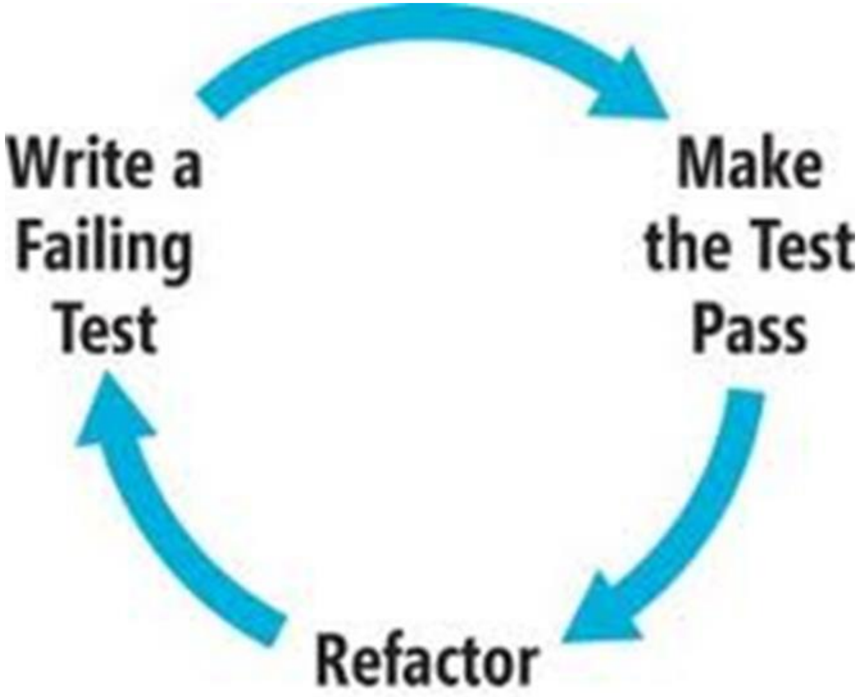
Android TDD

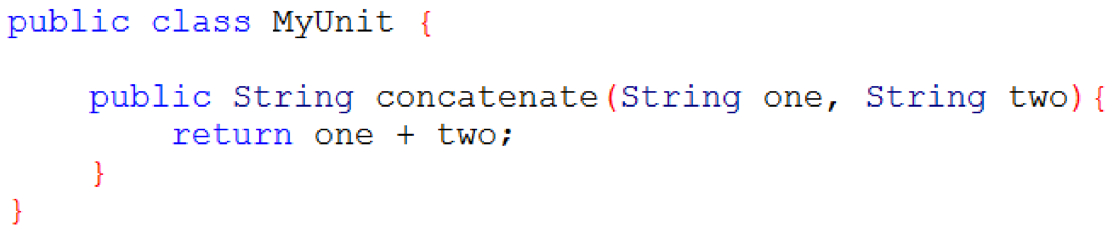
Godfrey Nolan



Agenda

- Intro
- Writing Android Unit Tests
- Other Tools
- TDD
- Apply TDD to an existing project

Intro
Unit Test Example



```
public class MyUnit {

    public String concatenate(String one, String two){
        return one + two;
    }
}
```

Intro
Unit Test Example

```
public class MyUnit {

    public String concatenate(String one, String two){
        return one + two;
    }
}
```

```
import org.junit.Test;
import static org.junit.Assert.*;

public class MyUnitTest {

    @Test
    public void testConcatenate() {
        MyUnit myUnit = new MyUnit();

        String result = myUnit.concatenate("one", "two");

        assertEquals("onetwo", result);

    }
}
```

Intro

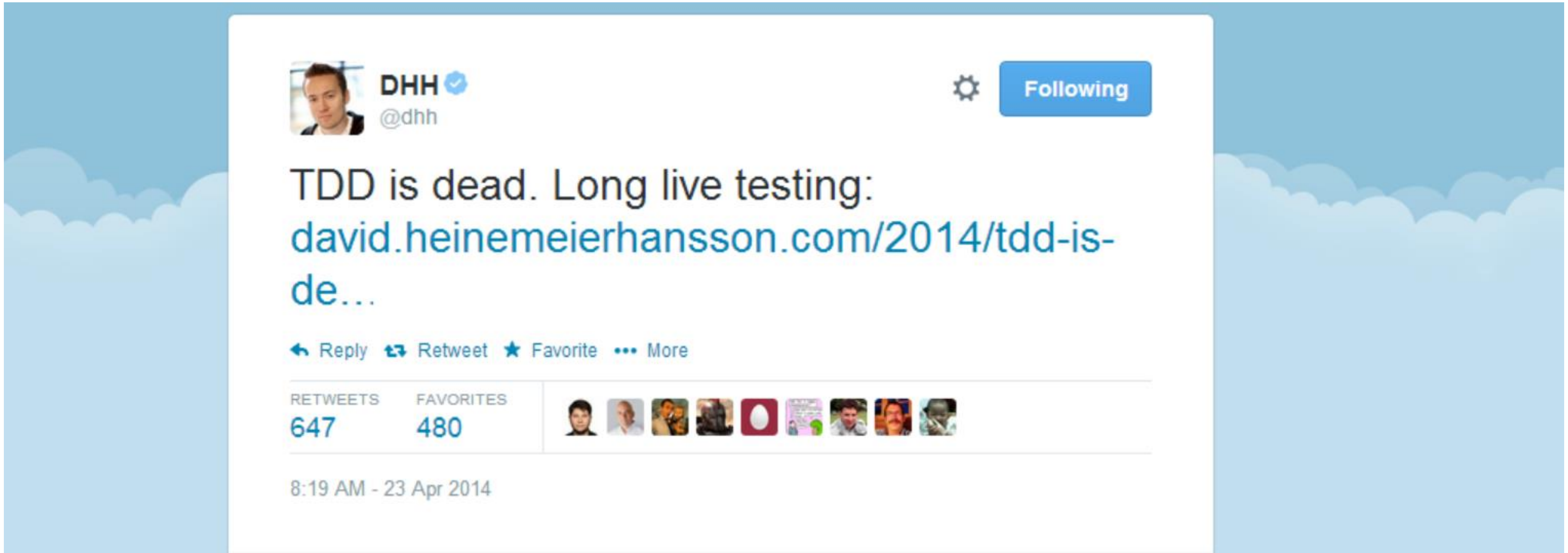
Value of Unit Tests

- Tests help us catch mistakes
- Gives you confidence to make changes
- Built in regression testing
- Increases life of codebase

Intro
Android Defects

- Application Installation Fails
- Application crashes during execution
- Scaling/Layout Problems
- Application hangs if some resources are not available
- Problem in landscape/portrait mode

Intro
Alternatives to Unit Testing



DHH
@dhh

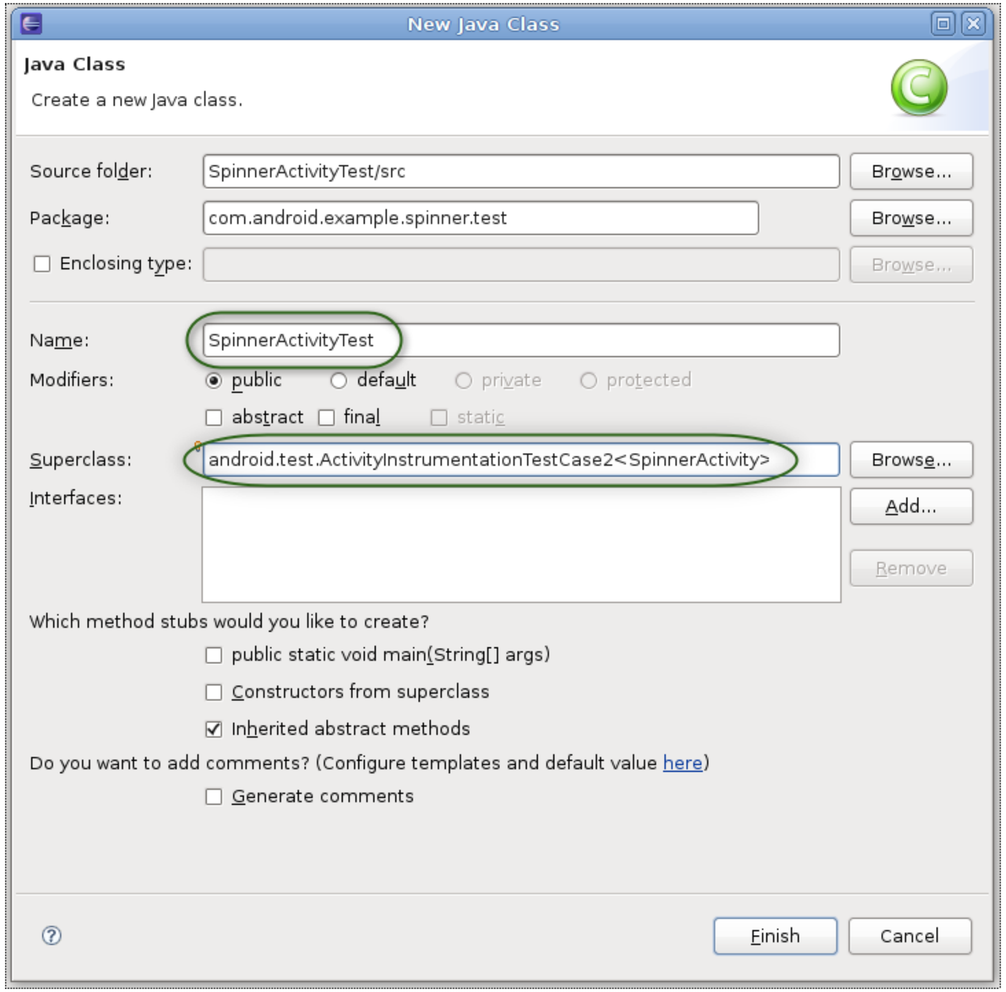
TDD is dead. Long live testing: david.heinemeierhansson.com/2014/tdd-is-de…

RETWEETS 647

FAVORITES 480

8:19 AM - 23 Apr 2014

Writing Android Unit Tests
Android Testing Framework



New Java Class

Java Class

Create a new Java class.

Source folder: SpinnerActivityTest/src

Package: com.android.example.spinner.test

□ Enclosing type:

Name: SpinnerActivityTest

Modifiers: ● public ○ default ○ private ○ protected

□ abstract □ final □ static

Superclass: android.test.ActivityInstrumentationTestCase2<SpinnerActivity>

Interfaces:

Which method stubs would you like to create?

□ public static void main(String[] args)

□ Constructors from superclass

☑ Inherited abstract methods

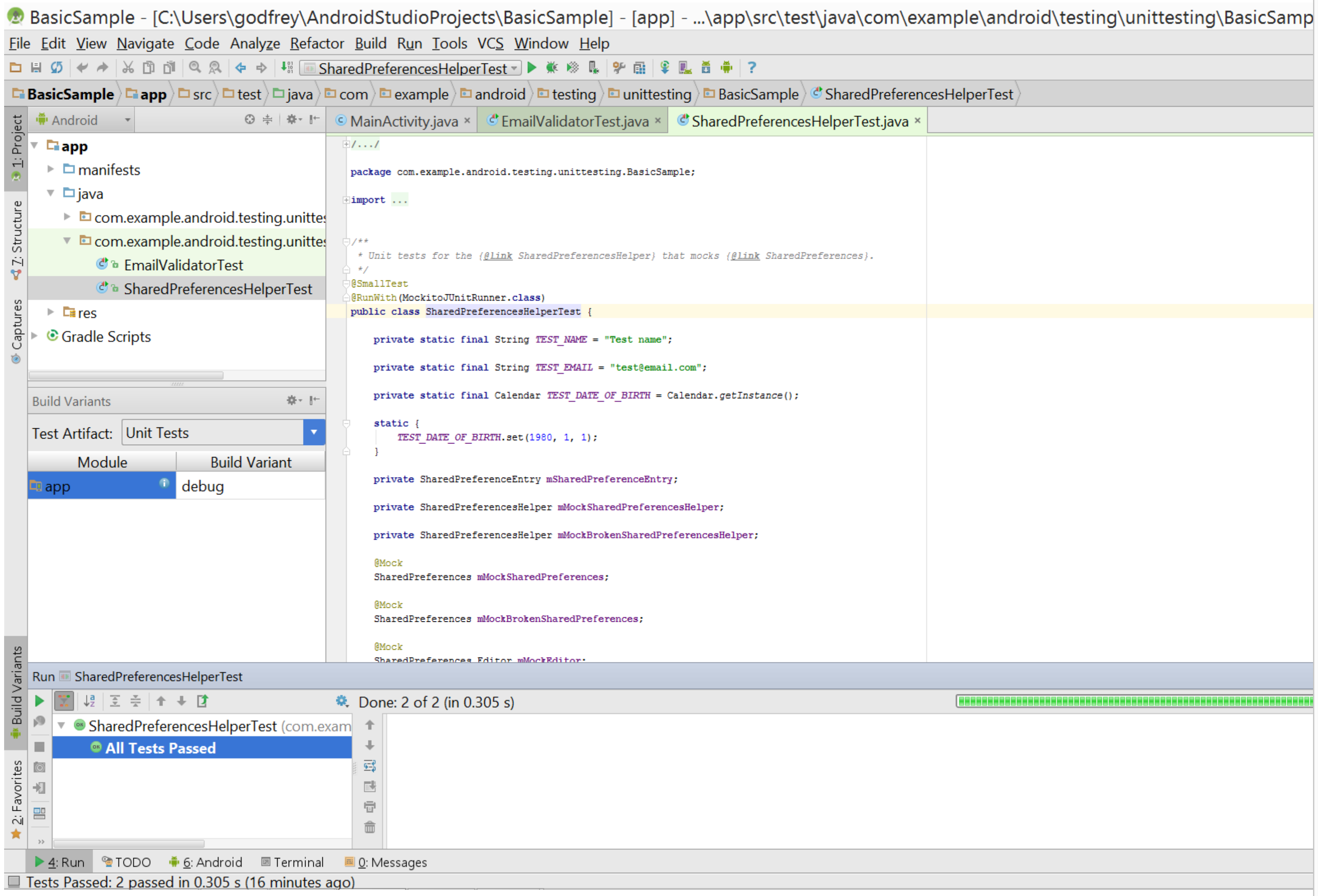
Do you want to add comments? (Configure templates and default value here)

□ Generate comments

Writing Android Unit Tests
Android Testing Framework

Writing Android Unit Tests
Android Unit Tests

Writing Android Unit Tests
Hello World



BasicSample - [C:\Users\godfrey\AndroidStudioProjects\BasicSample] - [app] - ...\app\src\test\java\com\example\android\testing\unittesting\BasicSamp

