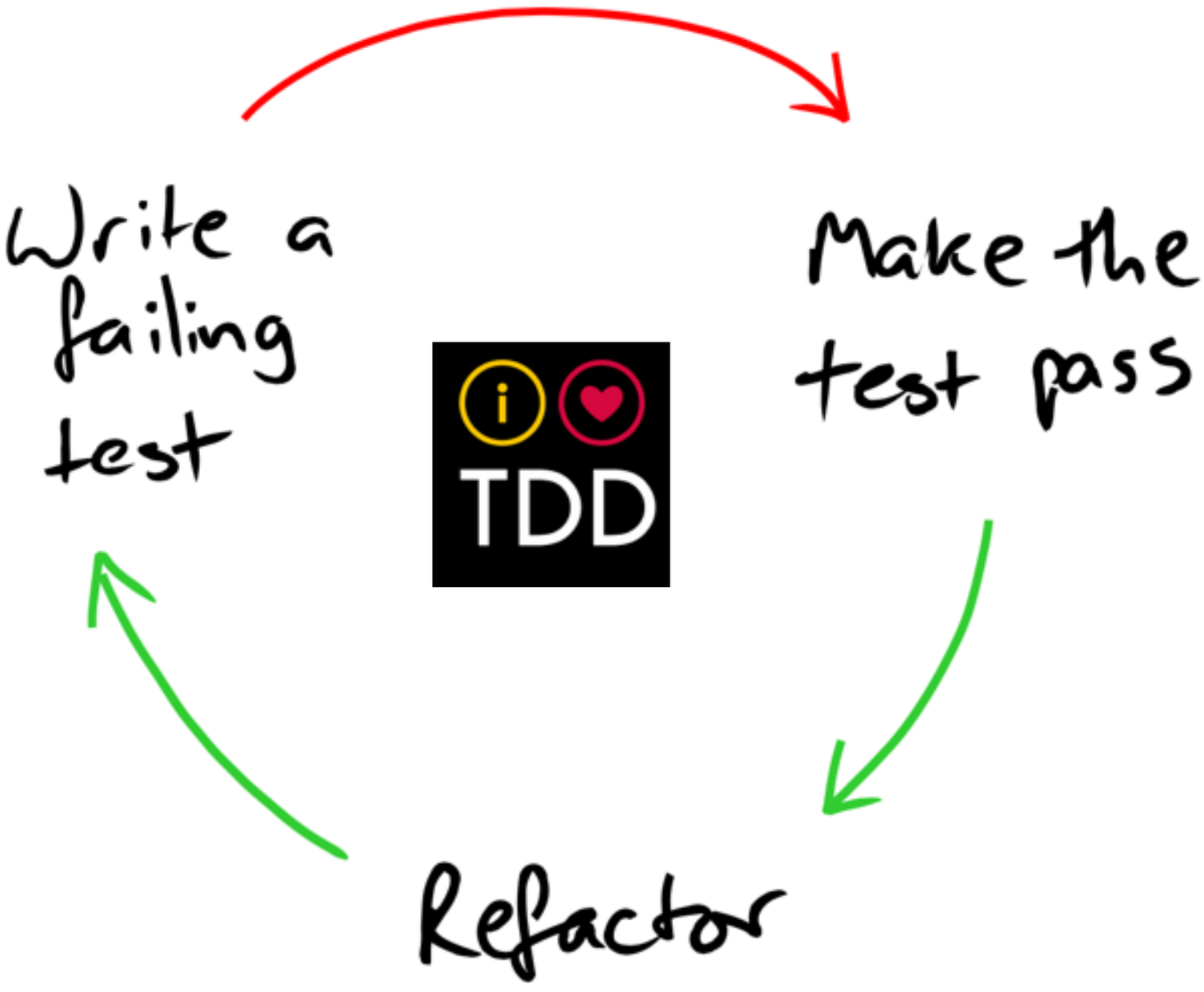
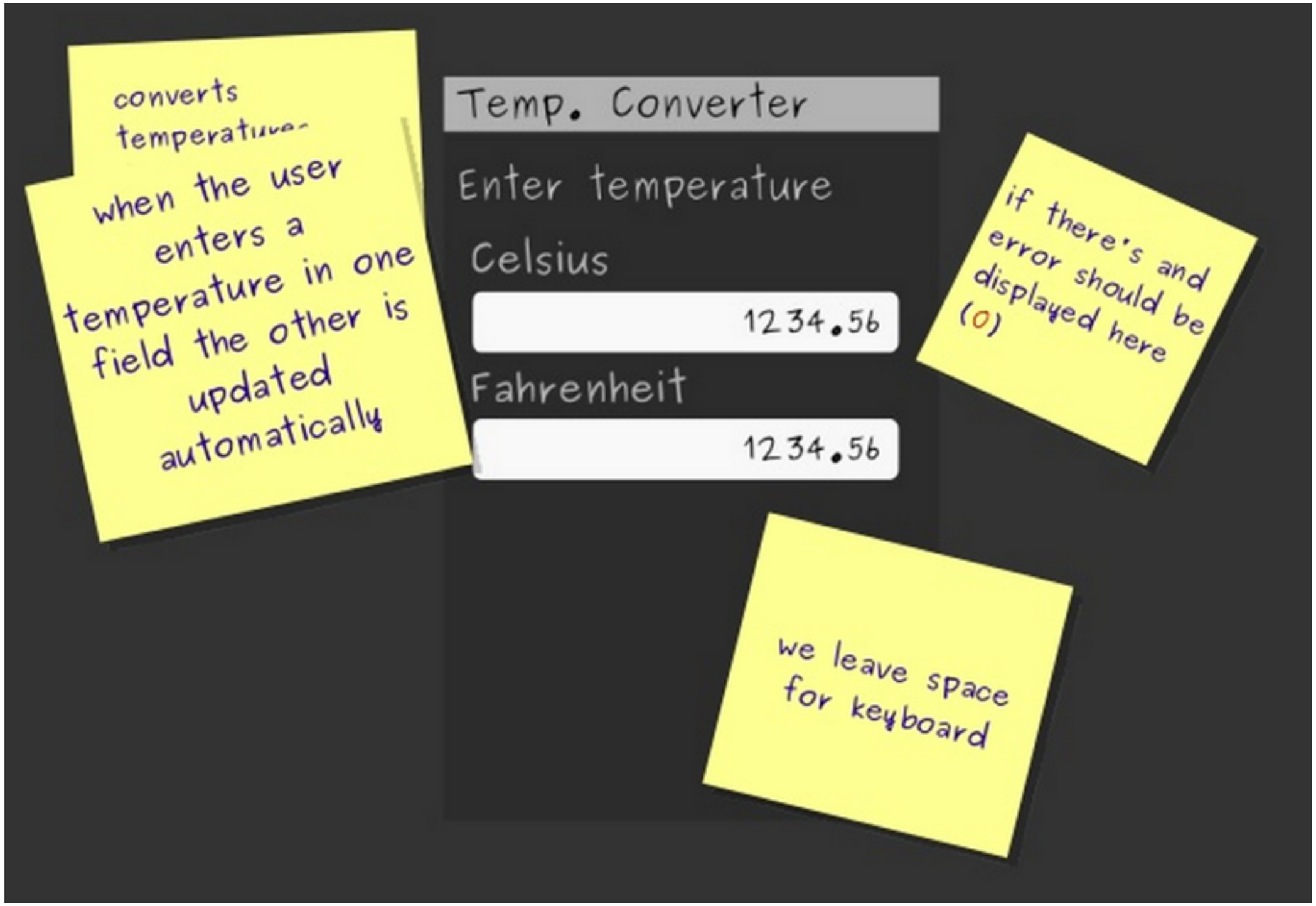
