[Updated] Android Studio + Robolectric + Gradle, getting it all to work

Posted by kostyay on September 4, 2013 Go to comments Leave a comment (25)

Updated: 24/1/2014 – Changed Robolectric test runner to work with Robolectric 2.2

Updated: 2/11/2013 – How to solve the "!!! JUnit version 3.8 or later expected:" error

In this post I will explain what worked for me in terms of getting Android Studio, Robolectric and Gradle to work in my Android Studio project.

Introduction

For the past few days I've been struggling to get the tests working in my Android project. The environment I use is the latest build of Android Studio (0.26 at the time of writing). My project was generated using the wizard.

After writing a bunch of code I needed to start testing it. After a quick search it turns out Robolectric is a framework to use for testing android applications.

There are loads of threads on stackoverflow on how to get it to work, unfortunately none of them fully worked for me. The tricky part is to get all these frameworks to work together in Android Studio and to convince Android Studio to let us debug and execute the tests in the environment.

How I got it to work

Here's the configuration that worked for me. While not perfect it provides sufficient tools to achieve the desired goal.
My goals were these:

1. Run tests from command line using a custom gradle task

2. Run a single test class from Android Studio (and also debug!! it), this has been by far the trickiest thing to get to work.

My source tree is as follows:

- App
  - src
    - main
      - java
      - AndroidManifest.xml
    - test
      - java

build.gradle

My gradle configuration looks like this:

```
buildscript {
    dependencies {
        repositories {
            mavenCentral()
            mavenLocal()
        }

        classpath 'com.android.tools.build:gradle:0.5.+'
    }
}

apply plugin: 'android'

repositories {
    mavenCentral()
    maven {
        url "https://oss.sonatype.org/content/repositories/snapshots"
    }
}

sourceSets {
    testLocal {
        java.srcDir file('src/test/java')
        resources.srcDir file('src/test/resources')
    }
}

android {
    compileSdkVersion 17
```

```
    buildToolsVersion "17.0.0"

    defaultConfig {
        minSdkVersion 15
        targetSdkVersion 17
    }

    // tell Android studio that the instrumentTest source set is located in the unit
    sourceSets {
        instrumentTest.setRoot('src/test')
    }
}

dependencies {
    compile 'com.android.support:support-v4:13.0.+'

    // Dependencies for the `testLocal` task, make sure to list all your global depen
    testLocalCompile 'junit:junit:4.11'
    testLocalCompile 'com.google.android:android:4.1.1.4'
    testLocalCompile 'com.android.support:support-v4:13.0.+'
    testLocalCompile 'org.robolectric:robolectric:2.1.+'

    // Android Studio doesn't recognize the `testLocal` task, so we define the same d
    // which is Android Studio's test task
    instrumentTestCompile 'junit:junit:4.11'
    instrumentTestCompile 'com.google.android:android:4.1.1.4'
    instrumentTestCompile 'com.android.support:support-v4:13.0.+'
    instrumentTestCompile 'org.robolectric:robolectric:2.1.+'

}

task localTest(type: Test, dependsOn: assemble) {
    testClassesDir = sourceSets.testLocal.output.classesDir

    android.sourceSets.main.java.srcDirs.each { dir -&gt;
        def buildDir = dir.getAbsolutePath().split('/')
        buildDir =  (buildDir[0..(buildDir.length - 4)] + ['build', 'classes', 'debug

        sourceSets.testLocal.compileClasspath += files(buildDir)
        sourceSets.testLocal.runtimeClasspath += files(buildDir)
    }

    classpath = sourceSets.testLocal.runtimeClasspath
}

check.dependsOn localTest
```

Configuring Robolectric

As described here you also need to define a custom Robolectric runner otherwise Robolectric won't find your AndroidManifest.xml. Unfortunately the solution described on that page didn't work for me (probably because I am not using the plugin), so I am using the following class instead.
You will probably need to play with it anyway to get it working (your paths may be different, but you get the

idea).

```
package com.xx.yyy;

import org.junit.runners.model.InitializationError;
import org.robolectric.AndroidManifest;
import org.robolectric.RobolectricTestRunner;
import org.robolectric.annotation.Config;
import org.robolectric.res.Fs;
import org.robolectric.res.FsFile;

public class RobolectricGradleTestRunner extends RobolectricTestRunner {
    public RobolectricGradleTestRunner(Class&lt;?&gt; testClass) throws Initializatio
        super(testClass);
    }

    @Override protected AndroidManifest getAppManifest(Config config) {
        String myAppPath = MyApplication.class.getProtectionDomain().getCodeSource().
        String manifestPath = myAppPath + "../../../src/main/AndroidManifest.xml";
        String resPath = myAppPath + "../../../src/main/res";
        String assetPath = myAppPath + "../../../src/main/assets";
        return createAppManifest(Fs.fileFromPath(manifestPath), Fs.fileFromPath(resPa
    }
}
```

Android Studio Configuration

Other things that I had to do is configure the following in the **Project Settings** screen for **Android Studio**:

1. Make sure the "Test Output Path" is configured correctly
2. In Modules>Dependencies make sure junit is at the top of the list, otherwise I was getting strange errors about junit being out of date.

