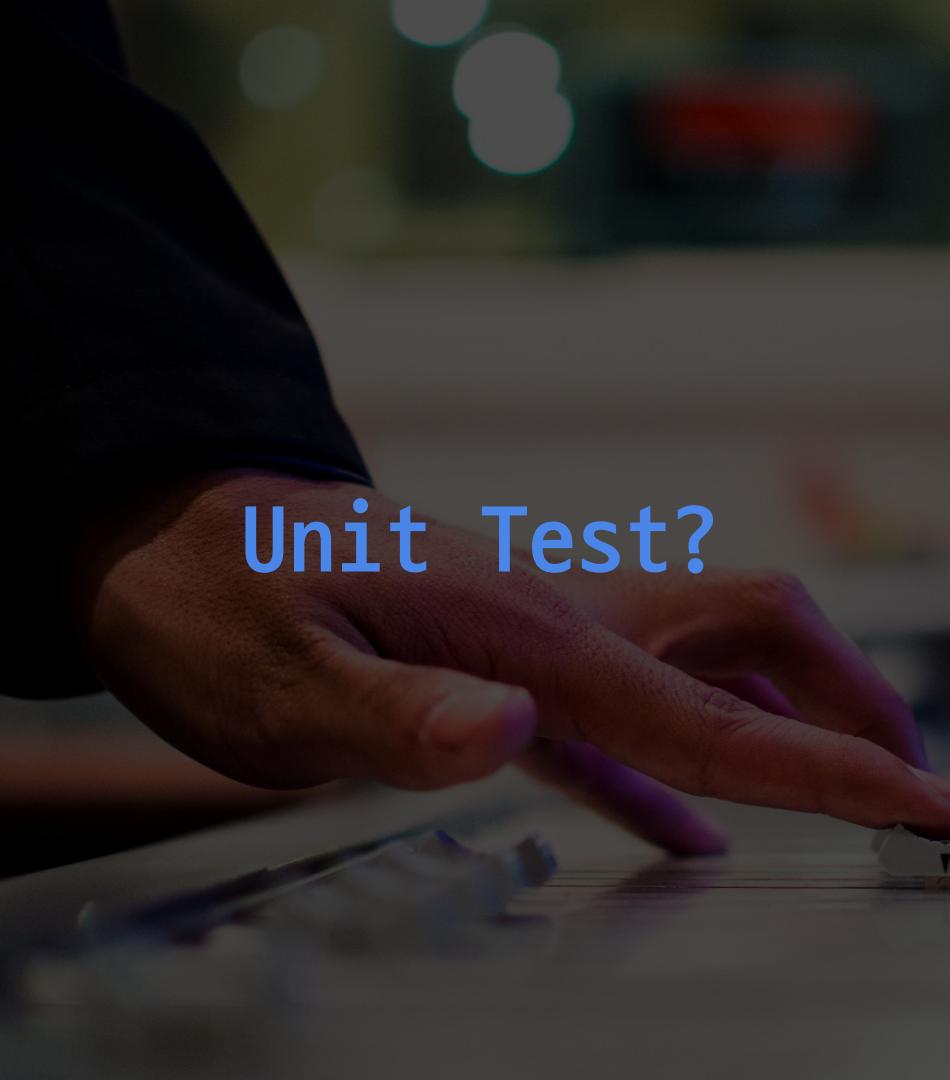



Android Unit Test UI Test Gradle

김병우



Unit Test?

유닛 테스트(unit test)는 컴퓨터 프로그래밍에서 소스 코드의 특정 모듈이 의도된 대로 정확히 작동하는지 검증하는 절차

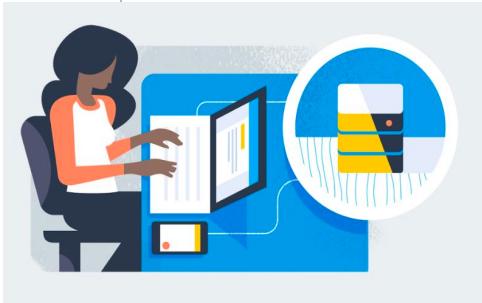
모든 함수와 메소드에 대한 테스트 케이스(Test case)를 작성하는 절차

코드 변경으로 인해 문제가 발생할 경우,
단시간 내에 이를 파악하고 바로 잡을 수
있도록 도와줌(SW 품질 향상 활동)

How it works

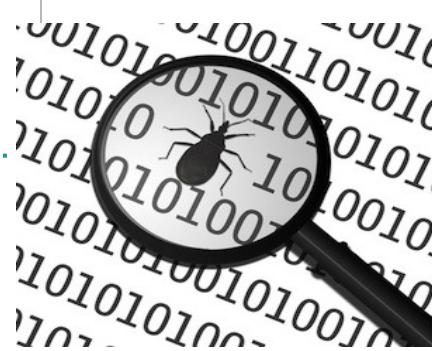
Step 1

소스코드 개발



Step 2

Test case 개발 및 수행

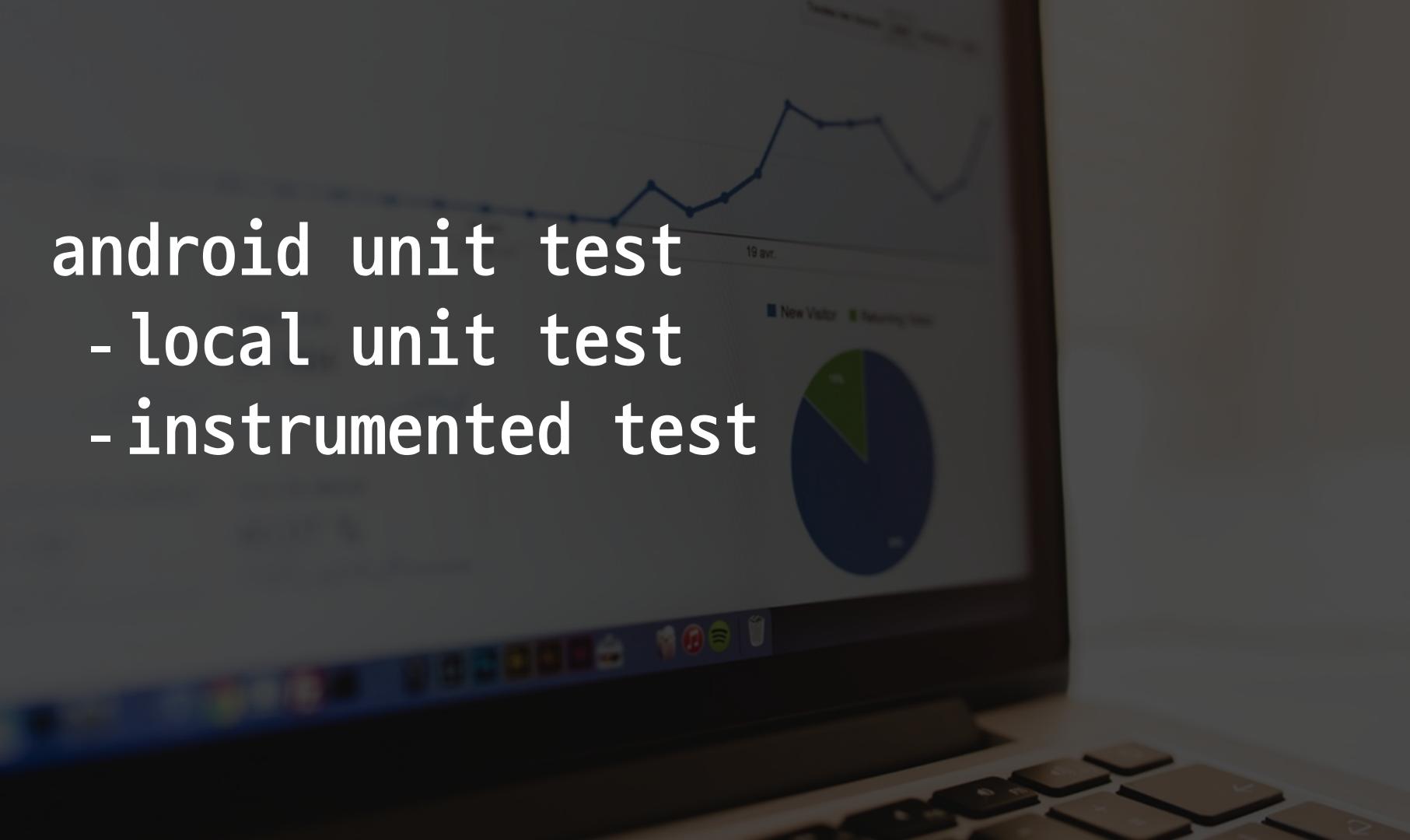


Step 3

결함 추적 및 수정



android unit test
- local unit test
- instrumented test



local unit test

소스 경로 *module-name/src/test/java/*

로컬 JVM(Java Virtual Machine)에서 실행되는 테스트

안드로이드에 대한 의존성이 없거나 단순한 테스트에 이용(Mockito 사용 가능)