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
SharedPreferencesHelperTest
BasicSample app src test java com example android testing unittesting BasicSample SharedPreferencesHelperTest
Android
app
manifests
java
com.example.android.testing.unittes
com.example.android.testing.unittes
EmailValidatorTest
SharedPreferencesHelperTest
res
Gradle Scripts
1: Project
7: Structure
Captures
Build Variants
2: Favorites
MainActivity.java
EmailValidatorTest.java
SharedPreferencesHelperTest.java

Build Variants

Test Artifact: Unit Tests

Module	Build Variant
app	debug

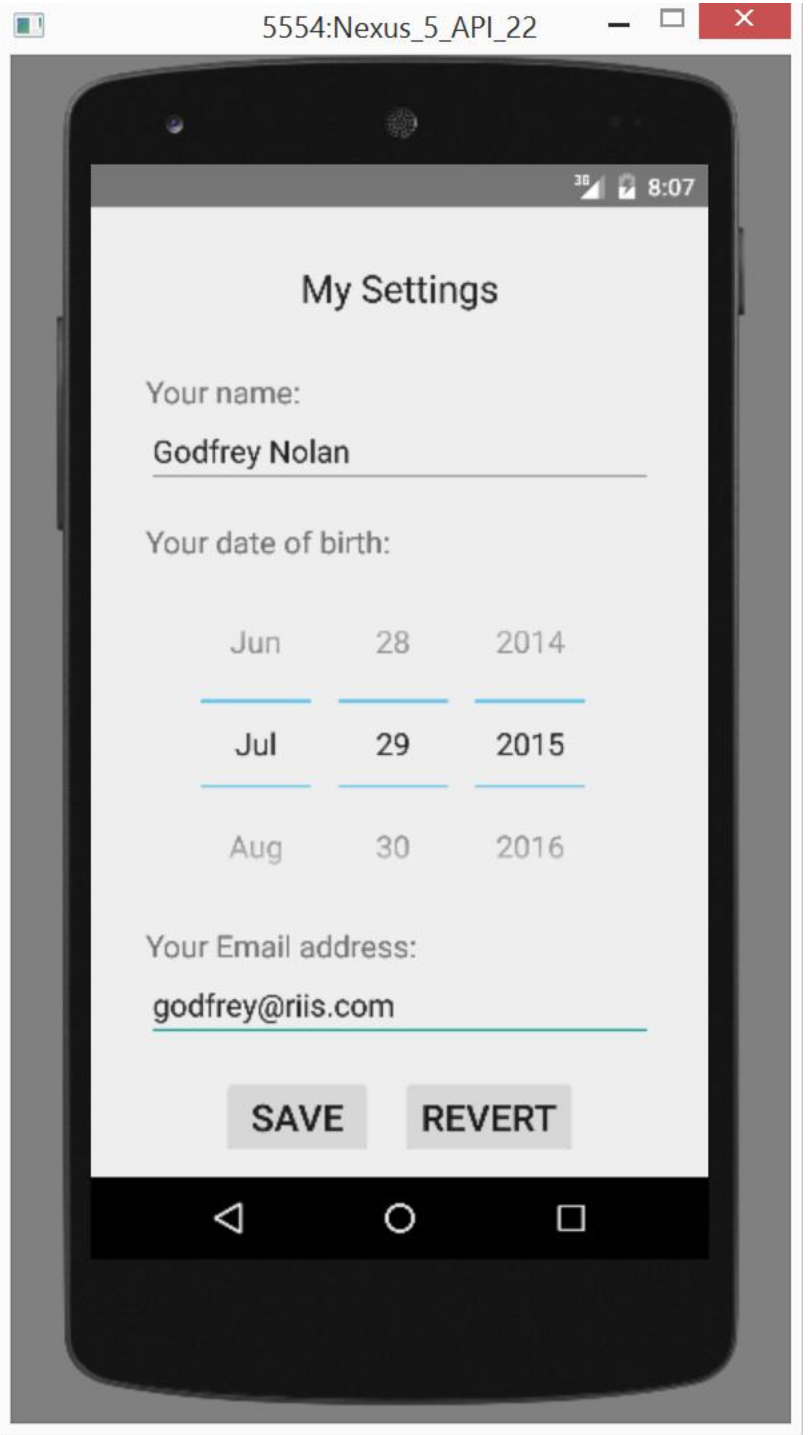
```
/.../

package com.example.android.testing.unittesting.BasicSample;

import ...


/**
 * Unit tests for the {@link SharedPreferencesHelper} that mocks {@link SharedPreferences}.
 */
@SmallTest
@RunWith(MockitoJUnitRunner.class)
public class SharedPreferencesHelperTest {

    private static final String TEST_NAME = "Test name";

    private static final String TEST_EMAIL = "test@email.com";

    private static final Calendar TEST_DATE_OF_BIRTH = Calendar.getInstance();

    static {
        TEST_DATE_OF_BIRTH.set(1980, 1, 1);
    }

    private SharedPreferenceEntry mSharedPreferenceEntry;

    private SharedPreferencesHelper mMockSharedPreferencesHelper;

    private SharedPreferencesHelper mMockBrokenSharedPreferencesHelper;

    @Mock
    SharedPreferences mMockSharedPreferences;

    @Mock
    SharedPreferences mMockBrokenSharedPreferences;

    @Mock
    SharedPreferences.Editor mMockEditor;
```

Run SharedPreferencesHelperTest

Done: 2 of 2 (in 0.305 s)

```
SharedPreferencesHelperTest (com.exam
    All Tests Passed
```

Tests Passed: 2 passed in 0.305 s (16 minutes ago)



5554:Nexus_5_API_22

My Settings

Your name:

Godfrey Nolan

Your date of birth:

Jun	28	2014
Jul	29	2015
Aug	30	2016

Your Email address:

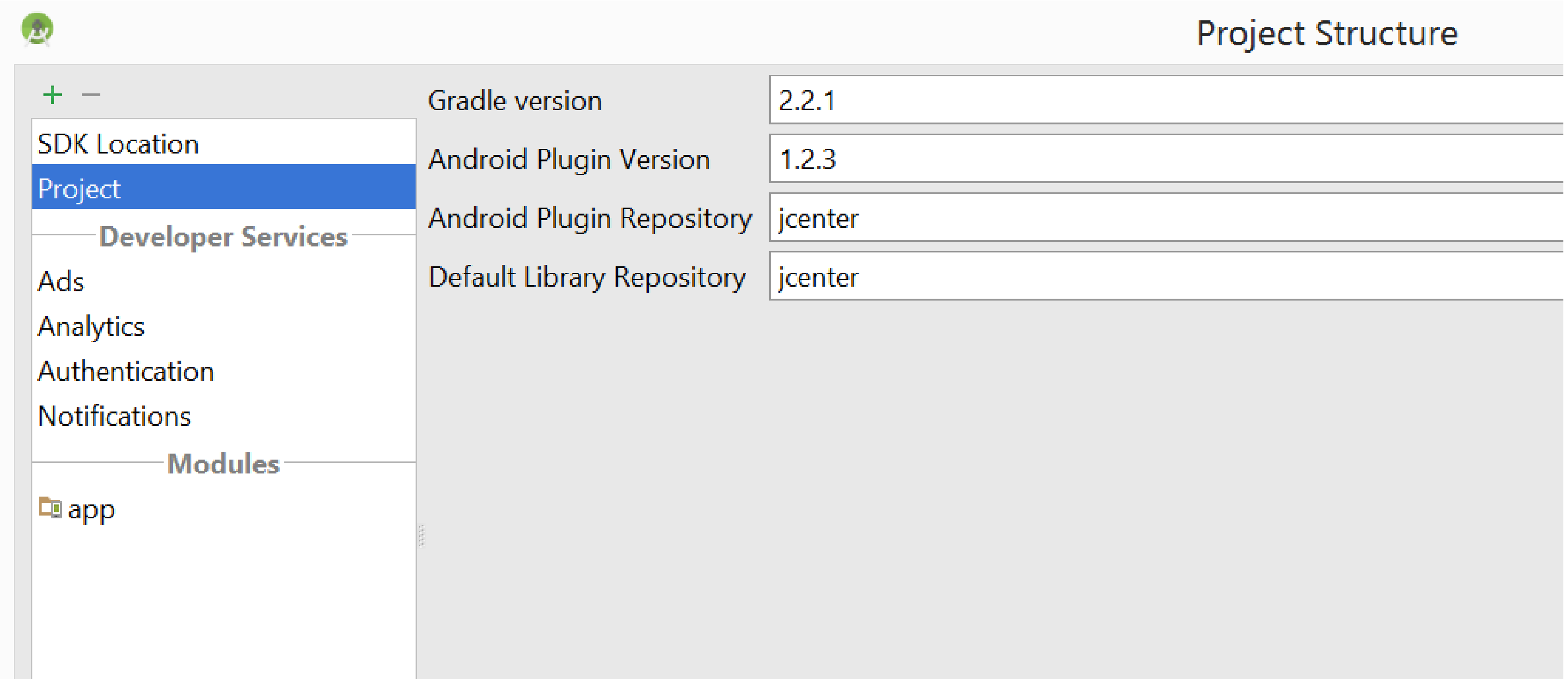
godfrey@riis.com

Writing Android Unit Tests
Hello World

- Prerequisites - Android Plug-in for Gradle version 1.1.x
- Create the src/test/java folders
- Add junit:4:12 dependency in build.gradle (app) file
- Create unit tests
- Choose Unit Tests' Test Artifact in Build Variant
- Right click on tests to run tests

