

极客学院
jikexueyuan.com

AndroidJUnitRunner

AndroidJUnitRunner—课程概要

- AndroidJUnitRunner介绍
- Instrument测试命令
- AndroidJUnitRunner过滤器与例子
- AndroidJUnitRunner参数

AndroidJUnitRunner介绍

AndroidJUnitRunner介绍

AndroidJUnitRunner

- 是一个测试运行器，用于运行JUnit3和JUnit4的Android测试包
- 替换InstrumentationTestRunner
- 支持InstrumentationTestRunner的所有特性，但又进行了扩展
- 保持所有InstrumentationTestRunner的命令格式

继承关系

java.lang.Object

android.app.Instrumentation

android.support.test.runner.MonitoringInstrumentation

android.support.test.runner.AndroidJUnitRunner

Instrument测试命令

Instrument测试命令

命令使用格式：

```
am instrument [-r] [-e <NAME> <VALUE>] [-p <FILE>] [-w]  
               [--user <USER_ID> | current]  
               [--no-window-animation] [--abi <ABI>] <COMPONENT>
```

COMPONENT: 包名/运行Runner

运行所有用例命令例子：

```
adb shell am instrument  
-w com.jikexueyuan.test/android.support.test.runner.AndroidJUnitRunner
```



Instrument测试命令

参数	说明
-r	打印原始结果(否则解码REPORT_KEY_STREAMRESULT)。使用[- e perf true]生成原始输出性能度量。
-e <NAME> <VALUE>	设置参数<NAME> <VALUE>。 为测试运行器常见的形式是 [-e <testrunner_flag> <value>[,<value>...]].
-p <FILE>	将profiling数据写入到文件中
-w	等待测试完成。这是一个必须参数
--user <USER_ID> current	指定用户运行， 如果不指定则使用当前用户运行
--no-window-animation	关掉窗口动画在运行
--abi <ABI>	通过所选的ABI启动进程。 选择支持的CPU架构

AndroidJUnitRunner过滤器与例子

AndroidJUnitRunner过滤器与例子

本课时我们将学习以下知识点：

- 过滤器
- 命令例子

熟悉执行命令，对编写运行框架与控制用例的运行时有非常帮助的

AndroidJUnitRunner过滤器与例子— 过滤器

过滤参数	说明
-e testFile <filePath>	运行文件中指定的用例
-e package <packageName>	运行这个包中的所有用例
-e size [small medium large]	运行注释SmallTest /MediumTest / LargeTest的用例
-e annotation <annotation>	运行指定注释的用例
-e notAnnotation <annotation>	运行不包含指定注释的用例
-e numShards <num>	将用例分割成不同的切片
-e shardIndex <id>	运行指定切片id的用例

AndroidJUnitRunner过滤器与例子— 例子

运行所有的用例:

```
adb shell am instrument -w  
com.android.foo/android.support.test.runner.AndroidJUnitRunner
```

运行一个类中的所有用例:

```
adb shell am instrument -w -e class com.android.foo.FooTest  
com.android.foo/android.support.test.runner.AndroidJUnitRunner
```

运行单个测试用例:

```
adb shell am instrument -w -e class com.android.foo.FooTest#testFoo  
com.android.foo/android.support.test.runner.AndroidJUnitRunner
```

运行多个类的所有用例:

```
adb shell am instrument -w -e class  
com.android.foo.FooTest,com.android.foo.TooTest  
com.android.foo/android.support.test.runner.AndroidJUnitRunner
```

AndroidJUnitRunner过滤器与例子— 例子

运行文件中的所有用例列表:

```
adb shell am instrument -w -e testFile /sdcard/tmp/testFile.txt  
com.android.foo/com.android.test.runner.AndroidJUnitRunner
```

文件应该包含一组行分离测试类和可选方法
指定格式为: com.android.foo.FooClassName#testMethodName

运行JAVA包中的所有测试用例:

```
adb shell am instrument -w -e package com.android.foo.bar  
com.android.foo/android.support.test.runner.AndroidJUnitRunner
```

调试测试,在代码中设置断点和传递: -e debug true 参数

运行指定测试切片的用例, 即在注释中指定的SmallTest /MediumTest / LargeTest:

```
adb shell am instrument -w -e size [small|medium|large]  
com.android.foo/android.support.test.runner.AndroidJUnitRunner
```

AndroidJUnitRunner过滤器与例子— 例子

过滤器通过给定的注释过滤运行用例:

```
adb shell am instrument -w -e annotation com.android.foo.MyAnnotation  
com.android.foo/android.support.test.runner.AndroidJUnitRunner
```

如果使用多个选项,结果测试运行将为两者的交集.

例如 “-e size large -e annotation com.android.foo.MyAnnotation” 将只运行包含两者LargeTest and “com.android.foo.MyAnnotation” 注释的用例

过滤器运行没有指定注释的用例:

```
adb shell am instrument -w -e notAnnotation com.android.foo.MyAnnotation  
com.android.foo/android.support.test.runner.AndroidJUnitRunner
```

如上所述,如果使用其他选项,结果测试运行将包含两个选项的交集. 例如: “-e size large -e notAnnotation com.android.foo.MyAnnotation” 将只允许含有 LargeTest 注释的用例 和 没有 “com.android.foo.MyAnnotation” 注释的用例

AndroidJUnitRunner过滤器与例子— 例子

过滤器运行没有指定注释列表中的用例:

```
adb shell am instrument -w -e notAnnotation  
com.android.foo.MyAnnotation,com.android.foo.AnotherAnnotation  
com.android.foo/android.support.test.runner.AndroidJUnitRunner
```

过滤器运行指定测试切片，切片数必须为1到numShards之间的整数

```
adb shell am instrument -w -e numShards 4 -e shardIndex 1  
com.android.foo/android.support.test.runner.AndroidJUnitRunner
```

AndroidJUnitRunner参数

AndroidJUnitRunner参数

参数	说明
-e log true	只在日志模式下运行 这个选项将加载和遍历所有测试的类和方法,但会绕过实际测试执行。用于快速获取信息要执行的测试的仪器命令
-e coverage true	生成代码覆盖率 这需要一个EMMA的构建。默认情况下,代码覆盖率结果文件将被保存在/data/ /coverage.ec 文件,除非有coverageFile标志
-e coverageFile <path>	指定代码覆盖率文件保存路径 -e coverageFile /sdcard/myFile.ec
-e listener <runlistener>	指定一个或多个RunListeners观察测试运行 -e listener com.foo.Listener,com.foo.Listener2
-e timeout_msec <int>	设置超时时间(以毫秒为单位),将被应用到每一个测试 -e timeout_msec 5000
-e disableAnalytics true	禁用谷歌分析

AndroidJUnitRunner参数

参数还可以在AndroidManifest文件中进行指定

如使用监听器

```
android:name="android.support.test.runner.AndroidJUnitRunner" ...
```

```
<meta-data android:name="listener"
```

```
android:value="com.foo.Listener,com.foo.Listener2" />
```

参数通过shell 命令传入将覆盖清单指定参数.

AndroidJUnitRunner

本套课程中我们学习了AndroidJUnitRunner。你应当掌握了以下知识：

- AndroidJUnitRunner介绍
- Instrument测试命令
- AndroidJUnitRunner过滤器
- AndroidJUnitRunner参数

你可以使用这些知识来测试你的应用，如果想继续提高，你可以继续在极客学院学习其他测试相关课程。

极客学院

jikexueyuan.com

中国最大的IT职业在线教育平台