예전에는 단순한 자바 코드를 테스트하기 위해 별도의 프로젝트를 만들어서 사용했음(불편했음)

junit4를 이용하는 방법과 거의 동일(<https://github.com/junit-team/junit4/wiki/참조>)

instrumented test

소스경로: *module-name/src/androidTest/java/*

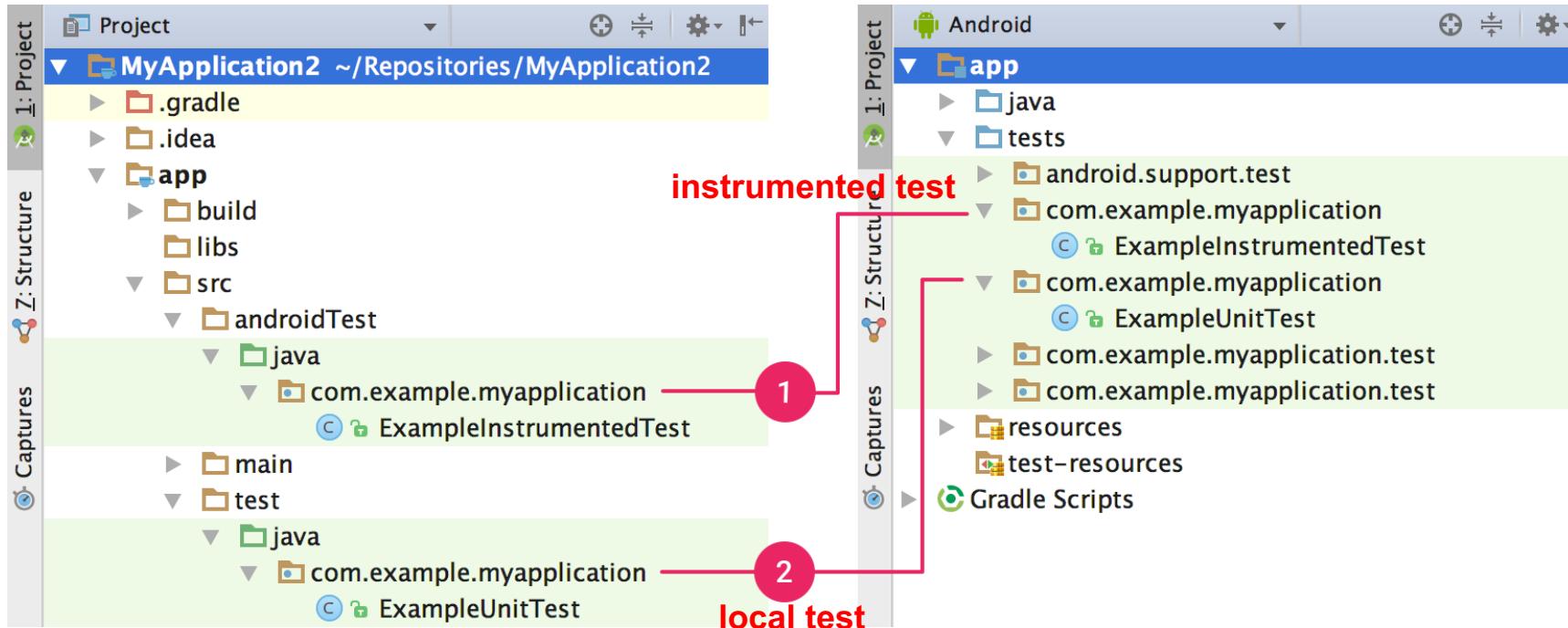
안드로이드 디바이스나 에뮬레이터에서 실행되는 테스트

테스트에서는 Instrumentation API에 액세스가능 테스트 앱의 Context 접근 가능

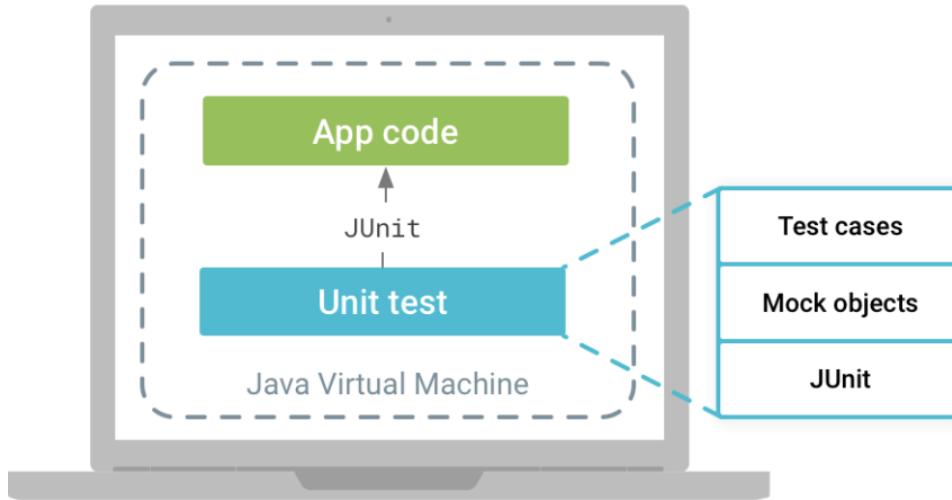
테스트에 Android 종속성이 있는 경우 사용(Activity, Service등)

APK(앱 APK와는 별개)로 빌드되므로 자체 AndroidManifest.xml 파일이 필요하나
Gradle이 빌드 과정에서 이 파일을 자동으로 생성(직접 추가도 가능)

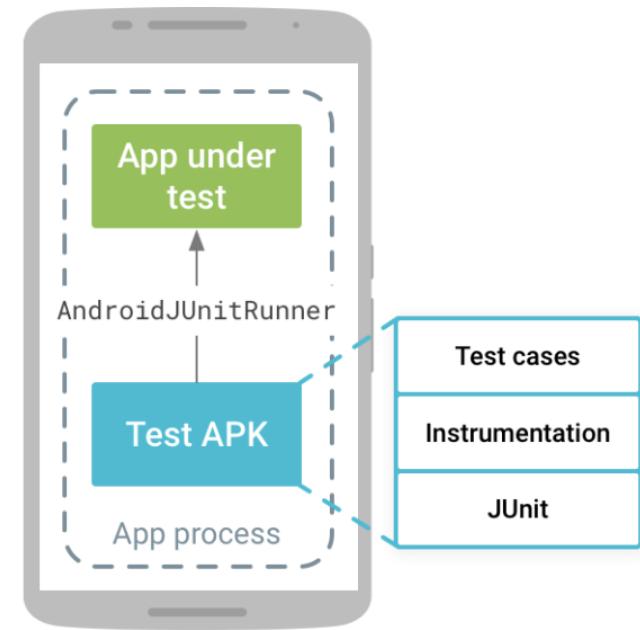
Unit Test 경로



안드로이드 Unit Test 작동방식



Local unit test
`src/test/java/`



Instrumented test
`src/androidTest/java/`

JUnit Assert

코드	설명
<code>assertEquals([message], expected, actual)</code>	<code>expected</code> 와 <code>actual</code> 값이 같은지 비교
<code>assertSame([message], expceted, actual)</code> <code>assertNotSame([message], expceted, actual)</code>	<code>expceted</code> 와 <code>actual</code> 객체가 동일한 객체인지 비교
<code>assertTrue([message], expceted)</code> <code>assertFalse([message], expceted)</code>	참/거짓 판별
<code>assertNull([message], expceted)</code> <code>assertNotNull([message], expceted)</code>	null여부 판단
<code>fail([message])</code>	테스트 실패로 판단

JUnit4 Annotation

annotation	설명
@BeforeClass	테스트 클래스 내에서 수행 전 한 번만 실행, static method 이여야함
@AfterClass	테스트 클래스 내에서 수행 후 한 번만 실행, static method 여야 함
@Before	테스트 케이스 수행 전 반복실행
@After	테스트 케이스 수행 후 반복실행
@Test	수행할 테스트 메소드 지정

The background of the slide features a nighttime city skyline, likely New York City, with numerous skyscrapers and buildings illuminated by their internal lights. The Empire State Building is prominently visible in the center, its Art Deco spire reaching towards a sky filled with scattered clouds. The colors of the sunset or twilight are reflected in the windows of the buildings, creating a warm glow against the dark night.