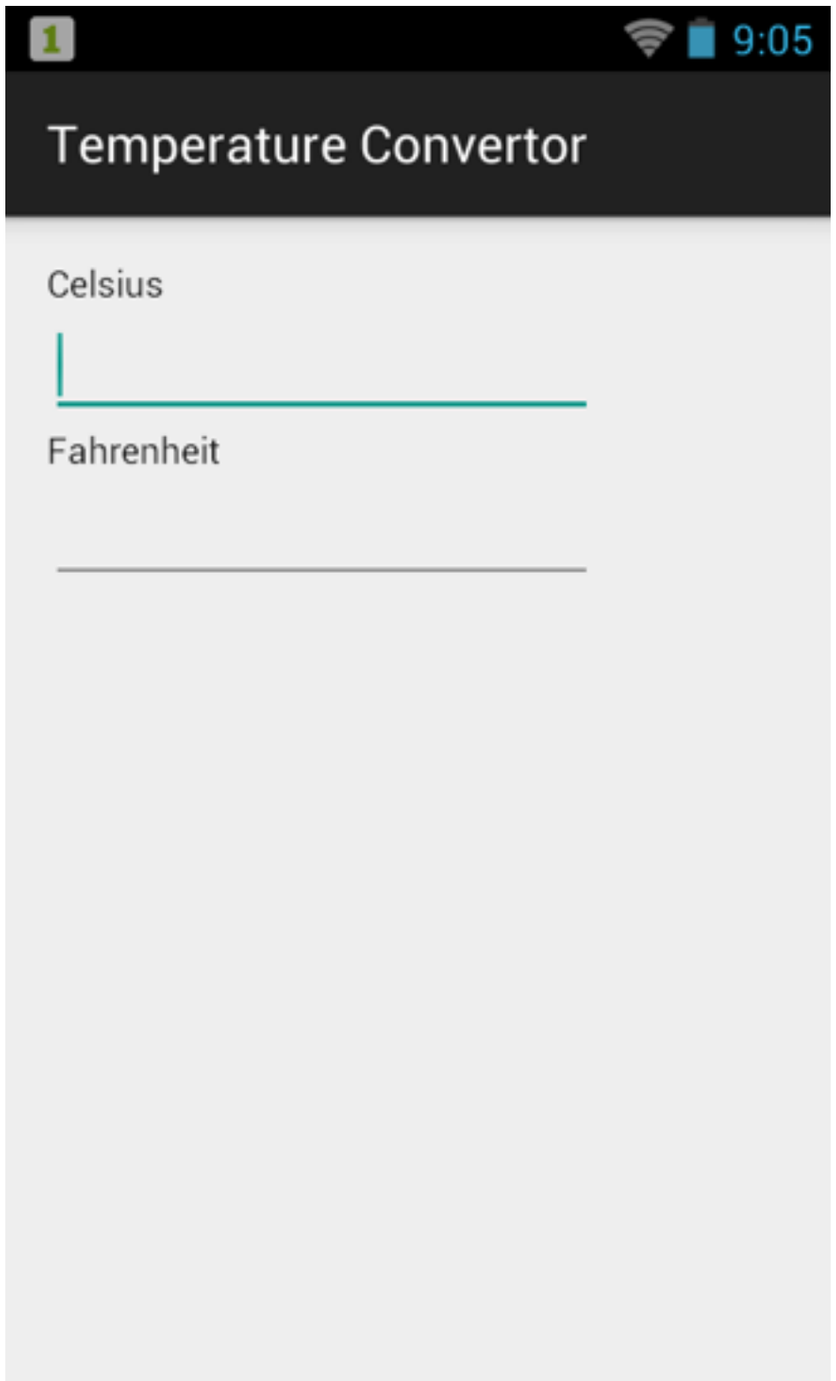
WORKSHOP #2



