

# Android UI Testing



왜!!!

테스트를 해야 하는가?

**1. Espresso**

**2. etc test framework**

# 1. Espresso

<https://developer.android.com/training/testing/ui-testing/espresso-testing.html>

# 1. Espresso

Sample Code

[https://github.  
com/googlesamples/android-  
testing](https://github.com/googlesamples/android-testing)

# 1. Espresso Gradle Setting

```
dependencies {  
    compile fileTree(include: ['*.jar'], dir: 'libs')  
    compile 'com.android.support:appcompat-v7:22.2.1'  
    compile 'com.jakewharton:butterknife:7.0.1'  
  
    //Support Annotations  
    androidTestCompile 'com.android.support:support-annotations:22.2.1'  
  
    //Espresso  
    androidTestCompile 'com.android.support.test:runner:0.3'  
    androidTestCompile 'com.android.support.test:rules:0.3'  
    androidTestCompile 'com.android.support.test.espresso:espresso-core:2.2'  
  
    //UiAutomator  
    androidTestCompile 'com.android.support.test.uiautomator:uiautomator-v18:2.1.1'  
  
    // add this for intent mocking support / Intent Mocking  
    androidTestCompile 'com.android.support.test.espresso:espresso-intents:2.2'  
    // add this for webview testing support / WebView Mocking  
    androidTestCompile 'com.android.support.test.espresso:espresso-web:2.2'  
}
```

# 1. Espresso Gradle Setting

```
defaultConfig {  
    applicationId "backgomeproject.espressosample"  
    minSdkVersion 18  
    targetSdkVersion 22  
    versionCode 1  
    versionName "1.0"  
  
    //테스트 runner 설정  
    testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"  
}  
  
//라이선스 키  
packagingOptions {  
    exclude 'LICENSE.txt'  
}
```





```
onView(ViewMatcher)  
    .perform(ViewAction)  
    .check(ViewAssertion);
```

```
onData(ObjectMatcher)  
    .DataOptions  
    .perform(ViewAction)  
    .check(ViewAssertion);
```

## View Matchers

### USER PROPERTIES

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

### UI PROPERTIES

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

### OBJECT MATCHER

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

### 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  
Layout matchers
```

## Data Options

```
inAdapterView(Matcher)  
atPosition(Integer)  
onChildView(Matcher)
```

## 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)
```

**intended(IntentMatcher);**

**intending(IntentMatcher)**  
**.respondWith(ActivityResult);**

## *Intent Matchers*

### **INTENT**

- hasAction(...)
- hasCategories(...)
- hasData(...)
- hasComponent(...)
- hasExtra(...)
- hasExtras(Matcher)
- hasExtraWithKey(...)
- hasType(...)
- hasPackage()
- toPackage(String)
- hasFlag(int)
- hasFlags(...)
- isInternal()

### **URI**

- hasHost(...)
- hasParamWithName(...)
- hasPath(...)
- hasParamWithValue(...)
- hasScheme(...)
- hasSchemeSpecificPart(...)

### **BUNDLE**

- hasEntry(...)
- hasKey(...)
- hasValue(...)

### **COMPONENT NAME**

- hasClassName(...)
- hasPackageName(...)
- hasShortClassName(...)
- hasMyPackageName()

## 2. Selendroid

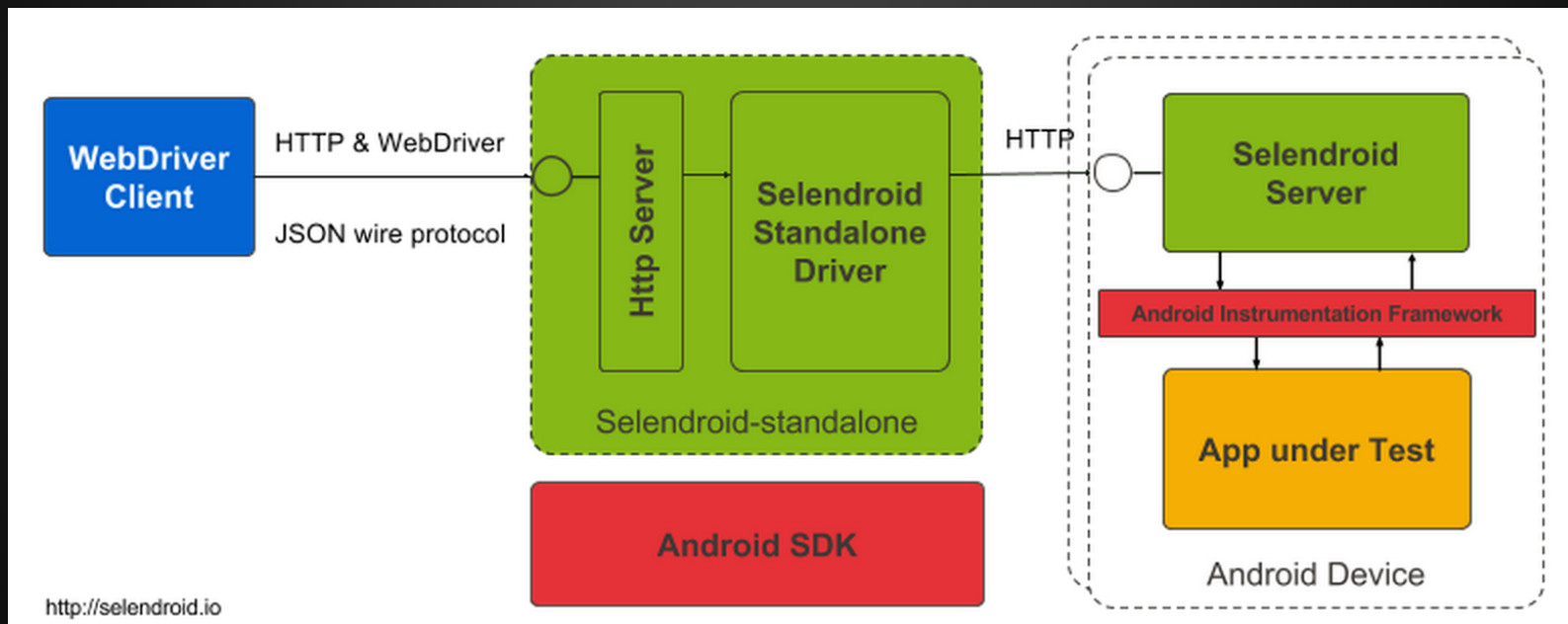
<http://selendroid.io/>

<http://appium.io/>

## 2. Selendroid

- 단위 테스트
- 인수 테스트
- Cross Platform 지원 (iOS, Android)
- Emulator 지원
- 비용 Free
- Script Recoding (Inspacter)
- WebView 테스트 지원
- HTTP 프로토콜 이용

## 2. Selendroid



# Top5 Android Testing Framework

	Robotium	uiautomator	Espresso	Appium	Calabash
Android	Yes	Yes	Yes	Yes	Yes
iOS	No	No	No	Yes	Yes
Mobile web	Yes (Android)	Limited to x.y clicks	No	Yes (Android & iOS)	Yes (Android)
Scripting Language	Java	Java	Java	Almost any	Ruby
Test creation tools	Testdroid Recorder	UI Automator viewer	Hierarchy Viewer	Appium.app	CLI
Supported API levels	All	16 =>	8, 10, 15 =>	All	All
Community	Contributors	Google	Google	Active	Pretty quiet

# 마지막으로 Android Unit Testing

<http://robolectric.org/>

<http://junit.org/>