Writing Android Unit Tests
Hello World

- Prerequisites - Android Plug-in for Gradle version 1.1.x



Gradle version: 2.2.1

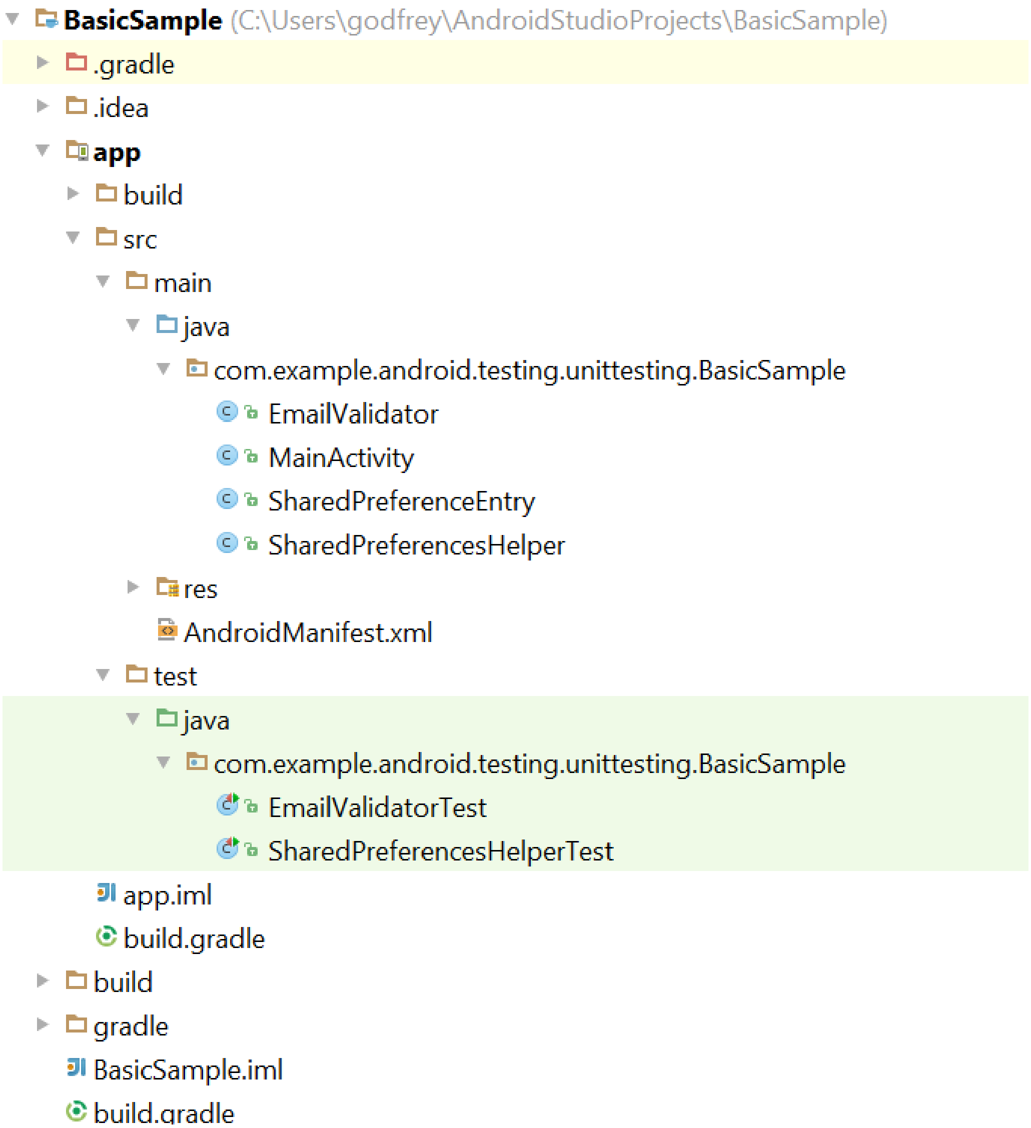
Android Plugin Version: 1.2.3

Android Plugin Repository: jcenter

Default Library Repository: jcenter

Writing Android Unit Tests
Hello World

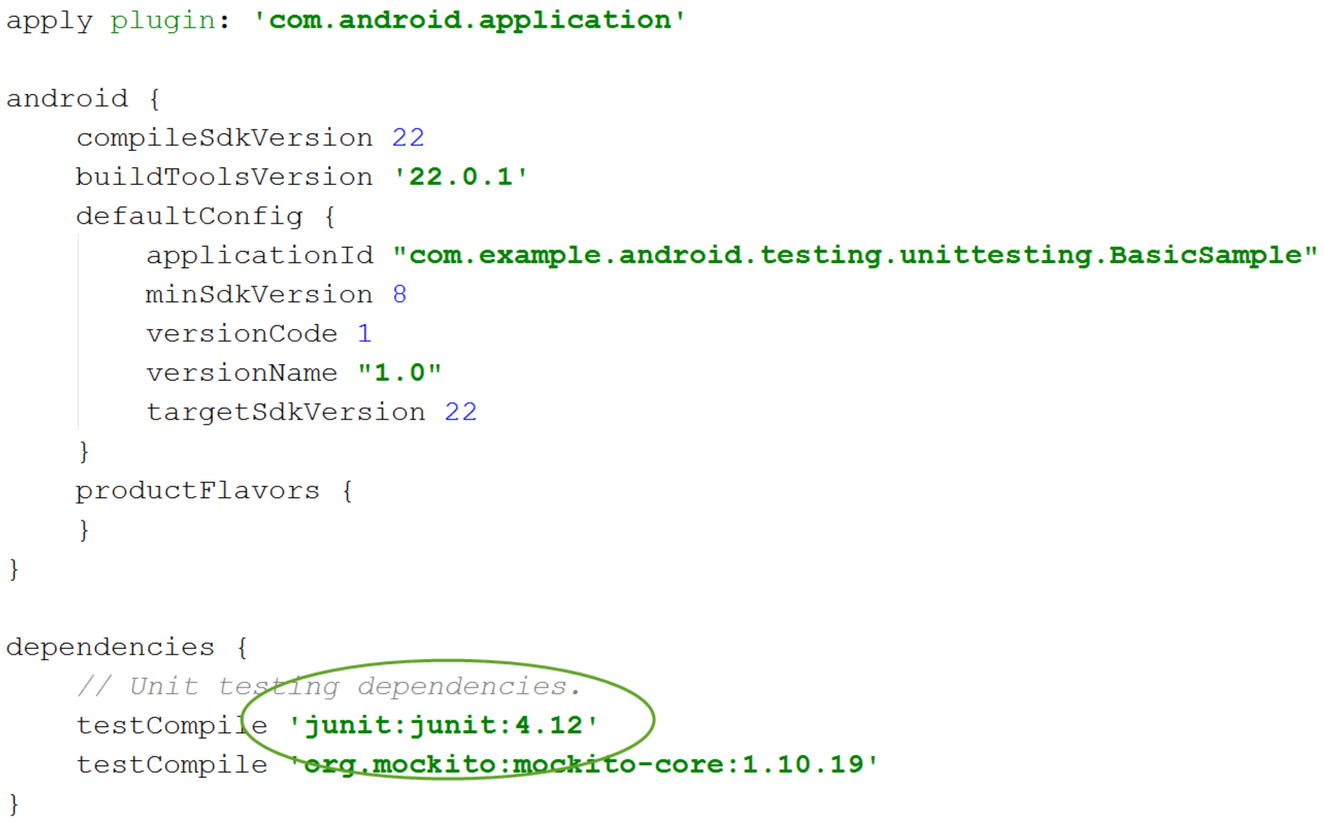
- ○ Create the src/test/java folders (change to Project view)



```
▼ BasicSample (C:\Users\godfrey\AndroidStudioProjects\BasicSample)
  ▶ .gradle
  ▶ .idea
  ▼ app
    ▶ build
    ▼ src
      ▼ main
        ▼ java
          ▼ com.example.android.testing.unittesting.BasicSample
              EmailValidator
              MainActivity
              SharedPreferenceEntry
              SharedPreferencesHelper
        ▶ res
          AndroidManifest.xml
      ▼ test
        ▼ java
          ▼ com.example.android.testing.unittesting.BasicSample
              EmailValidatorTest
              SharedPreferencesHelperTest
      app.iml
      build.gradle
  ▶ build
  ▶ gradle
    BasicSample.iml
    build.gradle
```

Writing Android Unit Tests
Hello World

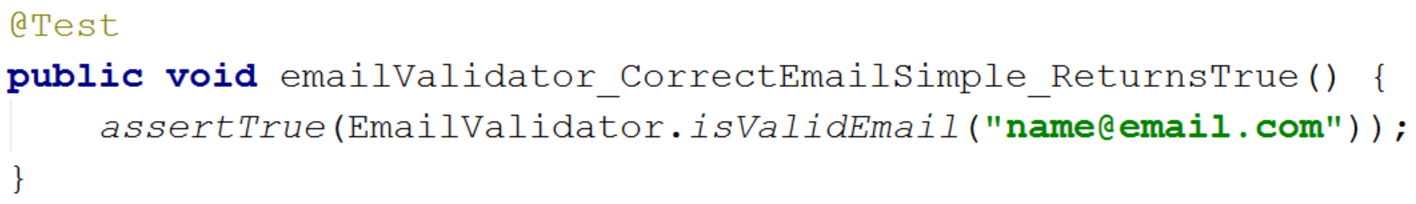
- Add junit:4:12 dependency in build.gradle (app) file



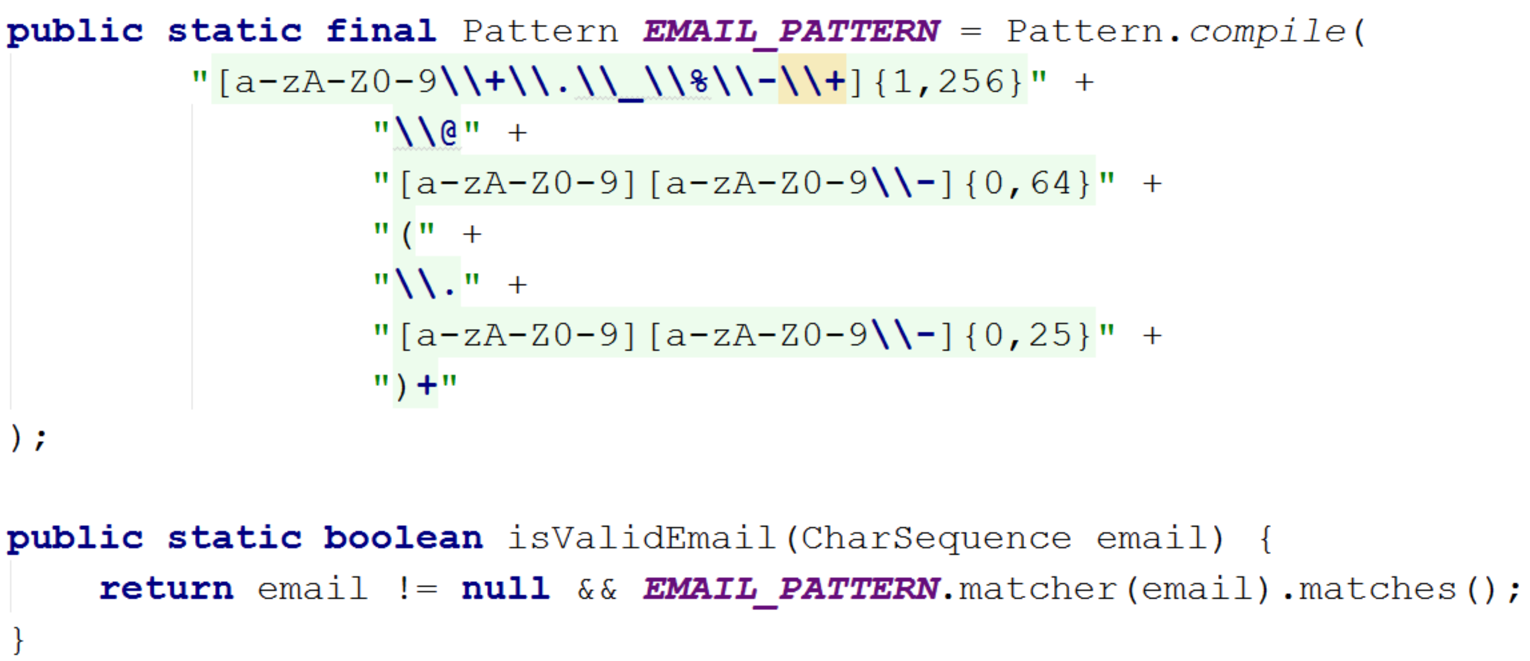
```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 22
    buildToolsVersion '22.0.1'
    defaultConfig {
        applicationId "com.example.android.testing.unittesting.BasicSample"
        minSdkVersion 8
        versionCode 1
        versionName "1.0"
        targetSdkVersion 22
    }
    productFlavors {
    }
}

dependencies {
    // Unit testing dependencies.
    testCompile 'junit:junit:4.12'
    testCompile 'org.mockito:mockito-core:1.10.19'
}
```

Writing an Android Unit Test
Hello World

- Create unit tests



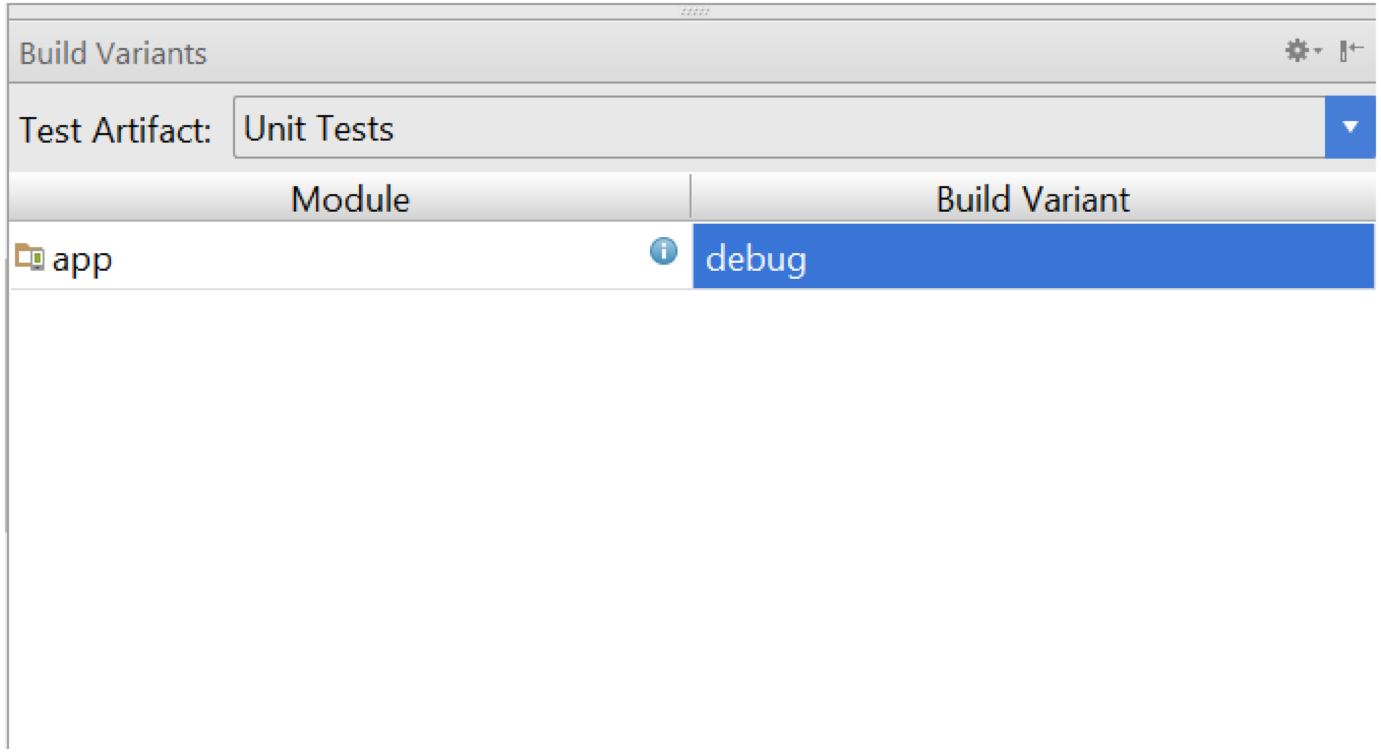
```
@Test
public void emailValidator_CorrectEmailSimple_ReturnsTrue() {
    assertTrue(EmailValidator.isValidEmail("name@email.com"));
}
```



```
public static final Pattern EMAIL_PATTERN = Pattern.compile(
        "[a-zA-Z0-9\\+\\.\\_\\%\\-\\+]{1,256}" +
                "\\@" +
                "[a-zA-Z0-9][a-zA-Z0-9\\-]{0,64}" +
                "(" +
                "\\." +
                "[a-zA-Z0-9][a-zA-Z0-9\\-]{0,25}" +
                ")+"
);

public static boolean isValidEmail(CharSequence email) {
    return email != null && EMAIL_PATTERN.matcher(email).matches();
}
```

