Robotium

Android Testing with Robotium

Cheng-Zen Yang

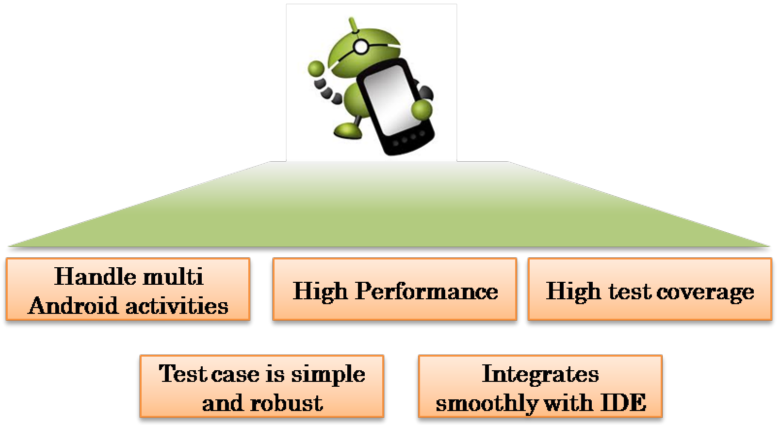
Prev. TA: Peng Lu, Chia-Hui Lin

User scenario testing for Android

- Robotium is an Android test automation framework that has full support for native and hybrid applications.
- Robotium makes it easy to write powerful and robust automatic black-box UI tests for Android applications.
- With the support of Robotium, test case developers can write function, system and user acceptance test scenarios, spanning multiple Android activities.

Benefits

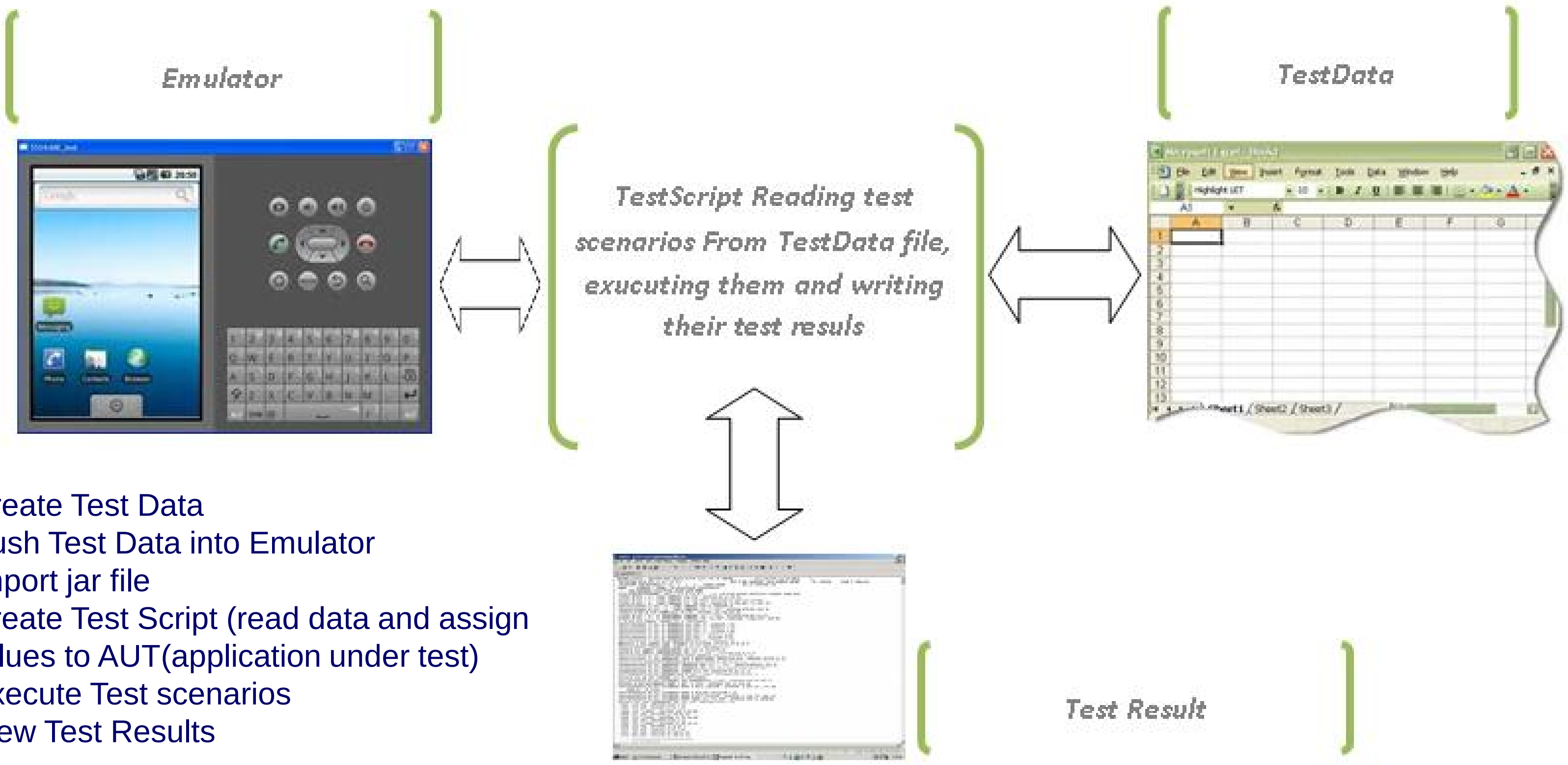
- Test Android apps, both native and hybrid.
- Requires minimal knowledge of the application under test.
- The framework handles multiple Android activities automatically.
- Minimal time needed to write solid test cases.
- Readability of test cases is greatly improved, compared to standard instrumentation tests.
- Test cases are more robust due to the run-time binding to UI components.
- Fast test case execution.
- Integrates smoothly with Maven, Gradle or Ant to run tests as part of continuous integration.


Handle multi Android activities
High Performance
High test coverage
Test case is simple and robust
Integrates smoothly with IDE

http://www.360logica.com/blog/wp-content/uploads/2014/09/Robotium.png

A typical scenario

- Data Driven Testing Architecture


Emulator
TestData
TestScript Reading test scenarios From TestData file, executing them and writing their test resuls
Test Result

* Create Test Data
* Push Test Data into Emulator
* Import jar file
* Create Test Script (read data and assign values to AUT(application under test)
* Execute Test scenarios
* View Test Results

http://2.bp.blogspot.com/-eAVOiJ8y1w8/TWgCO1DUFqI/AAAAAAAAAHw/5QkjKvaI_dc/s1600/Flow.jpg

The steps

- Installation of Robotium
  - To use Robotium in your Android test project, you need to add a dependency to the latest Robotium release to your build file.
  - Robotium tests inherit from ActivityInstrumentationTestCase2 and allows you to define test cases across Android activities.
- Implementation of test cases

The context in Robotium Project

```
@RunWith(AndroidJUnit4.class)
public class NotePadTest {

    private static final String NOTE_1 = "Note 1";
    private static final String NOTE_2 = "Note 2";

    @Rule
    public ActivityTestRule<NotesList> activityTestRule =
            new ActivityTestRule<>(NotesList.class);

    private Solo solo;
    @Before
    public void setUp() throws Exception {
      …
    }
    @After
    public void tearDown() throws Exception {
      …
    }
    public void testAddNote() throws Exception {
       …
    }
    public void testEditNoteTitle() throws Exception {
        …
    }
    public void testRemoveNote() throws Exception {
        …
    }
}
```

The main class for testing with Robotium is Solo. It is initialized with the instrumentation of the test case and the first activity to test.

- setUp
- tearDown
- testXXX(testcase)

The content of Robotium Project

- setUp()
  - setUp() is run before a test case is started.
  - This is where the solo object is created.
- tearDown()
  - tearDown() is run after a test case has finished.
  - finishOpenedActivities() will finish all the activities that have been opened during the test execution.
- testXXX (Your Test Case)

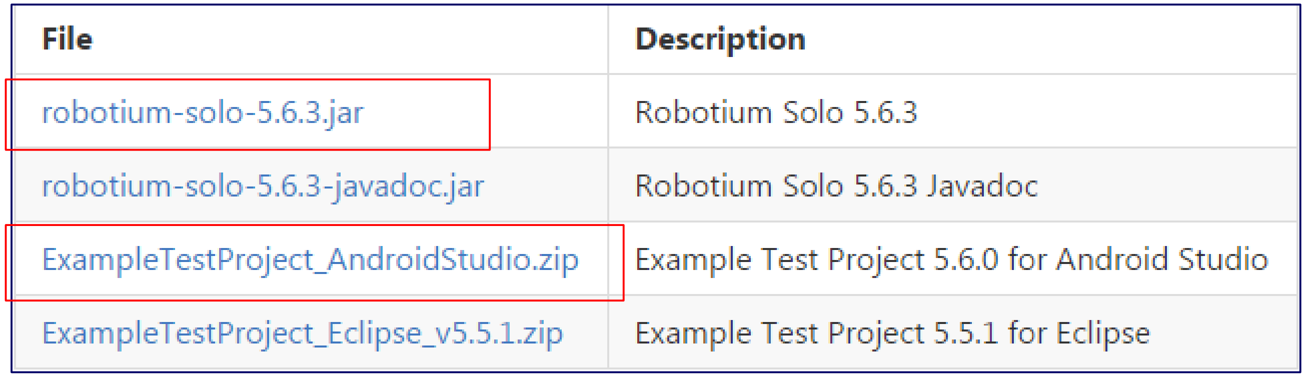
How to install Robotium ?

Step by step

- Download the robotium-solo-5.6.3.jar on the website.
- Open project.
- Import the JAR to your project.
- Start to code your Robotium Project .
- Run the test.

Download

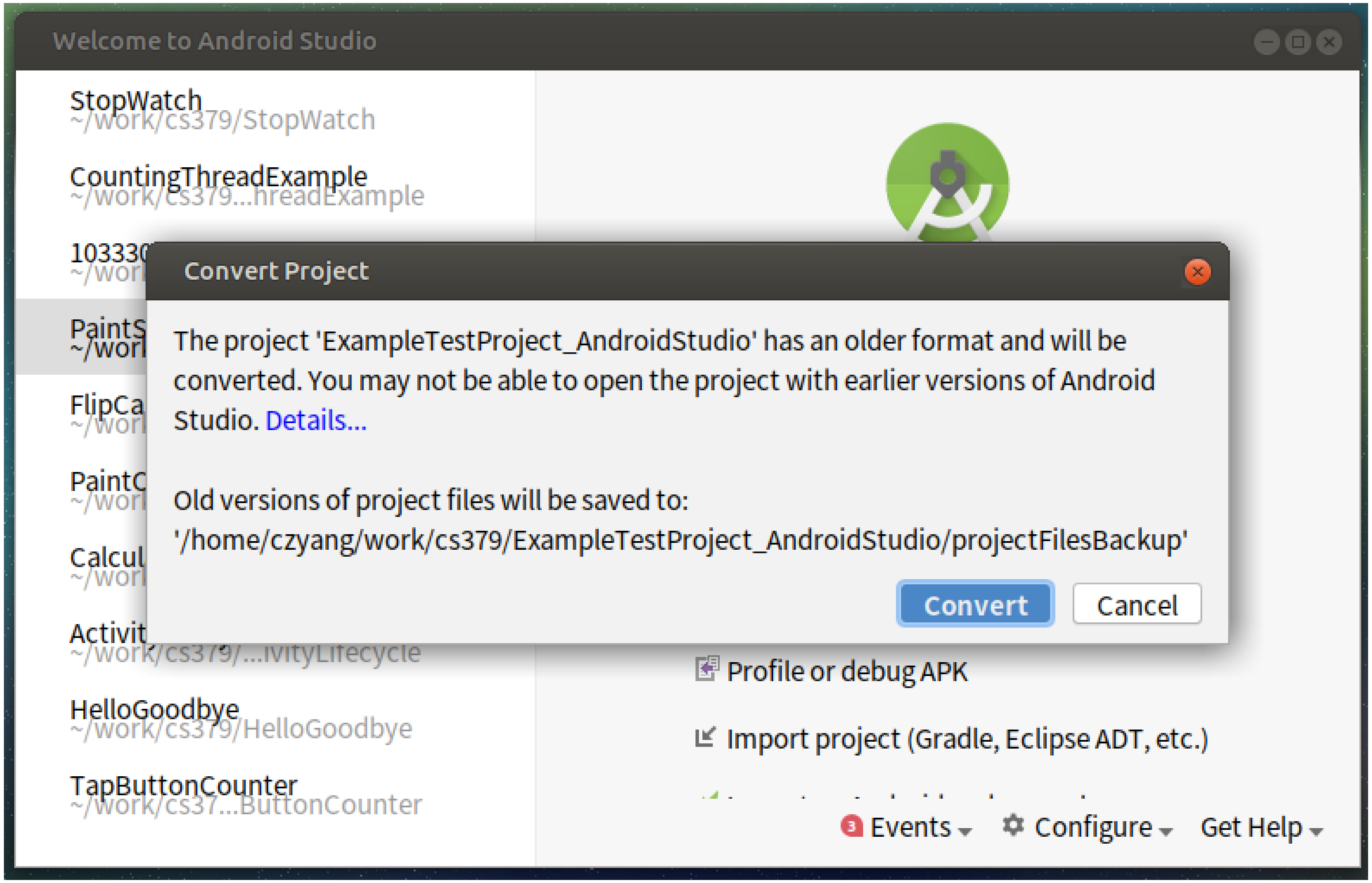
- Download page of robotium.
  https://github.com/RobotiumTech/robotium/wiki/Downloads
  1. robotium-solo-5.6.3.jar
  2. ExampleTestProject_AndroidStudio.zip