Temperature conversion

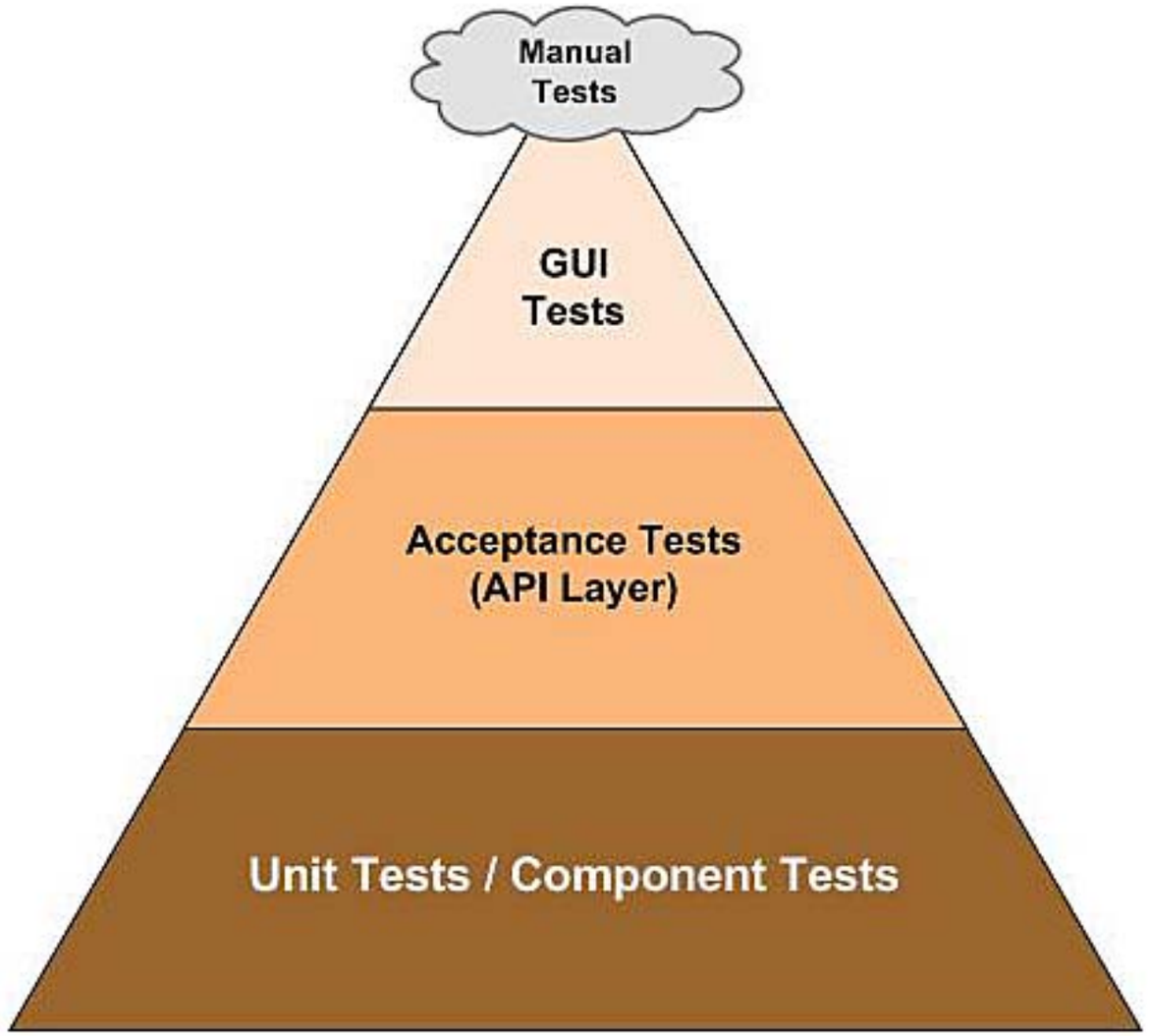


My application



How to test ?

Android Testing



Monkey

Espresso

Android unit test