Writing Android Unit Tests
Hello World

- Choose Unit Tests' Test Artifact in Build Variant



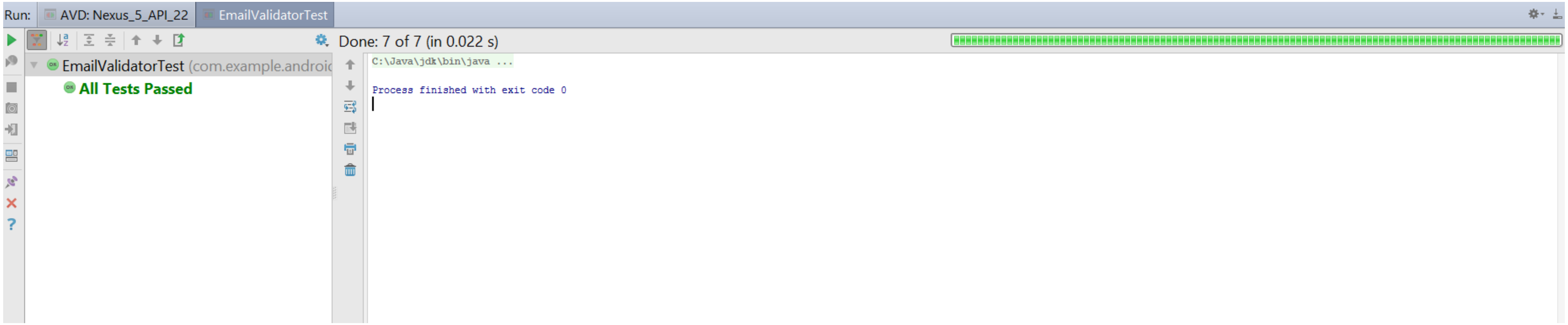
Build Variants

Test Artifact: Unit Tests

Module	Build Variant
app	debug

Writing Android Unit Tests
Hello World

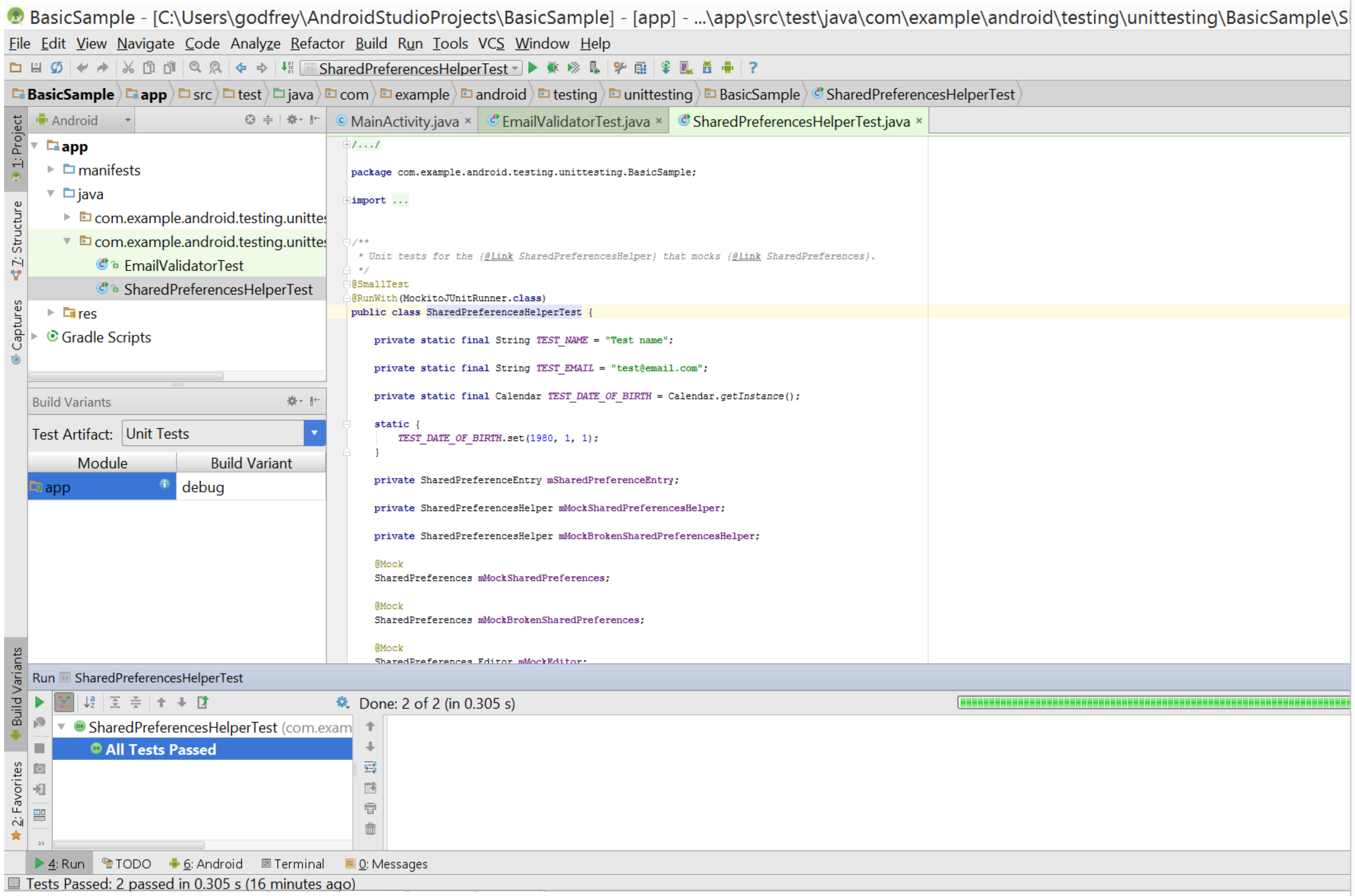
- Right click on test file to run tests



Done: 7 of 7 (in 0.022 s)

```
C:\Java\jdk\bin\java ...

Process finished with exit code 0
```

Writing Android Unit Tests
Hello World



BasicSample - [C:\Users\godfrey\AndroidStudioProjects\BasicSample] - [app] - ...\app\src\test\java\com\example\android\testing\unittesting\BasicSample\S

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
SharedPreferencesHelperTest
BasicSample app src test java com example android testing unittesting BasicSample SharedPreferencesHelperTest
1: Project
Z: Structure
Captures
Build Variants
2: Favorites
Android
app
manifests
java
com.example.android.testing.unittes
com.example.android.testing.unittes
EmailValidatorTest
SharedPreferencesHelperTest
res
Gradle Scripts
MainActivity.java
EmailValidatorTest.java
SharedPreferencesHelperTest.java
Build Variants

Test Artifact: Unit Tests

Module	Build Variant
app	debug

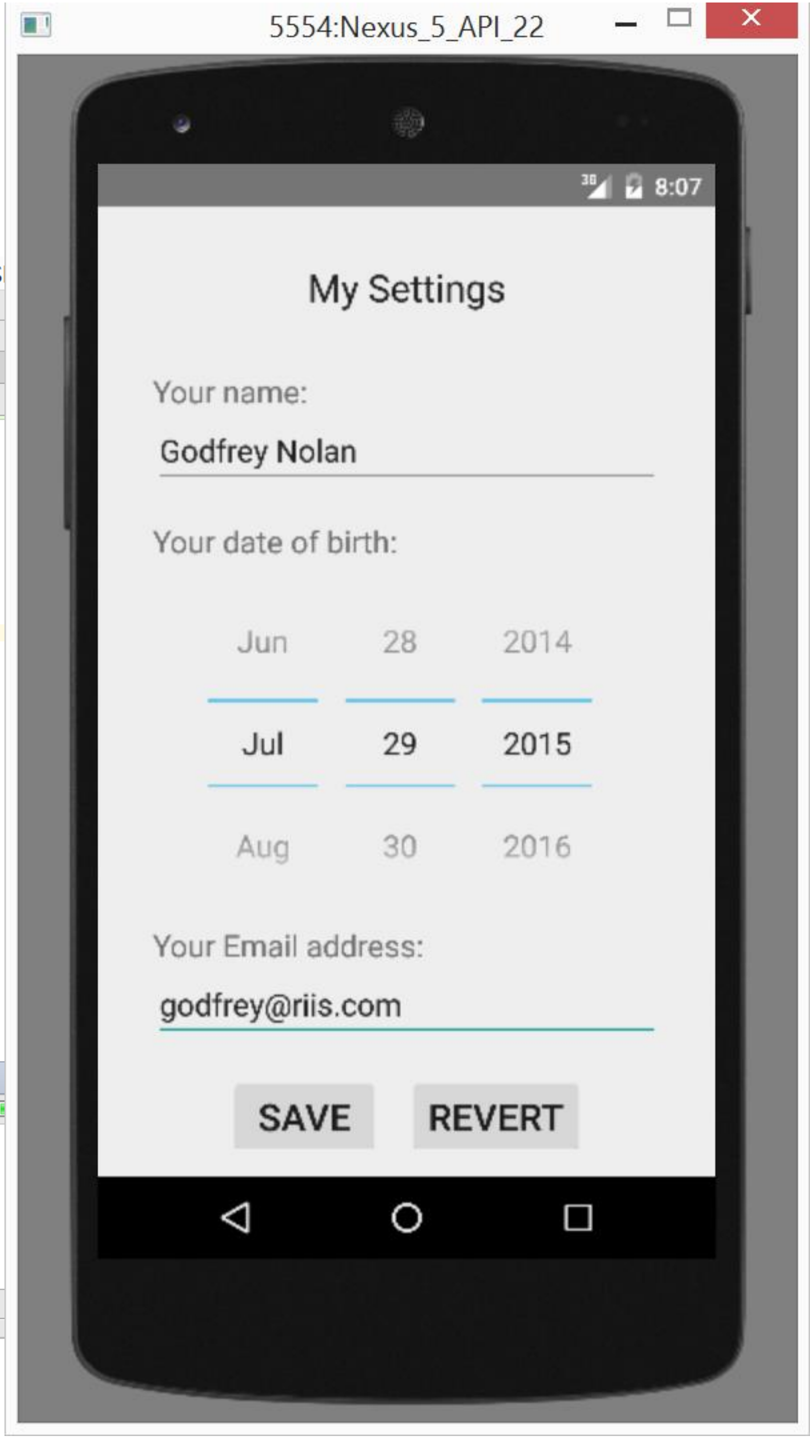
```
/.../

package com.example.android.testing.unittesting.BasicSample;

import ...


/**
 * Unit tests for the {@link SharedPreferencesHelper} that mocks {@link SharedPreferences}.
 */
@SmallTest
@RunWith(MockitoJUnitRunner.class)
public class SharedPreferencesHelperTest {

    private static final String TEST_NAME = "Test name";

    private static final String TEST_EMAIL = "test@email.com";

    private static final Calendar TEST_DATE_OF_BIRTH = Calendar.getInstance();

    static {
        TEST_DATE_OF_BIRTH.set(1980, 1, 1);
    }

    private SharedPreferenceEntry mSharedPreferenceEntry;

    private SharedPreferencesHelper mMockSharedPreferencesHelper;

    private SharedPreferencesHelper mMockBrokenSharedPreferencesHelper;

    @Mock
    SharedPreferences mMockSharedPreferences;

    @Mock
    SharedPreferences mMockBrokenSharedPreferences;

    @Mock
    SharedPreferences.Editor mMockEditor;
```

Run SharedPreferencesHelperTest

Done: 2 of 2 (in 0.305 s)

```
SharedPreferencesHelperTest (com.exam
    All Tests Passed
```

Tests Passed: 2 passed in 0.305 s (16 minutes ago)



My Settings

Your name:

Godfrey Nolan

Your date of birth:

Jun	28	2014
Jul	29	2015
Aug	30	2016

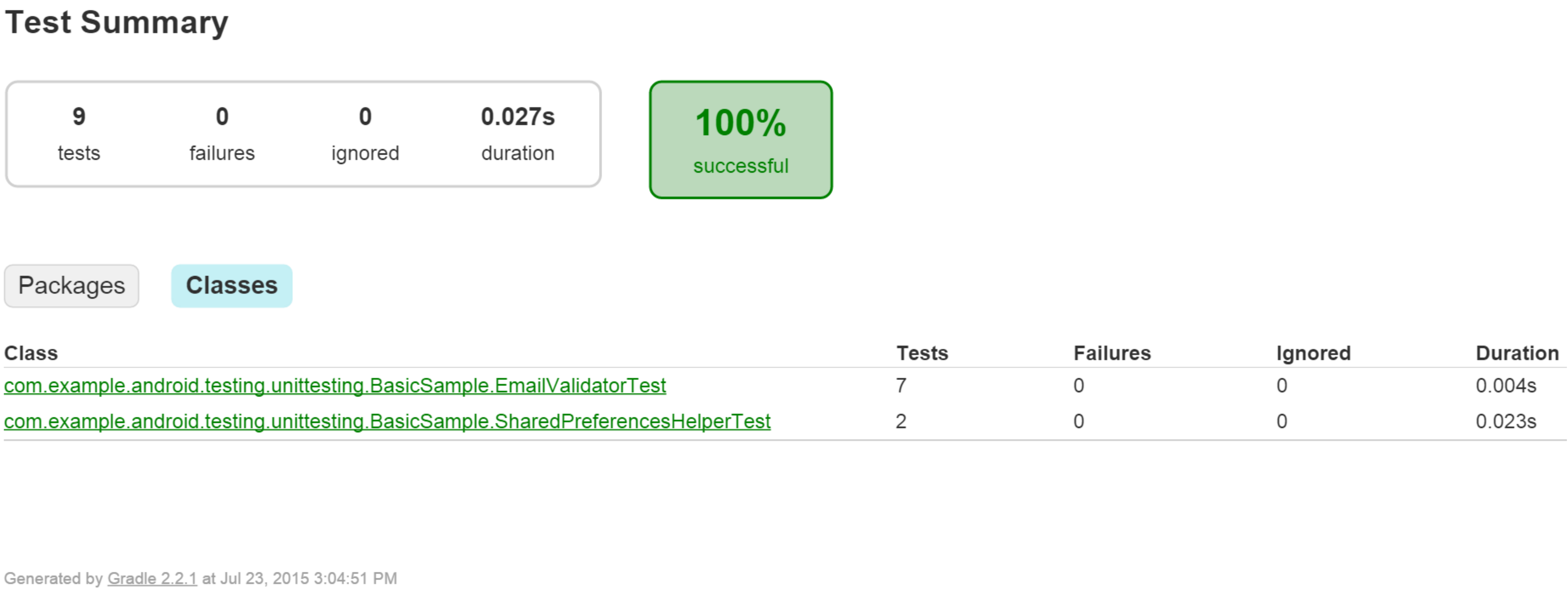
Your Email address:

godfrey@riis.com

Writing Android Unit Tests
Command Line

```
C:\Users\godfrey\AndroidStudioProjects\BasicSample>gradlew test --continue
Downloading https://services.gradle.org/distributions/gradle-2.2.1-all.zip
................................................................................
...................................................
Unzipping C:\Users\godfrey\.gradle\wrapper\dists\gradle-2.2.1-all\6dibv5rcnnqlfb
q9klf8imrndn\gradle-2.2.1-all.zip to C:\Users\godfrey\.gradle\wrapper\dists\grad
le-2.2.1-all\6dibv5rcnnqlfbq9klf8imrndn
Download https://jcenter.bintray.com/com/google/guava/guava/17.0/guava-17.0.jar
Download https://jcenter.bintray.com/com/android/tools/lint/lint-api/24.2.3/lint-api-24.2.3.jar
Download https://jcenter.bintray.com/org/ow2/asm/asm-analysis/5.0.3/asm-analysis-5.0.3.jar
Download https://jcenter.bintray.com/com/android/tools/external/lombok/lombok-ast/0.2.3/lombok-ast-0.2.3.jar
:app:preBuild UP-TO-DATE
:app:preDebugBuild UP-TO-DATE
:app:checkDebugManifest
:app:prepareDebugDependencies
:app:compileDebugAidl
:app:compileDebugRenderscript
.
.
.
:app:compileReleaseUnitTestSources
:app:assembleReleaseUnitTest
:app:testRelease
:app:test

BUILD SUCCESSFUL

Total time: 3 mins 57.013 secs
```