File	Description
robotium-solo-5.6.3.jar	Robotium Solo 5.6.3
robotium-solo-5.6.3-javadoc.jar	Robotium Solo 5.6.3 Javadoc
ExampleTestProject_AndroidStudio.zip	Example Test Project 5.6.0 for Android Studio
ExampleTestProject_Eclipse_v5.5.1.zip	Example Test Project 5.5.1 for Eclipse

Open the project ExampleTestProject

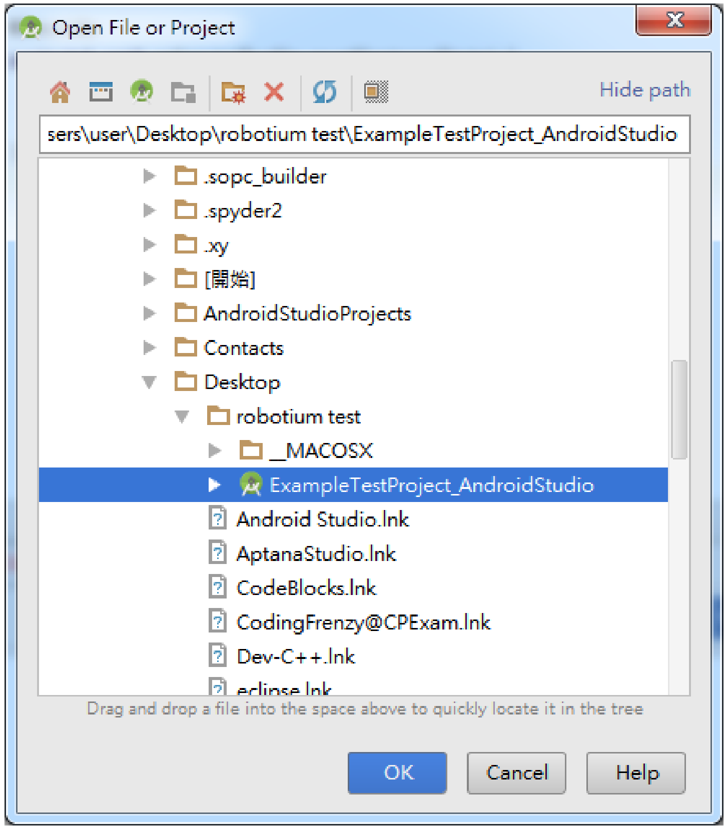
Project conversion



The project 'ExampleTestProject_AndroidStudio' has an older format and will be converted. You may not be able to open the project with earlier versions of Android Studio. Details...

Old versions of project files will be saved to:
'/home/czyang/work/cs379/ExampleTestProject_AndroidStudio/projectFilesBackup'

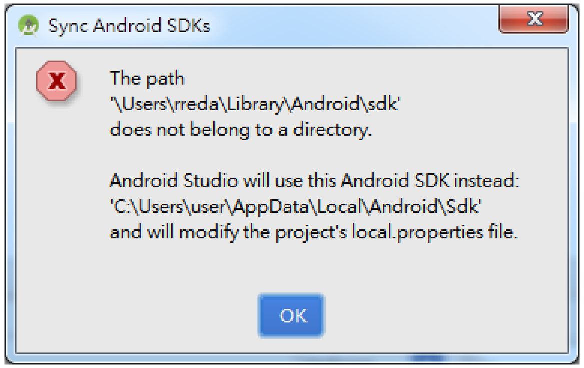
Open the test project



sers\user\Desktop\robotium test\ExampleTestProject_AndroidStudio

```
.sopc_builder
.spyder2
.xy
[開始]
AndroidStudioProjects
Contacts
Desktop
  robotium test
    __MACOSX
    ExampleTestProject_AndroidStudio
  Android Studio.lnk
  AptanaStudio.lnk
  CodeBlocks.lnk
  CodingFrenzy@CPExam.lnk
  Dev-C++.lnk
  eclipse.lnk
```

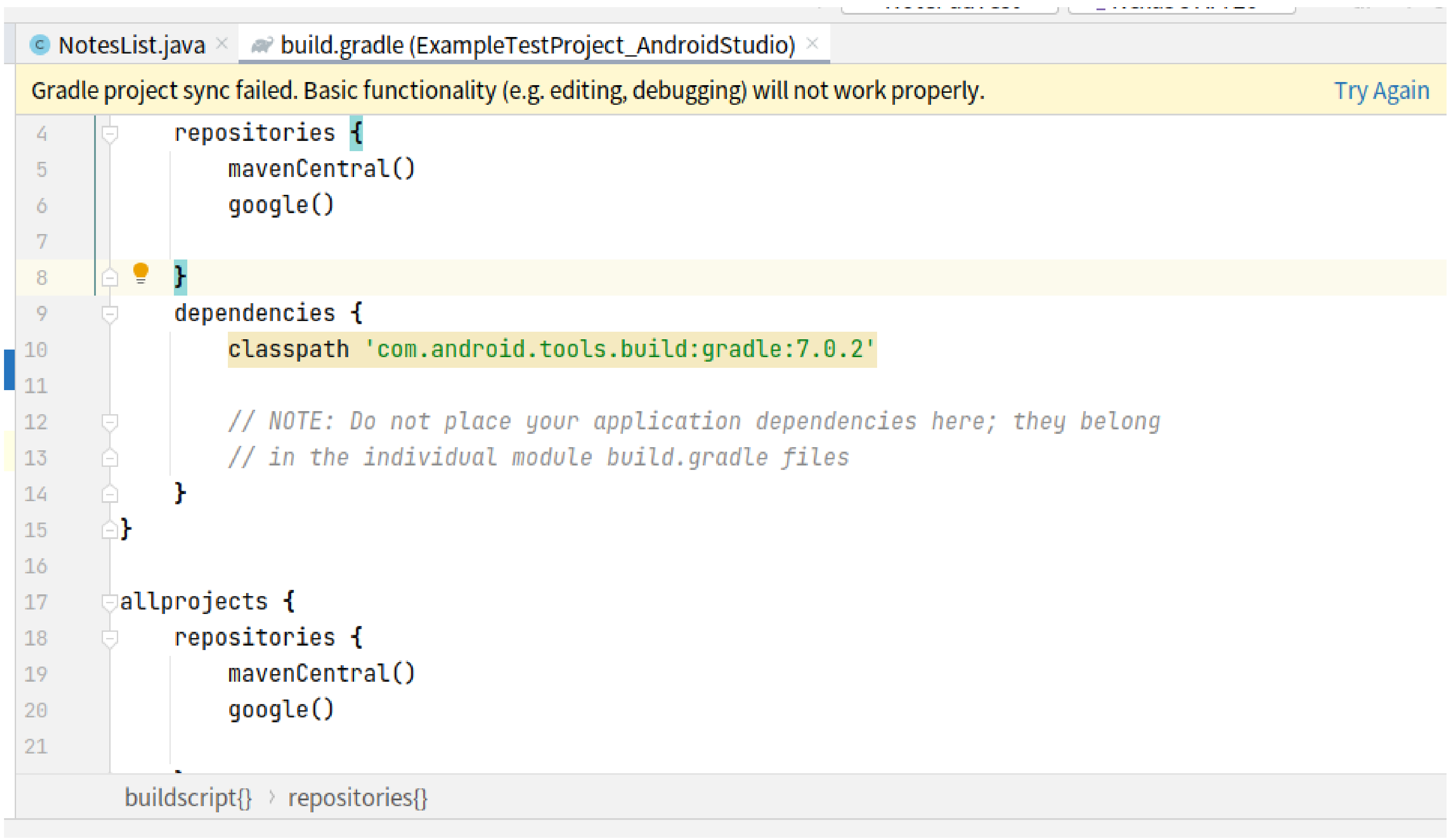
Drag and drop a file into the space above to quickly locate it in the tree



The path
'\Users\rreda\Library\Android\sdk'
does not belong to a directory.

Android Studio will use this Android SDK instead:
'C:\Users\user\AppData\Local\Android\Sdk'
and will modify the project's local.properties file.

Modify build.gradle

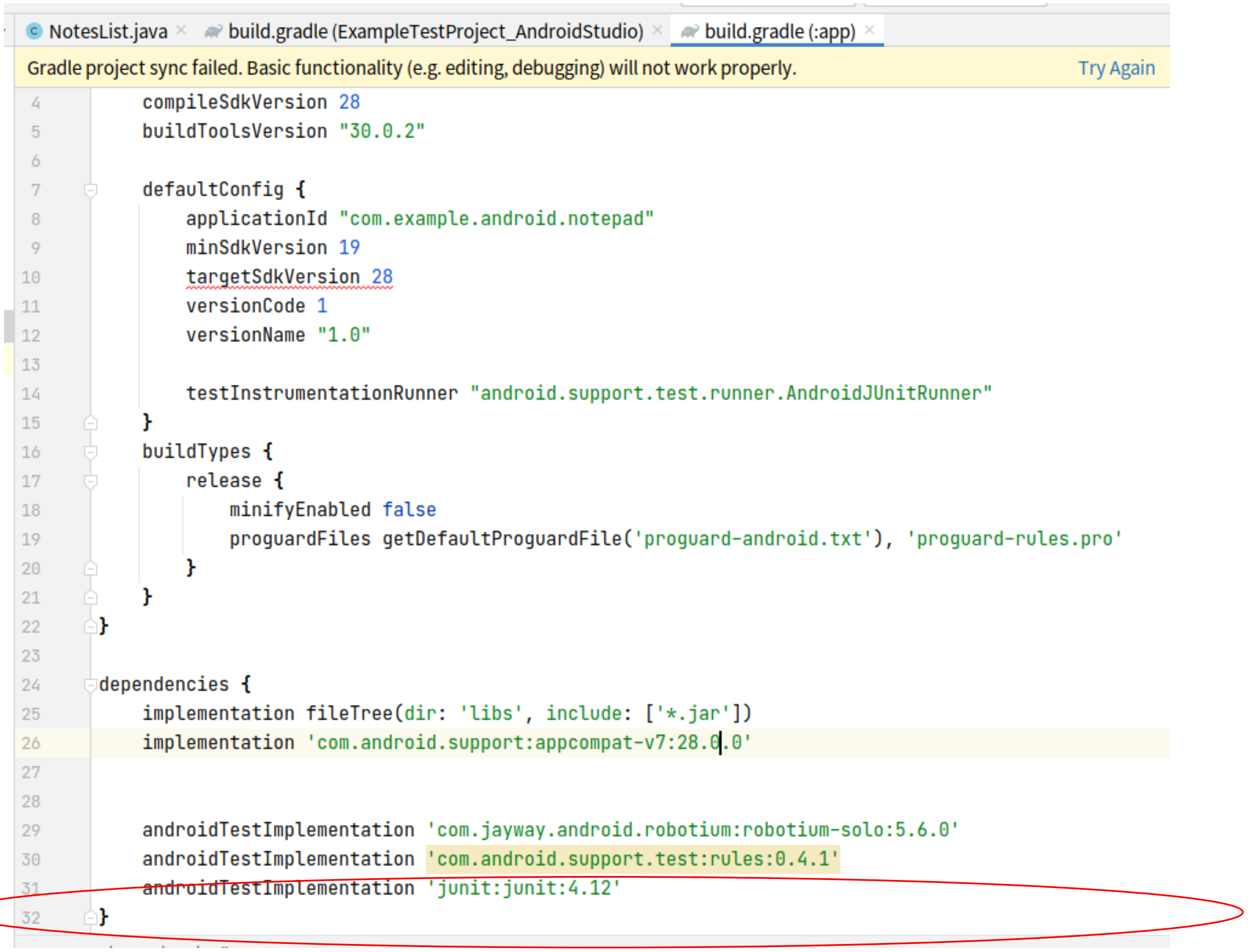

NotesList.java
build.gradle (ExampleTestProject_AndroidStudio)