Unit Test 작성하기 : local unit test

필수 라이브러리와 설정:build.gradle

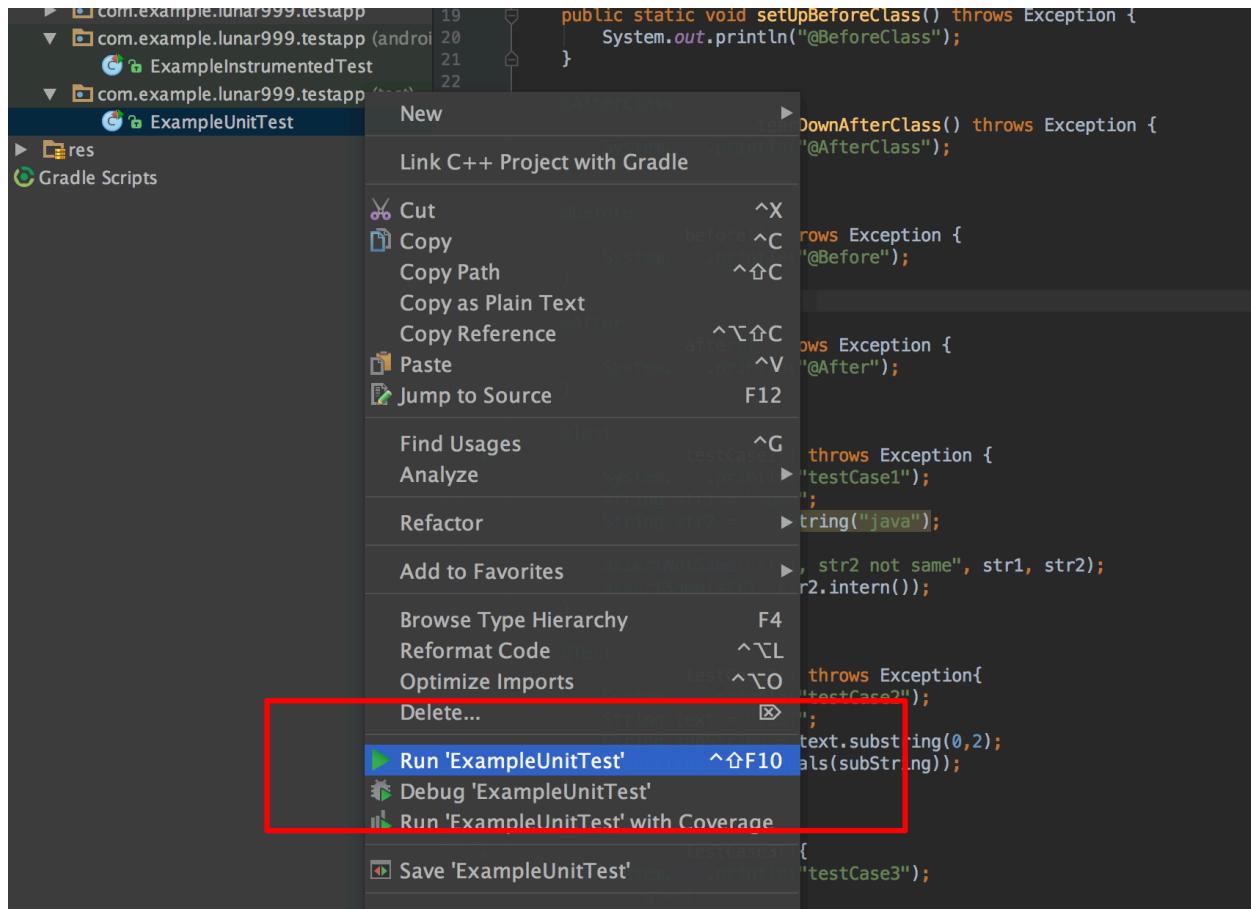
```
dependencies {  
    androidTestCompile 'com.android.support:support-annotations:24.0.0'  
    androidTestCompile 'com.android.support.test:runner:0.5'  
    androidTestCompile 'com.android.support.test:rules:0.5'  
    // Optional -- Hamcrest library  
    androidTestCompile 'org.hamcrest:hamcrest-library:1.3'  
    // Optional -- UI testing with Espresso  
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {  
        exclude group: 'com.android.support', module: 'support-annotations'  
    })  
    // Optional -- UI testing with UI Automator  
    androidTestCompile 'com.android.support.test.uiautomator:uiautomator-v18:2.1.2'  
}  
  
android {  
    defaultConfig {  
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"  
    }  
}
```

local unit test example

```
public class ExampleUnitTest {  
    @BeforeClass  
    public static void setUpBeforeClass() throws Exception{  
        System.out.println("@BeforeClass");  
    }  
  
    @AfterClass  
    public static void tearDownAfterClass() throws Exception{  
        System.out.println("@AfterClass");  
    }  
  
    @Before  
    public void before() throws Exception {  
        System.out.println("@Before");  
    }  
  
    @After  
    public void after() throws Exception {  
        System.out.println("@After");  
    }  
}
```

```
@Test  
public void testCase1() throws Exception {  
    System.out.println("testCase1");  
    String str1 = "java";  
    String str2 = new String("java");  
    assertNotSame("str1, str2 not same", str1,  
    str2);  
    assertEquals(str1, str2.intern());  
}  
  
@Test  
public void testCase2() throws Exception{  
    System.out.println("testCase2");  
    String text = "TEST";  
    String subString = text.substring(0,2);  
    assertTrue("TE".equals(subString));  
}  
  
@Test  
public void testCase3(){  
    System.out.println("testCase3");  
    assertEquals(1+5, 7);  
}
```

Test case run



Test case 결과화면(성공)

The screenshot displays the Android Studio interface during the execution of unit tests. The top half shows the code editor with `ExampleUnitTest.java` open, containing Java test annotations like `@BeforeClass`, `@AfterClass`, `@Before`, `@After`, and `@Test`. The bottom half shows the `Run` tool window with the results of the test run.

Test Results:

- ExampleUnitTest (com.example.lunar999) 2ms
 - testCase1 2ms
 - testCase2 0ms
 - testCase3 0ms

All 3 tests passed - 2ms

Process finished with exit code 0

Bottom status bar: Tests Passed: 3 passed (moments ago)

Test case 결과화면(실패)

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The project is named "TestApp" and contains an "app" module. The "app" module has a "src" directory with "java" and "test" sub-directories. Inside "test/java", there is a file named "ExampleUnitTest.java".
- Code Editor:** The code editor displays the "ExampleUnitTest.java" file. It contains three test methods: "testCase1()", "testCase2()", and "testCase3()". The "testCase3()" method is currently selected.
- Run Tab:** The bottom tab bar shows the "Run" tab is active. The output window shows the results of the test run:
 - 3 tests done: 1 failed - 30ms
 - ExampleUnitTest (com.example.lunar999.testapp) 30ms
 - testCase1 3ms
 - testCase2 0ms
 - testCase3 27ms
 - java.lang.AssertionError: Values should be different. Actual: 7**
 - <1 internal calls>
 - at org.junit.Assert.failEquals(*Assert.java:185*) <2 internal calls>
 - at com.example.lunar999.testapp.ExampleUnitTest.testCase3(*ExampleUnitTest.java:61*) <31 internal calls>
- Bottom Status Bar:** The status bar at the bottom indicates "Tests Failed: 2 passed, 1 failed (a minute ago)".

command line 실행 local unit test

Project root에서 \$>./gradlew test 실행

```
1. bash
lunar999@lunar999-MacBook:~/Desktop/TestApp$ ./gradlew test
Starting a Gradle Daemon, 1 incompatible Daemon could not be reused, use --status for details
Incremental java compilation is an incubating feature.
:app:preBuild UP-TO-DATE
:app:preDebugBuild UP-TO-DATE
:app:checkDebugManifest
:app:preReleaseBuild UP-TO-DATE
:app:prepareComAndroidSupportAnimatedVectorDrawable2531Library
:app:prepareComAndroidSupportAppcompatV72531Library
:app:prepareComAndroidSupportConstraintLayout102Library
:app:prepareComAndroidSupportDesign2531Library
:app:prepareComAndroidSupportRecyclerviewV72531Library
:app:prepareComAndroidSupportCompat2531Library
:app:prepareComAndroidSupportSupportCoreUi2531Library
:app:prepareComAndroidSupportSupportCoreUtils2531Library
:app:prepareComAndroidSupportSupportFragment2531Library
```

command line 실행 결과 #1

app/build/reports/tests/testDebugUnitTest(testReleaseUnitTest)/index.html

▼	app	오늘 오전 11:32	--	폴더
	app.iml	오늘 오전 11:32	11KB	도큐멘트
▼	build	2017년 8월 12일 오후 11:42	--	폴더
▶	generated	2017년 8월 12일 오후 5:50	--	폴더
▶	intermediates	2017년 8월 12일 오후 6:02	--	폴더
▶	outputs	2017년 8월 12일 오후 5:50	--	폴더
▼	reports	어제 오후 9:56	--	폴더
▼	tests	어제 오후 9:56	--	폴더
▼	testDebugUnitTest	2017년 8월 12일 오후 11:39	--	폴더
▶	classes	2017년 8월 12일 오후 11:39	--	폴더
▶	css	2017년 8월 12일 오후 11:39	--	폴더
▶	index.html	오늘 오후 12:34	2KB	HTML 5
▶	js	2017년 8월 12일 오후 11:39	--	폴더
▶	packages	2017년 8월 12일 오후 11:39	--	폴더
▶	testReleaseUnitTest	2017년 8월 12일 오후 11:39	--	폴더
▶	test-results	2017년 8월 12일 오후 11:39	--	폴더

command line 실행 결과 #2

file:///Users/lunar999/Desktop/TestApp/app/build/reports/tests/testDebugUnitTest/index.htm

Studio 단위테스트(Unit Test) 하기 from the Command Line | Android S... UIAutomator2와 DeviceFarm을 활용한... RecyclerView 오류 – Bansook Nam –... Building Instrumented Unit Tests | A... Test results - Test Summary +

Test Summary

3	0	0	0.001s
tests	failures	ignored	duration

100%
successful

Packages Classes

Package	Tests	Failures	Ignored	Duration	Success rate
com.example.lunar999.testapp	3	0	0	0.001s	100%

Generated by Gradle 3.3 at 2017, 8, 15 오후 12:44:37

command line 실행 결과 #3

app/build/test-results/testDebugUnitTest/TEST-패키지명.클래스명.xml

The screenshot shows a Mac OS X Finder window titled "TestApp". The left sidebar lists various locations like iCloud Drive, Applications, Desktop, Documents, Downloads, Pictures, git, and specific user folders (lunar999). The main pane displays a file tree under the "app" folder. The "test-results" folder contains subfolders for "testDebugUnitTest" and "testReleaseUnitTest". The "testDebugUnitTest" folder is selected, and its contents are shown in the table below. The XML file "TEST-com.ex...eUnitTest.xml" is highlighted with a blue selection bar.