jUnit

UI TESTING FOR ANDROID

espresso

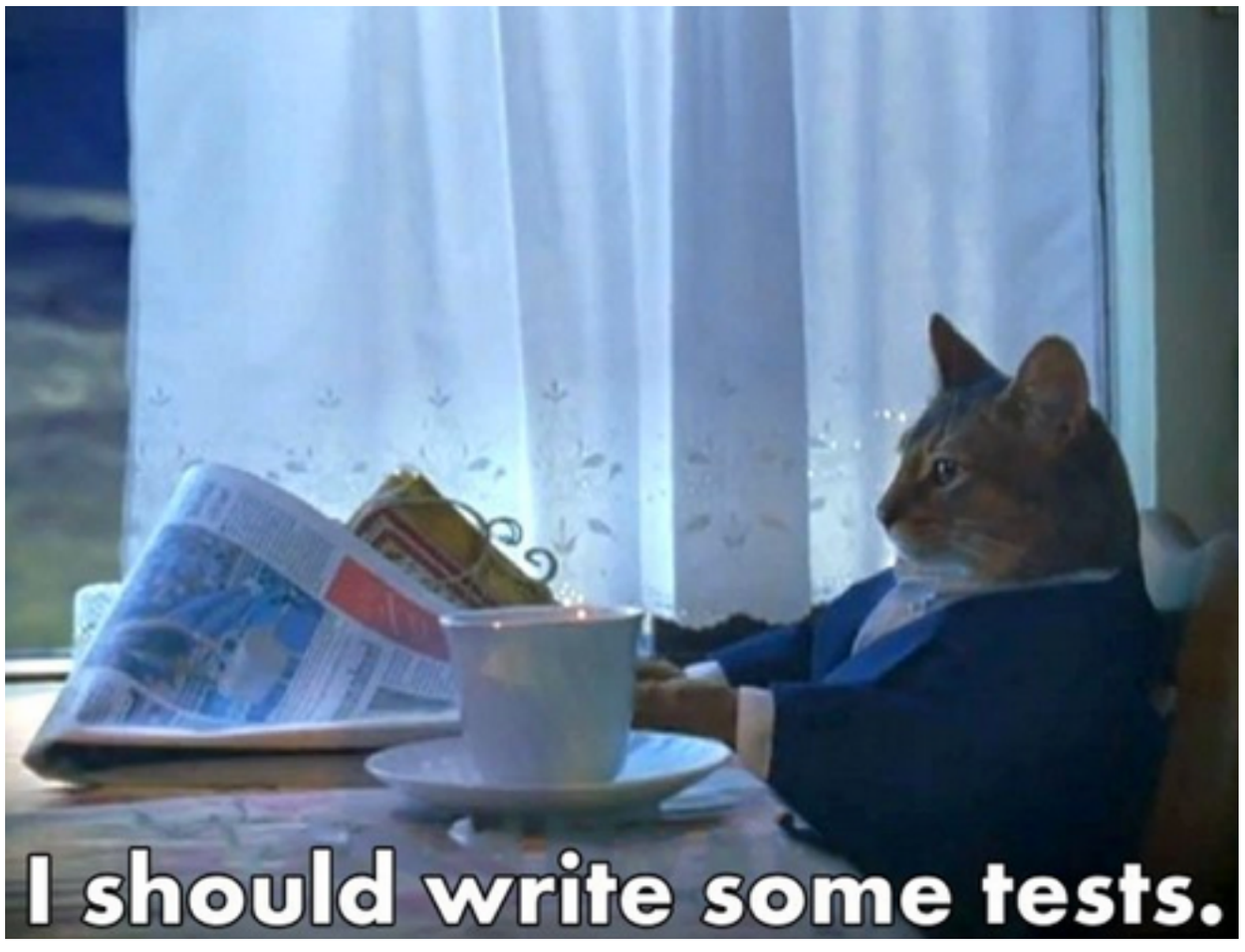
https://code.google.com/p/android-test-kit

Espresso ?

