavaBuild

	type filte text	, 'ava Build Path
Name	 ▷ P esc ince Builders Javin⊇nild Path ▷ Java Count Style 	 ✓ Source ➢ Projects ➡ Libraries ↔ Order and Export JARs and class folders on the build path: ➤ JRE System Library [JavaSE-1.7]
 libs CHANGELOG selenium-java-2.33.0.jar selenium-java-2.33.0-srcs.jar 	Java Compiler Ja.:- E-litor avadoc Location Project References Run/Debug Settin Task Tags XML Syntax	> ■ JUnit 3
	< > ?	







- 点Add External JARs...
- 加入: selenium-java2.33.0.jar和selenium-java2.33.0-srcs.jar



	LAP. Selectio.		X	$\Leftrightarrow \bullet \Rightarrow \bullet \bullet \bullet$
🗲 🌛 🗸 🕇 👢 « sel	enium-2.33.0) selenium-2.35.0	Search selenium-2.33.0	Q	
Organize New fold	ler		0	Add JARs
▷ 📙 live-network-con ^	Name	Date modified	Type dd	External JARs
LStyle MSOCache	ibs	2015/6/12 10:38	File fol A	dd Variable
▶ ↓ MyDownloads	sele. im-java-2.33.0.jar	2013/5/22 11:33	Executi A	dd Library
Myproject	selenium-server-standalone-2.33.	0.jar 2015/6/12 13:54	Execution do	l Class Folder
▷ ▶ Prosram Files			Ext	ernal Class Folder
▷ 🗼 Pr∴gran Files (xŧ				Edit
► ► rog.amDa				Remove
RavBin			۸ic	grate JAR File
D selenium-2.33.0				
SHDownload	<		```	Cancel
File na	ame: "selenium-java-2.33.0-srcs.jar" "selen	nium-ja 💙 *.jar;*.zip Open Can	✓ cel	







安装

- 再次点Add External JARs...
- 进入libs目录
- 加入所有jar包

	JAR Selection				· • •
→ ↑ « selor		~ ¢	Search libs	Q	
ganize • Nev. fo. ler					
Desktop (?)	Name		Date modified	Тур ^	
Hamestown	🛓 apache-mime4j-0.6.jar		2013/5/22 11:33	Exe	
Home group	bsh-1.3.0.jar		2013/5/22 11:33	Exe	
我的心心	🛓 cglib-nodep-2.1_3.jar		2013/5/22 11:33	Exe	
	🕌 commons-codec-1.6.jar		2013/5/22 11:33	Exe	
Desktop	🛓 commons-collections-3.2.1.jar		2013/5/22 11:33	Exe :	ler
Documents	commons-exec-1.1.jar		2013/5/22 11:33	Exe	
Downloads	commons-io-2.2.jar		2013/5/22 11:33	Exe	
Music	commons-jxpath-1.3.jar		2013/5/22 11:33	Exe	
Pictures	commons-lang3-3.1.jar		2013/5/22 11:33	Exe	
Videos	🕌 commons-logging-1.1.1.jar		2013/5/22 11:33	Exe	
Windows (C:)	cssparser-0.9.9.jar		2013/5/22 11:33	Exe	
☑ 文档 (D:)	🔔 quava-14 0 iar		2013/5/22 11.33	Fxe 💙	
¥	<			>	
File nan	ne:		✓ *.jar;*.zip	~	
			Open	col	
			Open Ca	icei .	