이름	수정일	크기	종류
app	오늘 오전 11:32	--	폴더
app.iml	오늘 오전 11:32	11KB	도큐멘트
build	오늘 오후 12:44	--	폴더
generated	2017년 8월 12일 오후 5:50	--	폴더
intermediates	2017년 8월 12일 오후 6:02	--	폴더
outputs	오늘 오후 5:23	--	폴더
reports	오늘 오후 12:44	--	폴더
test-results	오늘 오후 5:26	--	폴더
testDebugUnitTest	오늘 오후 12:44	--	폴더
binary	오늘 오후 12:44	--	폴더
TEST-com.ex...eUnitTest.xml	오늘 오후 12:44	736바이트	XML 문서
testReleaseUnitTest	오늘 오후 12:44	--	폴더
tmp	2017년 8월 12일 오후 11:39	--	폴더
build.gradle	2017년 8월 12일 오후 10:23	1KB	도큐멘트
libs	2017년 8월 12일 오후 5:35	--	폴더
proguard-rules.pro	2017년 8월 12일 오후 5:35	937바이트	도큐멘트
src	2017년 8월 12일 오후 5:35	--	폴더
build	2017년 8월 12일 오후 11:40	--	폴더
build.gradle	2017년 8월 12일 오후 5:35	498바이트	도큐멘트

command line 실행 결과 #4

```
testsuite system-out
<?xml version="1.0" encoding="UTF-8"?>
<testsuite name="com.example.lunar999.testapp.ExampleUnitTest" tests="3" skipped="0" failures="0" errors="0" timestamp="2017-08-15T03:44:37" hostname="lunar999-MacBook.local" time="0.002">
    <properties/>
    <testcase name="testCase1" classname="com.example.lunar999.testapp.ExampleUnitTest" time="0.001"/>
    <testcase name="testCase2" classname="com.example.lunar999.testapp.ExampleUnitTest" time="0.0"/>
    <testcase name="testCase3" classname="com.example.lunar999.testapp.ExampleUnitTest" time="0.0"/>
    <system-out><! [CDATA[@BeforeClass
@Before
testCase1
@After
@Before
testCase2
@After
@Before
testCase3
@After
@AfterClass
]]></system-out>
    <system-err><! [CDATA[]]></system-err>
</testsuite>
```

Unit Test 작성하기: instrumented unit test Espresso, UIAutomator 사용

instrumented test: app code

App (MainActivity)

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    txtView = (TextView) findViewById(R.id.hello_text);
    btnClick = (Button) findViewById(R.id.btnClick);

    RxView.clicks(btnClick)
        .map(event -> count++)
        .subscribe(value -> {
            txtView.setText("count:" + value.toString());
        }, throwable -> {
            Log.e(TAG, "Error: " + throwable.getMessage());
            throwable.printStackTrace();
        });
}
```

instrumented test: UI test code(Espresso 사용)

Test code(ExampleInstrumentedTest)

```
@RunWith(AndroidJUnit4.class)
public class ExampleInstrumentedTest {
    @Rule
    public ActivityTestRule<MainActivity> mActivityRule = new ActivityTestRule<>(MainActivity.class);

    @Test
    public void useApplicationContext() throws Exception {
        Context appContext = InstrumentationRegistry.getTargetContext();
        assertEquals("com.example.lunar999.testapp", appContext.getPackageName());
    }

    @Test
    public void test1() throws Exception {
        onView(withId(R.id.btnClick)).perform(click());
        onView(withId(R.id.hello_text)).check(matches(withText("count:0")));

        onView(withId(R.id.btnClick)).perform(click());
        onView(withId(R.id.hello_text)).check(matches(withText("count:1")));
    }
}
```

Espresso: 함수 설명

`onView` : View 처리(`TestView`, `EditText`, `Button` 등)

`withText` : 해당 텍스트 속성으로 가지는 View를 찾음

`withId` : `R.id.view`에 해당하는 View를 찾음

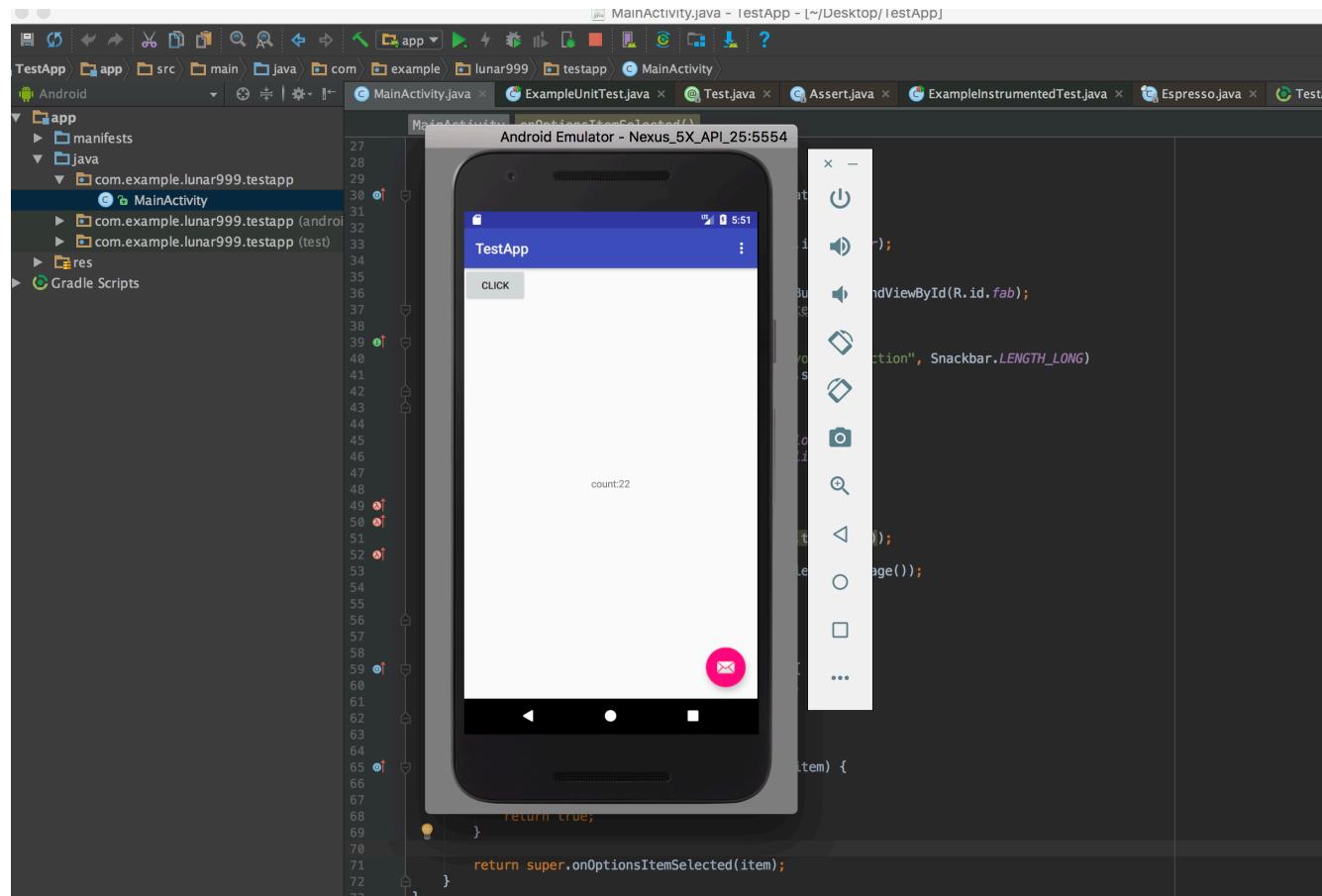
`perform` : `ViewAction`을 실행

`click` : 아이템을 클릭하는 이벤트 실행

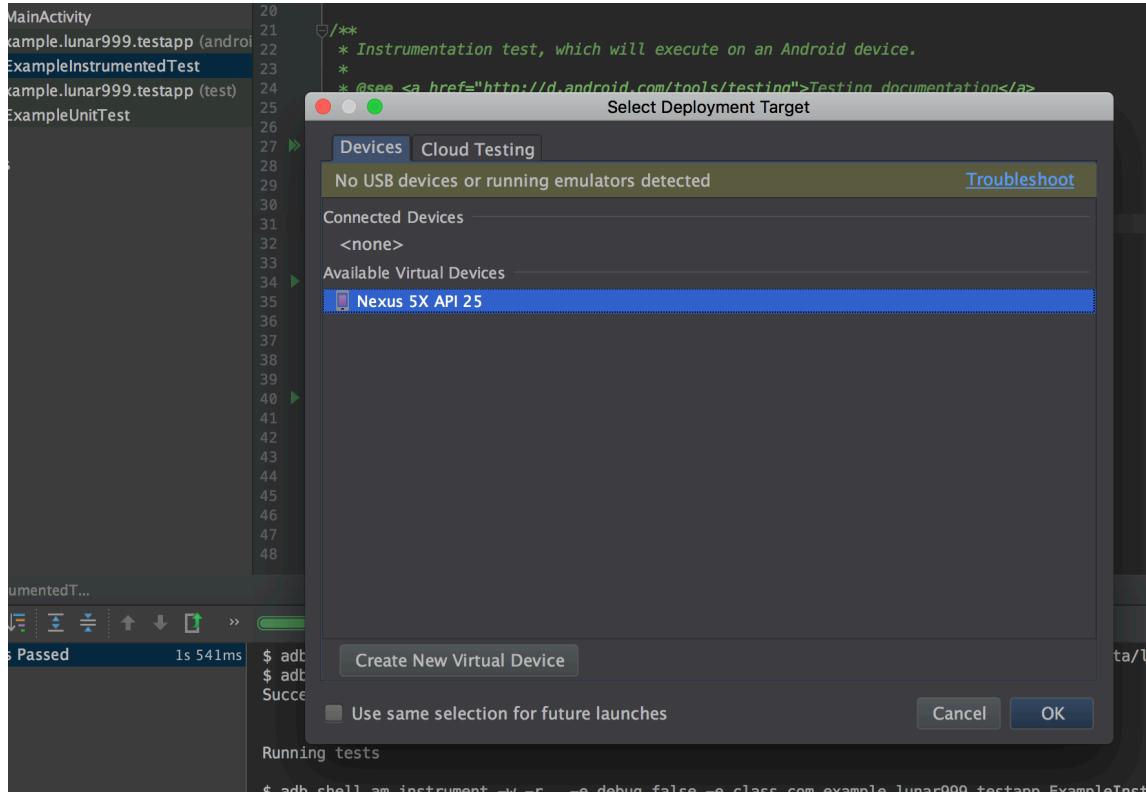
`check(matches())` : `ViewAssertion` 유효성을 체크하고, 맞는지 확인

`isDisplayed` : 화면에 보이는 상태를 확인

App 실행화면



instrumented test 실행



에뮬레이터나 디바이스를 선택하여 실행

실행시에는 테스트용으로 빌드된 앱이 설치가 되어 실행

실행시에 실제로 해당 앱이 실행되는 모습을 볼수 있음

instrumented unit test 결과 화면

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The project is named "TestApp" and contains an "app" module with "src" and "java" directories. Inside "java", there are three packages: "com.example.lunar999.testapp", "com.example.lunar999.testapp (andro)", and "com.example.lunar999.testapp (test)". The "ExampleInstrumentedTest.java" file is selected in the code editor.
- Code Editor:** The code for "ExampleInstrumentedTest.java" is displayed. It includes imports, a Javadoc comment, annotations (@RunWith, @Rule, @Test), and two test methods: "useApplicationContext()" and "test1()".
- Run Tab:** The "Run" tab is active, showing the test results:
 - All 2 tests passed - 1s 541ms
 - adb push /Users/lunar999/Desktop/TestApp/app/build/outputs/apk/app-debug-androidTest.apk /data/local/tmp/com.example.lunar999.testapp.test
 - adb shell pm install -r "/data/local/tmp/com.example.lunar999.testapp.test"
 - Success
- Bottom Bar:** The bottom bar shows various icons for running, debugging, and monitoring the application.

command line 실행 instrumented unit test

Project root에서 \$>./gradlew connectedAndroidTest 실행(에뮬, 기기 연결 필요)