- A funny little Android UI test APIs
- Created by Google
- Easy APIs

Why Espresso ?

“Developer Developer Developer”



Why Espresso ?

- Developer need ...
  - easy
  - reliable
  - durable

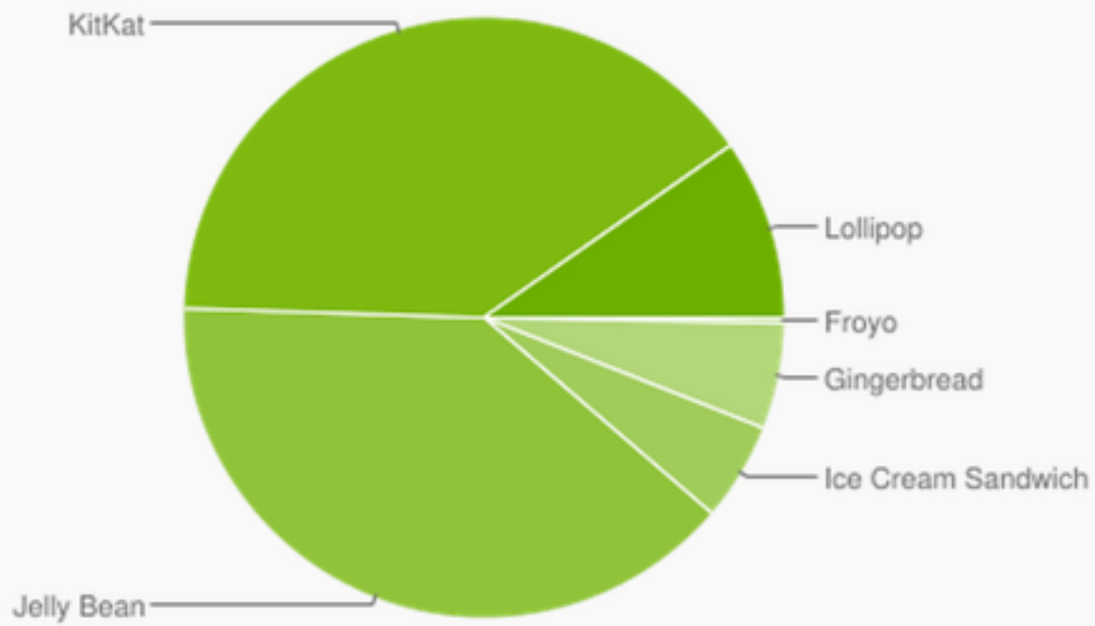
Why Espresso ?

- On all API Levels

Codename	API
Froyo	8
Gingerbread	10
Ice Cream Sandwich	15
Jelly Bean	16,17,18
KitKat	19
Lollipop	21

Platform version

Version	Codename	API	Distribution
2.2	Froyo	8	0.3%
2.3.3 - 2.3.7	Gingerbread	10	5.7%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	5.3%
4.1.x	Jelly Bean	16	15.6%
4.2.x		17	18.1%
4.3		18	5.5%
4.4	KitKat	19	39.8%
5.0	Lollipop	21	9.0%
5.1		22	0.7%