占 Add Libarary	0	Properties N WebTest	_ □ ×
$\gamma \sim \gamma \sim$	type filter text	Java Build Path	$\Leftrightarrow \bullet \Rightarrow \bullet$
选择Junt	Resource	Source Projects シンションションション・シント Order and Export	
然 仁 法 坏 Luni+1	Java Build Path	JARs and class focters on the build path:	
然后近年Junit4	Java Code Style	operadriver 1.3.jar - C:\selenium-2.33.0\selenium-2.33.0\seler	Add JARs
	 Java Compiler Java Editor 	hanton isdriver-1.0.3.jar - C:\selenium-2.33.0\selenium-2.33.	Add External JARs
	Javadoc Location	ac-1.∋ jar - C:\selenium-2.33.0\selenium-2.33.0\selenium-2.3	Add Variable
it Library	Project References Run/Debug Sattin Task Tajis XML Syntar	Image: Selenium-java-2.33.0-srcs.jar C:\selenium-2.33.0\selenium	Add Library
ect the JUnit version to use in this project.		seienium-Java-2.33.0.jar - C:\selenium-2.33.0\selenium-2.33.0\selenium-2.33.0	Add Class Folder
		Extra construction in the second selenium selenium in the second selenium is the second selenium in the second selenium is the second selenium second selenium is the second selenium second sec	Add External Class Folder
library version: JUnit 4		 ^a a xalan-2.7.1.jar - C:\selenium-2.33.0\seleni	Edit
ent location: junit.jar - C:\ADT\eclipse\plugins\org.junit_4.10.0.\		xml-apis-1.4.01.jar - C:\selenium-2.33.0\selenium-2.33.0\seler	Remove
ce location: Not found		A JRE System Library [JavaSE-1.7]	Minute IAD File
			wigrate JAR File
	< >		
	?	0	K Cancel
	-		
			ununu Otaatina
		203	www.stesting
			Copyright @20



安装

- 2. Install the AndroidDriver APK by using platform-tools

 a. list divce name:
 >adb devices
 b. download AndroidDriver APK:
 - http://code.google.com/p/selenium/downloads/list
 - c. install AndroidDriver APK:

>adb -s emulator-5554 -e instali -r c.\android-server-2.9.apk

d. start the Android WebDriver application

>adb -s emulator-5554 shell am start -a android.intent.action.MAIN -n

org.openqa.selenium.android.app/.MainActivity

e. setup the port forwarding in order to forward traffic from the host machine to the emulator

>adb -s emulator-5554 forward tcp:8080 tcp:8080

Selenium/WebDriver的使用









打开页面

// And now use the driver to visit baidu driver.get("https://www.baidu.com");





- 1, 如何找到页面元素
- <input type="text" name="passwd"id="passwd-id" />
- WebElement element = driver.findElement(**By.id**(**"passwd-id"**));
- WebElement element = driver.findElement(<a>Byname("passwd"));
- WebElement element =driver.findElement(Ey.xpath("//input[@id='passwd-id']"));
- <div class="cheese">Cheddar</div><div</pre>
- class="cheese">Gouda</div>
- WebElement cheeses = driver.findElements(By.className("cheese"));
- cheese
- WebElement chees e: -: driver.findElement(By.linkText("cheese"));



- 2, 如何对页面元素进行操作
- 2.1输入框 (text field or textarea)
- 找到输入框元素:
- WebElement element = driver.findElement(By.id("passwd-id"));
- •在输入框中输入内容:
- element.sendKeys("test");
- •将输入框清空:

element.clear();

•获取输入框的文本内容:

element.getText();





- 2.2下拉选择框(Select)
- 找到下拉选择框的元素:

Select select = new Select(driver.findElement(By.id("select"));;

•选择对应的选择项:

select.selectByVisibleText("mediaAgencyA");或 select.selectByValue("MA_ID_001");

•不选择对应的选择项:

select.deselectAll();
select.deselectByValue("MA_ID_COL');
select.deselectByVisibleText("mediaAgencyA");

• 或者获取选择项的值:

select.getAllSelectedOptions();
select.getFirstSelectedOption();



- 2.3单选项(Radio Button)
- •找到单选框元素:

WebElement bookMode =driver.findElen:erit(Fy.id("BookMode"));

•选择某个单选项:

bookMode.click();

•清空某个单选项:

bookMode.clear();

•判断某个单选项灵否已经被选择:

bookMode.isSelected();





2.4多选项(checkbox)

•多选项的操作和单选的差不多:

WebElement checkbox =driver.findElement(@y.id("myCheckbox.")); checkbox.click(); checkbox.clear(); checkbox.isSelected(); checkbox.isEnabled();





- 2.5 按钮(button)
- 找到按钮元素:

WebElement saveButton = driver.findElement(By.id("save"));

- •点击按钮:
- saveButton.click();
- •判断按钮是否enable:
- saveButton.isEnabled ();







- •2.6 左右选择框
- •也就是左边是可供选择项,选择后移动到右边的框中,反之亦然。例如:

Select lang = new Select(driver.findElement(By.id("languages"))); lang.selectByVisibleText("English"); WebElement addLanguage =driver.findElement(By.id("addButton")); addLanguage.click();

•2.7 弹出对话框(Popup dialogs)

Alert alert = driver.switch?c().alert(); alert.accept(); alert.dismiss(); alert.getText();



2.8 表单(Form)

Form中的元素的操作和其它的元素操作一样,对元素操作完成后 对表单的提交可以:

WebElement approve = driver.findElement(Ey id("approve")); approve.click();//Type=Button 或

approve.submit();//Type=Submit







- 2.9 上传文件(Upload File)
- •上传文件的元素操作:

WebElement adFileUpload = driver.findElement(By.id("WAP-upload"));
String filePath = "C:\\test\\uploadfile\\media_ads\\test.jpg";
adFileUpload.sendKeys(filePath);





- 2.10 Windows 和Frames之间的切换
- •一般来说,登录后建议是先进入默认frame:

driver.switchTo().defaultContent();

- 切换到某个Frames: driver.switchTo().frame("leftFrame";;
- •从一个Frames切换到另一个Frames:

driver.switchTo().frame("mainFrame");

• 切换到某个window:

driver.switchTo().window("windowName");



2.11 拖拉(Drag and Drop)

WebElement element =driver.findElement(By.eame("source")); WebElement target = driver.findElement(By.name("target")); (new Actions(driver)).dragAndDrop(element, target).perform(); 2.12 导航(Navigationand History) 打开一个新的页面: driver.navigate().to("http://www.example.com"); 通过历史导航返回原页面. driver.navigate().forward(); driver.navigate().back();

Selenium/WebDriver的使用







高级使用

1,设置Cookies 我们经常要对的值进行读取和设置。 增加cookie:

// Now set the cookie. This one's valid for the entire domain Cookie cookie = new Cookie("key", "value"); driver.manage().addCookie(cookie);





高级使用

2,获取cookie的值:

- // And now output all the available cookies for the current URL
- Set<Cookie> allCookies = driver.manage().getCookies();
- for (Cookie loadedCookie : allCookies) {
 - System.out.println(String.format("%s -> %s",loadedCookie.getName(),
 loadedCookie.getValue()));
- •根据某个cookie的name获取cookie的值:
- driver.manage().getCookieNamed("mmsid");





高级使用



3, 删除 cookie:

- // You can delete cookies in 3 ways
- // By name
- driver.manage().deleteCookieNamed("CookieName");
- // By Cookie
- driver.manage().deleteCookie(loadedCookie);
- // Or all of them
- driver.manage().deleteAllCockies();





高级使用

4,调用Java Script

Web driver对Java Script的调用是通过JavascriptExecutor来实现的, 例如:

javascriptExecutor js = (JavascriptExecutor) driver;

```
js.executeScript("(function(){inventoryGricMgr.setTableFieldValue("'+
inventoryId + "','" + fieldName + "','"
+ value + "');})()");
```





Selenium/WebDriver的使用





勾

- **Create test case and running:**
- package com.jerry;
- import junit.framework.TestCase;
- import org.openqa.selenium.By;
- import org.openqa.selenium.WebDriver;
- import org.openqa.selenium.WebElement;
- import org.openqa.selenium.android.AndroidDriver;
- public class FirstDemoForWebdriver extends TestCase{
- public static void main(String[] args) {
- String inputString="Software Testing",
- WebDriver driver = new Ano oidDriver();
- // And now use this to visit baidu
- driver.get("https://m.baidu.com/?from=844b&vit=fps");