```
1. bash
lunar999@lunar999-MacBook:~/Desktop/TestApp$ ./gradlew connectedAndroidTest
Incremental java compilation is an incubating feature.
:app:preBuild UP-TO-DATE
:app:preDebugBuild UP-TO-DATE
:app:compileDebugNdk UP-TO-DATE
:app:checkDebugManifest
:app:preReleaseBuild UP-TO-DATE
:app:prepareComAndroidSupportAnimatedVectorDrawable2531Library
:app:prepareComAndroidSupportAppcompatV72531Library
:app:prepareComAndroidSupportConstraintConstraintLayout102Library
:app:prepareComAndroidSupportDesign2531Library
:app:prepareComAndroidSupportRecyclerviewV72531Library
:app:prepareComAndroidSupportCompat2531Library
:app:prepareComAndroidSupportSupportCoreUi2531Library
:app:prepareComAndroidSupportSupportCoreUtils2531Library
:app:prepareComAndroidSupportSupportFragment2531Library
:app:prepareComAndroidSupportSupportMediaCompat2531Library
:app:prepareComAndroidSupportSupportV42531Library
:app:prepareComAndroidSupportSupportVectorDrawable2531Library
:app:prepareComAndroidSupportTransition2531Library
:app:prepareComJakewhartonRxbinding2Rxbinding200Library
:app:prepareComJakewhartonRxbinding2RxbindingAppcompatV7200Library
:app:prepareIoReactivexRxjava2Rxandroid201Library
:app:prepareDebugDependencies
:app:compileDebugAidl
:app:compileDebugRenderscript
:app:generateDebugBuildConfig
:app:generateDebugResValues
```

command line 실행 결과 #1

app/build/reports/AndroidTests/connected/패키지명.앱이름.html

▼	app		오늘 오전 11:3
	app.iml		오늘 오전 11:3
▼	build		오늘 오후 12:4
▶	generated		2017년 8월 1
▶	intermediates		2017년 8월 1
▼	outputs		오늘 오후 5:4
▼	androidTest-results		오늘 오후 5:4
▼	connected		오늘 오후 5:4
▶	TEST-SM-G906S - 6.0.1-app-.xml		오늘 오후 5:4
▶	apk		오늘 오후 5:4
▶	code-coverage		오늘 오후 5:2
▶	logs		2017년 8월 1
▼	reports		오늘 오후 5:4
▼	androidTests		오늘 오후 5:4
▼	connected		오늘 오후 5:4
▶	com.example.lunar999.testapp.ExampleInstrumentedTest.html		오늘 오후 5:4
▶	com.example.lunar999.testapp.html		오늘 오후 5:4
▶	css		오늘 오후 5:4

command line 실행 결과 #2

Class com.example.lunar999.testapp.ExampleInstrumentedTest

[all](#) > [com.example.lunar999.testapp](#) > ExampleInstrumentedTest



Tests

Test	SM-G906S - 6.0.1
test1	passed (35.121s)
useApplicationContext	passed (0.240s)

command line 실행 결과 #3

app/build/outputs/androidTest-results/cpnnceted/TEST-단말명-버전-app.xml

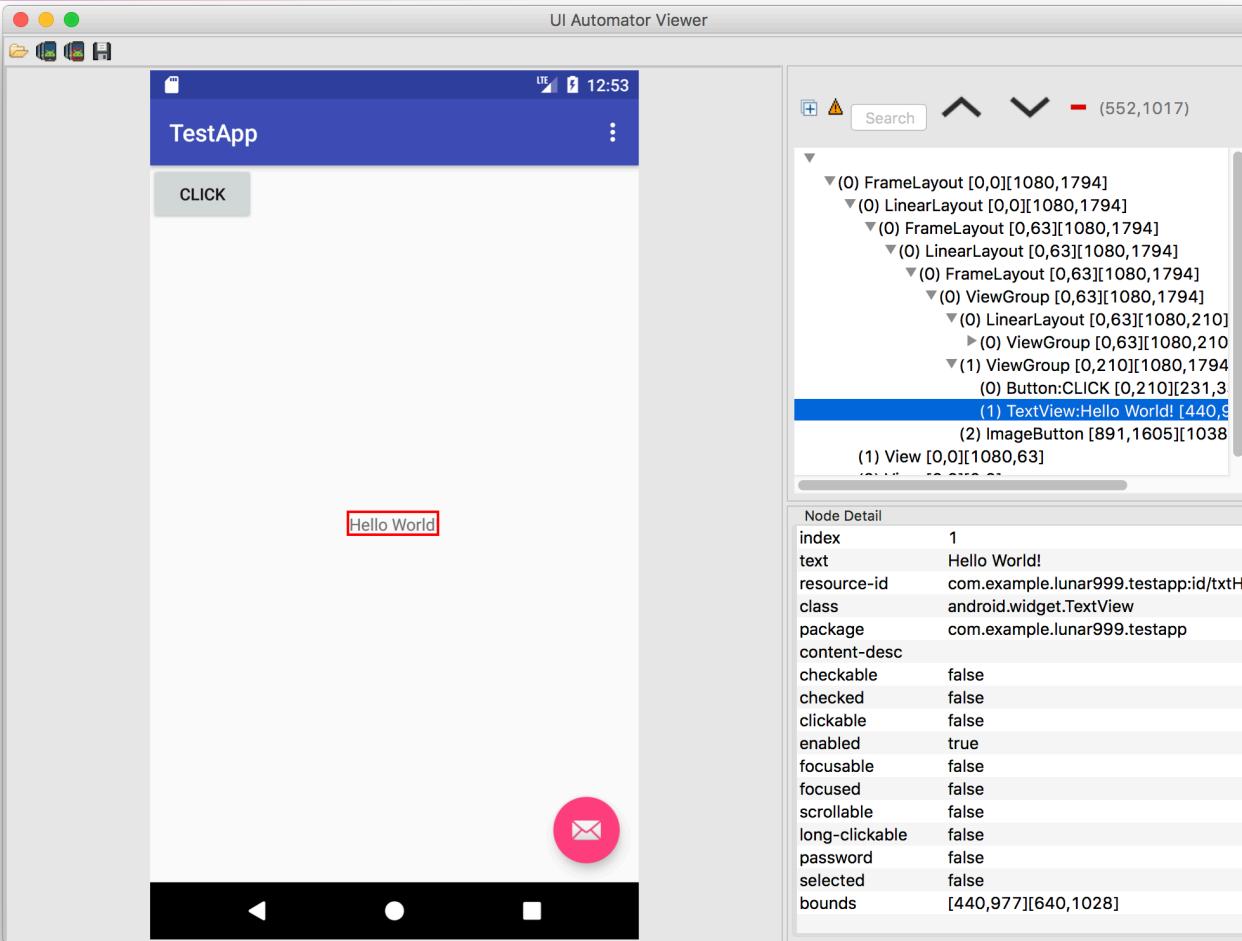
▼	app	오늘 오전
	app.iml	오늘 오전
▼	build	오늘 오전
▶	generated	2017년
▶	intermediates	2017년
▼	outputs	오늘 오전
▶	androidTest-results	오늘 오전
▼	connected	오늘 오전
	TEST-SM-G906S - 6.0.1-app-.xml	오늘 오전
▶	apk	오늘 오전
▶	code-coverage	오늘 오전
▶	logs	2017년
▼	reports	오늘 오전
▼	androidTests	오늘 오전
▼	connected	오늘 오전

command line 실행 결과 #4