Gradle project sync failed. Basic functionality (e.g. editing, debugging) will not work properly. Try Again

```
    repositories {
        mavenCentral()
        google()

    }
    dependencies {
        classpath 'com.android.tools.build:gradle:7.0.2'

        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        mavenCentral()
        google()

```

buildscript{} › repositories{}

Modify app/build.gradle

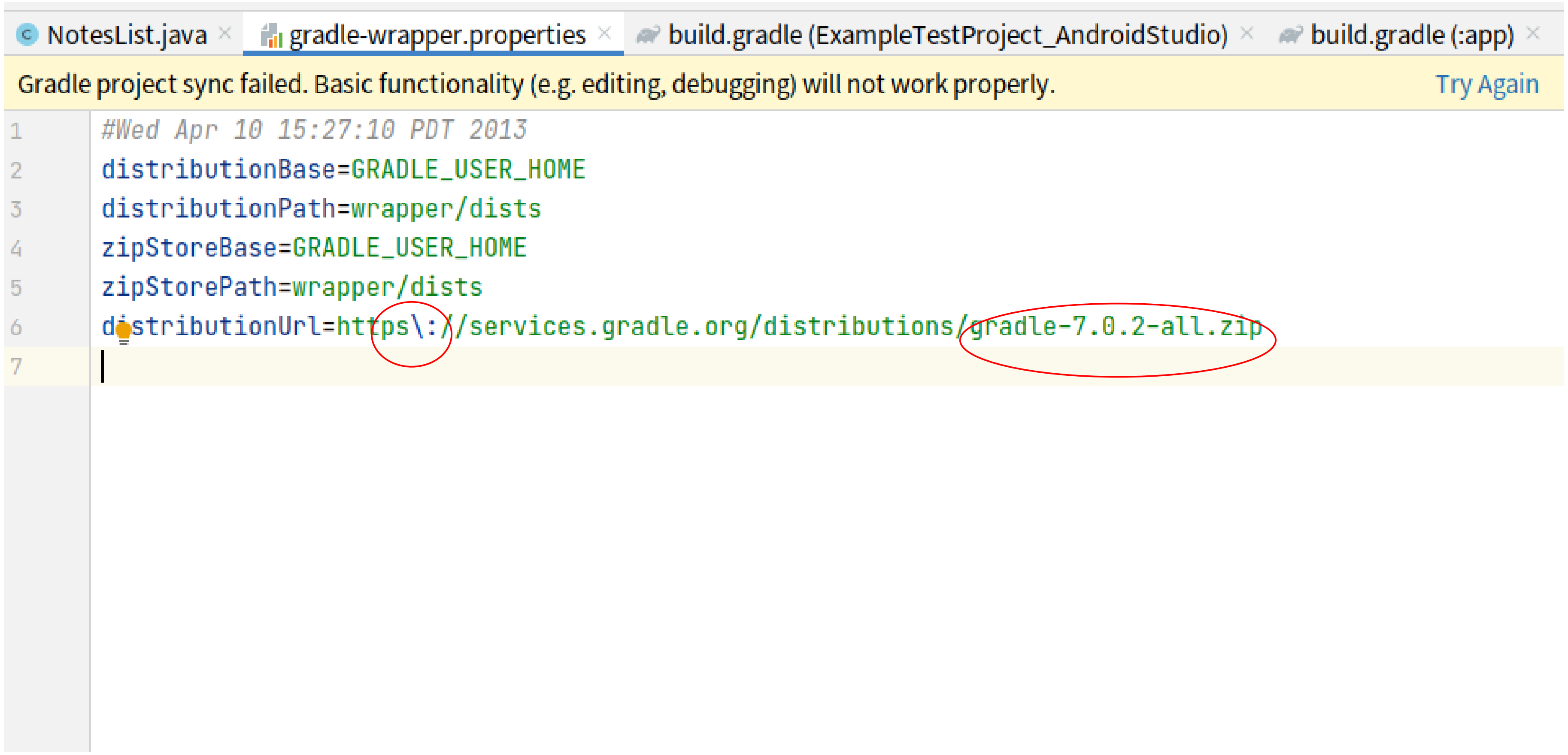


NotesList.java × build.gradle (ExampleTestProject_AndroidStudio) × build.gradle (:app) ×

Gradle project sync failed. Basic functionality (e.g. editing, debugging) will not work properly. Try Again

```
    compileSdkVersion 28
    buildToolsVersion "30.0.2"

    defaultConfig {
        applicationId "com.example.android.notepad"
        minSdkVersion 19
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'


    androidTestImplementation 'com.jayway.android.robotium:robotium-solo:5.6.0'
    androidTestImplementation 'com.android.support.test:rules:0.4.1'
    androidTestImplementation 'junit:junit:4.12'
}
```

Modify gradle-wrapper.properties



NotesList.java × | gradle-wrapper.properties × | build.gradle (ExampleTestProject_AndroidStudio) × | build.gradle (:app) ×

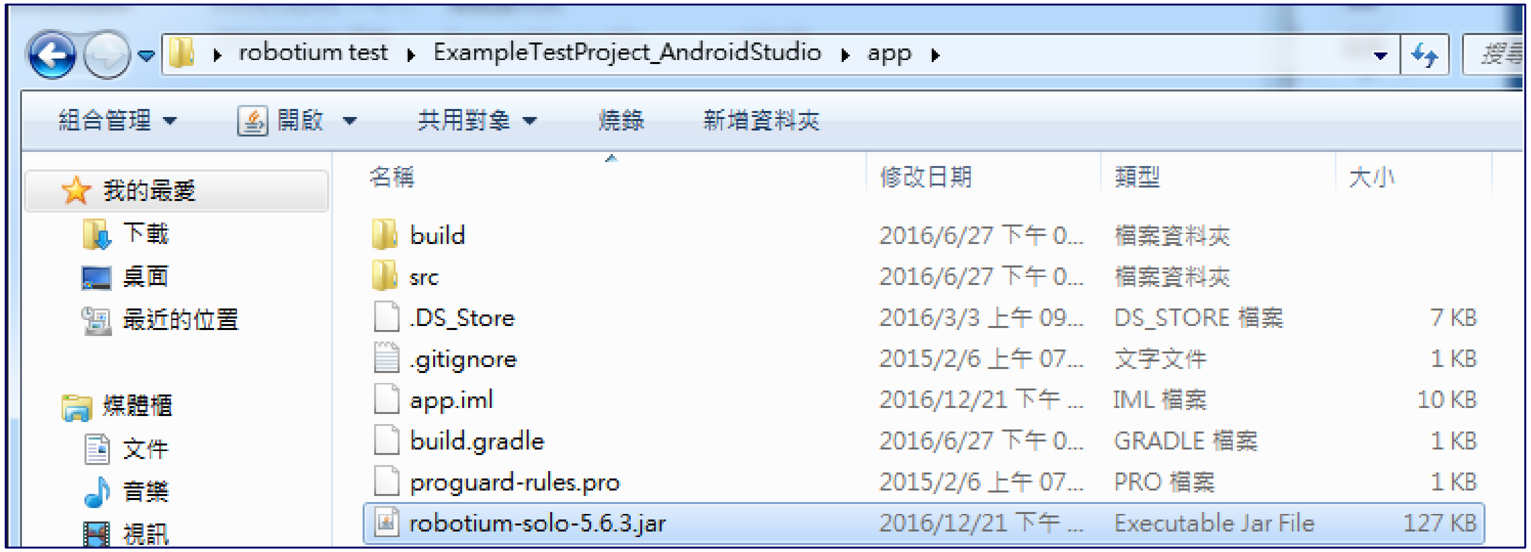
Gradle project sync failed. Basic functionality (e.g. editing, debugging) will not work properly. Try Again

```
#Wed Apr 10 15:27:10 PDT 2013
distributionBase=GRADLE_USER_HOME
distributionPath=wrapper/dists
zipStoreBase=GRADLE_USER_HOME
zipStorePath=wrapper/dists
distributionUrl=https\://services.gradle.org/distributions/gradle-7.0.2-all.zip
```

Import the JAR to Your Project

Copy the robotium-solo-5.6.3.jar

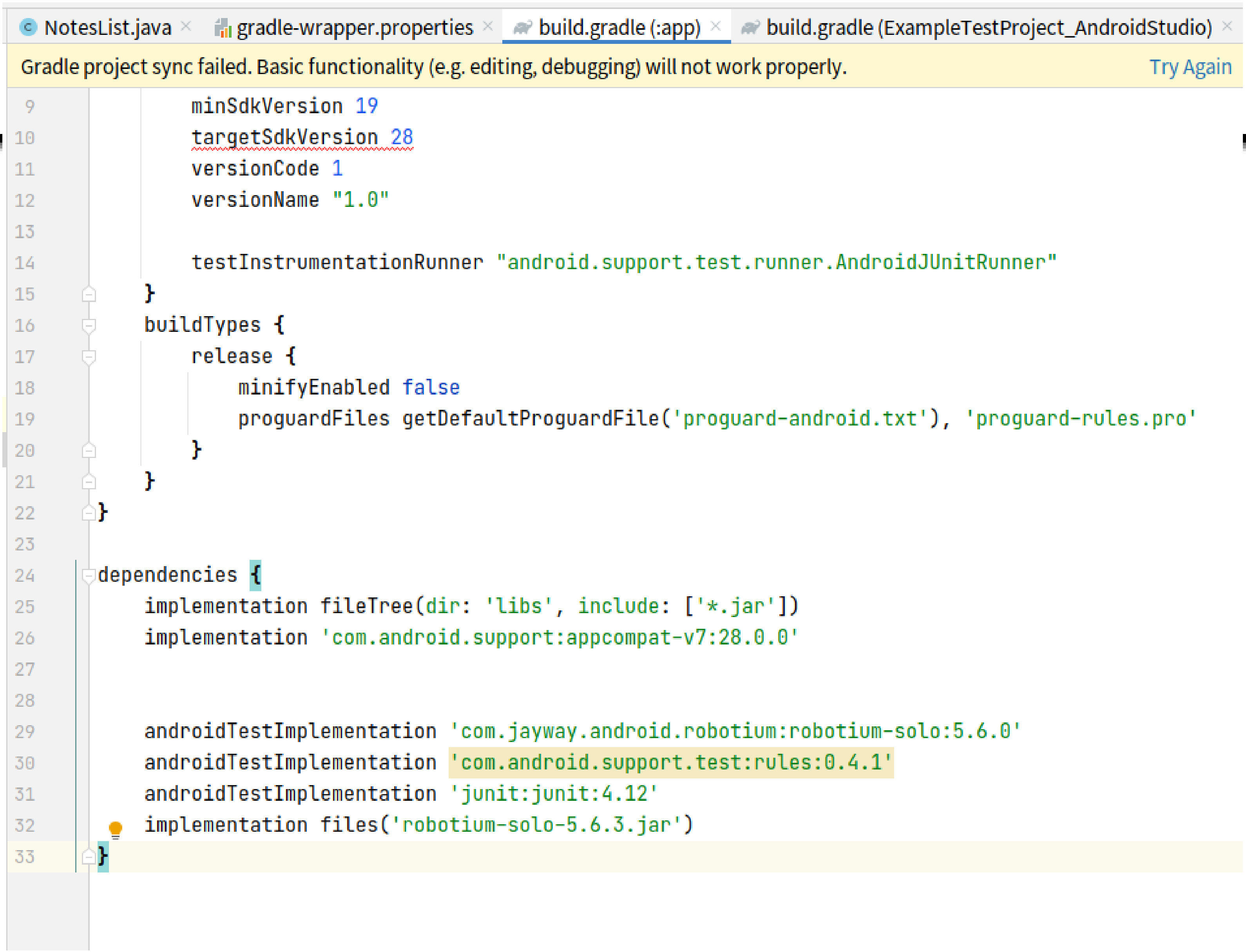
- Copy the robotium-solo-5.6.3.jar to [ProjectPath]/app