妥用Java Application来测试



实例

```
// 用下面代码也可以实现
driver.navigate().to("https://www.baidu.com");
// 获得百度首页的Title
System.out.println("1 Page title is: " + driver.getTitle());
// 通过 id 找到 input 的 DOM
WebElement element = driver.findElement(By.id("index-kw"));
11 输入关键字
element.clear();
element.sendKeys(inputStrir.g);
WebElement mybutton = driver.findElement(By.id("index-bn"));
mybutton.click();
try {
    Thread. sleep(2000);
```

} catch (InterruptedException e) {





} catch (InterruptedException e) { e.printStackTrace();

//获得查询后的Title System.out.println("2 Page title is: * ・ chiver.getTitle()); //关闭浏览器 driver.quit();



目录



•移动软件测试技巧 •安卓系统测试 E ▶Andriod 各种UI测试框架介绍 ≻UiAutomator UI自动化测试框架 ▶Android黑盒自动化测试Monrey/MonkeyRunner ▶Selenium/WebDriver的使用 录 ▶DDMS使用 ▶多平台测试 ▶星云测试 www.3testing.com • APP测试讨论 Copyright @2005-2016

- 启动虚拟机
- ・启动DDMS
- .\sdk\tools



w.3testing.com Copyright @2005-2016

啄木鸟软件测试培训网



• 列出包名

Name			
🛛 🚇 Jerry [emulator-5554]	Online		Jerr; [4].4
system_process	386	**	6003
com.android.inputmethod.la	503	*	8607.
com.android.settings	516	**	2603
com.android.phone	523	27.7.4 7.7.4	8604
com.android.launcher	5.14	**	8605
com.android.printspooler	563	***\$	8606
android.process.aco. 9	<u>し</u>93	**	8607
com.android.music	671	**	8608
com.android.system, ii	695	**	8601
android.proccss.media	758	**	8609
com.ano.roira.dia ler	833	**	8610
comandroir providers.calen	853	**	8611
corn.and pid.mms	874	**	8612
com.android.calendar	900	**	8613
com.android.deskclock	921	**	8614
com.android.email	938	**	8615
com.android.exchange	958	**	8616

www.3testing.com Copyright @2005-2016



www.3testing.com

Copyright @2005-2016

• 查看当前包名运行的线程

File Edit Actions Device

8 84 0 🔹 🛠 👳				Info	Thre	eads VM He	ap Alic	antion T	Fracker
Name				ID	79	Statu:	utinie	stime	Name
▲ Jerry [emulator-5554]	Online		Jerry [4.4	1	544	Native	462	80	main
system_process	386	**	8600	12	547	Vr.Wait	1	16	GC
com.android.inputmethod.la	503	**	8602	*3	549	Vr. Wait	3	5	Signal
com.android.settings	516	**	8603	*4	5.70	Runnable	1	4	JDWP
com.android.phone	528	**	8604	*5	551	VmWait	33	18	Comp
com.android.launcher	544	**	8605 / 87.	"6	552	Wait	1	0	Refere
com.android.printspooler	563	**	8606	,7	553	Wait	5	4	Finaliz
android.process.acore	583	**	8607	*2	554	Wait	0	0	Finaliz
com.android.music	671	**	86,78	a	556	Nativo	2	2	Rindor
com.android.systemui	695	\$\$	8601		550	Native	2	2	Diridei
android.process.media	758	**	8609	Refr	esh				
com.android.dialer	833	*	8610						
com.android.providers.calen	853	***	8611						
com.android.mms	874	**	8612						
com.android.calendar	300	**	8613						
com android deskcle sk	Q5 1	**	8617	I					



使用DDMS模拟打电话

	Darvik Debug Morritor								
l	Info	Threads	VM Hea	ap Alloc	ation Tracker	Sysinfo	Network	Emulator	
l	Telepho	ony Statu	5						
I	Voice:	home	~	Speed:	Full 🗸				
l	Data:	home	~	Latency:	None 🗸 🗸				
	Telepho	ony Actio	าร						
ł	Incomi	ing numb	er: 136	8173259	6				
ł	● Voi	ce							
l		S							
l	Messa	ge:							
l									
	Call	lang Up							
l	Location	Controls							
l	Manual	GPX	KML						
l	Dec	imal							
l	○ Sexagesimal								
	Longitu	ıde -122.	084095						
l	Latitud	e 37.42	2006						
l	Send								
1									

cepts Java regexes. Prefix with pid:, app:, tag: or text to limit scope.

