

Desarrollo de aplicaciones móviles con Android



Tipos de pruebas

Android Studio está diseñada para simplificar las pruebas. Con solo algunos clics, puedes establecer una prueba JUnit que se ejecute en el JVM local o una prueba instrumental que se ejecute en un dispositivo. Por supuesto, también puedes extender tus capacidades de pruebas integrando frameworks de prueba como [Mockito](#), para probar llamadas de Android API en tus pruebas de unidades locales, y [Espresso](#) o [UI Automator](#) para ejercitar la interacción con usuarios en tus pruebas instrumentadas. Puedes generar pruebas Espresso automáticamente usando la [grabadora de pruebas Espresso](#).

Tipos de pruebas y ubicación

Pruebas de unidad local

Ubicación: `module-name/src/test/java/`.

Estas son pruebas que se ejecutan en la máquina virtual Java (JVM) local de tu máquina. Usa estas pruebas para minimizar el tiempo de ejecución cuando tus pruebas no tengan dependencias de marco de Android o cuando puedas simular las dependencias del marco de Android.

En el tiempo de ejecución, estas pruebas se ejecutan en una versión modificada de `android.jar` en la que se quitan todos los modificadores `final`. Esto te permite usar bibliotecas de simulación populares, como Mockito.

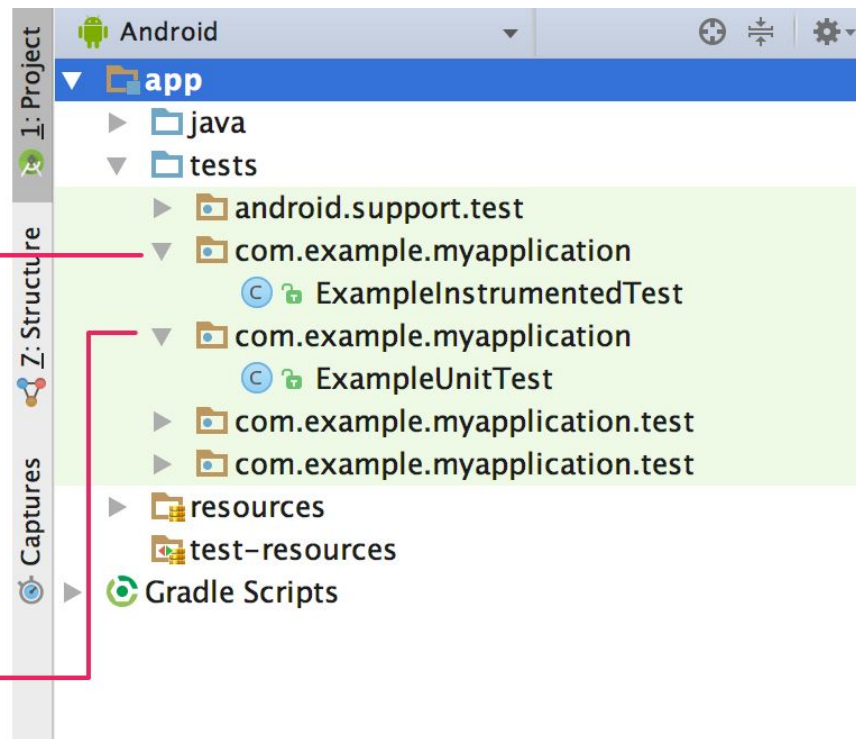
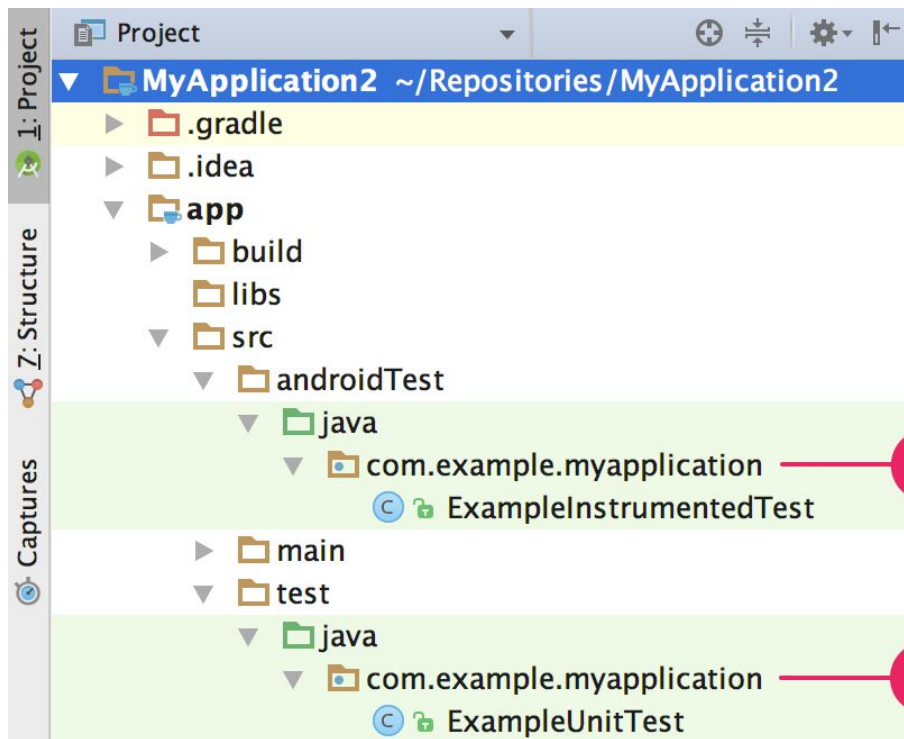
Tipos de pruebas y ubicación