Writing Android Unit Tests
HTML output



Test Summary

9
tests

0
failures

0
ignored

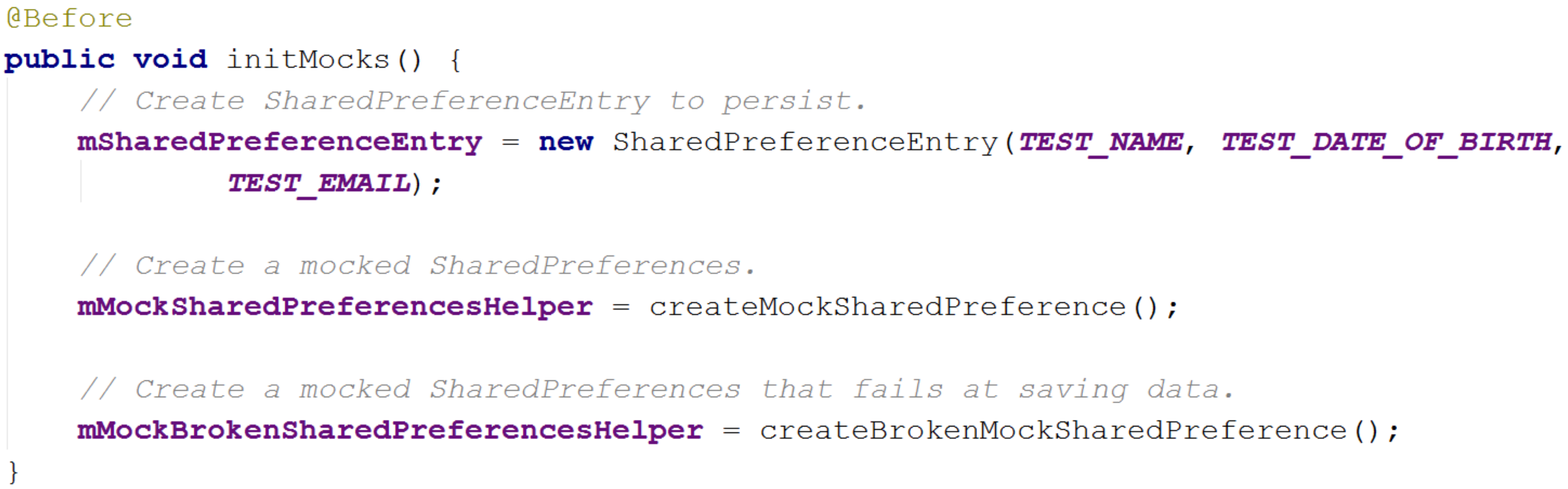
0.027s
duration

100%
successful

Class	Tests	Failures	Ignored	Duration
com.example.android.testing.unittesting.BasicSample.EmailValidatorTest	7	0	0	0.004s
com.example.android.testing.unittesting.BasicSample.SharedPreferencesHelperTest	2	0	0	0.023s

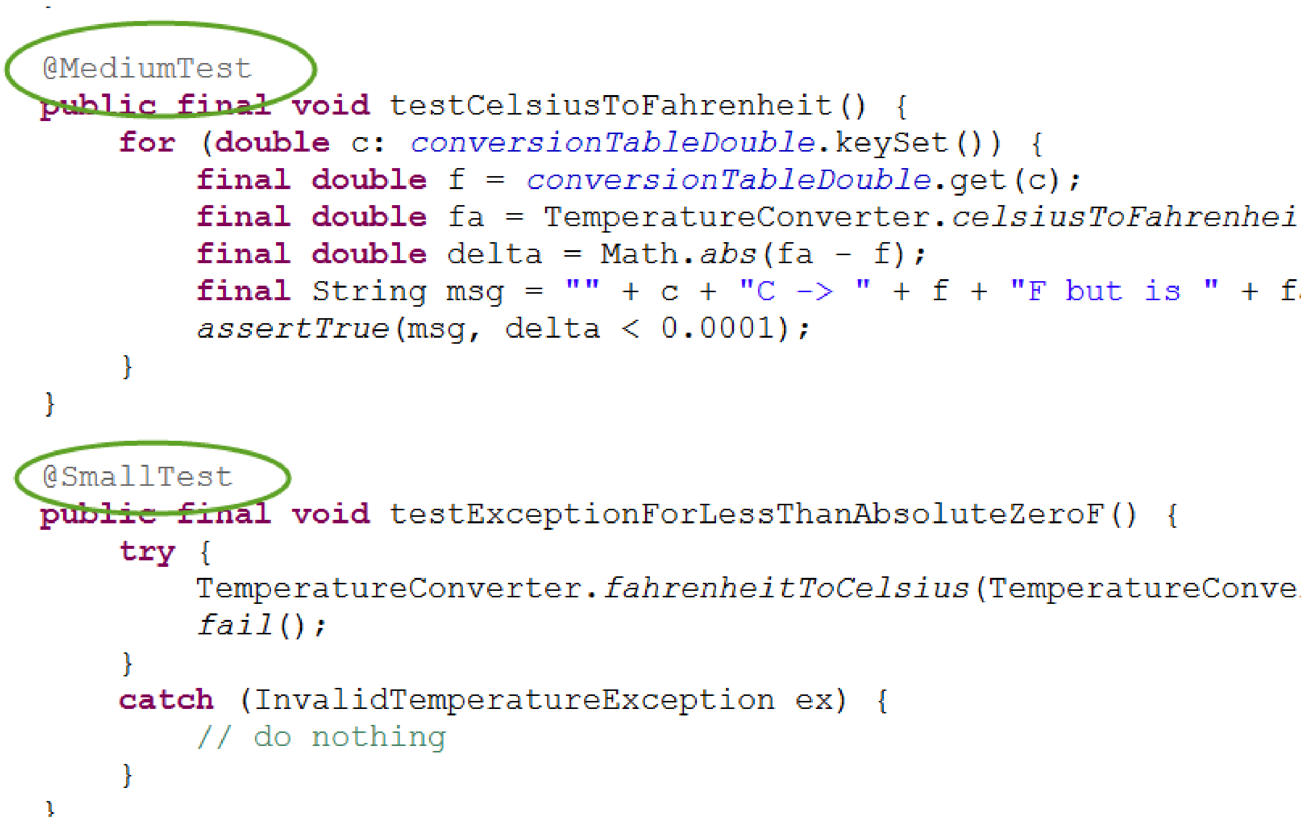
Generated by Gradle 2.2.1 at Jul 23, 2015 3:04:51 PM

Writing Android Unit Tests
Setup and Teardown

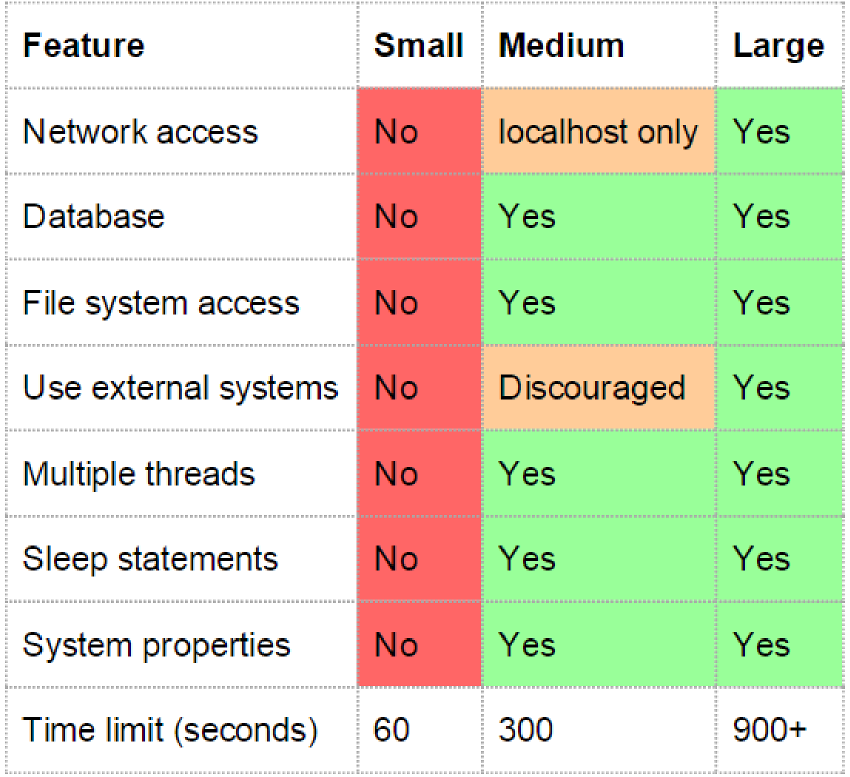


```
@Before
public void initMocks() {
    // Create SharedPreferenceEntry to persist.
    mSharedPreferenceEntry = new SharedPreferenceEntry(TEST_NAME, TEST_DATE_OF_BIRTH,
            TEST_EMAIL);

    // Create a mocked SharedPreferences.
    mMockSharedPreferencesHelper = createMockSharedPreference();

    // Create a mocked SharedPreferences that fails at saving data.
    mMockBrokenSharedPreferencesHelper = createBrokenMockSharedPreference();
}
```

Writing Android Unit Tests
Grouping Tests



```
@MediumTest
public final void testCelsiusToFahrenheit() {
    for (double c: conversionTableDouble.keySet()) {
        final double f = conversionTableDouble.get(c);
        final double fa = TemperatureConverter.celsiusToFahrenhei
        final double delta = Math.abs(fa - f);
        final String msg = "" + c + "C -> " + f + "F but is " + f
        assertTrue(msg, delta < 0.0001);
    }
}

@SmallTest
public final void testExceptionForLessThanAbsoluteZeroF() {
    try {
        TemperatureConverter.fahrenheitToCelsius(TemperatureConve
        fail();
    }
    catch (InvalidTemperatureException ex) {
        // do nothing
    }
}
```



Feature	Small	Medium	Large
Network access	No	localhost only	Yes
Database	No	Yes	Yes
File system access	No	Yes	Yes
Use external systems	No	Discouraged	Yes
Multiple threads	No	Yes	Yes
Sleep statements	No	Yes	Yes
System properties	No	Yes	Yes
Time limit (seconds)	60	300	900+

Writing Android Unit Tests
Assertions

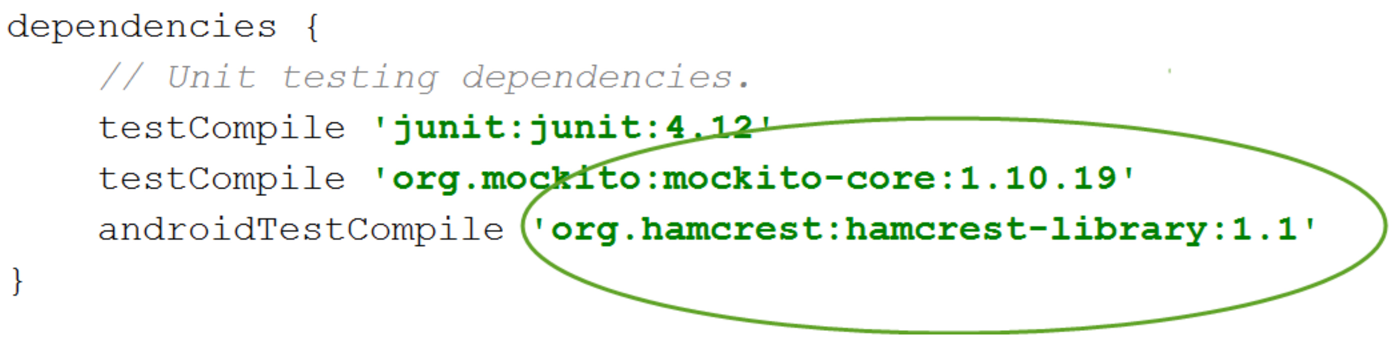
- assertEquals
- assertFalse
- assertNotNull
- assertNotSame
- assertNull
- assertSame
- assertTrue
- fail

- assertBaselineAligned
- assertBottomAligned
- assertGroupContains
- assertGroupIntegrity
- assertGroupNotContains
- assertHasScreenCoordinates
- assertHorizontalCenterAligned
- assertLeftAligned
- assertOffScreenAbove
- assertOffScreenBelow
- assertOnScreen
- assertRightAligned
- assertTopAligned
- assertVerticalCenterAligned

- assertAssignableFrom
- assertContainsRegex
- assertContainsInAnyOrder
- assertContainsInOrder
- assertEmpty
- assertMatchesRegex
- assertNotContainsRegex
- assertNotEmpty
- assertNotMatchesRegex
- checkEqualsAndHasCodeMethods

Writing Android Unit Tests
More Assertions - Hamcrest

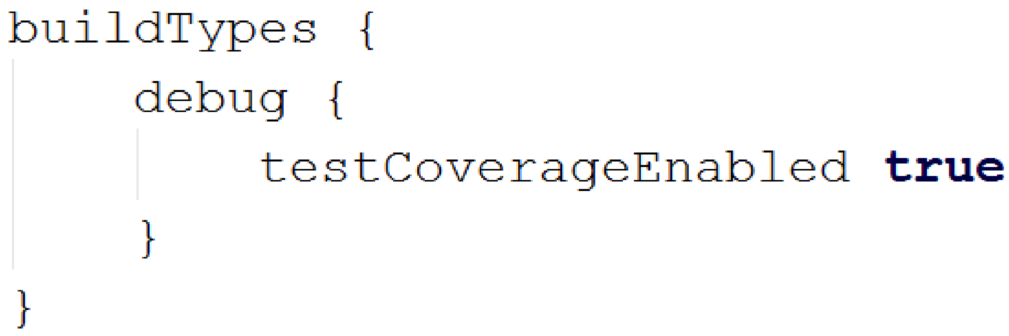
- assertThat, is, anything etc.
- Custom Matchers
- add lib to dependencies



```
dependencies {
    // Unit testing dependencies.
    testCompile 'junit:junit:4.12'
    testCompile 'org.mockito:mockito-core:1.10.19'
    androidTestCompile 'org.hamcrest:hamcrest-library:1.1'
}
```