•	55	54:Jerry	\checkmark
			13:38 E
XXX	UNE 12	АМ	;

www.3testing.com Copyright @2005-2016



使用DDMS模拟打电话





使用DDMS模拟打电话




DDMS使用

使用DDMS模拟发短信





DDMS使用

练习:10分钟

- 学习使用DDMS
- 查看进程等信息
- 模拟打电话
- 模拟发短信



目录



•移动软件测试技巧 •安卓系统测试 E ▶Andriod 各种UI测试框架介绍 ≻UiAutomator UI自动化测试框架 ▶Android黑盒自动化测试Monkey/MonkeyRunner ▶Selenium/WebDriver的使用 录 ▶DDMS使用 ▶多平台测试 ▶星云产品 • APP测试讨论 www.3testing.com Copyright @2005-2016



名平台测试

- http://www.testin.cn/
- Testin特性
- 真机测试:终端云 节省测试设备购买租赁成本
- Testin云测试基于云端部署超过300款3000多示 云流的Pad、Phone、Touch、Smart TV等 智能移动设备,实时上架最新终端,免去测试终端的购买、租赁等诸多烦恼。
- 自动化测试: 高效率节省测试人员成公及时间
- 彻底告别原始的人工测试,5分钟内自动完成安装/卸载、启动/运行、UI适配等枯燥手 工测试,保障App应用高质量快速迭代,按期发布最新版本。
- 云测试: 云测试 服务全球移动互联网开发者
- 7×24小时不间断服务, 全球任何国家和地区均可在线选择真机进行App应用与终端之间的自动化兼容适配测试及功能测试,一键提交,自动出具规范化的测试报告。



多平台测试

测试类型

1) 兼容测试

①安装卸载测试:测试App在指定终端上是否可证常安装、正常卸载,准确定位错误原因。 ②遍历测试:自动识别App可执行的功能,在---定时间内遍历App的不同功能界面,通过 截图记录操作路径并输出日志、定位异常现象。

③运行稳定性测试:类似Monkey的随机性压力测试,测试App运行期的稳定性。

④UI适配测试:测试App的UI与目标终端的屏幕是否适配,记录是否存在渲染失败、错位、

黑边框、黑白屏等现象。







多平台测试

- 测试类型
- 2) 性能测试
- ①启动时间检测:检测App在终端上首次启动时间。
- ②内存、CPU耗用检测:检测App在终端上运行时不同时段占用内存、CPU情况。
- ③流量耗用检测:检测App在终端上运行时的网络流量消耗情况。
- ④电池温度检测:检测App在终端上运行时,对终端的电池温度等性能指标的影响情况。