```
<?xml version='1.0' encoding='UTF-8' ?>
<testsuite name="com.example.lunar999.testapp.ExampleInstrumentedTest" tests="2" failures="0" errors="0" skipped="0" time="35.35" timestamp="2023-06-20T10:45:23+09:00">
  <properties>
    <property name="device" value="SM-G906S - 6.0.1" />
    <property name="flavor" value="" />
    <property name="project" value="app" />
  </properties>
  <testcase name="test1" classname="com.example.lunar999.testapp.ExampleInstrumentedTest" time="35.121" />
  <testcase name="useApplicationContext" classname="com.example.lunar999.testapp.ExampleInstrumentedTest" time="0.24" />
</testsuite>
```

.xml

uiautomatorviewer



Android SDK에 포함되어
있음([Android/sdk/tools/bin/uiautomatorviewer](#))

uiautomatorviewer는
Android 기기에 현재
표시되는 UI 구성 요소를
검사하고 분석하기에
편리한 GUI를 제공

레이아웃 계층 구조를
검사하고 기기 화면에
표시되는 UI 구성 요소의
속성 확인 가능

instrumented test: UI test code(UIAutomator 사용)

```
@RunWith(AndroidJUnit4.class)
@SdkSuppress(minSdkVersion = 18)
public class ExampleUiAutomatorTest {
    private static final String BASIC_SAMPLE_PACKAGE = "com.example.lunar999.testapp";
    private static final int LAUNCH_TIMEOUT = 5000;
    private static final String STRING_TO_BE_TYPED = "count:1";
    private UiDevice mDevice;

    @Before
    public void startMainActivityFromHomeScreen() {
        // Initialize UiDevice instance
        mDevice = UiDevice.getInstance(InstrumentationRegistry.getInstrumentation());
        // Start from the home screen
        mDevice.pressHome();
        // Wait for launcher
        final String launcherPackage = getLauncherPackageName();
        assertThat(launcherPackage, notNullValue());
        mDevice.wait(Until.hasObject(By.pkg(launcherPackage).depth(0)), LAUNCH_TIMEOUT);

        // Launch the blueprint app
        Context context = InstrumentationRegistry.getContext();
        final Intent intent = context.getPackageManager()
            .getLaunchIntentForPackage(BASIC_SAMPLE_PACKAGE);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK);      // Clear out any previous instances
        context.startActivity(intent);

        // Wait for the app to appear
        mDevice.wait(Until.hasObject(By.pkg(BASIC_SAMPLE_PACKAGE).depth(0)), LAUNCH_TIMEOUT);
    }
}
```

instrumented test: UI test case #1

```
@Test
public void testCase1() {
    mDevice.findObject(By.res(BASIC_SAMPLE_PACKAGE,
"txtHello")).setText(STRING_TO_BE_TYPED);
    mDevice.findObject(By.res(BASIC_SAMPLE_PACKAGE, "btnClick")).click();
    mDevice.findObject(By.res(BASIC_SAMPLE_PACKAGE, "btnClick")).click();

    UiObject2 changedText =
mDevice.wait(Until.findObject(By.res(BASIC_SAMPLE_PACKAGE, "txtHello")), 500);
    assertThat(changedText.getText(), is(equalTo(STRING_TO_BE_TYPED)));
}
```

instrumented test: UI test case #2

```
@Test
public void testCase2() throws UiObjectNotFoundException {
    mDevice.pressHome();
    UiObject appsTab = mDevice.findObject(
        new UiSelector().resourceId("com.google.android.apps.nexuslauncher:id/all_apps_handle"));
    appsTab.click();
    UiScrollable appViews = new UiScrollable(new UiSelector().scrollable(true));
    appViews.setAsHorizontalList();
    UiObject settingsApp = appViews.getChildByText(
        new UiSelector().className(android.widget.TextView.class.getName()), "Settings");
    settingsApp.clickAndwaitForNewWindow();