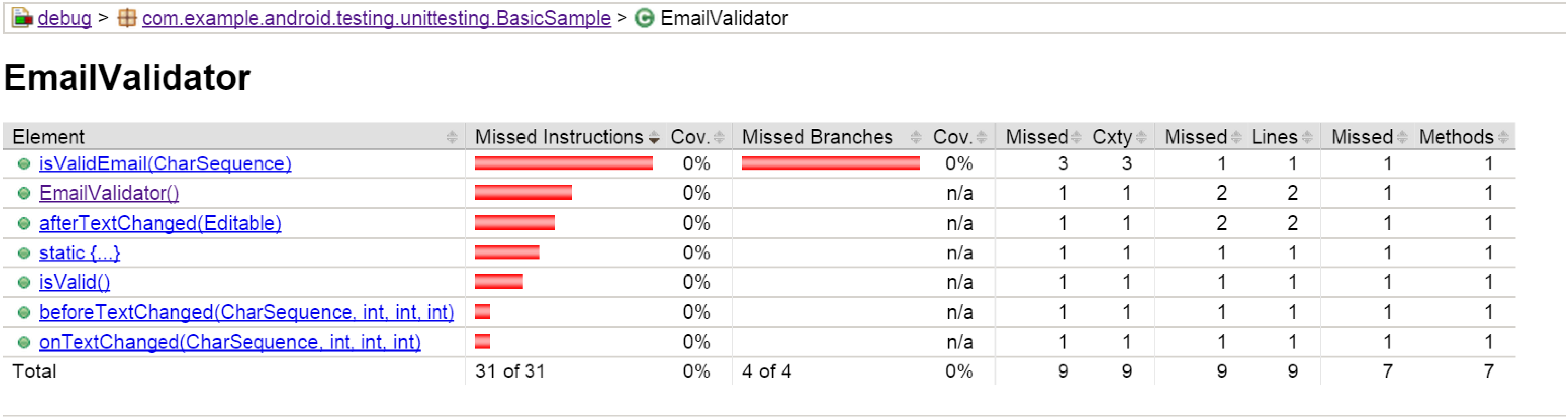
Other tools
Code Coverage

- Gradle includes jacoco for code coverage
- Add testCoverageEnabled = true in debug build
- run from command line using gradlew createDebugCoverageReport
- coverage reports are in build/reports/coverage/debug



```
buildTypes {
    debug {
        testCoverageEnabled true
    }
}
```

Other tools
Code Coverage

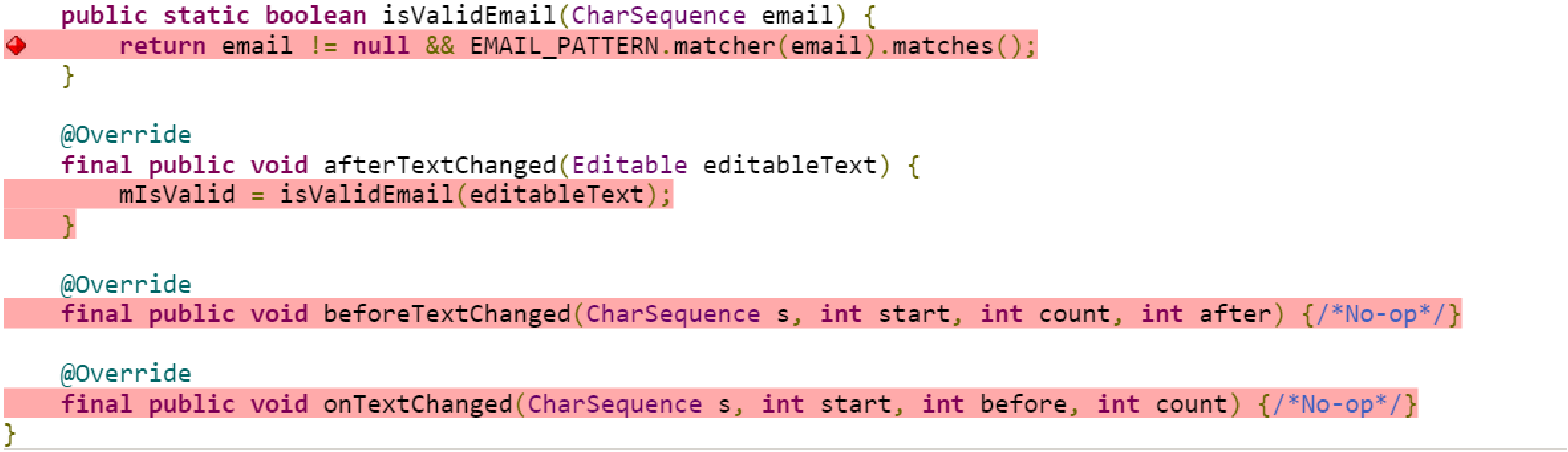


debug > com.example.android.testing.unittesting.BasicSample > EmailValidator

EmailValidator

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
isValidEmail(CharSequence)	🟥	0%	🟥	0%	3	3	1	1	1	1
EmailValidator()	🟥	0%		n/a	1	1	2	2	1	1
afterTextChanged(Editable)	🟥	0%		n/a	1	1	2	2	1	1
static {...}	🟥	0%		n/a	1	1	1	1	1	1
isValid()	🟥	0%		n/a	1	1	1	1	1	1
beforeTextChanged(CharSequence, int, int, int)	🟥	0%		n/a	1	1	1	1	1	1
onTextChanged(CharSequence, int, int, int)	🟥	0%		n/a	1	1	1	1	1	1
Total	31 of 31	0%	4 of 4	0%	9	9	9	9	7	7

Other tools
Code Coverage



```java
    public static boolean isValidEmail(CharSequence email) {
        return email != null && EMAIL_PATTERN.matcher(email).matches();
    }

    @Override
    final public void afterTextChanged(Editable editableText) {
        mIsValid = isValidEmail(editableText);
    }

    @Override
    final public void beforeTextChanged(CharSequence s, int start, int count, int after) {/*No-op*/}

    @Override
    final public void onTextChanged(CharSequence s, int start, int before, int count) {/*No-op*/}
}
```

Other Tools
Dagger

○ Dependency Injection

○ Compile time not runtime

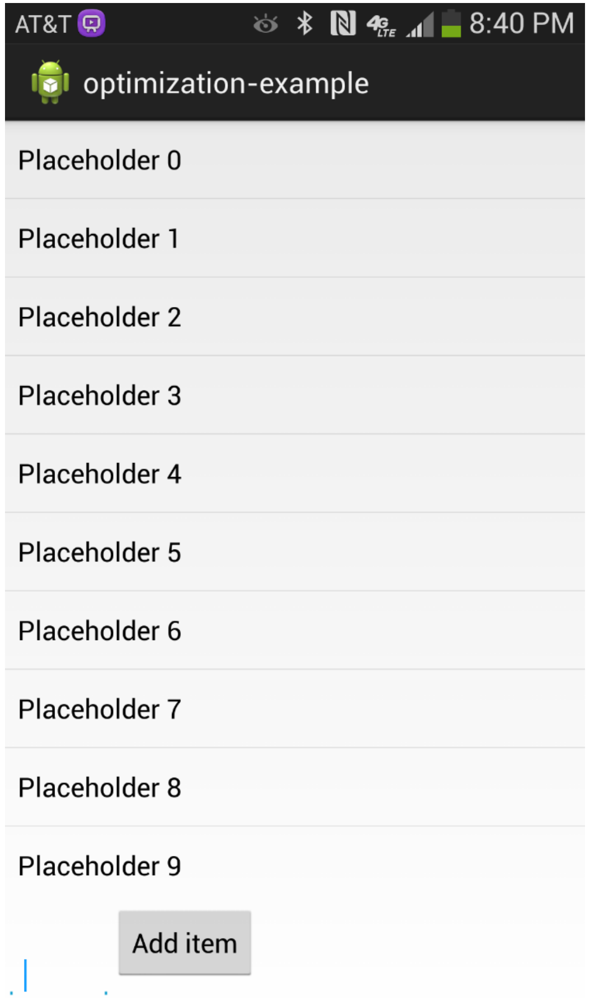
○ Used for mocking and/or decoupling Android

○ Example

  Mock out Data Providers in ToDoList

  Replace SQLite database with fake stub

Other Tools
Dagger - ToDoList



optimization-example

Placeholder 0

Placeholder 1

Placeholder 2

Placeholder 3

Placeholder 4

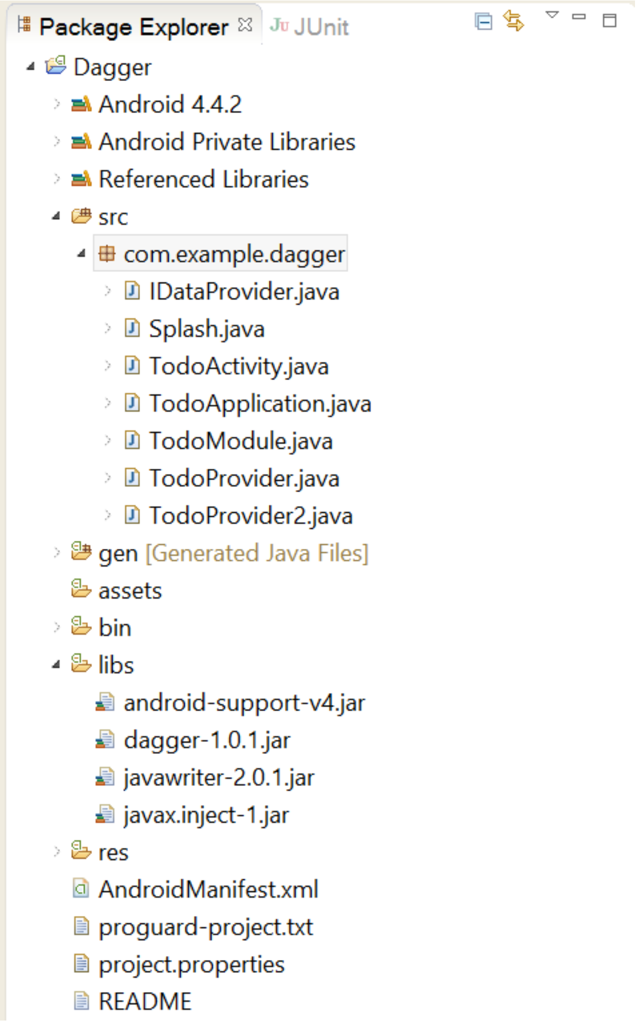
Placeholder 5

Placeholder 6

Placeholder 7

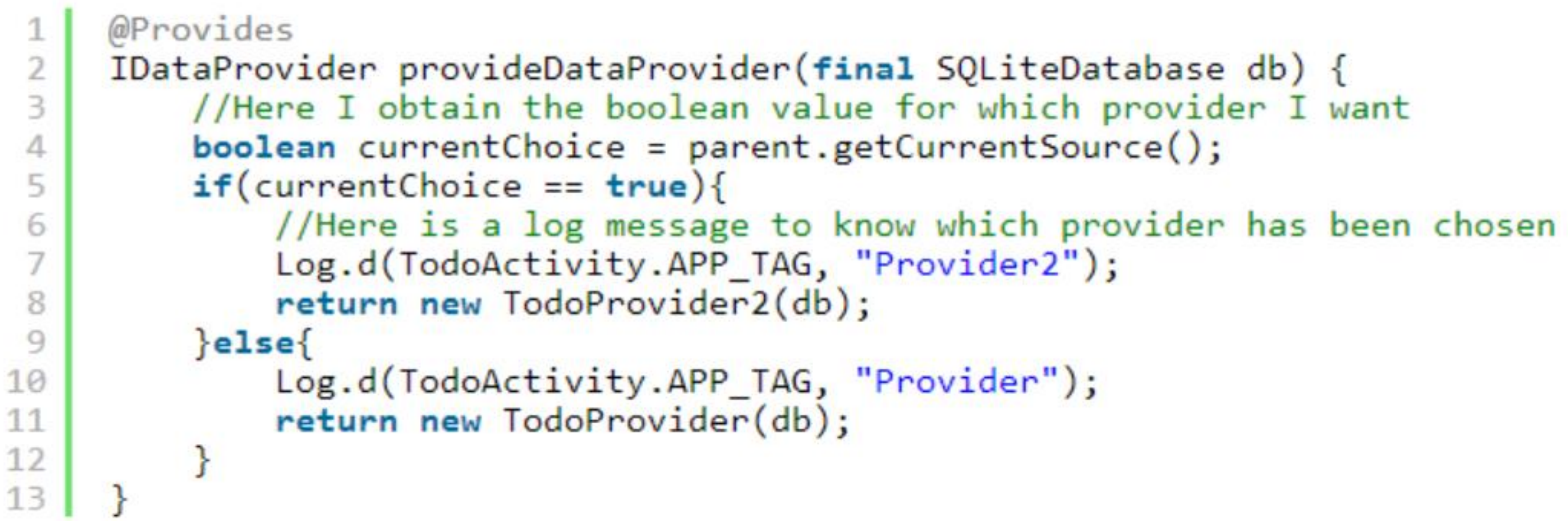
Placeholder 8

Placeholder 9



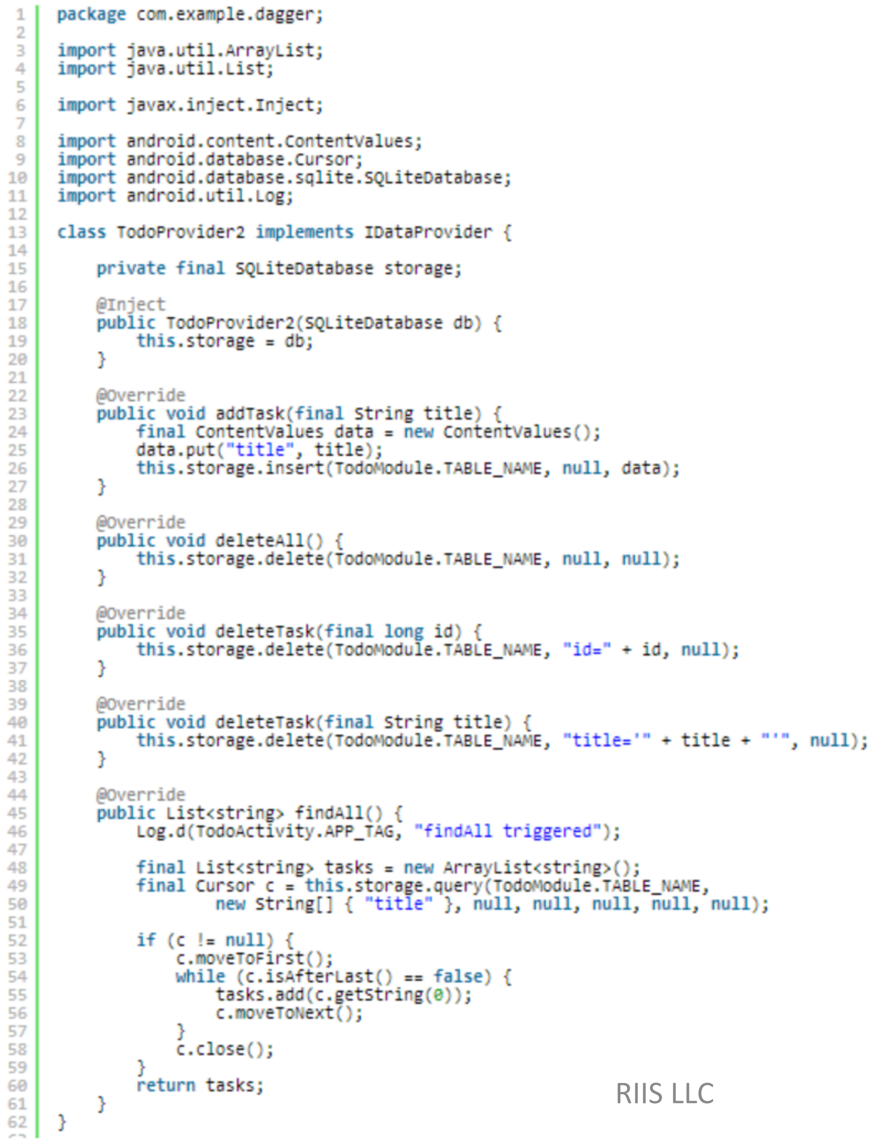
- Dagger
  - Android 4.4.2
  - Android Private Libraries
  - Referenced Libraries
  - src
    - com.example.dagger
      - IDataProvider.java
      - Splash.java
      - TodoActivity.java
      - TodoApplication.java
      - TodoModule.java
      - TodoProvider.java
      - TodoProvider2.java
  - gen [Generated Java Files]
  - assets
  - bin
  - libs
    - android-support-v4.jar
    - dagger-1.0.1.jar
    - javawriter-2.0.1.jar
    - javax.inject-1.jar
  - res
  - AndroidManifest.xml
  - proguard-project.txt
  - project.properties
  - README