多平台测试

测试类型

3) 功能测试

①自定义脚本测试:上传自定义脚本,脚本下公出准确的测试方法,能自动定位错误及反

馈出错原因, 能在结果报告中呈现测试过程出现的bug并提供重现步骤。利用JUnit快速定

位代码错误,帮助您正确改善产品质量,

②执行结果判定:比对每个用例的测试结果,未通过用例给出准确的日志分析。

③支持Robotium、淘宝Athrun框架编写的自动化测试脚本。







多平台测试

测试步骤



多平台测试





www.3testing.com Copyright @2005-2016



多平台测试





多平台测试

选择被测平台,分辨率

选择测试终端						
Top 300 0						
16.15.16 ×						
© Android 15.3.5-RT-2	C Android 22.1	E Android 222	Android 2.2	0 E# (167	8/48	
Android 2.3.7	Android 2.3	III Android 3.1	Android 3.2.1	□ <u>=</u> <u>A</u> (<u>A</u> (<u>y</u> '(<u>b</u> , (b)) 2 □ <u>=</u> <u>H</u> (<u>b</u>) (<u>b</u> , (b), (b), (b)) 2	R/16#1 80940.st #38	
E Android 3.2	C Android 4.0.0	C Android 4.0.1	🗍 Android 4.0.3 💴	U == 5Ph D. 14 40.4		
C Android 40.4	C Android 4.0	Android 4.1.1	🛙 Android 4.1.2 📟	2 == GAL TID 370 (SM.		
C Android 4.1.4	C Android 4.1	C Android 4.2.1	III Android 4.2.2	7 = Galaxy Fit (GT-55670) -		
C Android 4.2	Android 4.3.0	C Android 4.3.1	🗍 Android 4.3 📟	GALAXY Trend (SCH-36		
C Android 4.4.2	C Android 4.4.4	III Android 5.0	Android 6517,4.0.0	回 三星 GALAXY S4 Mini (GT-L.		
E AliyunOs 2.0.0	III AliyunOs 2.0.1	AliyunOs 2012.3	C Android 30564	三重 Galaxy Trend Duos II (
SHE				回 三重 Galaxy Mega 6.3 (GT-L		
(Taxana)	10 march 10	10 ann an 1		E GALAXY S4 LTE-A (SH		
0 240'320	E 256-320	E 420°480	W 492'728	回 三星 Galaxy SII (GT-89100G)		
10 480°800	W 480-850	0 4827854	2 A T960	回 三種 Galaxy S Duos (SCH-30		
UI 540'888	W 540'900	UI 544'960		回 三星 Galaxy Tab 2 (GT-P511		
1024-600	E 600-1024	00 640 960	Constantin	回 三星 SM-G18190 41.2		
10 1024-768	······································	120°1280	0 79/21280	回 三重 GT-19502 4.4.2		
0 800*1280	10 1280°800	10 2080° 305	1080°1800	E E Galaxy Trend 3 (SM-G		
1080°1920	1152-1920	C 1920*1200	W 1200-1920	回 三星 Galaxy Note 10.1 (GT		
10/ 2048-2640	···· 1220.752	= 2048'7,490	iii 1440-2300	□ 三星 GALAXY SⅢ (SO+-1535		A
				三三里 Galaxy Grand (GT-8912		No. 10 Annual Departies

www.3testing.com Copyright @2005-2016



多平台测试

选择被测平台

NECCO	SP BH	1589940	HARM	584-0	(6)						+ marmar
意动于机祥州	(1.3.4) - 26178	in.									
Æ	276920	用即手约(平向 (1.3.4)			文明大手	11.88 MB			18-212-71	2014-1-2-2_58-48	
\leq	RICHS	#@@it			Ridiks	Richit			R*A.*	100 / 311 (11)第1年353行	
重点服务	再业报告	2.0754	15161716	HRAD	TER:	· 应用评慎	凝重分析			R.	O WHENE
安耕东款(2) 原面(ウ 143 75	运获着和ISA用户(用户要推到保護	2.2011世第)						亚洲 安留失敗 经结
1875 C8813 4	113	<u>19872-0</u>	620-172-4.3								
运行共教(5) 聚素白 約万1	法获载和过渡用户(月	10世史出於17年4	(型は約注意)							重测 运行失败 终端
\$875 Y300C A	112	三篇的	CH 2992LA3		ER SM-	G3508L4.3			43	三篇 SM-T2558 4.3	
通过(293)	原並约 13281 7	5 活动繁荣的高度中。	(用户要盖款很多	1.1.2.4.1.1.1.1)						
HTC HILE 4.1	12	HTCS	00843.2		HTC 8064	412		HTC 609d 4.1.2		HTC 7088 4.1.2	
HTC 709d 4.3	12	HTC.8	24.4.2.2		HTC 8024	LU.		HTC 8160 4.3		HTC M85w 4.4.2	
HTC 5720# 4	0.4	H7C.57	7201411		HTC TA	104		HTC T327: 4.0.3		HTC 13284 4.0.3	
HICTION 4	63	HIGH	128w 4.0.3		HICD			HTC 1528+ 4.0.4		HTC 87204413	
HTC 2550# 4	0.3	0890	N1W 4.2.2		OPPO NS	11.41		OPPO 82017 4.3		OPPO #6007.4.3	
OFPO RADEZ	4.1	OFRO.	88091.42.1		OPPO RA	11.9 /18		OPPO RELST 4.2	L	OPPO 88197,42,1	
OPPO RAZIT	4.2.2	0790	R823T 42.1		1000 P	01422		OFPO RE29T 4.2	2	OPPO 8430 4.2.2	