Current Limitations:

1. Currently you need to manually rebuild the test project from command line using **gradle compileTest** each time you modify your test files.
    1. **Update:** You can run it from inside Android Studio, no need to have the Terminal running. Click Run>Edit Configurations, Add a new Application configuration to execute gradle compileTest.
    2. As Graeme Irwin suggested in the comments you can configure gradle to compile the tests after each successful build of the application.

```
configurations {
  testLocalCompile {
    extendsFrom compile
  }
}
```

2. I still didn't manage to get "Run All Tests" feature in Android Studio do work, but it doesn't seem too important at the moment and hopefully the guys from Roboelectric/Android Studio will fix this in future builds.

Work around for "!!! JUnit version 3.8 or later expected:"

At some point (I believe in version 0.2.7 or 0.2.8) Google removed the Dependencies screen from the Project Settings. This made it impossible to run and debug a single test from inside Android Studio. This is a major feature that I was using massively and I just couldn't live without it. At the point Google returned the Dependencies screen, but I have yet to figure out how to use it. After poking around with this for a while I came up with an ugly hack that achieves the desired effect: Being able to run a single test suite from inside Android Studio and debug it.

Here's how to get it to work:

1. Get Android Studio create a Junit configuration for your test suite
    1. Right click on the test suite class in the Project Structure window
    2. Choose Run>MyTestSuiteTest, choose the one with the JUnit icon (2 arrows <>)
    3. This will fail but a new configuration will be created, you will see a similiar output:

```
/System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/
!!! JUnit version 3.8 or later expected:

java.lang.RuntimeException: Stub!
        at junit.runner.BaseTestRunner.(BaseTestRunner.java:5)
        at junit.textui.TestRunner.(TestRunner.java:54)
        at junit.textui.TestRunner.(TestRunner.java:48)
        at junit.textui.TestRunner.(TestRunner.java:41)
        at com.intellij.rt.execution.junit.JUnitStarter.junitVersionC
        at com.intellij.rt.execution.junit.JUnitStarter.canWorkWithJU
        at com.intellij.rt.execution.junit.JUnitStarter.main(JUnitSta
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAc
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(Delegating
        at java.lang.reflect.Method.invoke(Method.java:597)
        at com.intellij.rt.execution.application.AppMain.main(AppMain
```

2. As you can see it fails because it thinks that the JUnit version is not up to date. This stackoverflow question goes into detail why this happens.
3. Unfortunately Android Studio no longer has the screen that allows to reorder the dependencies.
4. Go ahead and copy the -classpath to a text editor and move the junit dependency to be in the beginning of the classpath. Also find out where the build results of your project go and append

that directory to the end of the class pathes (class pathes are delimited by :). The end result should look somewhat like this:

```
307e8a14ea34684d/asm-4.1.jar:/Users/kostyay/.gradle/caches/artifacts-26/filest
```

5. Notice that the first item in class path is jquery and the last one is the output folder for the built classes
6. Now go to Run>Edit Configurations, find your test configuration and edit the VM Options setting. Paste the above there
7. Try running the tests and it should succeed.
8. Hacky and ugly but works.

Hope this helps.

Programming android, android studio, gradle, junit, robolectric

← Quick tip: Django read only admin pages

Using github to manage a WordPress based website →

Kostya Yegorov Mod → Jack Weber • a month ago

Hey

I don't normally run the tests from AS. I use AS just to debug them. But I guess if debugging works so should running the tests.

Seems like you are having some issues with android support libraries and robolectric.

Make sure you are running the latest version of robolectric (2.2). Also it seems like you are trying to test some activities. Make sure you are using RE activity builder (look here for more information: http://robolectric.org/activit...