    UiObject settingsValidation = new UiObject(
        new UiSelector().packageName("com.android.settings"));
    assertThat(settingsValidation.exists(), equalTo(true));
}
```

instrumented test: UIAutomator 함수 설명

UiCollection: 표시되는 텍스트 또는 콘텐츠 설명 속성별로 하위 요소를 카운트하거나 표적화할 목적으로 컨테이너의 UI 요소를 열거

UiObject: 기기에서 눈에 보이는 UI 요소를 표시

UiScrollable: 스크롤 가능한 UI 컨테이너에 있는 항목을 검색하기 위한 지원 기능을 제공

UiSelector: 기기에 있는 하나 이상의 대상 UI 요소에 대한 쿼리를 표시함

Configurator: UI Automator 테스트를 실행하기 위한 주요 매개변수를 설정

instrumented test: 실행 결과

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar shows the project structure under the `TestApp` root, specifically the `app` module. It includes files like `AndroidManifest.xml`, `MainActivity.java`, and `content_main.xml`.
- Code Editor:** The main editor window displays `ExampleUiAutomatorTest.java`. The code contains two test methods: `testCase1()` and `testCase2()`. `testCase1()` performs UI interactions on a device object. `testCase2()` launches the app from the home screen, finds a settings app, and performs validation.
- Run Tab:** The bottom tab bar has the "Run" tab selected, which is highlighted with a red border. The status bar indicates "All 3 tests passed - 24s 773ms".
- Test Results:** The "Test Results" section in the bottom-left shows the execution details:
 - Test started at PM 8:04 ...
 - 08/18 20:04:32: Launching ExampleUiAutomatorTest
 - adb push /Users/lunar999/Desktop/TestApp/app/build/outputs/apk/app-debug.apk /data/local/tmp/com.example.lunar999.testapp
 - adb shell pm install -r "/data/local/tmp/com.example.lunar999.testapp"
 - Success
 - Test Results: 24s 773ms
 - com.example.lunar999.t 24s 773ms
 - testCase1 8s 75ms
 - testCase2 10s 215ms
 - checkPreconditions 6s 483ms
- Event Log:** The bottom right corner shows the "Event Log" tab, which is currently active. It displays the command-line output of the test execution.

UIAutomator, Espresso 차이점

둘다 UI테스트를 위한 비슷한 기능을 제공하는 테스트 프레임워크

Espresso는 하나의 앱만 실행가능

뷰의 일치(클래스명, R.id등), 클릭, 스와이프, 텍스트입력, 링크열기 등

UIAutomator는 여러가지의 앱을 실행가능

기기상태접근(wifi on/off), Home버튼, back버튼, 메뉴버튼 누르기, 기기회전, 알림창 열기, 스크린샷 찍기

A photograph showing a person from behind, climbing a set of dark wooden stairs. The person is wearing a blue t-shirt, light-colored shorts, and flip-flops. They are holding onto a blue railing. The stairs are made of dark wood planks.

Gradle

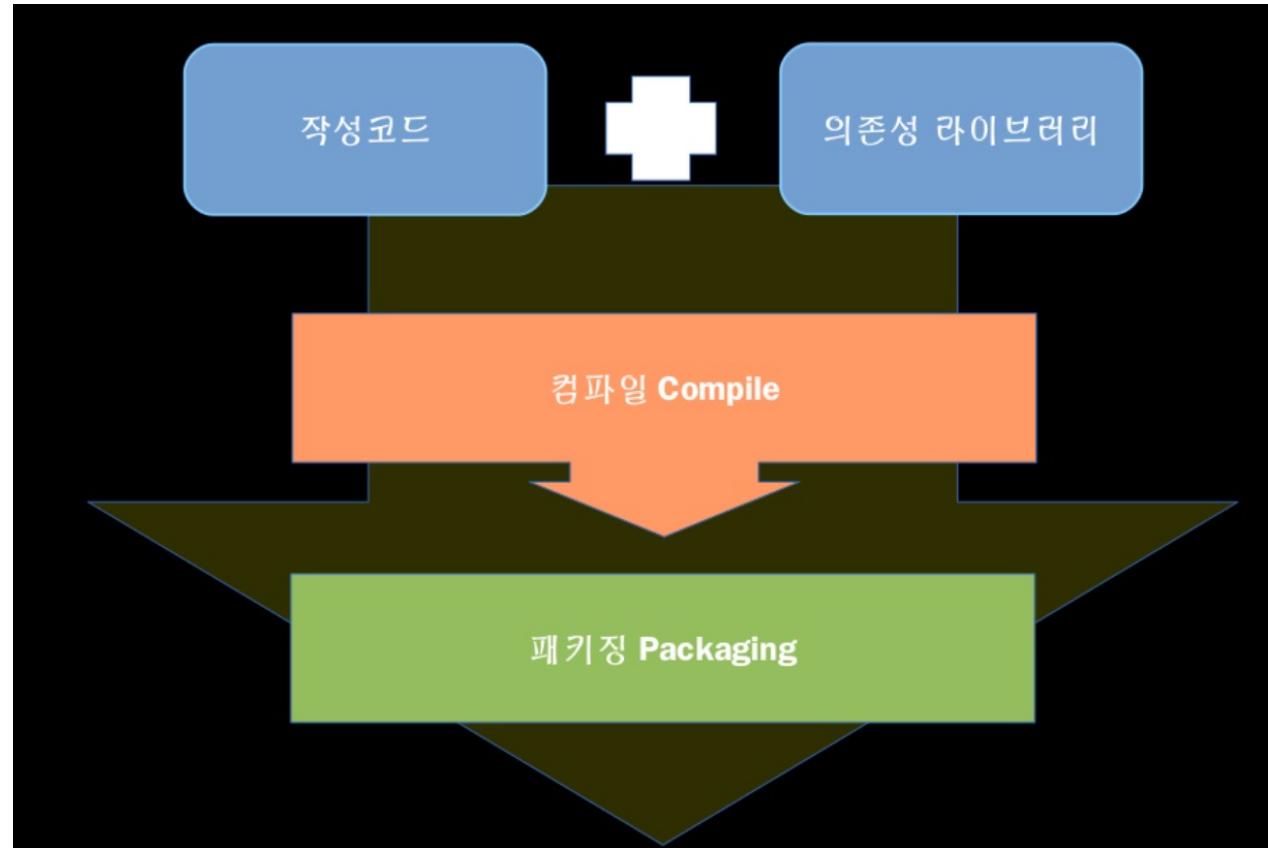
The Gradle logo consists of a large, dark teal silhouette of an elephant facing right. A white outline highlights the elephant's trunk, eye, and foot. Above the elephant's head, a white circle contains a small green bird with a yellow beak and a black outline.

Gradle

빌드시스템이란?

소스코드를 컴파일 하고
패키징 하는 모든 단계

빌드툴 : ant, maven,
gradle 등등등



maven

프로젝트 객체모델(Project Object Model)이라는 개념으로 프로젝트 의존성, 라이브러리, 프로젝트 생명주기 관리기능 등을 제공하는 프로젝트 관리 도구

빌드 기능외에 테스트, 레포팅, 문서작성 기능을 제공함

pom.xml을 활용하여 정의함

정적인 XML파일에 동적인 빌드라는 것을 정의하는 방법에 한계가 있음



gradle

진화된 빌드툴, 테스트, 배포, 개발등에 자동화를 지원

균일한 빌드 환경을 만들어줌

Multi-Project 빌드(모듈)

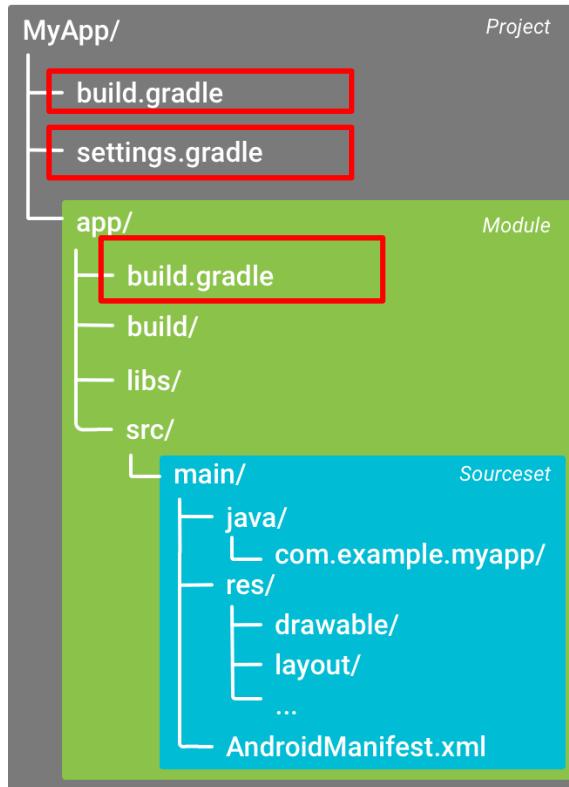
의존성 관리의 다양한 방법 제공

Ant의 유연성과 Maven의 편리성을 조합한 시스템

XML을 사용하지 않고 Groovy 기반의 DSL(Domain Specific Language)로 설정

최근 많은 오픈소스들의 빌드시스템으로 사용됨(Spring, Andoird 등)

안드로이드 gradle 빌드 시스템



안드로이드 프로젝트 생성시 3개의 gradle파일 생성

1. 프로젝트 빌드 설정 파일: build.gradle
2. app 모듈용 빌드 설정 파일: app/build.gradle
3. 빌드 설정: settings.gradle

build.gradle: Project

```
buildscript { //외부 라이브러리 포함할때 사용
    repositories {
        jcenter() //예전에는 mavenCentral()을 많이 썼으나 jcenter를 더 많이 사용
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:2.3.3' //사용라이브러리 지정
            //버전이 분명하지 않으면 +를 이용하여 표시가능(가장 높은 버전이 지정됨
            //com.android.tools.build:gradle:2.3.+"
    }
}
allprojects {
    repositories {
        jcenter() //모든 프로젝트(서브포함)에 jcenter를 사용
    }
}

task clean(type: Delete) {
    delete rootProject.buildDir //clean시 build디렉토리를 삭제
}
```

build.gradle: app

```
apply plugin: 'com.android.application' // com.android.application plug-in을 사용
android {
    compileSdkVersion 25      //빌드에 사용할 SDK버전
    buildToolsVersion "26.0.0"  //앱빌드시 사용하는 빌드툴 버전
    defaultConfig {
        applicationId "com.example.lunar999.testapp"    //애플리케이션의 아이디 패키지 이름과 동일하게 사용
        minSdkVersion 23
        targetSdkVersion 25
        versionCode 1      //app 버전코드
        versionName "1.0"   //app 버전명(앱사용자에게 보이는 버전)
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false //프로가드 활성화를 위한 값
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro' //프로가드사용시 필요한 설정
        }
    }
    compileOptions{
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
}
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:25.3.1'
    androidTestCompile 'com.android.support.test.uiautomator:uiautomator-v18:2.1.2'
    testCompile 'junit:junit:4.12'
}
```

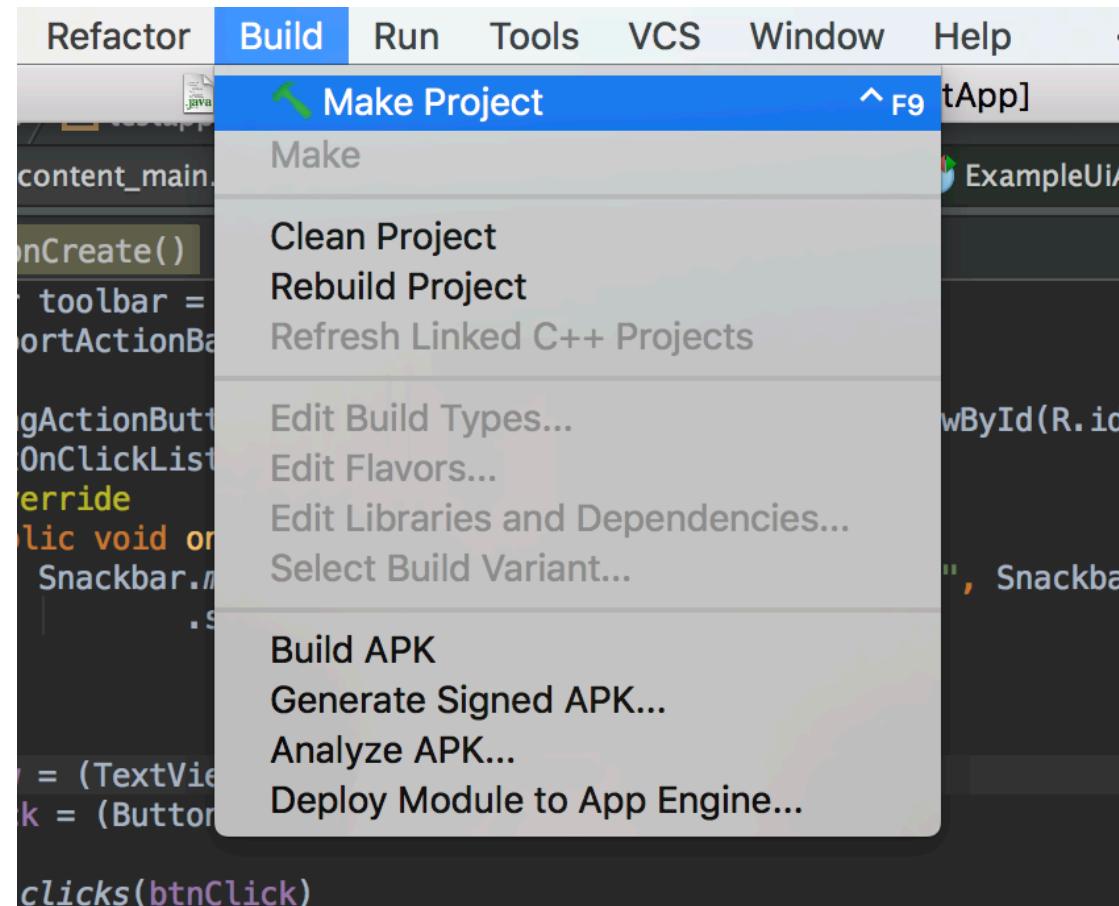
1.compile: 일반 앱 빌드를 위한 라이브러리 지정.
2.androidTestCompile:안드로이드 테스트에 사용하는 라이브러리
3.testCompile - 유닛 테스트에 사용할 라이브러리

gradle build

Memu > build > Make

Project를 선택하거나 app을 run하면 빌드

Clean project: gradle clean수행



gradle build

Gradle task 목록 출력

Project root \$>./gradlew task

주요 task

- clean
- androidDependencies
- assembleDebug
- installDebug
- test
- lint

```
1. bash
-----
All tasks runnable from root project
-----
Android tasks
-----
androidDependencies - Displays the Android dependencies of the project.
signingReport - Displays the signing info for each variant.
sourceSets - Prints out all the source sets defined in this project.