多平台测试

功能测试结果

VEDITIR CLUMP - MILLIN	ER,								
0.00010	809078(114)		2410	11.89.548			18.1 ¥7 €	2014-11-25 23.58-48	
ecets.	wate.		Ricks	8055			842.88	300.7.111 (21)#04(0)/E	
AND ARNO	2.0112	-	1650	40180748	8.275	1			Ó Rom
		(HICK)		0.0775		0.078			
rents	293	0		2		5		818	17.6. %
416	97.7%	0.0%		0.7%		1.7%			
ECHALALI HUMA RETE WITHIN (ALIO) E 5420-572.A.1 HEM	A SAA JUBPA Jacomerska stokasa III. SA JUBPA UID. Oranasto								in matter
the second se									





多平台测试

结果出来后查看性能





目录

•移动软件测试技巧 •安卓系统测试 E ▶Andriod 各种UI测试框架介绍 ▶UiAutomator UI自动化测试框架 ▶Android黑盒自动化测试Monkey.MonkeyRunner ▶Selenium/WebDriver的使用 录 ▶DDMS使用 ▶多平台测试 ▶星云产品 • APP测试讨论 www.3testing.com Copyright @2005-2016

啄木鸟软件测试培训网

星云测试



Copyright @2005-2016

软件示波器:无线、自动采集被测应用的代码运行逻辑和指标数据



软件示波器,在实际测试的过程中,测试员
可以看到类似于心电图的数据获取模式,让
软件测试过程因此变得有趣、可视、精准。
星云示波器的Console模式优点:
> 可视化观察程序运行指标,辅助测试人员
对被测试程序进行分析,例如等价类的辅

助划分、探索性测试等。 ➤ 示波器可以实现WildTest(场测),无需

· 示波器可以实现Wildlest(场测),元需 与PC的连线通过wifi就可以实现高速获取 被测数据。



星云测试

星云代码级崩溃捕获:捕获所有闪退,无需重观, 视大减少定位问题时间

控制流程图 已经法行	代码 函 块:2437行:3 块:2439行:3 块:2440行:3 块:2442行:3 块:2442行:3 块:2445行:3 块:2445行:3 块:2445行:3 块:2445行:3	887 (888 (888 (888 (888 (888 (888 (900 (901 (902 (903 (904 (905 (905 (907 (未送行 tch (#ViewFosition) ViewUtil, USER: (#UserInfo, getUgeI) :UserInfo, closeUge() :e exit(): :ak: ViewUtil, HOWE:	47887 自动记录崩溃发生时刻之 前程序执行的详细路径信息,捕获难以重现的缺陷 并快速解决: 最后50个代码块。
测试用例追溯 块号	函数遣溯 文件号	代码追溯 块首行	依赖用例追溯 执行序号	最后期推荐 最后50个判定执行
1 2464	24 41	19	3269	
2 2461	24 41	16	3268	
4 2451	24 41	99	3266	
5 6877	53 35	52	3265	
6 2448	24 39	97	3264	
7 2447	24 39	97	3263	
8 2446	24 39	96	3262	www.3testing.com
9 2438	24 38	38	3261	Convright @2005_2016
块	条件	判定		Copyright @2005-2016