Other Tools
Dagger – ToDoModule.java



```java
@Provides
IDataProvider provideDataProvider(final SQLiteDatabase db) {
    //Here I obtain the boolean value for which provider I want
    boolean currentChoice = parent.getCurrentSource();
    if(currentChoice == true){
        //Here is a log message to know which provider has been chosen
        Log.d(TodoActivity.APP_TAG, "Provider2");
        return new TodoProvider2(db);
    }else{
        Log.d(TodoActivity.APP_TAG, "Provider");
        return new TodoProvider(db);
    }
}
```

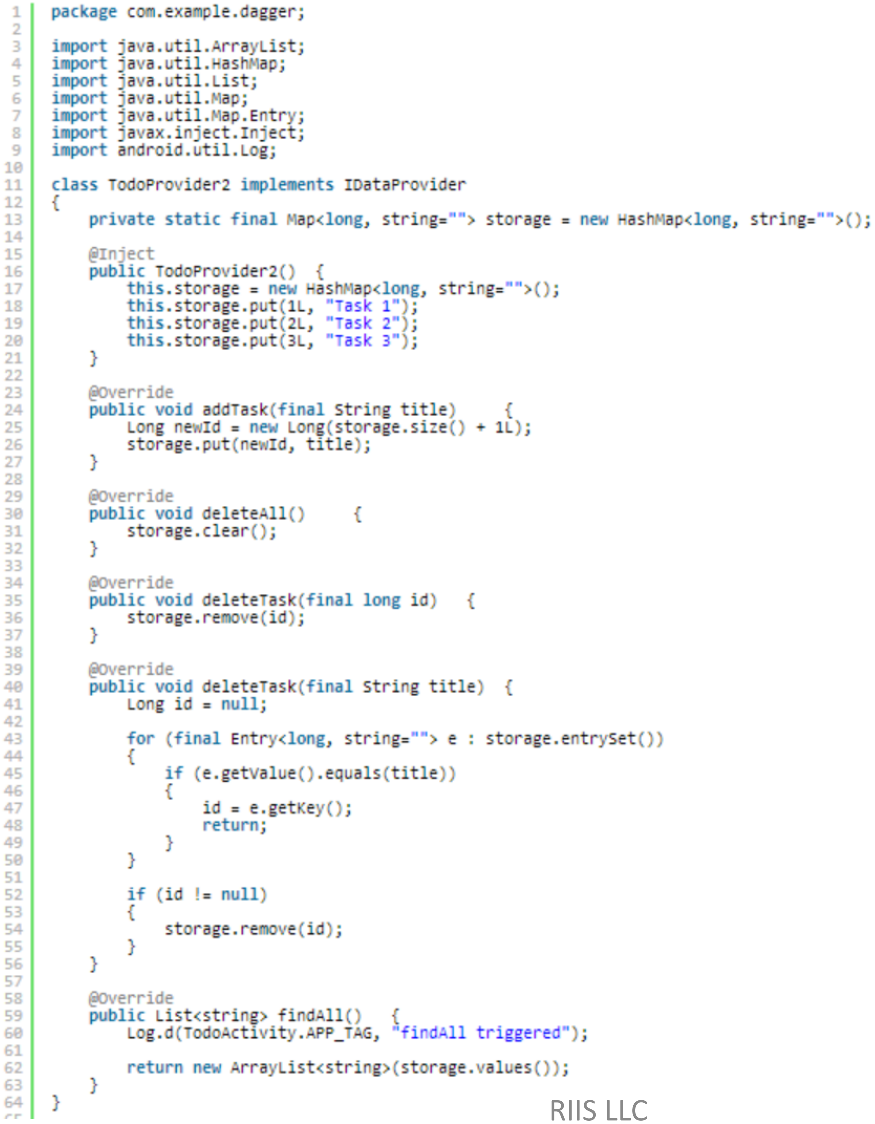
Other Tools
Dagger – ToDoProvider.java (SQLite)



```java
package com.example.dagger;

import java.util.ArrayList;
import java.util.List;

import javax.inject.Inject;

import android.content.ContentValues;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.util.Log;

class TodoProvider2 implements IDataProvider {

    private final SQLiteDatabase storage;

    @Inject
    public TodoProvider2(SQLiteDatabase db) {
        this.storage = db;
    }

    @Override
    public void addTask(final String title) {
        final ContentValues data = new ContentValues();
        data.put("title", title);
        this.storage.insert(TodoModule.TABLE_NAME, null, data);
    }

    @Override
    public void deleteAll() {
        this.storage.delete(TodoModule.TABLE_NAME, null, null);
    }

    @Override
    public void deleteTask(final long id) {
        this.storage.delete(TodoModule.TABLE_NAME, "id=" + id, null);
    }

    @Override
    public void deleteTask(final String title) {
        this.storage.delete(TodoModule.TABLE_NAME, "title='" + title + "'", null);
    }

    @Override
    public List<string> findAll() {
        Log.d(TodoActivity.APP_TAG, "findAll triggered");

        final List<string> tasks = new ArrayList<string>();
        final Cursor c = this.storage.query(TodoModule.TABLE_NAME,
                new String[] { "title" }, null, null, null, null, null);

        if (c != null) {
            c.moveToFirst();
            while (c.isAfterLast() == false) {
                tasks.add(c.getString(0));
                c.moveToNext();
            }
            c.close();
        }
        return tasks;
    }
}
```

Other Tools

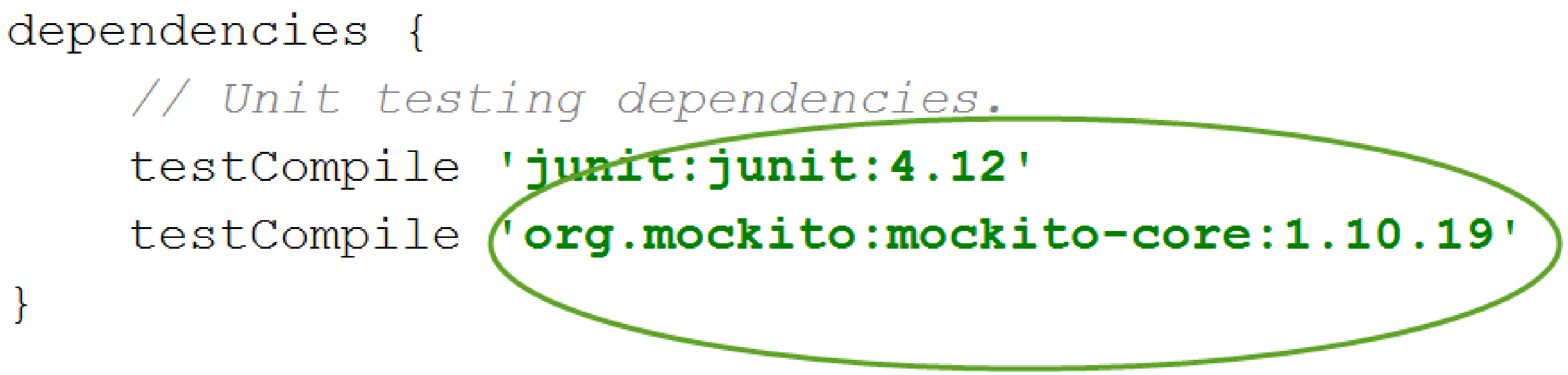
Dagger – ToDoProvider2.java (Fake)



```java
package com.example.dagger;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;
import javax.inject.Inject;
import android.util.Log;

class TodoProvider2 implements IDataProvider
{
    private static final Map<long, string=""> storage = new HashMap<long, string="">();

    @Inject
    public TodoProvider2()  {
        this.storage = new HashMap<long, string="">();
        this.storage.put(1L, "Task 1");
        this.storage.put(2L, "Task 2");
        this.storage.put(3L, "Task 3");
    }

    @Override
    public void addTask(final String title)     {
        Long newId = new Long(storage.size() + 1L);
        storage.put(newId, title);
    }

    @Override
    public void deleteAll()     {
        storage.clear();
    }

    @Override
    public void deleteTask(final long id)   {
        storage.remove(id);
    }

    @Override
    public void deleteTask(final String title)  {
        Long id = null;

        for (final Entry<long, string=""> e : storage.entrySet())
        {
            if (e.getValue().equals(title))
            {
                id = e.getKey();
                return;
            }
        }

        if (id != null)
        {
            storage.remove(id);
        }
    }

    @Override
    public List<string> findAll()   {
        Log.d(TodoActivity.APP_TAG, "findAll triggered");

        return new ArrayList<string>(storage.values());
    }
}
```