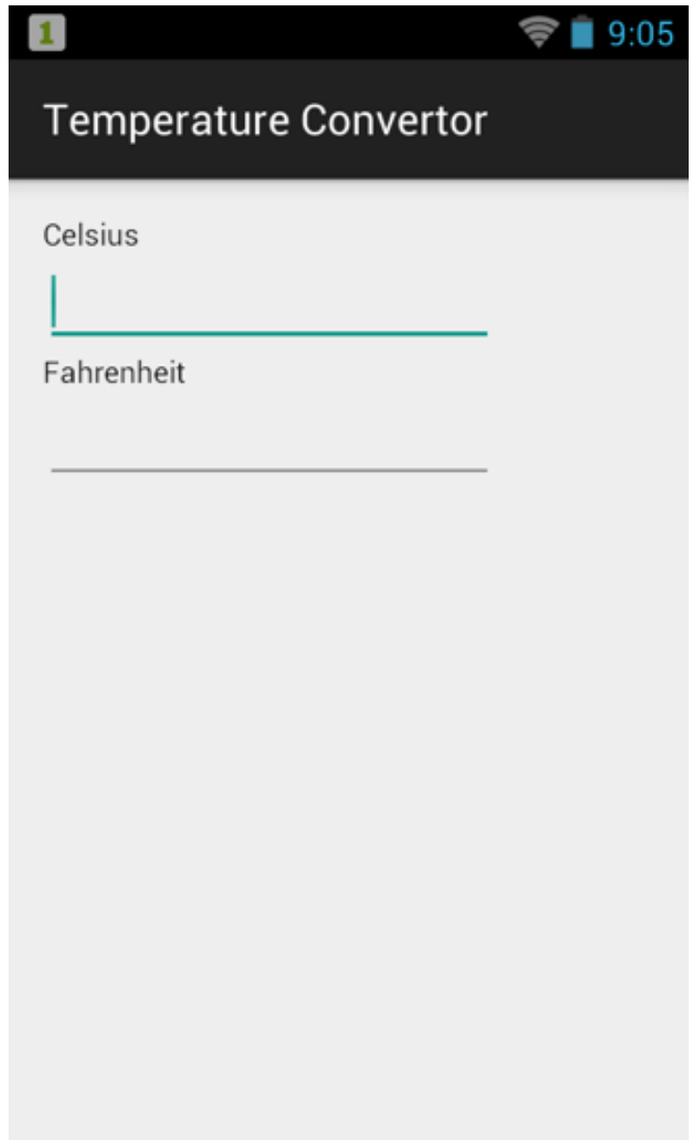
Data collected during a 7-day period ending on May 4, 2015.
Any versions with less than 0.1% distribution are not shown.

http://developer.android.com/about/dashboards/index.html#Platform

Easy :: API for testing

What would a device user do ?

- Find a view
- Do something with it
- Check some state



Find view & Do

```
onView(withId(R.id.greet_button))
        .perform(click());
```

Check something

```
onView(withText("Hello Steve!"))
        .check(matches(isDisplayed()));
```

Easy :: Framework APIs

onView(Matcher<View>)

perform(ViewAction)

check(ViewAssertion)

Easy :: Framework APIs

onView(**Matcher<View>**)

- withId
- withText
- withContentDescription
- isDisplay
- hasFocus
- hasSibling
- custom

perform(ViewAction)

check(ViewAssertion)

Easy :: Framework APIs

onView(Matcher<View>)

perform(**ViewAction**)

- click
- longClick
- doubleClick
- typeText
- scrollTo
- custom

check(ViewAssertion)

Easy :: Framework APIs

onView(Matcher<View>)

perform(ViewAction)

check(**ViewAssertion**)