Build tasks
-----
assemble - Assembles all variants of all applications and secondary packages.
assembleAndroidTest - Assembles all the Test applications.
assembleDebug - Assembles all Debug builds.
assembleRelease - Assembles all Release builds.
build - Assembles and tests this project.
buildDependents - Assembles and tests this project and all projects that depend on it.
buildNeeded - Assembles and tests this project and all projects it depends on.
clean - Deletes the build directory.
cleanBuildCache - Deletes the build cache directory.
compileDebugAndroidTestSources
compileDebugSources
compileDebugUnitTestSources
compileReleaseSources
compileReleaseUnitTestSources
mockableAndroidJar - Creates a version of android.jar that's suitable for unit tests.

Build Setup tasks
-----
init - Initializes a new Gradle build. [incubating]
wrapper - Generates Gradle wrapper files. [incubating]

Help tasks
-----
buildEnvironment - Displays all buildscript dependencies declared in root project 'TestApp'.
components - Displays the components produced by root project 'TestApp'. [incubating]
dependencies - Displays all dependencies declared in root project 'TestApp'.
dependencyInsight - Displays the insight into a specific dependency in root project 'TestApp'.
dependentComponents - Displays the dependent components of components in root project 'TestApp'. [incubating]
help - Displays a help message.
model - Displays the configuration model of root project 'TestApp'. [incubating]
projects - Displays the sub-projects of root project 'TestApp'.
properties - Displays the properties of root project 'TestApp'.
tasks - Displays the tasks runnable from root project 'TestApp' (some of the displayed tasks may belong to subprojects).

Install tasks
-----
installDebug - Installs the Debug build.
```

gradle build

```
$>./gradlew androidDependencies
```

앱에서 사용하는 라이브러리
참조정보를 트리형태로 보여줌

```
1. bash
Total time: 6.912 secs
lunar999@lunar999-MacBook:~/Desktop/TestApp$ gradle androidDependencies

> Configure project :app
The setTestClassesDir(File) method has been deprecated and is scheduled to be removed in Gradle 5.0. Please use the setTestClassesDirs(FileCollection) method instead.
The getTestClassesDir() method has been deprecated and is scheduled to be removed in Gradle 5.0. Please use the getTestClassesDirs() method instead.
The ConfigurableReport.setDestination(Object) method has been deprecated and is scheduled to be removed in Gradle 5.0. Please use the method ConfigurableReport.setDestination(File) instead.

> Task :app:androidDependencies
debug
+-- com.android.support:appcompat-v7:25.3.1@aar
|   +--- com.android.support:support-annotations:25.3.1@jar
|   +--- com.android.support:support-v4:25.3.1@aar
|   |   +--- com.android.support:support-compat:25.3.1@aar
|   |   |   \--- com.android.support:support-annotations:25.3.1@jar
|   |   +--- com.android.support:support-media-compat:25.3.1@aar
|   |   |   +--- com.android.support:support-annotations:25.3.1@jar
|   |   |   \--- com.android.support:support-compat:25.3.1@aar
|   |   \--- com.android.support:support-annotations:25.3.1@jar
|   +--- com.android.support:support-core-utils:25.3.1@aar
|   |   +--- com.android.support:support-annotations:25.3.1@jar
|   |   \--- com.android.support:support-compat:25.3.1@aar
|   |       \--- com.android.support:support-annotations:25.3.1@jar
|   +--- com.android.support:support-core-ui:25.3.1@aar
|   |   +--- com.android.support:support-annotations:25.3.1@jar
|   |   \--- com.android.support:support-compat:25.3.1@aar
|   |       \--- com.android.support:support-annotations:25.3.1@jar
|   +--- com.android.support:support-fragment:25.3.1@aar
|   |   +--- com.android.support:support-compat:25.3.1@aar
|   |   |   \--- com.android.support:support-annotations:25.3.1@jar
|   |   +--- com.android.support:support-media-compat:25.3.1@aar
|   |   |   +--- com.android.support:support-annotations:25.3.1@jar
|   |   |   \--- com.android.support:support-compat:25.3.1@aar
|   |   \--- com.android.support:support-annotations:25.3.1@jar
|   +--- com.android.support:support-core-ui:25.3.1@aar
|   |   +--- com.android.support:support-annotations:25.3.1@jar
|   |   \--- com.android.support:support-compat:25.3.1@aar
|   |       \--- com.android.support:support-annotations:25.3.1@jar
|   \--- com.android.support:support-vector-drawable:25.3.1@aar
    +--- com.android.support:support-annotations:25.3.1@jar
    \--- com.android.support:support-compat:25.3.1@aar
        \--- com.android.support:support-annotations:25.3.1@jar
```

gradle build

\$> ./gradlew installDebug -> 디버그키로 만든 apk를 연결될 디바이스에 설치

\$> ./gradlew assembleDebug -> 디버그키로 app build함

\$> ./gradlew :app:asD -> 위와 동일(단축명령어)

\$> ./gradlew test -> 안드로이드 테스트케이스를 실행

\$> ./gradlew lint -> lint를 실행하고 결과를 html이나 xml로 저장함

gradle lint 결과

The screenshot shows a web browser window displaying the Lint Report for an Android application. The URL in the address bar is `file:///Users/lunar999/Desktop/TestApp/app/build/reports/lint-results.html`. The browser interface includes standard navigation buttons (back, forward, search) and tabs for different developer tools.

The main content area is titled "Lint Report: 11 warnings". A timestamp at the top right indicates the check was performed on Saturday, August 19, 2017, at 00:26:28 KST.

The report is organized into sections:

- Overview**: A summary of the findings.
- Correctness**:
 - 2 **OldTargetApi**: Target SDK attribute is not targeting latest version
 - 1 **GradleDependency**: Obsolete Gradle Dependency
 - 2 **GradleOverrides**: Value overridden by Gradle build script
- Security**:
 - 1 **AllowBackup**: AllowBackup/FullBackupContent Problems
- Usability**:
 - 1 **GoogleAppIndexingWarning**: Missing support for Firebase App Indexing
- Internationalization**:
 - 3 **SetTextI18n**: TextView Internationalization
 - 1 **HardcodedText**: Hardcoded text
- Disabled Checks (21)**: A list of 21 disabled checks.

A "DISMISS" button is located at the bottom of the main report area.

A detailed view of a specific warning is shown in a modal dialog:

Target SDK attribute is not targeting latest version

`./src/main/AndroidManifest.xml`:7: Not targeting the latest versions of Android; compatibility modes apply. Consider testing and updating this version. Consult the `android.os.Build.VERSION_CODES` javadoc for details.

gradle: tip

Android studio에서 빌드가 오류가 생기면 우선 gradle sync를 눌러 로그를 확인

Dependency의 순서를 지정못함(gradle 알아서 해줌)

버전을 명시할때 +를 사용하기보다는 정확한 버전을 사용하는게 좋음

gradle 을 daemon을 실행하면 수행시간은 빨라지나 메모리 사용량이 증가
\$>./gradlew -daemon 옵션사용

gradle: tip

Apk파일명에 버전명 추가하기

```
buildTypes {  
    applicationVariants.all{  
        variant->variant.outputs.each {  
            output ->output.outputFile = new File(  
                output.outputFile.parent,  
                output.outputFile.name.replace(".apk",  
                    "-${variant.versionName}.apk"))  
        }  
    }  
}
```

```
lunar999@lunar999-MacBook:~/Desktop/TestApp/app/build/outputs/apk$ ll  
total 12808  
drwxr-xr-x  5 lunar999  staff   170  8 19 00:45 ./  
drwxr-xr-x  7 lunar999  staff   238  8 15 17:42 ../  
-rw-r--r--  1 lunar999  staff  2681122  8 19 00:45 app-debug-1.0.apk  
-rw-r--r--  1 lunar999  staff  1187898  8 18 20:04 app-debug-androidTest.apk  
-rw-r--r--  1 lunar999  staff  2681153  8 19 00:34 app-debug.apk
```

```
lunar999@lunar999-MacBook:~/Desktop/TestApp/app/build/outputs/apk$ █
```