名稱	修改日期	類型	大小
build	2016/6/27 下午 0...	檔案資料夾	
src	2016/6/27 下午 0...	檔案資料夾	
.DS_Store	2016/3/3 上午 09...	DS_STORE 檔案	7 KB
.gitignore	2015/2/6 上午 07...	文字文件	1 KB
app.iml	2016/12/21 下午 ...	IML 檔案	10 KB
build.gradle	2016/6/27 下午 0...	GRADLE 檔案	1 KB
proguard-rules.pro	2015/2/6 上午 07...	PRO 檔案	1 KB
robotium-solo-5.6.3.jar	2016/12/21 下午 ...	Executable Jar File	127 KB

Modify app/build.gradle

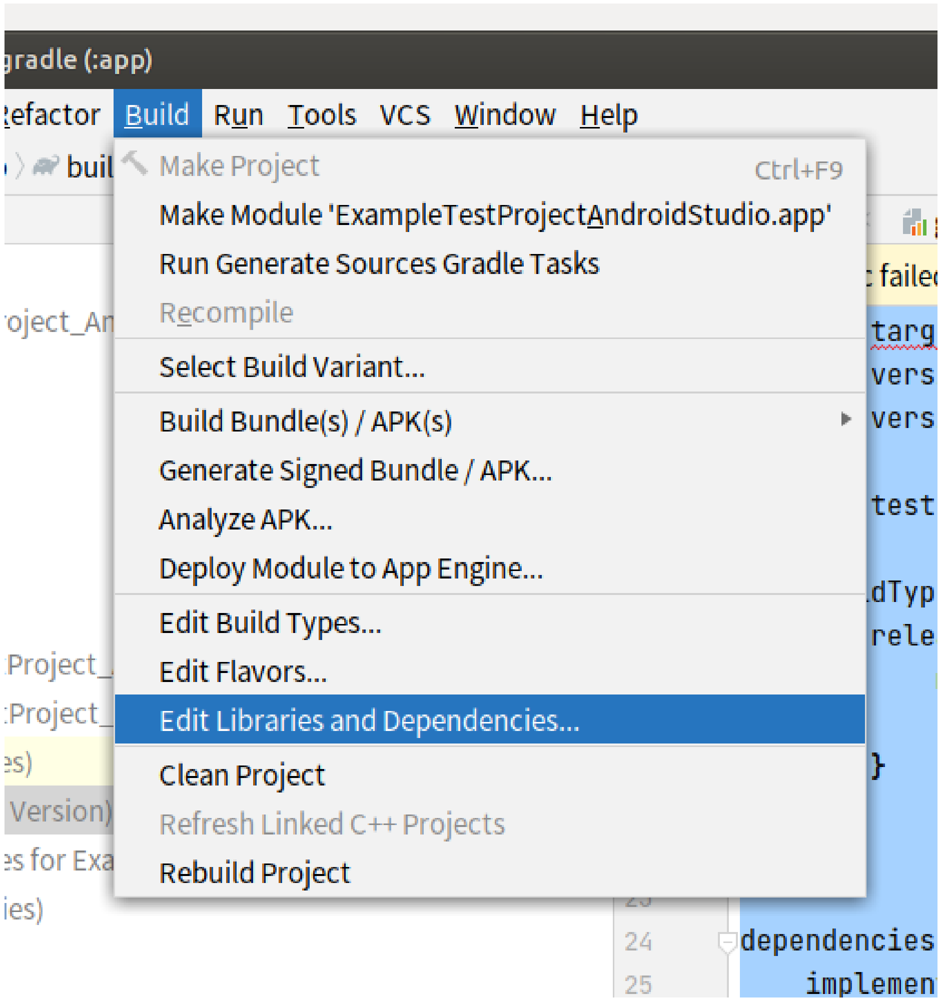
```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 28
    buildToolsVersion "30.0.2"

    defaultConfig {
        applicationId "com.example.android.notepad"
        minSdkVersion 19
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
                        implementation fileTree(dir: 'libs', include: ['*.jar'])
                        implementation 'com.android.support:appcompat-v7:28.0.0'


    androidTestImplementation 'com.jayway.android.robotium:robotium-solo:5.6.0'
    androidTestImplementation 'com.android.support.test:rules:0.4.1'
    androidTestImplementation 'junit:junit:4.12'
    implementation files('robotium-solo-5.6.3.jar')
}
```


NotesList.java
gradle-wrapper.properties
build.gradle (:app)
build.gradle (ExampleTestProject_AndroidStudio)

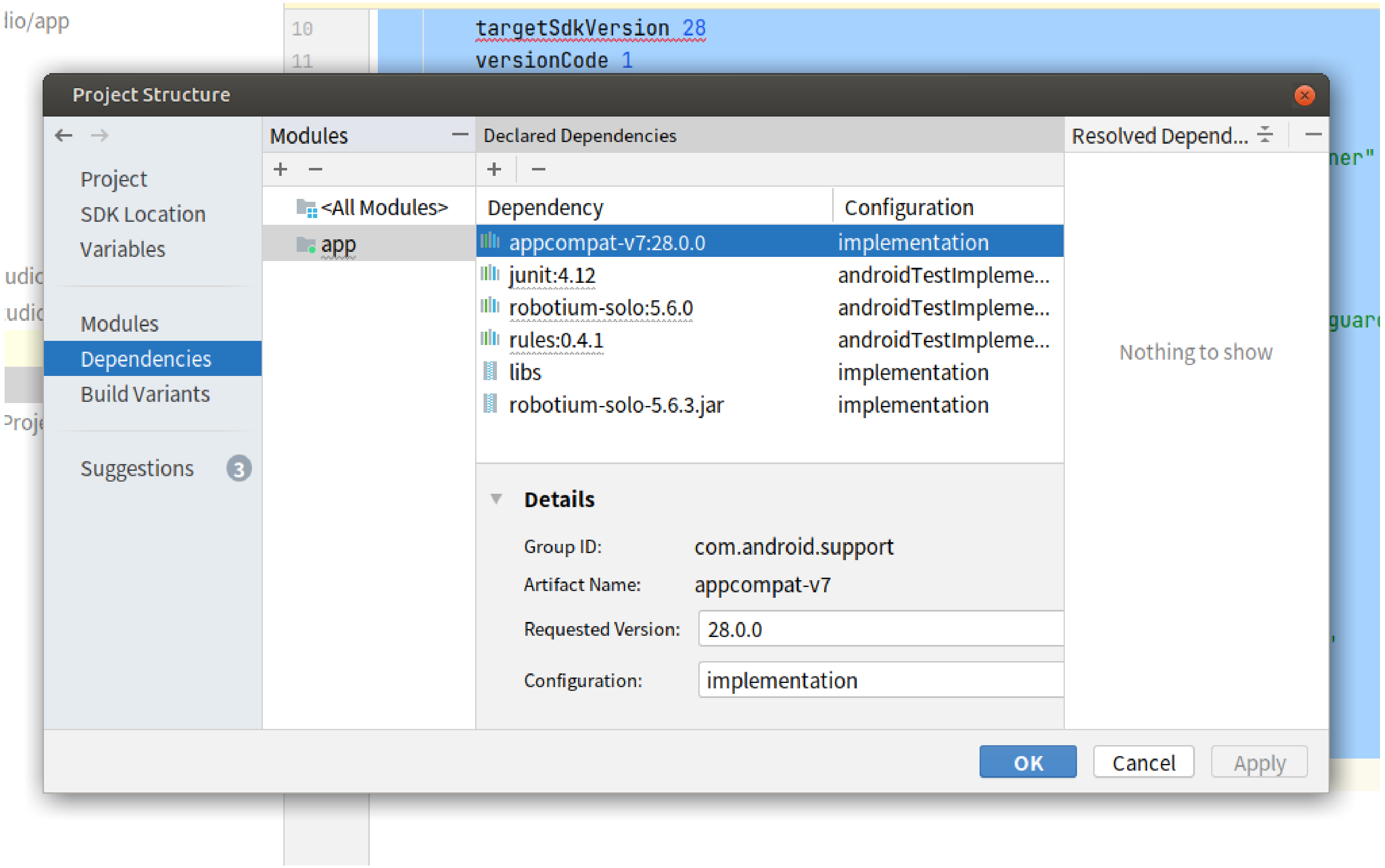
Gradle project sync failed. Basic functionality (e.g. editing, debugging) will not work properly. Try Again

```
        minSdkVersion 19
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'


    androidTestImplementation 'com.jayway.android.robotium:robotium-solo:5.6.0'
    androidTestImplementation 'com.android.support.test:rules:0.4.1'
    androidTestImplementation 'junit:junit:4.12'
    implementation files('robotium-solo-5.6.3.jar')
}
```

Setting Lib dependencies (1/3)


gradle (:app)
efactor Build Run Tools VCS Window Help
Make Project Ctrl+F9
Make Module 'ExampleTestProjectAndroidStudio.app'
Run Generate Sources Gradle Tasks
Recompile
Select Build Variant...
Build Bundle(s) / APK(s)
Generate Signed Bundle / APK...
Analyze APK...
Deploy Module to App Engine...
Edit Build Types...
Edit Flavors...
Edit Libraries and Dependencies...
Clean Project
Refresh Linked C++ Projects
Rebuild Project

Setting Lib dependencies (2/3)



```
targetSdkVersion 28
versionCode 1
```

Project Structure

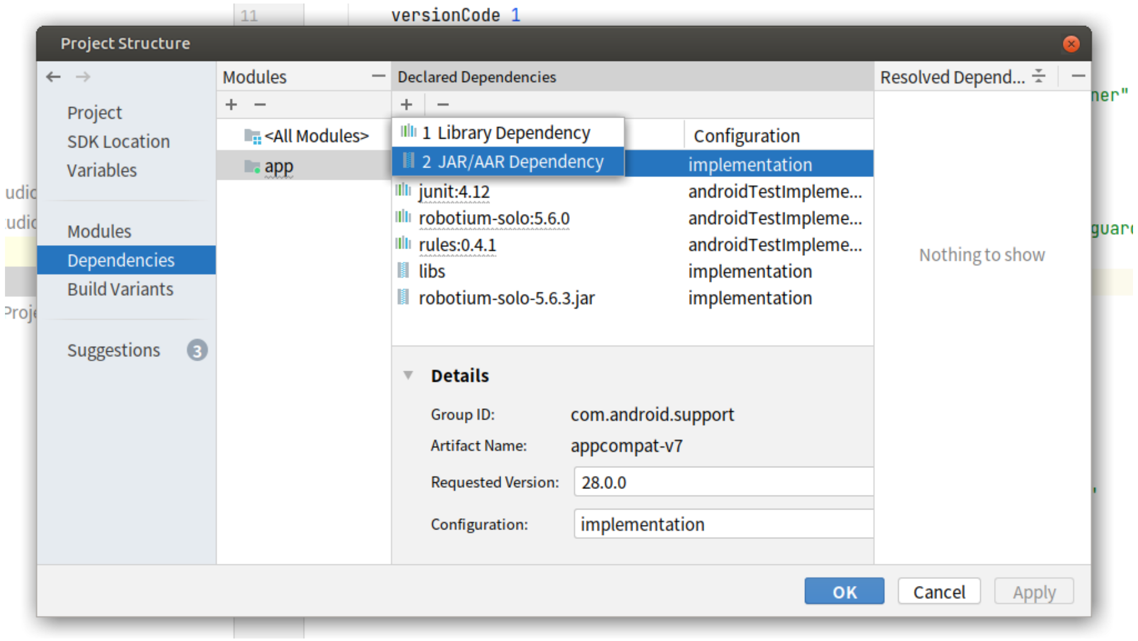
Dependency	Configuration
appcompat-v7:28.0.0	implementation
junit:4.12	androidTestImpleme...
robotium-solo:5.6.0	androidTestImpleme...
rules:0.4.1	androidTestImpleme...
libs	implementation
robotium-solo-5.6.3.jar	implementation

Details

Group ID:	com.android.support
Artifact Name:	appcompat-v7
Requested Version:	28.0.0
Configuration:	implementation

Setting Lib dependencies (3/3)

- If there is no robotium-solo-5.6.3.jar
  - Click”add”
  - Click “Jar dependency”
  - Add robotium-solo-5.6.3.jar

