Android 平台开发静态 jar单体测试示例

Last edited by **caoquanli** 1 month ago

Android平台开发静态jar单体测试示例

Table of Contents

在Android定制开发过程中，有时候我们会添加自己的jar，为了质量要求会对这些自定义的jar 进行单体测试，并提供测试覆盖率的报告。本文就这一问题提供解决手顺。

1. 构建Jar和测试APK

这个例子将会以external目录下面创建一个jar为起点展开描述。

1. 首先在external/下面创建如下目录： `foo/src/com/example/foo`

2. 在上面这个目录下面添加Foo.java文件

```
 package com.example.foo;

public class Foo {
    public int getNumber(int i) {
        return i;
    }
}
```

3. 在external/foo目录下面添加Android.mk文件：

```
 LOCAL_PATH := $(call my-dir)
include $(CLEAR_VARS)

LOCAL_MODULE := com.example.foo

LOCAL_SRC_FILES := $(call all-java-files-under, src)

include $(BUILD_STATIC_JAVA_LIBRARY)

include $(call all-makefiles-under,$(LOCAL_PATH))
```

4. 在external/foo下面创建一个test目录存放测试代码

   - 首先创建测试文件：src/com/example/foo/test/FooTest.java

```
 package com.example.foo.test;
import com.example.foo.Foo;

import android.support.test.runner.AndroidJUnit4;
import android.util.Pair;
import org.junit.Test;
import org.junit.runner.RunWith;
import static org.hamcrest.Matchers.is;
import static org.junit.Assert.assertThat;


@RunWith(AndroidJUnit4.class)
public class FooTest {
    @Test
    public void testGetNumber() {
        Foo foo = new Foo();
        int i = foo.getNumber(3);
        assertThat(i,is(2));
    }
}
```

   - 创建AndroidManifest.xml

```
 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.foo.test">
    <application>
        <uses-library android:name="android.test.runner" />
    </application>

    <instrumentation android:name="android.support.test.runner.AndroidJUnitRunner"
        android:targetPackage="com.example.foo.test"
        android:label="Tests for com.example.foo" />
</manifest>
```

- 编写mk文件

```
 LOCAL_PATH:= $(call my-dir)
include $(CLEAR_VARS)

LOCAL_MODULE_TAGS := tests

LOCAL_SRC_FILES := $(call all-java-files-under, src)

LOCAL_PACKAGE_NAME := FooTest

LOCAL_JAVA_LIBRARIES := android.test.runner

LOCAL_STATIC_JAVA_LIBRARIES := ub-uiautomator junit legacy-android-test android-supp

LOCAL_CERTIFICATE := platform

LOCAL_JACK_COVERAGE_INCLUDE_FILTER := com.example.foo.*

LOCAL_PRIVATE_PLATFORM_APIS := true

include $(BUILD_PACKAGE)

include $(call all-makefiles-under,$(LOCAL_PATH))
```

- 编写AndroidTest.xml，主要用于TF框架生成覆盖率报告,具体的信息可以参考: Android Trade Federation(TF)框架单体测试

```
 <configuration description="Runs Tests for foo test">
    <target_preparer class="com.android.tradefed.targetprep.InstallBuildEnvApkSetup'
        <option name="apk-name" value="FooTest.apk" />
    </target_preparer>
     <option name="test-suite-tag" value="Footest" />
    <test class="com.android.tradefed.testtype.CodeCoverageTest" >
        <option name="package" value="com.example.foo.test" />
        <option name="runner" value="android.support.test.runner.AndroidJUnitRunner'
        <option name="metadata-file" value="out/target/common/obj/APPS/FooTest_inter
        <option name="source-dir" value="external/foo/src" />  <!--被测试代码的包的目录
    </test>
</configuration>
```

2. 生成coverage.em文件

在源码根目录下面运行 `EMMA_INSTRUMENT_STATIC=true  mmma -j8 external/foo/` .上述命令会生成 out/target/common/obj/APPS/FooTest_intermediates/coverage.em