- matches
- doesNotExist
- custom

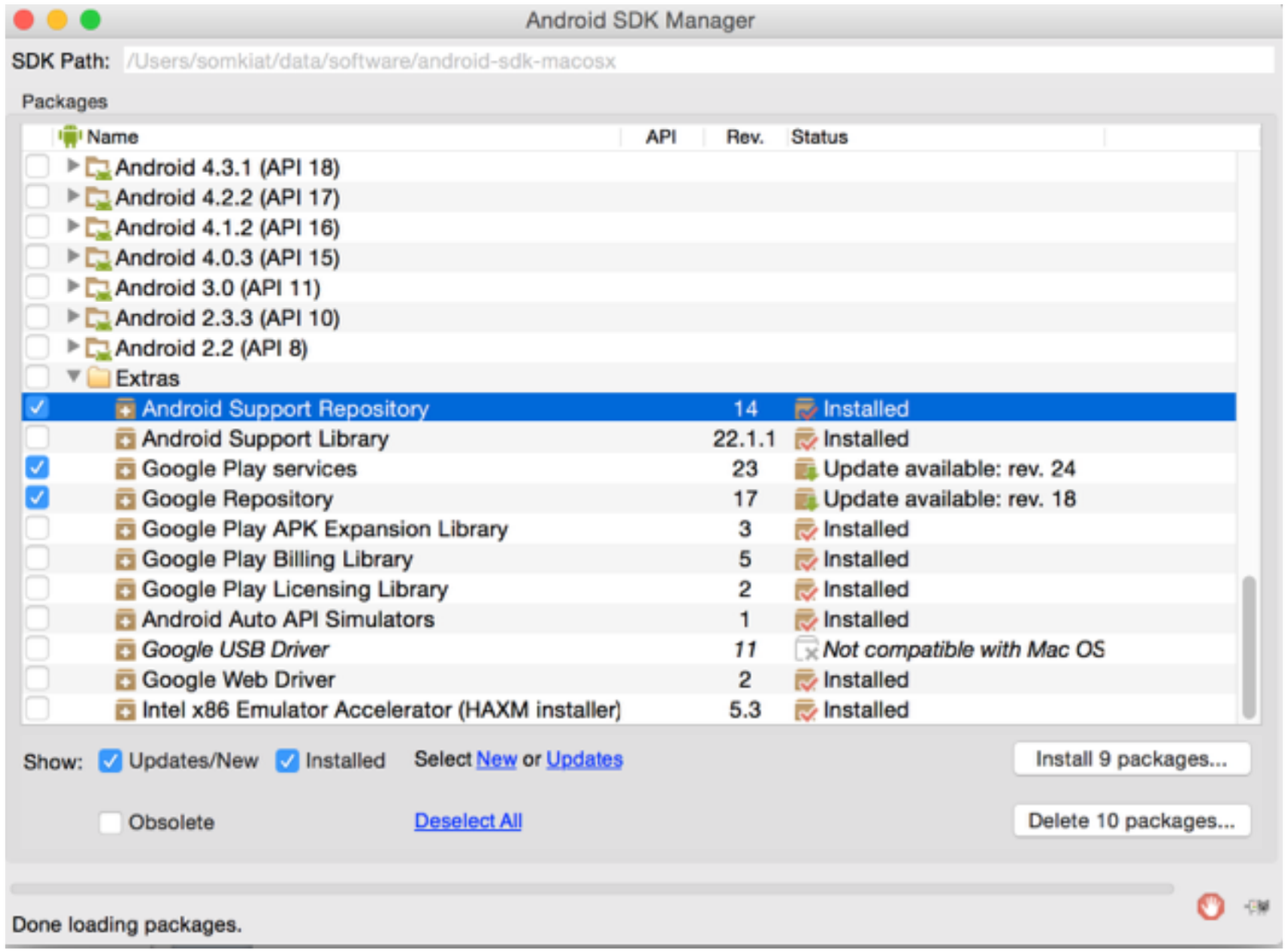
Easy :: To install

How to install espresso ?

https://code.google.com/p/android-test-kit/wiki/EspressoSetupInstructions

Install Android suport repository

Tools -> Android -> SDK Manager



Config app/build.gradle

```
dependencies {

    compile fileTree(dir: 'libs', include: ['*.jar'])

    compile 'com.android.support:appcompat-v7:22.1.1'

    androidTestCompile 'com.android.support.test:runner:0.2'

    androidTestCompile 'com.android.support.test:rules:0.2'

    androidTestCompile 'com.android.support.test.espresso:espresso-core:2.1'

}
```

Config app/build.gradle

```
android {

    ...

    defaultConfig {

        ...

        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"

    }

    ...

}
```

Create first test !!

Default in **src/androidTest/java/<your package>**

- HelloEspressoTest.java

Style #1

```
import static android.support.test.espresso.Espresso.onView;
import static android.support.test.espresso.assertion.ViewAssertions.matches;
import static android.support.test.espresso.matcher.ViewMatchers.isDisplayed;
import static android.support.test.espresso.matcher.ViewMatchers.withText;

import workshop.temperatureconvertor.MainActivity;

@LargeTest
public class HelloEspressoTest extends ActivityInstrumentationTestCase2<MainActivity> {
    public HelloEspressoTest(Class<MainActivity> activityClass) {
        super(activityClass);
    }

    @Override
    public void setUp() throws Exception {
        super.setUp();
        getActivity();
    }

    public void testShowCelsius() {
        onView(withText("Celsius")).check(matches(isDisplayed()));
    }
}
```