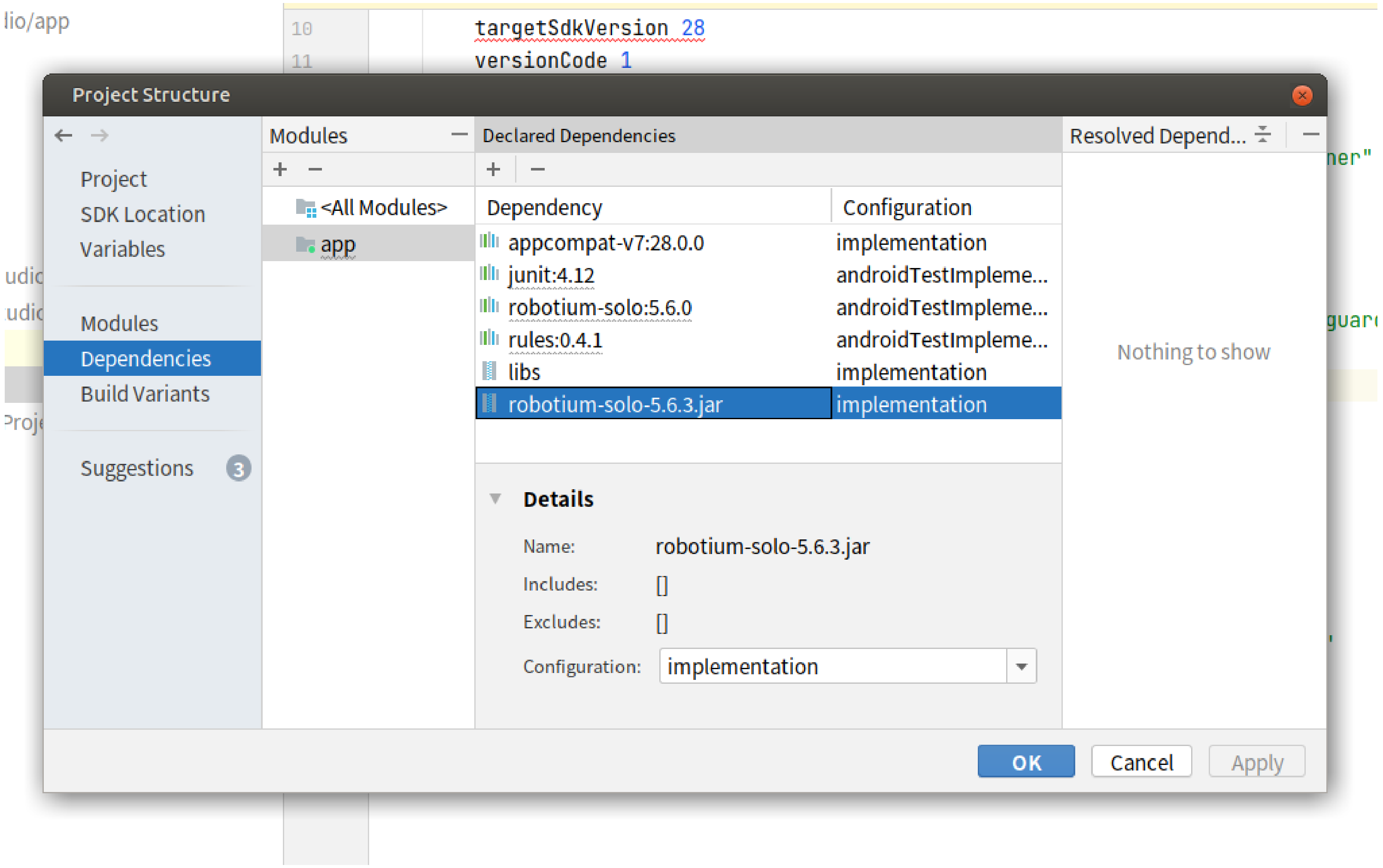


	Configuration
junit:4.12	androidTestImpleme...
robotium-solo:5.6.0	androidTestImpleme...
rules:0.4.1	androidTestImpleme...
libs	implementation
robotium-solo-5.6.3.jar	implementation

Details

Group ID:	com.android.support
Artifact Name:	appcompat-v7
Requested Version:	28.0.0
Configuration:	implementation

Setting Lib dependencies



```
targetSdkVersion 28
versionCode 1
```

Project Structure

- Project
- SDK Location
- Variables
- Modules
- Dependencies
- Build Variants
- Suggestions 3

Modules

- <All Modules>
- app

Declared Dependencies

Dependency	Configuration
appcompat-v7:28.0.0	implementation
junit:4.12	androidTestImpleme...
robotium-solo:5.6.0	androidTestImpleme...
rules:0.4.1	androidTestImpleme...
libs	implementation
robotium-solo-5.6.3.jar	implementation

Details

Name:	robotium-solo-5.6.3.jar
Includes:	[]
Excludes:	[]
Configuration:	implementation

Resolved Depend...

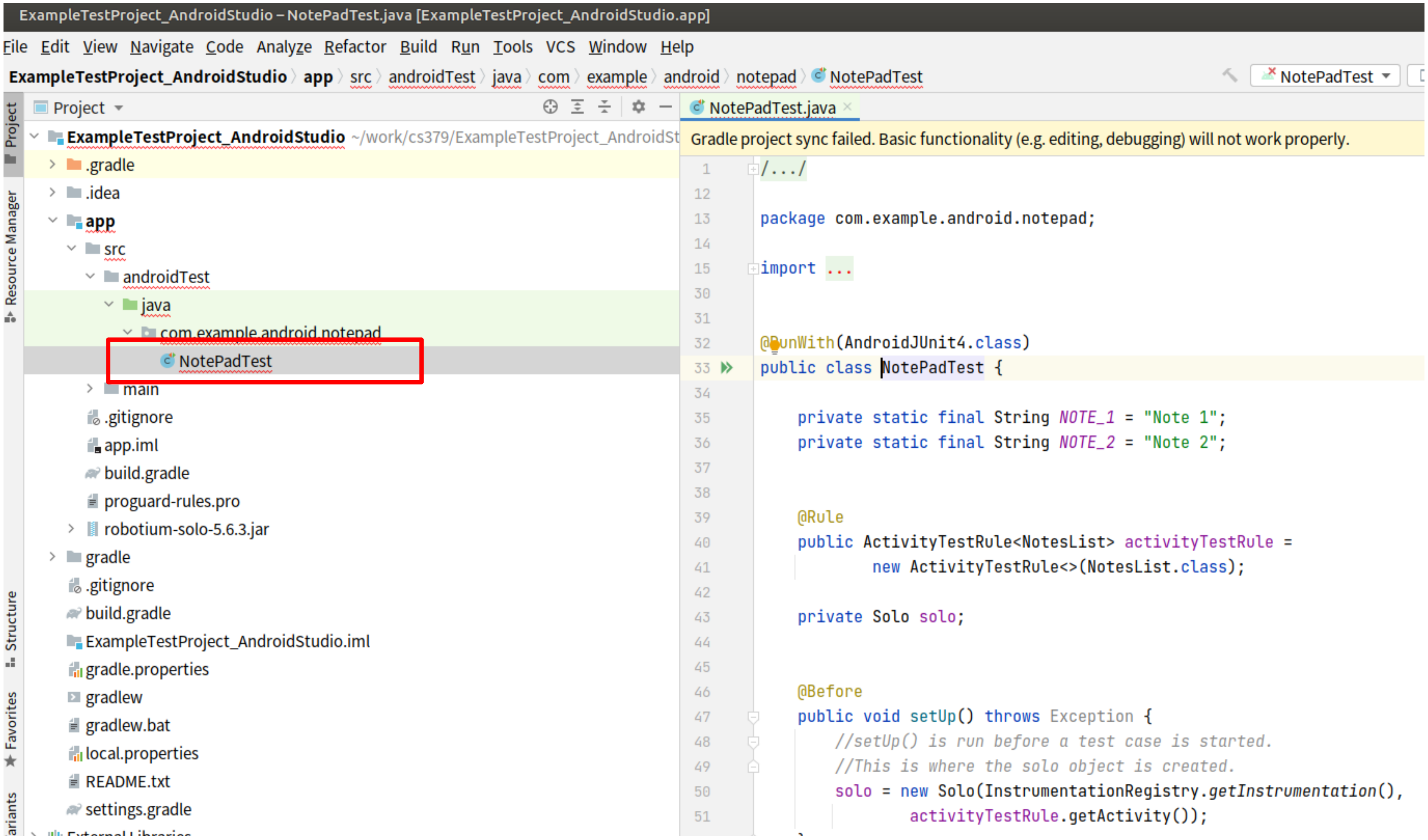
Nothing to show

Possible modification

```
…

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:23.0.1'


    androidTestImplementation 'com.jayway.android.robotium:robotium-solo:5.6.3'
    androidTestImplementation 'com.android.support.test:rules:1.0.2'
    androidTestImplementation 'junit:junit:4.13.1'
    implementation files('robotium-solo-5.6.3.jar')
    implementation 'com.android.support:support-annotations:28.0.0'
}
```

Start to Code your Robotium Project

Create the testing code

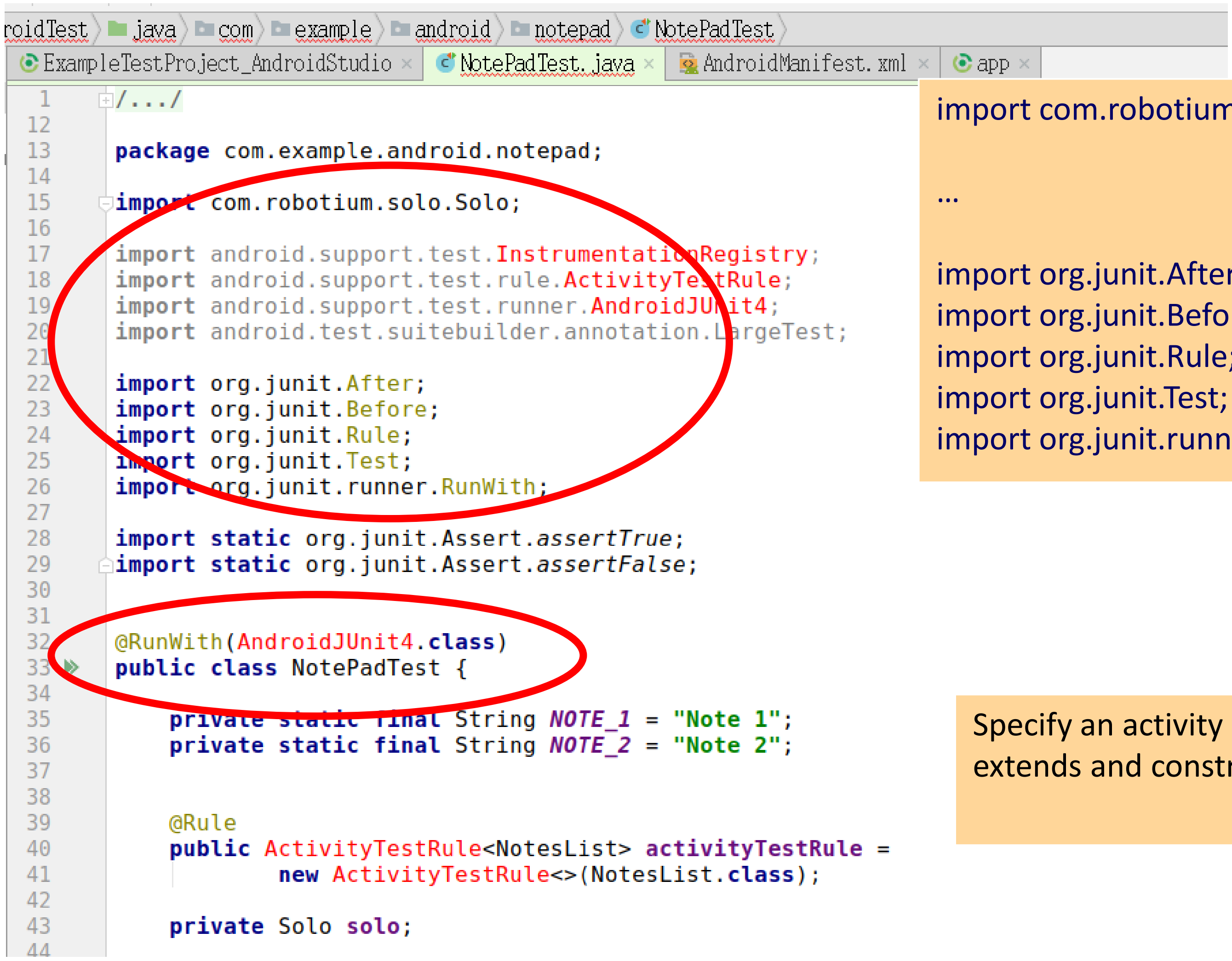

ExampleTestProject_AndroidStudio – NotePadTest.java [ExampleTestProject_AndroidStudio.app]