▲ ▼ • Reply • Share ›

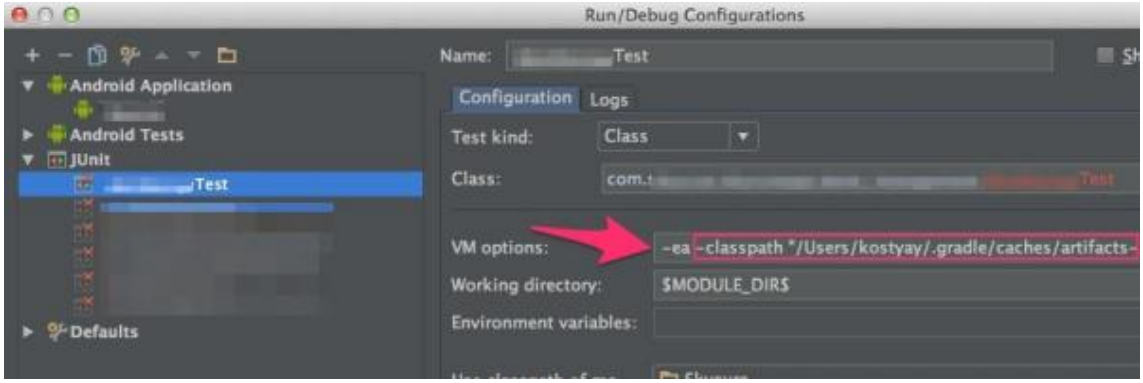
Diego Costantini • a month ago

Hi,
where can I find the classpath to change it manually?
So far I only found the references in the .iml file and it is restored every time (and I couldn't get past the classnotfound issue by moving the output tag).

▲ ▼ • Reply • Share ›

Kostya Yegorov Mod → Diego Costantini • a month ago

Hey
Its under Run>Edit Configurations



▲ ▼ • Reply • Share ›

Diego Costantini → Kostya Yegorov • a month ago

ok, I got until there, but I assumed you have a classpath with a list somewhere and then you just move junit to the top and add output at the end.
If that's not the case, did you have to copy the classpath list from a (failing) run and paste it reordered there?

▲ ▼ • Reply • Share ›

Kostya Yegorov Mod → Diego Costantini • a month ago

Thanks right, you copy it from the failing run and paste it

reordered there (dont forget to add the build output folder in the end as well)

Reply · Share ›

Diego Costantini → Kostya Yegorov · a month ago

wow, that hurts, i wonder why it is taking so long to account for such an essential feature in android studio.
now i can continue fighting against different dependency versions conflicting with each other...
thanks man

Reply · Share ›

Diego Costantini → Diego Costantini · a month ago

hey, did you happen to have issues with resource location?
"android.content.res.Resources$NotFoundException: unknown resource 2131034112"

my test classes output is in bin/classes, but apparently not the right R.class.
I picked one of those resource numbers and I found it only in build/source/.../R.java, but I don't know how to find and tell where the resulting R.class is...

Reply · Share ›

Kostya Yegorov Mod → Diego Costantini · a month ago

I didn't have any issues with resources.
Did you build the tests from command line before attempting to debug them?
It appears that AS doesn't recompile everything when you try to run a test.
Try to run 'gradle localTest' from command line and see if that helps.

Reply · Share ›

Diego Costantini → Kostya Yegorov · a month ago

i was not using any custom sourceSet, and only the res was a problem for some of the tests.
now i am trying to make it work with robolectric 2.3, since it is supposed to converge to android studio.
unfortunately it is an abyss far from my 2.2 snapshot and most of the tests fail, especially due to the big refactory they did on SQLite.
now i am considering in which direction i should put my efforts...

Reply · Share ›

Kostya Yegorov Mod ➚ Diego Costantini • a month ago

We are currently working with Robolectric 2.2 and for database we use the ORMLite package for db access. Although it took a while to get there, everything works fine for us now.

▲ ▼ • Reply • Share ›

юрий рюрик • a month ago

Hi, I am trying to run my robolectric test from android studio, I did the hack with the classpath and the error about junit 3.8 vanished:) but I received new error "Class not found: "com.mypackage.myActivity" " Can you help me ? please?:)

▲ ▼ • Reply • Share ›

Kostya Yegorov ➚ юрий рюрик • a month ago

Hi

Make sure to append the path to your build output directory in the end of the classpath string:

In my case it is:
-classpath
"....:/Users/kostyay/android/SecretApp/Secret/build/classes/testLocal"

▲ ▼ • Reply • Share ›

юрий рюрик ➚ Kostya Yegorov • a month ago

thank you:) it helped me

▲ ▼ • Reply • Share ›

юрий рюрик ➚ юрий рюрик • a month ago

P.S. for Robolectric 2.0+ runner class should be alittle different

```
import org.junit.runners.model.InitializationError;
import org.robolectric.AndroidManifest;
import org.robolectric.RobolectricTestRunner;
import org.robolectric.annotation.Config;
import org.robolectric.res.Fs;

public class RobolectricGradleTestRunner extends
RobolectricTestRunner {
public RobolectricGradleTestRunner(Class testClass) throws
InitializationError {
super(testClass);
}

@Override
protected AndroidManifest getAppManifest(Config config) {
```

```
String myAppPath =
RobolectricGradleTestRunner.class.getProtectionDomain().getCc
String manifestPath = myAppPath +
"../../../src/main/AndroidManifest.xml";
String resPath = myAppPath + "../../../src/main/res";
String assetPath = myAppPath + "../../../src/main/assets";
return createAppManifest(Fs.fileFromPath(manifestPath),
Fs.fileFromPath(resPath), Fs.fileFromPath(assetPath));
}
}
```

Reply · Share ›

Kostya Yegorov Mod ➔ юрий рюрик · a month ago

Thanks, we've just updated to Roboelectric 2.2 and that helped :)

Reply · Share ›

Kostya Yegorov Mod · 3 months ago

Hey, I have updated the guide. It now explains how to get a single test suite to run using the newer android studio versions. Hope this helps!