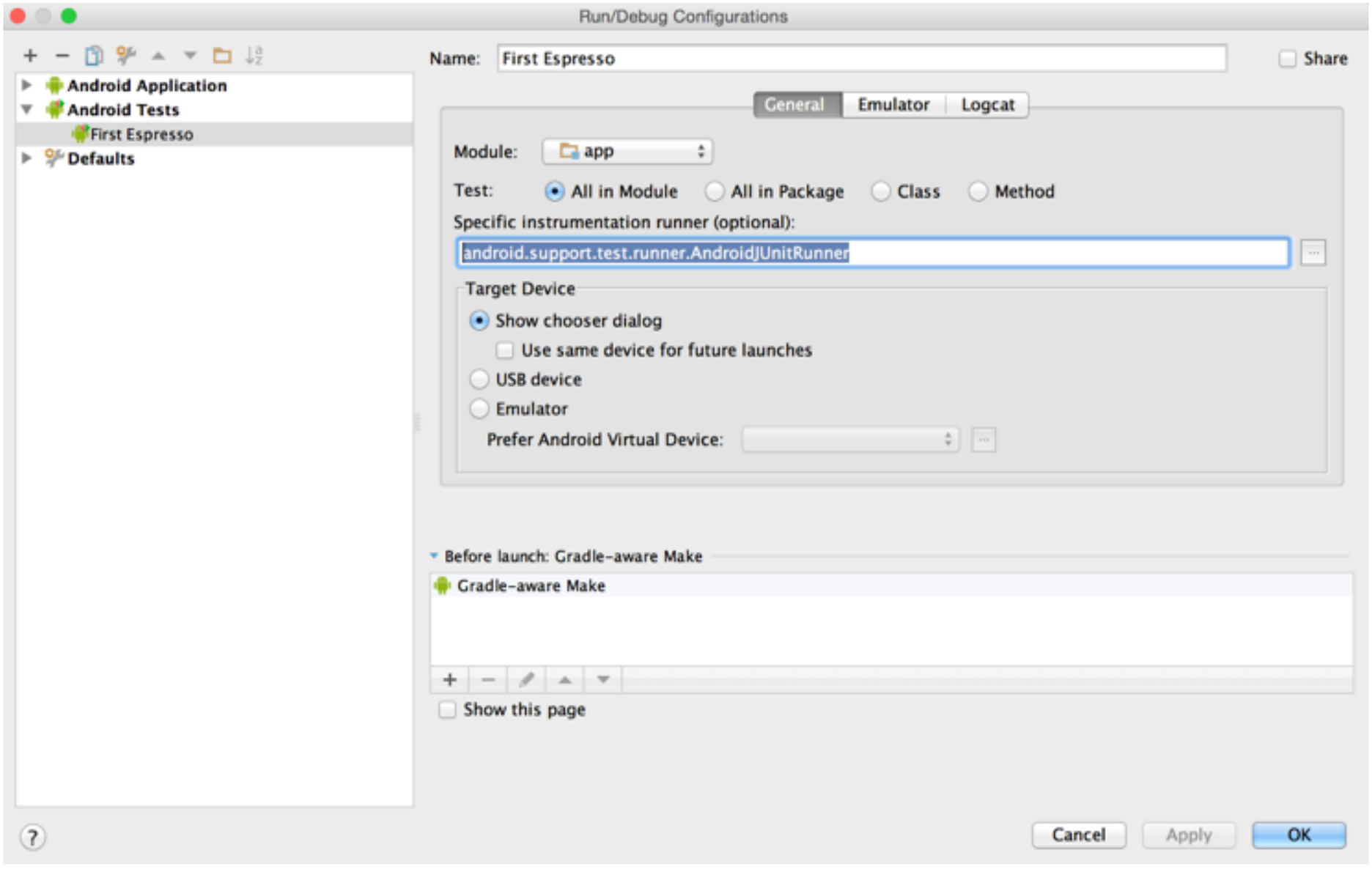
Style #2

```
import static android.support.test.espresso.Espresso.onView;
import static android.support.test.espresso.assertion.ViewAssertions.matches;
import static android.support.test.espresso.matcher.ViewMatchers.isDisplayed;
import static android.support.test.espresso.matcher.ViewMatchers.withText;

@RunWith(AndroidJUnit4.class)
@LargeTest
public class HelloEspressoTest {

    @Rule
    public ActivityTestRule activityActivityTestRule = new ActivityTestRule(MainActivity.class);

    @Test
    public void listGoesOverTheFold() {
        onView(withText("Celsius")).check(matches(isDisplayed()));
    }

}
```

How to run test ?

Run -> Edit configuration -> Add android tests

Runner = **android.support.test.runner.AndroidJUnitRunner**



How to run test ?

On command line or terminal

$./gradlew :App:connectedAndroidTest

For your app

- Fill in data in **Celcius**
- Check result in **Fahrenheit**

- Fill in data in **Fahrenheit**
- Check result in **Celcius**

Try by yourself

Demo

```
@Test
public void convertDataFromCelsiusToFahrenheit() {
    onView(withId(R.id.celsius_value)).perform(ViewActions.typeText("0"));
    onView(withId(R.id.fahrenheit_value)).check(matches(withText("32.0")));
}
```

```
@Test
public void convertDataFromFahrenheitToCelsius() {
    onView(withId(R.id.fahrenheit_value)).perform(ViewActions.typeText("32"));
    onView(withId(R.id.celsius_value)).check(matches(withText("0.0")));
}
```

Many data test ?

Use Parameterized runner

```
@RunWith(Parameterized.class)
public class DemoParameterizedTest {

    private final String celsius;
    private final String fahrenheit;

    @Rule
    public ActivityTestRule activityActivityTestRule = new ActivityTestRule(MainActivity.class);

    @Parameters
    public static Iterable<Object[]> data() {
        return Arrays.asList(new Object[][] {
                {"0", "32.0"},
                {"1", "33.8"}
        });
    }

    public DemoParameterizedTest(String celsius, String fahrenheit) {
        this.celsius = celsius;
        this.fahrenheit = fahrenheit;
    }

    @Test
    public void convertDataFromCelsiusToFahrenheit() {
        onView(withId(R.id.celsius_value)).perform(ViewActions.typeText(this.celsius));
        onView(withId(R.id.fahrenheit_value)).check(matches(withText(this.fahrenheit)));
    }

}
```