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
ExampleTestProject_AndroidStudio › app › src › androidTest › java › com › example › android › notepad › NotePadTest
Project
ExampleTestProject_AndroidStudio ~/work/cs379/ExampleTestProject_AndroidSt
.gradle
.idea
app
src
androidTest
java
com.example.android.notepad
NotePadTest
main
.gitignore
app.iml
build.gradle
proguard-rules.pro
robotium-solo-5.6.3.jar
gradle
.gitignore
build.gradle
ExampleTestProject_AndroidStudio.iml
gradle.properties
gradlew
gradlew.bat
local.properties
README.txt
settings.gradle
NotePadTest.java

Gradle project sync failed. Basic functionality (e.g. editing, debugging) will not work properly.

```java
/.../

package com.example.android.notepad;

import ...


@RunWith(AndroidJUnit4.class)
public class NotePadTest {

    private static final String NOTE_1 = "Note 1";
    private static final String NOTE_2 = "Note 2";


    @Rule
    public ActivityTestRule<NotesList> activityTestRule =
            new ActivityTestRule<>(NotesList.class);

    private Solo solo;


    @Before
    public void setUp() throws Exception {
        //setUp() is run before a test case is started.
        //This is where the solo object is created.
        solo = new Solo(InstrumentationRegistry.getInstrumentation(),
                activityTestRule.getActivity());
```

Start to code your Robotium project



roidTest › java › com › example › android › notepad › NotePadTest
ExampleTestProject_AndroidStudio × | NotePadTest.java × | AndroidManifest.xml × | app ×

```
/.../

package com.example.android.notepad;

import com.robotium.solo.Solo;

import android.support.test.InstrumentationRegistry;
import android.support.test.rule.ActivityTestRule;
import android.support.test.runner.AndroidJUnit4;
import android.test.suitebuilder.annotation.LargeTest;

import org.junit.After;
import org.junit.Before;
import org.junit.Rule;
import org.junit.Test;
import org.junit.runner.RunWith;

import static org.junit.Assert.assertTrue;
import static org.junit.Assert.assertFalse;


@RunWith(AndroidJUnit4.class)
public class NotePadTest {

    private static final String NOTE_1 = "Note 1";
    private static final String NOTE_2 = "Note 2";


    @Rule
    public ActivityTestRule<NotesList> activityTestRule =
            new ActivityTestRule<>(NotesList.class);

    private Solo solo;
```

import com.robotium.solo.Solo;

…

import org.junit.After;
import org.junit.Before;
import org.junit.Rule;
import org.junit.Test;
import org.junit.runner.RunWith;

Specify an activity which you want to test in extends and construction

Start to code your Robotium project

```
@RunWith(AndroidJUnit4.class)
public class NotePadTest {
    private static final String NOTE_1 = "Note 1";
    private static final String NOTE_2 = "Note 2";
    @Rule
    public ActivityTestRule<NotesList> activityTestRule =
            new ActivityTestRule<>(NotesList.class);
  private Solo solo;
  @Before
    public void setUp() throws Exception {
      …
    }
  @After
    public void tearDown() throws Exception {
      …
    }

    @Test
    public void testAddNote() throws Exception {
      …
    }
  @Test
    public void testEditNote() throws Exception {
        …
    }
}
```

Test environment setting

Test case

Test Environment Setting Example

```
@Before
public void setUp() throws Exception {
  solo = new Solo(getInstrumentation(), getActivity());
}

@After
public void tearDown() throws Exception {
  solo.finishOpenedActivities();
}
```

Test Case Example (1/3)

```
@Test
public void testAddNote() throws Exception {
  //Unlock the lock screen
  solo.unlockScreen();
  solo.clickOnMenuItem("Add note");
  //Assert that NoteEditor activity is opened
  solo.assertCurrentActivity("Expected NoteEditor activity", "NoteEditor");
  //In text field 0, enter Note 1
  solo.enterText(0, "Note 1");
  solo.goBack();
  //Clicks on menu item
  solo.clickOnMenuItem("Add note");
  //In text field 0, type Note 2
  solo.typeText(0, "Note 2");
  //Go back to first activity
  solo.goBack();
  //Takes a screenshot and saves it in "/sdcard/Robotium-Screenshots/".
  solo.takeScreenshot();
  boolean notesFound = solo.searchText("Note 1") && solo.searchText("Note 2");
  //Assert that Note 1 & Note 2 are found
  assertTrue("Note 1 and/or Note 2 are not found", notesFound);

}
```

Test Case Example (2/3)

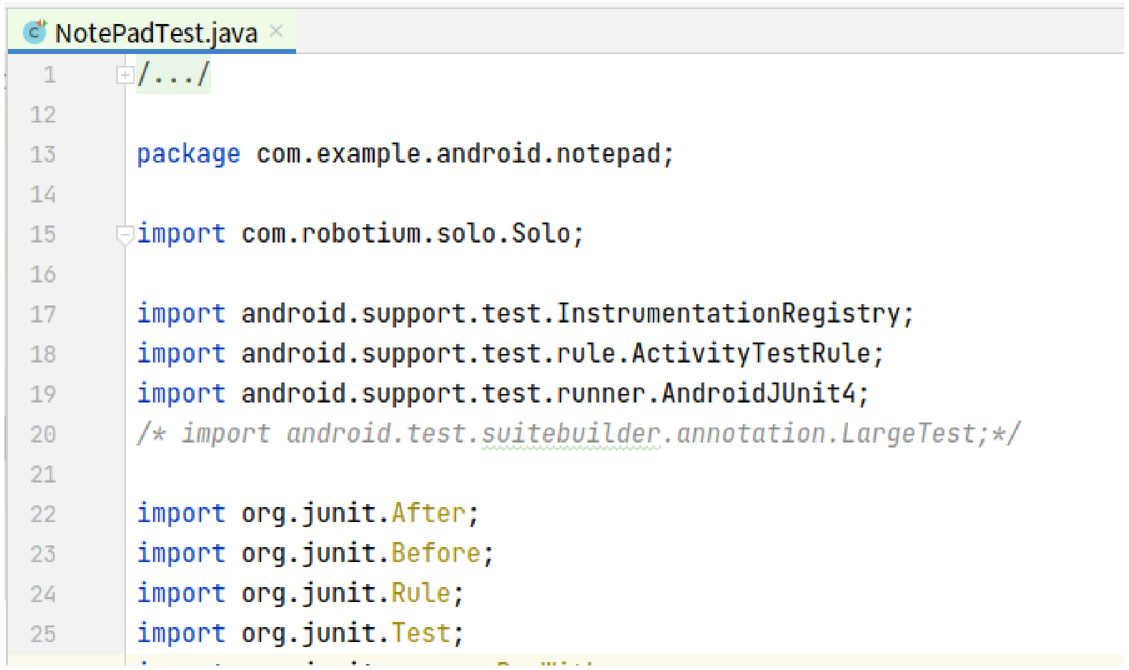
```
@Test
public void testEditNoteTitle() throws Exception {
//Click on add action menu item
      solo.clickOnView(solo.getView(com.example.android.notepad.R.id.menu_add));
      //In text field 0, enter Note 1
      solo.enterText(0, NOTE_1);
      //Press hard key back button
      solo.goBack();
      solo.clickOnText(NOTE_1);
      //Click on menu item "Edit title"
      solo.clickOnMenuItem("Edit title");
      //Clear the edit text field
      solo.clearEditText(1);
      //In the text field enter Note 2
      solo.enterText(1, NOTE_2);
      //Click on button "OK"
      solo.clickOnButton("OK");
      //Click on action menu item Save
      solo.clickOnView(solo.getView(com.example.android.notepad.R.id.menu_save));
      //Long click Note 2
      solo.clickLongOnText(NOTE_2);
      //Click on Delete
      solo.clickOnText("Delete");
      //Assert that Note 2 is deleted
      assertFalse("Note 2 is found", solo.searchText(NOTE_2));
}
```

Test Case Example (3/3)

```
@Test
public void deleteNotes() throws Exception {
//Click on first item in List
        solo.clickInList(1);
        //Click on delete action menu item
        solo.clickOnView(solo.getView(com.example.android.notepad.R.id.menu_delete));
        //Long click first item in List
        solo.clickLongInList(1);
        //Click delete
        solo.clickOnText(solo.getString(R.string.menu_delete));
}
```

Comment out some code

- Remove the following package
  - android.test.suitebuilder.annotation.LargeTest

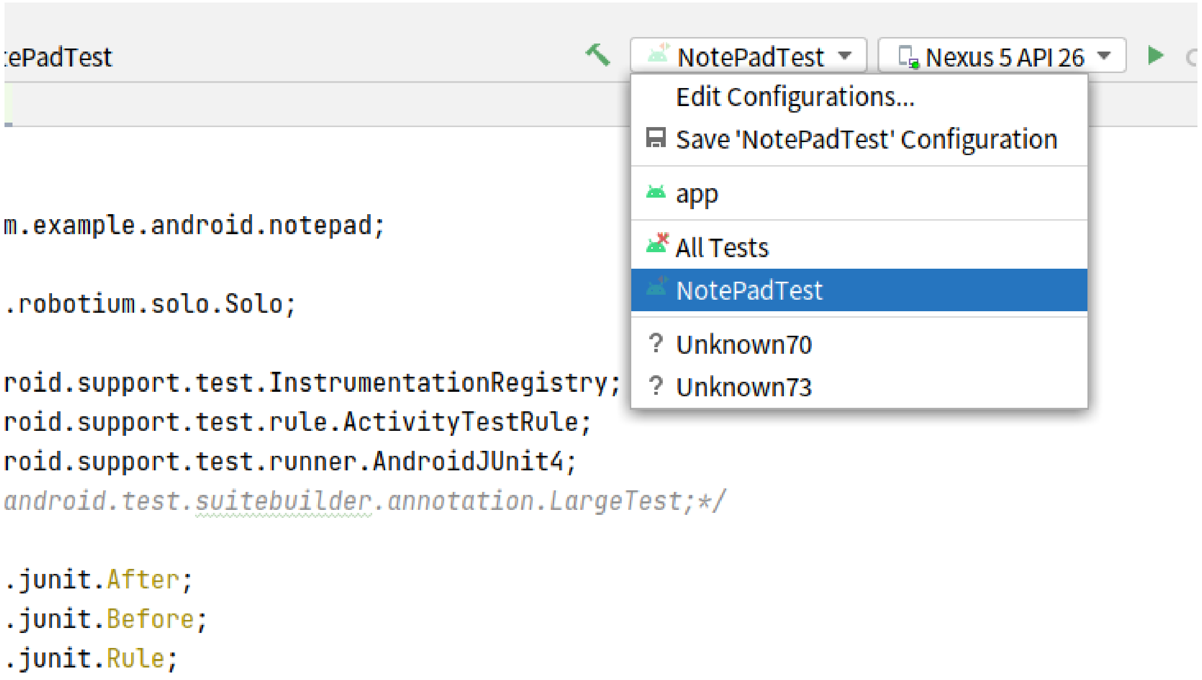


NotePadTest.java

```
/.../

package com.example.android.notepad;

import com.robotium.solo.Solo;

import android.support.test.InstrumentationRegistry;
import android.support.test.rule.ActivityTestRule;
import android.support.test.runner.AndroidJUnit4;
/* import android.test.suitebuilder.annotation.LargeTest;*/

import org.junit.After;
import org.junit.Before;
import org.junit.Rule;
import org.junit.Test;
```