3. 使用TF框架运行单体测试并生成单体测试报告

在源码路径下面运行 tradefed.sh，然后输入run external/foo/test/AndroidTest.xml命令:

```
tf >run external/foo/test/AndroidTest.xml
01-07 15:57:59 I/TestInvocation: Invocation was started with cmd: external/foo/test/Andro
idTest.xml
01-07 15:57:59 W/TestInvocation: Using the test-tag from the build_provider. Consider upd
ating your config to have no alias/namespace in front of test-tag.
01-07 15:57:59 I/TestInvocation: Starting invocation for 'stub' with '[ BuildInfo{bid=0,
target=stub, serial=GFEDCBA0987654321} on device 'GFEDCBA0987654321']
01-07 15:57:59 W/FileUtil: Chmod is not supported by this OS.
01-07 15:57:59 W/FileUtil: Chmod is not supported by this OS.
```

```
01-07 15:57:59 W/FileUtil: Chmod is not supported by this OS.
01-07 15:57:59 I/InstallBuildEnvApkSetup: Installing FooTest.apk on GFEDCBA0987654321
01-07 15:58:02 W/FileUtil: Chmod is not supported by this OS.
01-07 15:58:02 I/TextResultReporter: Saved device_logcat_setup log to /tmp/0/stub/inv_442
2312557002179652/device_logcat_setup_1663073802664629022.zip
01-07 15:58:02 I/RemoteAndroidTest: Running am instrument -w -r   -e timeout_msec 300000
-e coverage true com.example.foo.test/android.support.test.runner.AndroidJUnitRunner on u
nknown-car_on_mt2712-GFEDCBA0987654321
01-07 15:58:03 I/InvocationToJUnitResultForwarder: run com.example.foo.test started: 1 te
sts
tf >.
Test com.example.foo.test.FooTest#testGetNumber: failed
java.lang.AssertionError:
Expected: is (2)
but: was (3)
at org.hamcrest.MatcherAssert.assertThat(MatcherAssert.java:18)
at org.junit.Assert.assertThat(Assert.java:956)
at org.junit.Assert.assertThat(Assert.java:923)
at com.example.foo.test.FooTest.testGetNumber(FooTest.java:17)
at java.lang.reflect.Method.invoke(Native Method)
at org.junit.runners.model.FrameworkMethod$1.runReflectiveCall(FrameworkMethod.java:50)
at org.junit.internal.runners.model.ReflectiveCallable.run(ReflectiveCallable.java:12)
at org.junit.runners.model.FrameworkMethod.invokeExplosively(FrameworkMethod.java:47)
at org.junit.internal.runners.statements.InvokeMethod.evaluate(InvokeMethod.java:17)
at org.junit.internal.runners.statements.FailOnTimeout$CallableStatement.call(FailOnTimeo
ut.java:148)
at org.junit.internal.runners.statements.FailOnTimeout$CallableStatement.call(FailOnTimeo
ut.java:145)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at java.lang.Thread.run(Thread.java:764)
01-07 15:58:03 I/InvocationToJUnitResultForwarder: run ended 16 ms
tf >

01-07 15:58:03 I/CodeCoverageTest: coverage file from device: /tmp/coverage_7192723171902
557139.ec
01-07 15:58:03 W/FileUtil: Chmod is not supported by this OS.
01-07 15:58:03 I/TextResultReporter: Saved com.example.foo.test_runtime_coverage log to /
tmp/0/stub/inv_4422312557002179652/com.example.foo.test_runtime_coverage_5311353432635798
454.zip
01-07 15:58:03 I/CodeCoverageTest: create report dir: out/target/testcases/coverage/com.e
xample.foo.test successfully
01-07 15:58:03 I/CodeCoverageTest: Jacoco code coverage report generation success
01-07 15:58:03 W/FileUtil: Chmod is not supported by this OS.
01-07 15:58:03 I/TextResultReporter: Saved device_logcat_test log to /tmp/0/stub/inv_4422
312557002179652/device_logcat_test_3861513594486482280.zip
01-07 15:58:03 W/FileUtil: Chmod is not supported by this OS.
01-07 15:58:03 I/TextResultReporter: Saved device_logcat_teardown log to /tmp/0/stub/inv_
4422312557002179652/device_logcat_teardown_5620749441751906614.zip
01-07 15:58:03 W/FileUtil: Chmod is not supported by this OS.
01-07 15:58:03 I/TextResultReporter: Saved host_log log to /tmp/0/stub/inv_44223125570021
79652/host_log_7488144755445571044.zip
```