Pruebas instrumentadas

Ubicación: **module-name**/src/androidTest/java/.

Estas son pruebas que se ejecutan en un dispositivo o emulador de hardware. Estas pruebas tienen acceso a las API [Instrumentation](#), y te permiten acceder a información como el [Context](#) de la app que pruebas y controlar la app a prueba desde tu código de prueba. Usa estas pruebas cuando escribas pruebas de IU funcionales e integradoras para automatizar la interacción de usuarios, o cuando tus pruebas tengan dependencias de Android que los objetos ficticios que no puedan contemplar.

Directorio para pruebas



Asegurar que estén las dependencias

También asegúrate de especificar las dependencias de bibliotecas de pruebas en el archivo `build.gradle` del módulo de tu app:

```
dependencies {  
    // Required for local unit tests (JUnit 4 framework)  
    testCompile 'junit:junit:4.12'  
  
    // Required for instrumented tests  
    androidTestCompile 'com.android.support:support-annotations:24.0.0'  
    androidTestCompile 'com.android.support.test:runner:0.5'  
}
```

onView(**Matcher**)
 .perform(**ViewAction**)
 .check(**ViewAssertion**)



CHEAT SHEET
espresso 2.0

Matchers

USER PROPERTIES

```
withId()
withText()
withTagKey()
withTagValue()
hasContentDescription()
withContentDescription()
withHint()
withSpinnerText()
hasLinks()
hasEllipsizedText()
hasMultilineText()
```

UI PROPERTIES

```
isDisplayed()
isCompletelyDisplayed()
isEnabled()
hasFocus()
isClickable()
isChecked()
isNotChecked()
withEffectiveVisibility()
isSelected()
```

COMMON HAMCREST MATCHERS

```
allOf(Matchers)
anyOf(Matchers)
is(...)
not(...)
endsWith(String)
startsWith(String)
```

HIERARCHY

```
withParent(Matcher)
withChild(Matcher)
hasDescendant(Matcher)
isDescendantOfA(Matcher)
hasSibling(Matcher)
isRoot()
```

INPUT

```
supportsInputMethods()
hasImeAction()
```

CLASS

```
isAssignableFrom(...)
withClassName()
```

ROOT MATCHERS

```
isFocusable()
isTouchable()
isDialog()
withDecorView()
isPlatformPopup()
```

SEE ALSO

Preference matchers
Cursor matchers

View Actions

CLICK/PRESS

```
click()
doubleClick()
longClick()
pressBack()
pressImeActionButton()
pressKey([int/EspressoKey])
pressMenuKey()
closeSoftKeyboard()
openLink()
```

GESTURES

```
scrollTo()
swipeLeft()
swipeRight()
swipeUp()
swipeDown()
```

TEXT

```
clearText()
typeText(String)
typeTextIntoFocusedView(String)
replaceText(String)
```

View Assertions

```
matches(Matcher)
doesNotExist()
selectedDescendantsMatch()
```

LAYOUT ASSERTIONS

```
noEllipsizedText(Matcher)
noMultilineButtons()
noOverlaps([Matcher])
```

POSITION ASSERTIONS

```
isLeftOf(Matcher)
isRightOf(Matcher)
isLeftAlignedWith(Matcher)
isRightAlignedWith(Matcher)
isAbove(Matcher)
isBelow(Matcher)
isBottomAlignedWith(Matcher)
isTopAlignedWith(Matcher)
```