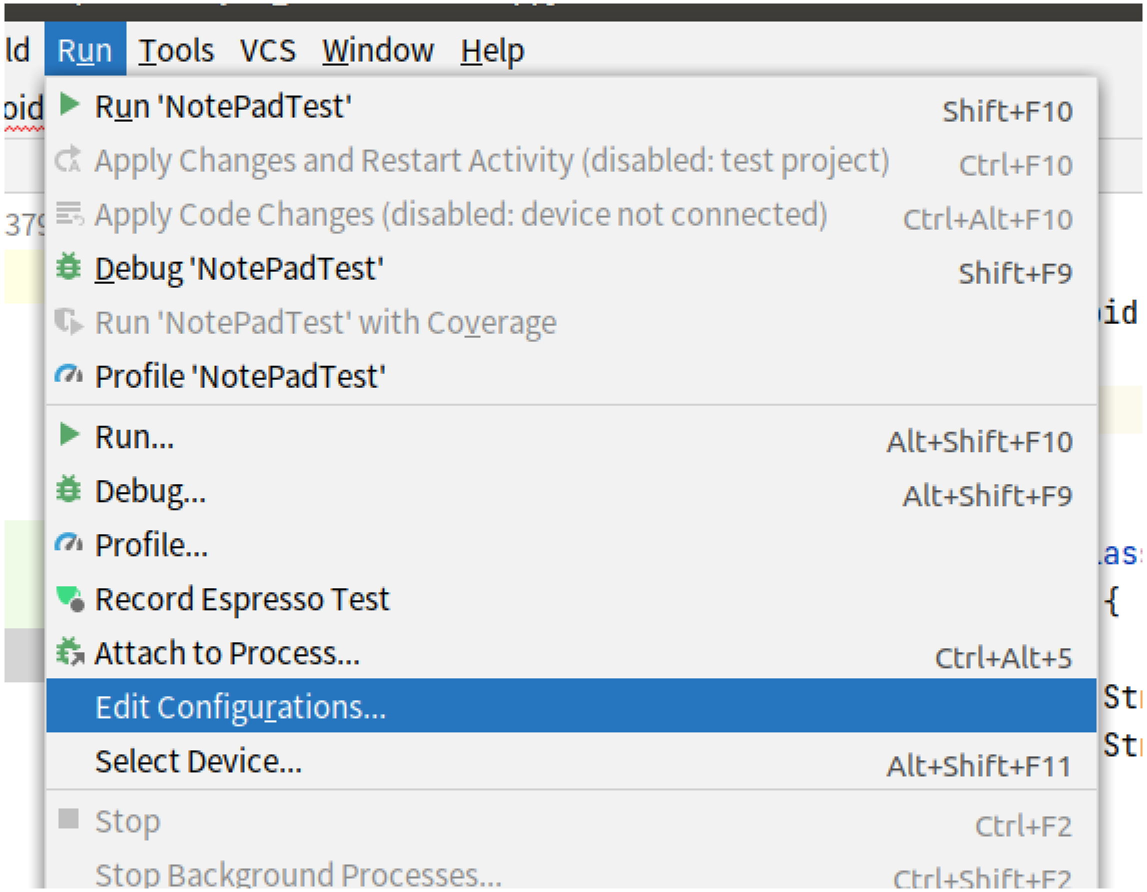
Run the testing code

- If NodePadTest is missing, add the Configuration



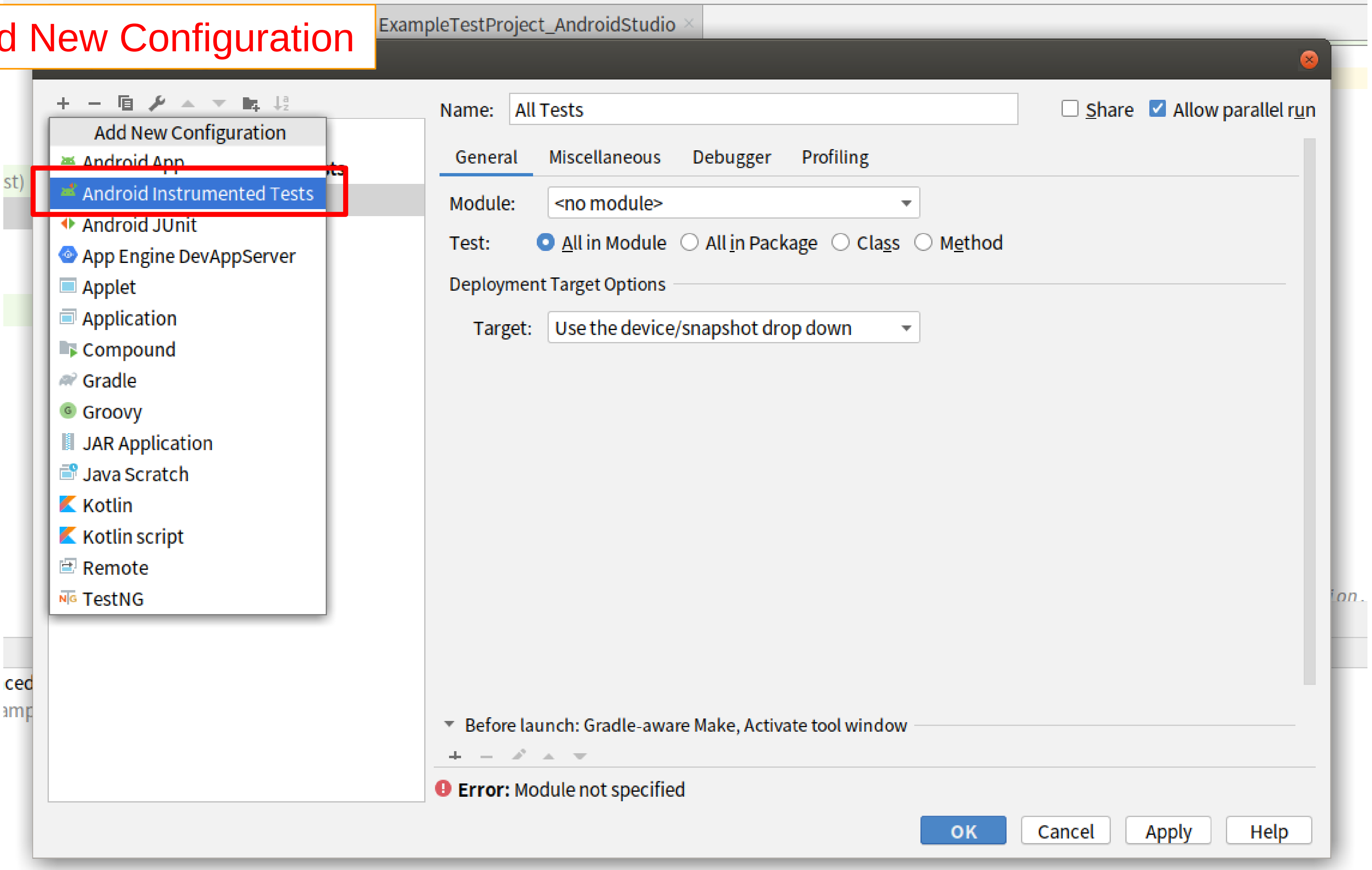
```
m.example.android.notepad;

.robotium.solo.Solo;

roid.support.test.InstrumentationRegistry;
roid.support.test.rule.ActivityTestRule;
roid.support.test.runner.AndroidJUnit4;
android.test.suitebuilder.annotation.LargeTest;*/

.junit.After;
.junit.Before;
.junit.Rule;
```

Run Test Configuration (1/3)


ld Run Tools VCS Window Help
Run 'NotePadTest' Shift+F10
Apply Changes and Restart Activity (disabled: test project) Ctrl+F10
Apply Code Changes (disabled: device not connected) Ctrl+Alt+F10
Debug 'NotePadTest' Shift+F9
Run 'NotePadTest' with Coverage
Profile 'NotePadTest'
Run... Alt+Shift+F10
Debug... Alt+Shift+F9
Profile...
Record Espresso Test
Attach to Process... Ctrl+Alt+5
Edit Configurations...
Select Device... Alt+Shift+F11
Stop Ctrl+F2
Stop Background Processes... Ctrl+Shift+F2

Run Test Configuration (2/3)

Add New Configuration


ExampleTestProject_AndroidStudio

Name: All Tests

[ ] Share [✓ Allow parallel run]

General | Miscellaneous | Debugger | Profiling

Module: <no module>

Test: [✓ All in Module] All in Package Class Method

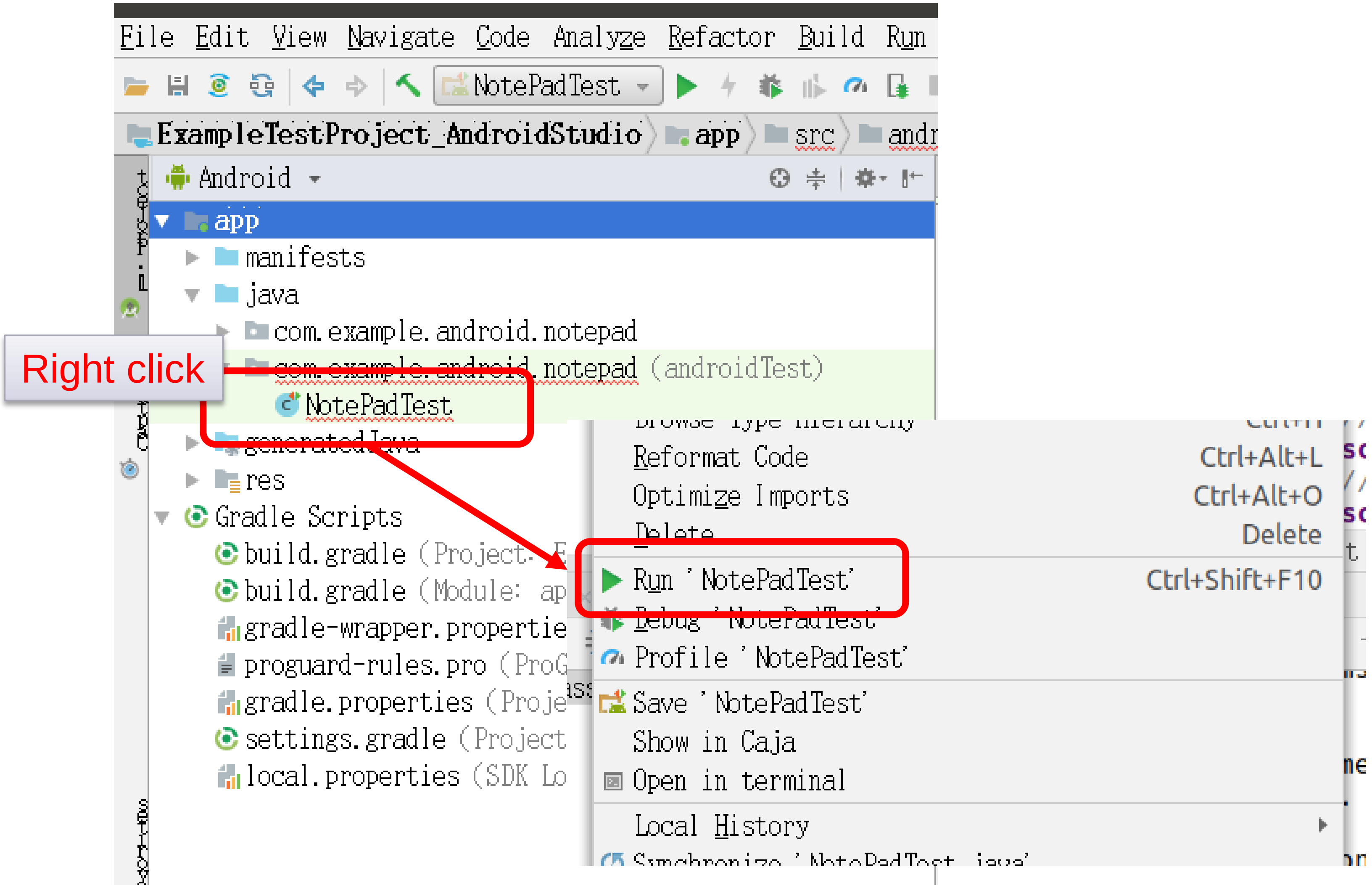
Deployment Target Options

Target: Use the device/snapshot drop down

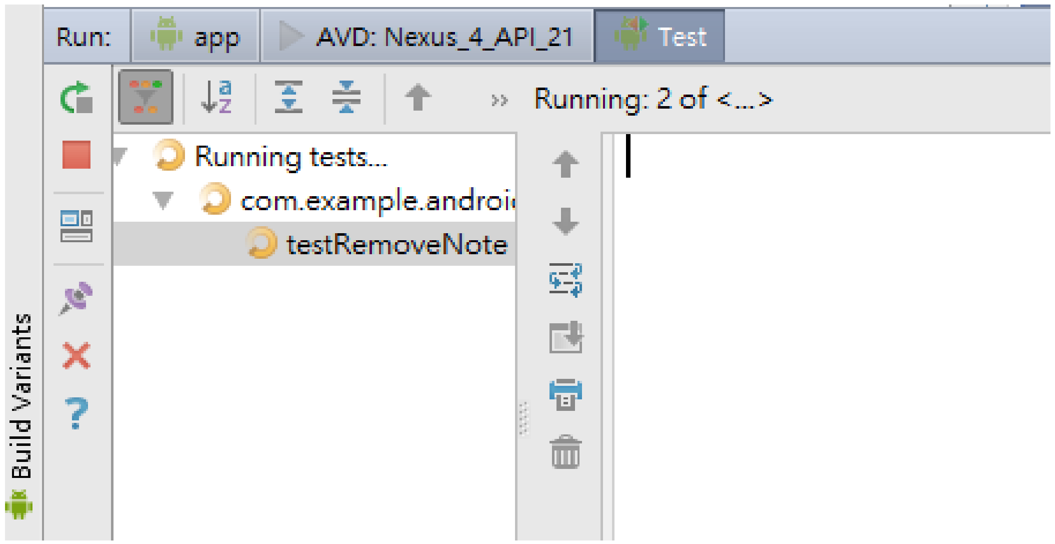
Before launch: Gradle-aware Make, Activate tool window