Reply · Share ›

Graeme Irwin · 3 months ago

To fix limitation number 1 (manually rebuild of test project with gradle) add this to your build.gradle:

assembleDebug.finalizedBy testLocalClasses

This does mean you are compiling tests all the time but I tend to do this anyways.

Also, you can avoid repeating all your compile dependencies by adding this:

```
configurations {
testLocalCompile {
extendsFrom compile
}
}
```

Reply · Share ›

Mick · 3 months ago

How do you achieve your second goal?

"Run a single test class from Android Studio (and also debug!! it), this has been by far the trickiest thing to get to work."

Reply · Share ›

Rod · 5 months ago

Hey Kostya,

I am trying to create my template as well, using testNG as my unit test framework and I have been running all sorts of issues.

Starting with, I am new to Android development (I did java very long time ago and at the moment I am catching up with everything else...)

So I did all your steps as followed:

1. Created a blank Android app project.
2. Added 'test' directory as you specified.
3. Almost copied and pasted your build.gradle updating my dependencies to point to testNG as oppose to junit:
testLocalCompile 'org.testng:testng:6.8.7'
testLocalCompile 'org.uncommons:reportng:1.1.4'
testLocalCompile 'org.mockito:mockito-all:1.9.5'
testLocalCompile 'org.easytesting:fest-assert-core:2.0M10'

Did the very same for instrumentTestCompile

4. Didn't configured everything else for roboelectric as I am not going to be using it (yet) and just want the sample project to test my core functionality.

then do "gradle check" getting a failure saying

Gradle: A problem occurred evaluating project ':Bootstrap'.
> Could not find method sourceSet() for arguments [build_682ufc701d0olbrjsrecmvqbgl$_run_closure4_closure9@212f9b2e] on project ':Bootstrap'.

Have you tried configuring your project for testNG?
Do you know what am I doing wrong?
Cheers.

 · Reply · Share ›

Kostya Yegorov Mod → Rod · 5 months ago

Paste your entire gradle configuration file on http://pastebin.com. I will take a look and try to help.

 · Reply · Share ›

Guest · 5 months ago

Hey, Hi.

I am trying to replicate this on my own and I am getting the following failure when I am running gradle check (or gradle build)

running gradle check (or gradle build)

$ gradle check

FAILURE: Build failed with an exception.
* Where:
Build file '/Users/-/git/gradleTestNG/gradleTestNG/build.gradle' line: 97

* What went wrong:
A problem occurred evaluating project ':GradleTestNG'.
> Could not find method sourceSet() for arguments [build_682ufc701d0olbrjsrecmvqbgl$_run_closure4_closure9@797266b7] on project ':Bootstrap'.
* Try:
Run with --stacktrace option to get the stack trace. Run with --info or --debug option to get more log output.
BUILD FAILED

Kostya Yegorov Mod → Guest • 5 months ago

Hey

Paste your entire gradle configuration file on http://pastebin.com. I will take a look and try to help.

Cayle Sharrock • 5 months ago

Thanks for this. I was 99% of the way there. Setting the test-classes folder was the missing ingredient.

I have found an issue with multi-project builds. The gradlew sets the current path to the subproject folder, while andoid-studio always sets it to the main project folder. As a result I have crappy little hacks like this in my source code to get hold of resource files:

File srcFile = new File("src/test/res/DemoData.db");
if (!srcFile.exists()) srcFile = new File("subProject/src/test/res/DemoData.db");

Have you come across this / found a more elegant solution?

Kostya Yegorov Mod → Cayle Sharrock • 5 months ago

Hey,
Unfortunately no.

The worse thing is that Google removed the screen that modified the class load order (the Modules>Dependencies tab) so at the moment I am no longer able

to run/debug the tests in the IDE. The IDE tries to load junit3.8 intead of 4 which is lower in the dependencies list.

 · Reply · Share ›

 Subscribe Add Disqus to your site

RSS Feed

Recent Posts

- Using github to manage a WordPress based website
- [Updated] Android Studio + Robolectric + Gradle, getting it all to work
- Quick tip: Django read only admin pages
- Website disappeared from Google search results
- How to get the old toolbar (Reset/Power off/Suspend) in VMWare Workstation 8/9

Categories

- Environment
- Programming
- SEO