星云测试



Copyright @2005-2016

回归测试用例选取算法

大型系统维护—智能的回归测试用例选取分析算法

在回归测试时,基于智能算法,完全自动 筛选测试用例,大大减少了回归测试的时

间。

计算每个测试用例受影响的程序,用户可 以根据此数据来进行用例回归测试的优先 级排序,把高风险的用例测试放到前面。

生成测试用、`数:	117			排序: 测试用例回归优势			
回'.4版4. Nid	回归版本测测,用例名称	测试用例最后运行版本	测试用例最后运行时间	测试用例最后运行状态	测试用例回归优先级	函数剖面变化比例	
28	` s st	15/11/13	2015-11-16 14:02:49	BUG	120	29.4%	详细
·	附近酒店和菜单切换(未登录)	15/11/13	2015-11-15 13:22:18	BUG	75	34.1%	详细
24	所有情趣酒店详细信息	15/11/13	2015-11-15 15:06:14	BUG	67	31.4%	详细
2	附近酒店和菜单切换(登录状 态)	15/11/13	2015-11-15 13:20:17	崩溃	63	29.2%	详细
27	测试补充	15/11/13	2015-11-16 12:21:13	崩溃	61	29.1%	详细
3	优惠券查看页面登录页面忘记密 码	15/11/13	2015-11-16 11:30:50	BUG	56	30.6%	详细
20	附近酒店列表 (登录和未登录状 态)	15/11/13	2015-11-15 13:49:53	BUG	48	37.9%	详细
5	酒店详细信息订单按钮	15/11/13	2015-11-15 14:26:08	BUG	47	29.9%	详细
0	刷新我的收藏	15/11/13	2015-11-15 14:50:24	BUG	45	35.2%	详细
10	立即支付	15/11/13	2015-11-15 14:45:22	崩溃	45	41.1%	详细
34	刷新我的订单列表	15/11/13	2015-11-15 14:46:37	BUG	43	41.7%	详细
1	附近酒店地图列表切换(登录)	15/11/13	2015-11-15 13:18:49	崩溃	42	42%	详细
29	筛选	15/11/13	2015-11-15 14:07:49	崩溃	42	32.5%	洋细
79	取消订单	15/11/13	2015-11-15 14:44:08	BUG	42	40.4%	详细
.0	附近酒店地图列表切换(未登录 状态)	15/11/13	2015-11-15 13:13:39	崩溃	41	30.8%	详细
16	选择城市后查看列表	15/11/13	2015-11-15 13:40:17	BUG	39	28.4%	详细



星云测试



全方位的技术指标,提供数字化验收支持

技术债务分析模型,分析软件内部设计在未来 维护中的附加成本。可以对软件交付和验收授 供直观的参考。

- ▶ 测试充分度
- ▶ 可维护性、可测试性
- ▶ 代码重复块、代码违规、代码注释
- > 测试贡献度分析、测试用例执行分析
- ▶ 崩溃的统计及代码级别崩溃分析

100	Alexand and				-
	лю	asta-	kabdin2.0		
					Q LILWE
	60940965837.43			经米结为 6年以前约3	
		6% Φ	Total A	11103 + HE	
	NUMBER OF	274280 HI 880 H 880 H	* 🔶 * 🔛	A2098- NO.+ 2	
	1.1744	dent) des	с		- 10.
	жан		609.8.4.9.54k	CELLERORI	10414532
	17	12	rinutex+ 74.2% 205日 	100064-11.0%	101/01/1 26,8% 000/10 1000 000/000 100
	80002			Designation of the second seco	SARCHS HT
	11.0 6 metty,	85		27-922/2010 101	
	609322;0:044			629Crash	
			- de - a = 20		



目录



啄木鸟软件测试培训网

结束



Thank You

顾翔

啄木鸟软件测试培训网 QQ: 2025344 微信号: xianggu0625 Email: xianggu625@126.com 网站: www.3testing.com 微信公众号: 见二维码