Error: Module not specified

Run Test Configuration (3/3)



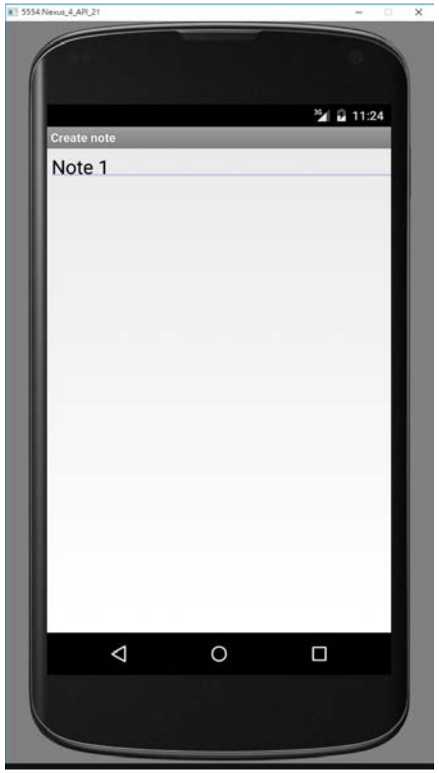
Testing



Run: app | AVD: Nexus_4_API_21 | Test

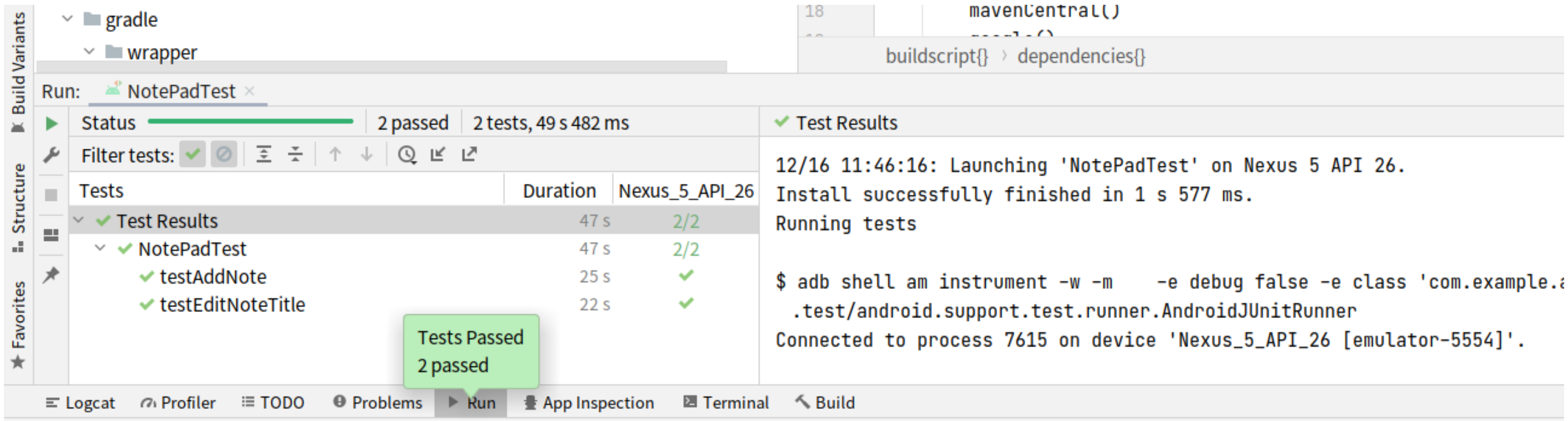
Running: 2 of <...>

```
Running tests...
  com.example.androi
    testRemoveNote
```



Testing Results

- All tests are passed!



Status 2 passed 2 tests, 49 s 482 ms

Filter tests:

Tests	Duration	Nexus_5_API_26
✓ Test Results	47 s	2/2
✓ NotePadTest	47 s	2/2
✓ testAddNote	25 s	✓
✓ testEditNoteTitle	22 s	✓

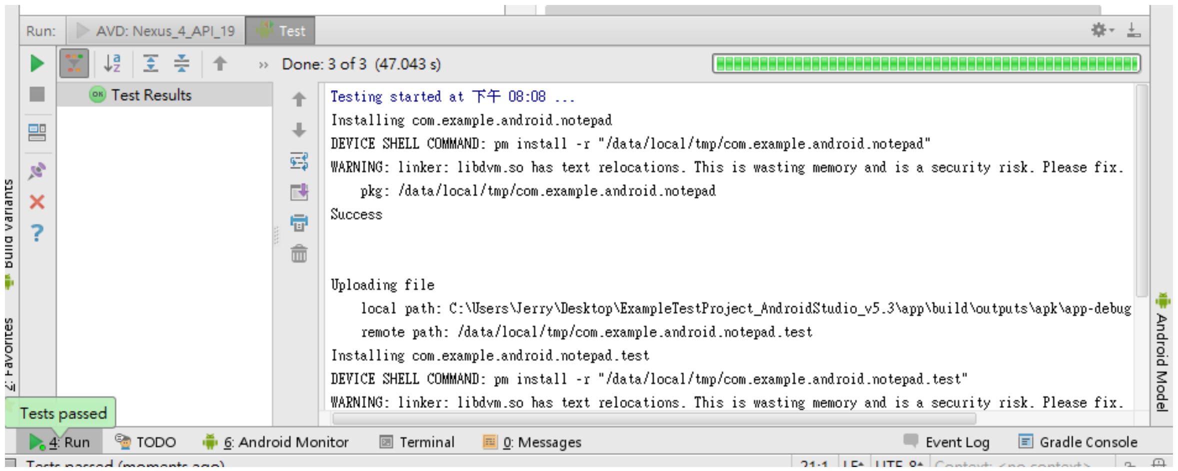
Tests Passed
2 passed

✓ Test Results

```
12/16 11:46:16: Launching 'NotePadTest' on Nexus 5 API 26.
Install successfully finished in 1 s 577 ms.
Running tests

$ adb shell am instrument -w -m    -e debug false -e class 'com.example.a
 .test/android.support.test.runner.AndroidJUnitRunner
Connected to process 7615 on device 'Nexus_5_API_26 [emulator-5554]'.
```

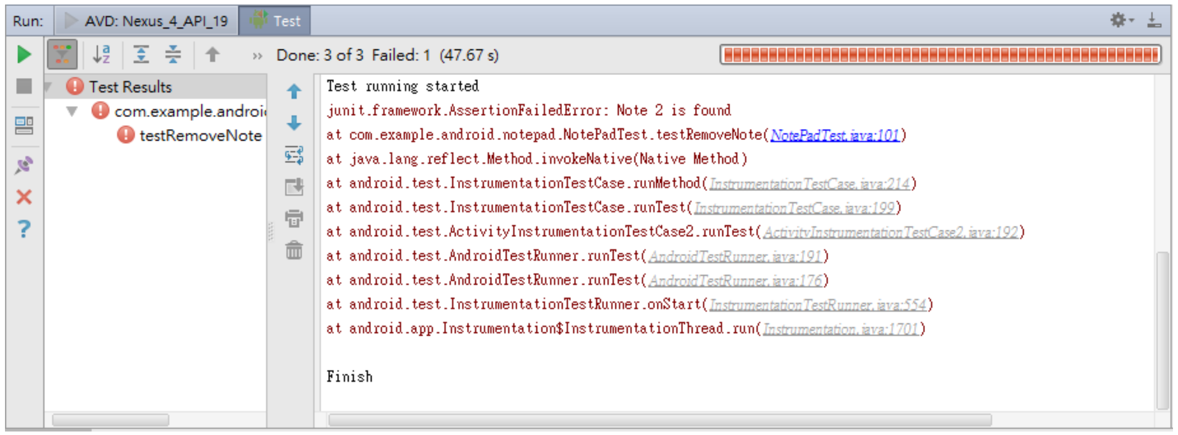
Testing Result

Pass



```
Testing started at 下午 08:08 ...
Installing com.example.android.notepad
DEVICE SHELL COMMAND: pm install -r "/data/local/tmp/com.example.android.notepad"
WARNING: linker: libdvm.so has text relocations. This is wasting memory and is a security risk. Please fix.
    pkg: /data/local/tmp/com.example.android.notepad
Success


Uploading file
    local path: C:\Users\Jerry\Desktop\ExampleTestProject_AndroidStudio_v5.3\app\build\outputs\apk\app-debug
    remote path: /data/local/tmp/com.example.android.notepad.test
Installing com.example.android.notepad.test
DEVICE SHELL COMMAND: pm install -r "/data/local/tmp/com.example.android.notepad.test"
WARNING: linker: libdvm.so has text relocations. This is wasting memory and is a security risk. Please fix.
```

Fail



```
Test running started
junit.framework.AssertionFailedError: Note 2 is found
at com.example.android.notepad.NotePadTest.testRemoveNote(NotePadTest.java:101)
at java.lang.reflect.Method.invokeNative(Native Method)
at android.test.InstrumentationTestCase.runMethod(InstrumentationTestCase.java:214)
at android.test.InstrumentationTestCase.runTest(InstrumentationTestCase.java:199)
at android.test.ActivityInstrumentationTestCase2.runTest(ActivityInstrumentationTestCase2.java:192)
at android.test.AndroidTestRunner.runTest(AndroidTestRunner.java:191)
at android.test.AndroidTestRunner.runTest(AndroidTestRunner.java:176)
at android.test.InstrumentationTestRunner.onStart(InstrumentationTestRunner.java:554)
at android.app.Instrumentation$InstrumentationThread.run(Instrumentation.java:1701)

Finish
```

Robotium API (5.4)

Method	Description
getView(int id)	Searches for the view with the specified ID in the current activity.
assertCurrentActivity(text, Activity.class)	Ensure that the current activity equals the second parameter.
getCurrentActivity() .getFragmentManager() .findFragmentById()	Searches for a fragment.
waitForText(text)	Waits for a text on the screen, default timeout 5 seconds.
clickOnButton(text)	Clicks on a button with the "text" text.
sendKey(Solo.MENU);	Sends the menu key event.
clickOnText(text)	Search for text in the current user interface and clicks on it.
enterText()	Enters a text.
searchText(text)	Searches for a text in the current user interface, return true if found.
searchButton(text)	Searches for a button with the text in the current user interface.
clickOnSearch()	Allows to click on part of the screen.
goBack()	Press the back button.
setDatePicker()	Sets the date in a DatePicker.
clickInList(x);	Click on item number x in a ListView
pressSpinnerItem(0,2);	Presses an item in a Spinner
isCheckBoxChecked()	Checks if the checkbox is checked.
takeScreenshot()	Saves a screenshot on the device in the /sdcard/Robotium-Screenshots/ folder. Requires the android.permission.WRITE_EXTERNAL_STORAGE permission in the AndroidManifest.xml of the application under test.
waitForActivity(SecondActivity.class, 2000)	Waits for the specified activity for 2 seconds

References

- https://github.com/robotiumtech/robotium
- https://github.com/RobotiumTech/robotium/wiki/Getting-Started
- http://www.vogella.com/tutorials/Robotium/article.html