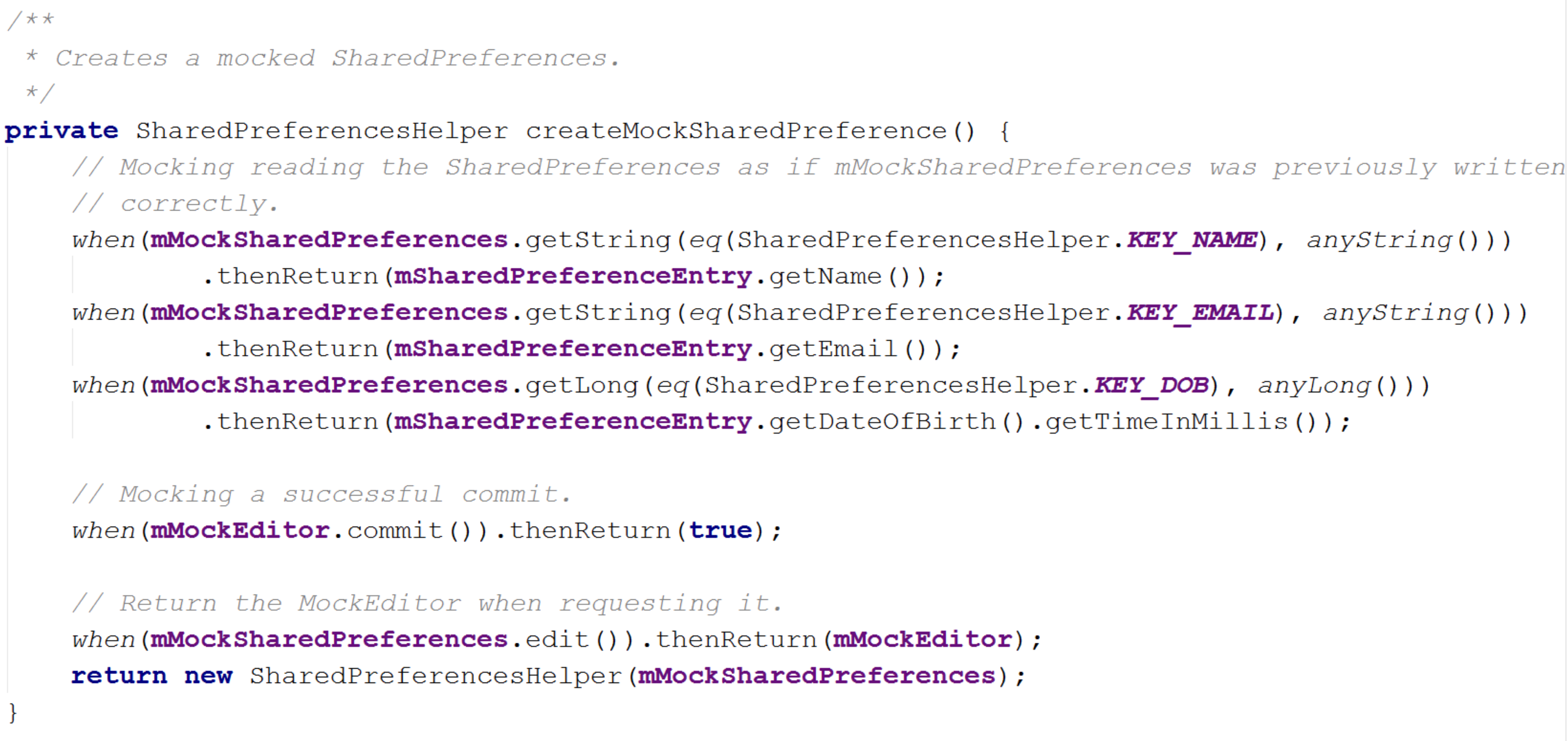
Other tools
Mocking

- Add mockito dependency in build.gradle (app) file



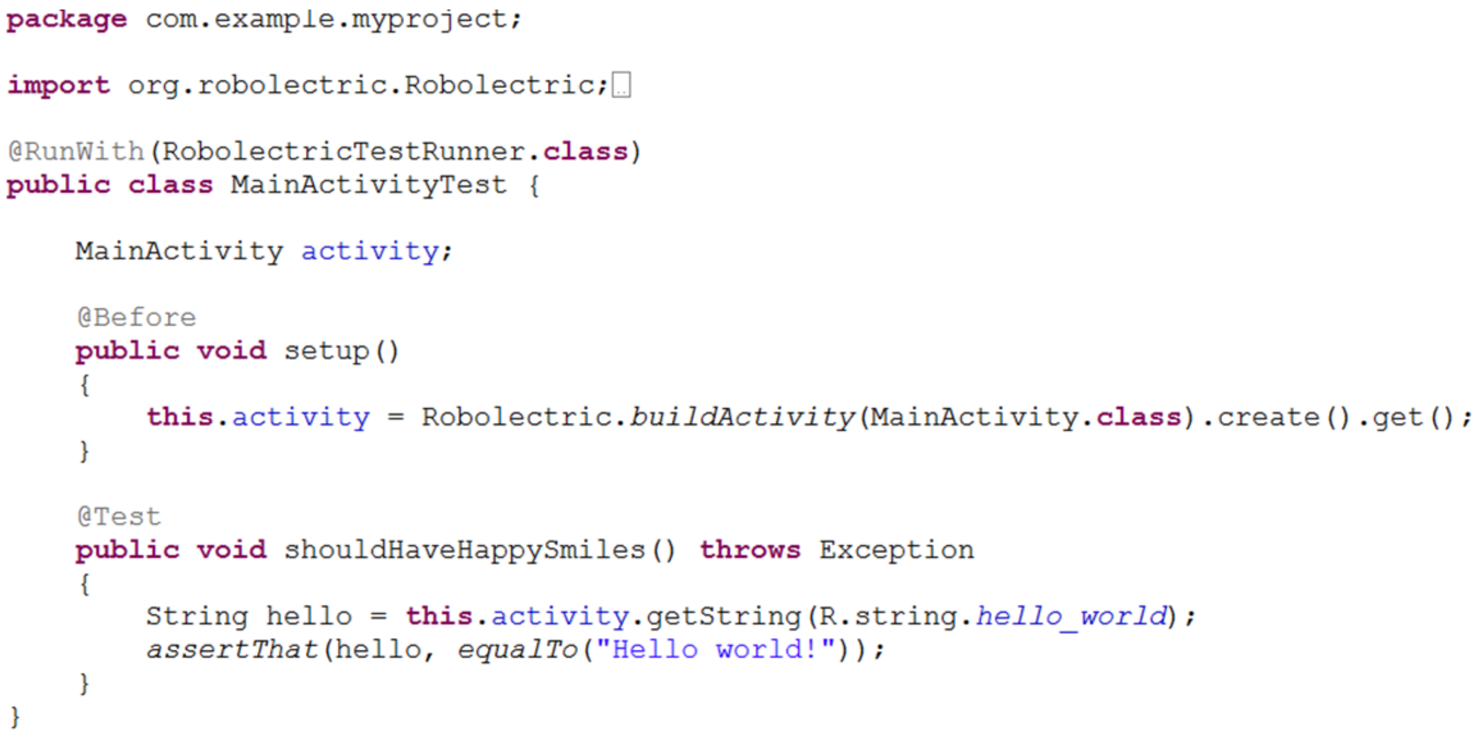
```
dependencies {
    // Unit testing dependencies.
    testCompile 'junit:junit:4.12'
    testCompile 'org.mockito:mockito-core:1.10.19'
}
```

Other tools
Mocking



```
/**
 * Creates a mocked SharedPreferences.
 */
private SharedPreferencesHelper createMockSharedPreference() {
    // Mocking reading the SharedPreferences as if mMockSharedPreferences was previously written
    // correctly.
    when(mMockSharedPreferences.getString(eq(SharedPreferencesHelper.KEY_NAME), anyString()))
            .thenReturn(mSharedPreferenceEntry.getName());
    when(mMockSharedPreferences.getString(eq(SharedPreferencesHelper.KEY_EMAIL), anyString()))
            .thenReturn(mSharedPreferenceEntry.getEmail());
    when(mMockSharedPreferences.getLong(eq(SharedPreferencesHelper.KEY_DOB), anyLong()))
            .thenReturn(mSharedPreferenceEntry.getDateOfBirth().getTimeInMillis());

    // Mocking a successful commit.
    when(mMockEditor.commit()).thenReturn(true);

    // Return the MockEditor when requesting it.
    when(mMockSharedPreferences.edit()).thenReturn(mMockEditor);
    return new SharedPreferencesHelper(mMockSharedPreferences);
}
```

Other tools
Robolectric

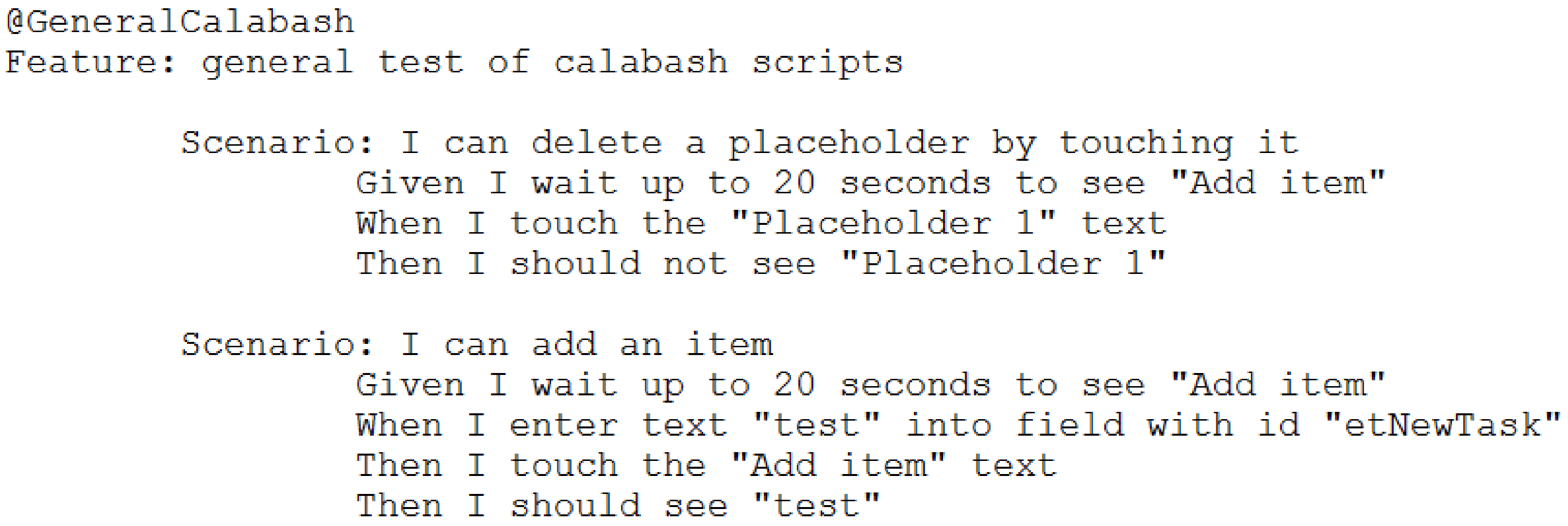


```java
package com.example.myproject;

import org.robolectric.Robolectric;

@RunWith(RobolectricTestRunner.class)
public class MainActivityTest {

    MainActivity activity;

    @Before
    public void setup()
    {
        this.activity = Robolectric.buildActivity(MainActivity.class).create().get();
    }

    @Test
    public void shouldHaveHappySmiles() throws Exception
    {
        String hello = this.activity.getString(R.string.hello_world);
        assertThat(hello, equalTo("Hello world!"));
    }
}
```

Other Tools
BDD

- Add Functional Testing or BDD
- Try Cucumber using Calabash
- Given/when/then Gherkin format

Other Tools
BDD

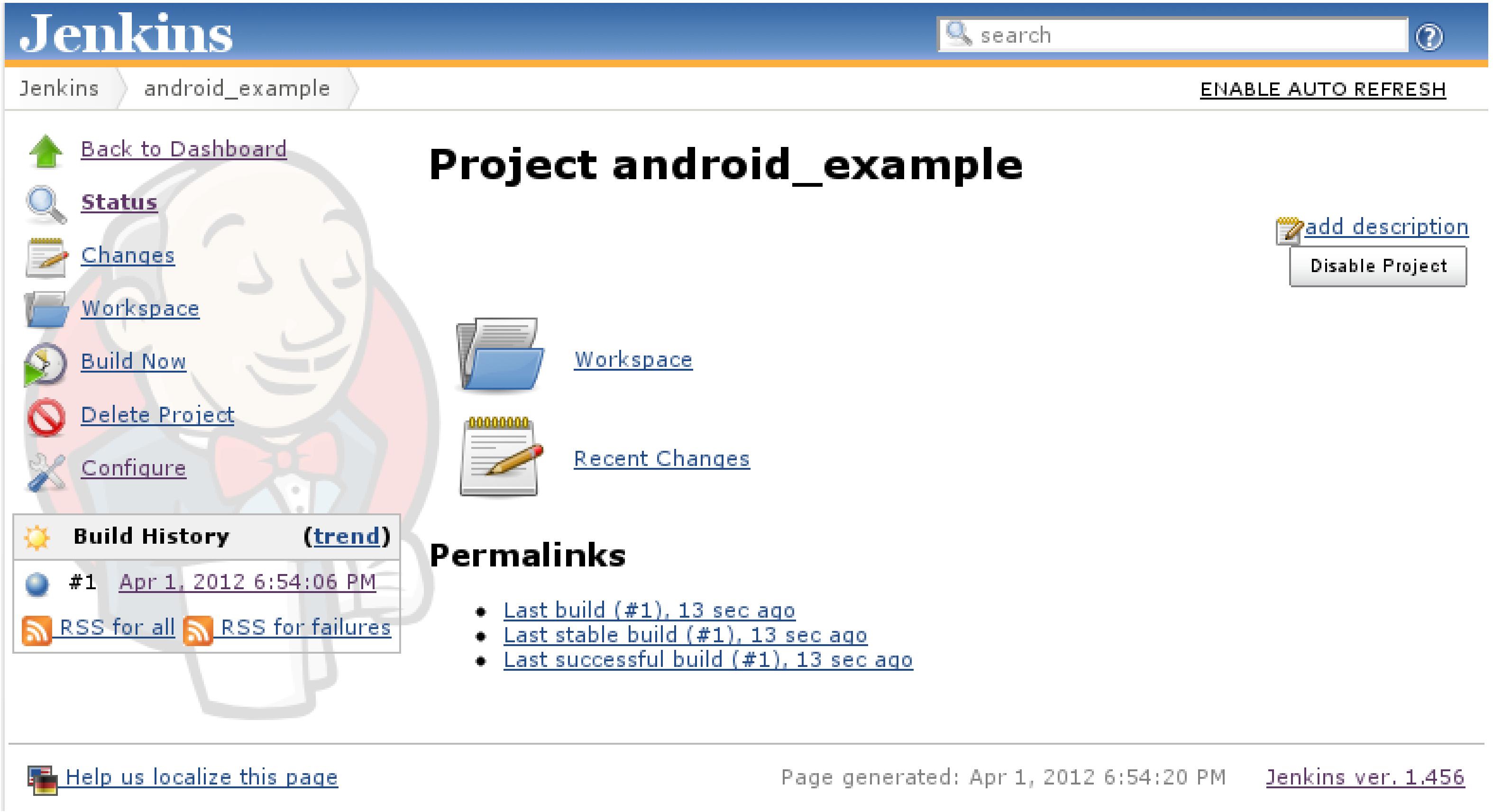


```
@GeneralCalabash
Feature: general test of calabash scripts

        Scenario: I can delete a placeholder by touching it
                Given I wait up to 20 seconds to see "Add item"
                When I touch the "Placeholder 1" text
                Then I should not see "Placeholder 1"

        Scenario: I can add an item
                Given I wait up to 20 seconds to see "Add item"
                When I enter text "test" into field with id "etNewTask"
                Then I touch the "Add item" text
                Then I should see "test"
```

Other Tools
BDD

- gem install calabash-android
- calabash-android gen (skeleton framework)
- Create your features
- If possible use canned steps otherwise some ruby required
- calabash-android run <test>.apk

Other tools
Jenkins



Jenkins

search

Jenkins › android_example

ENABLE AUTO REFRESH

- Back to Dashboard
- Status
- Changes
- Workspace
- Build Now
- Delete Project
- Configure

Build History (trend)

#1 Apr 1, 2012 6:54:06 PM

RSS for all RSS for failures

Project android_example

add description

Disable Project

Workspace

Recent Changes

Permalinks

- Last build (#1), 13 sec ago
- Last stable build (#1), 13 sec ago
- Last successful build (#1), 13 sec ago

Help us localize this page

Page generated: Apr 1, 2012 6:54:20 PM

Jenkins ver. 1.456

Other tools
Jenkins

- Get Jenkins.jar file from http://jenkins-ci.org
- Start and install gradle plugin
- Install the Android SDK
- Configure emulator
- Configure to build, run unit tests, calabash tests

Other Tools
ProGuard

- Add to ProGuard configuration file of the main project

```
-printmapping proguard-mapping.txt
```

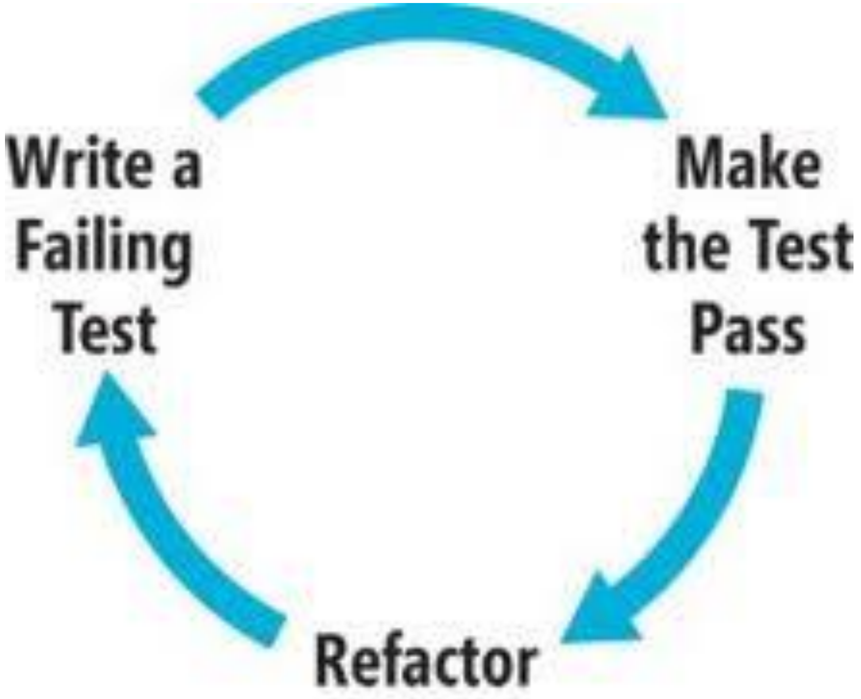
- Add to ProGuard configuration file of the test project

```
-applymapping ../mainproject/bin/proguard-mapping.txt
-injars ../mainproject/bin/classes
```

- Proguard will obfuscate test files the same as the original code ☺

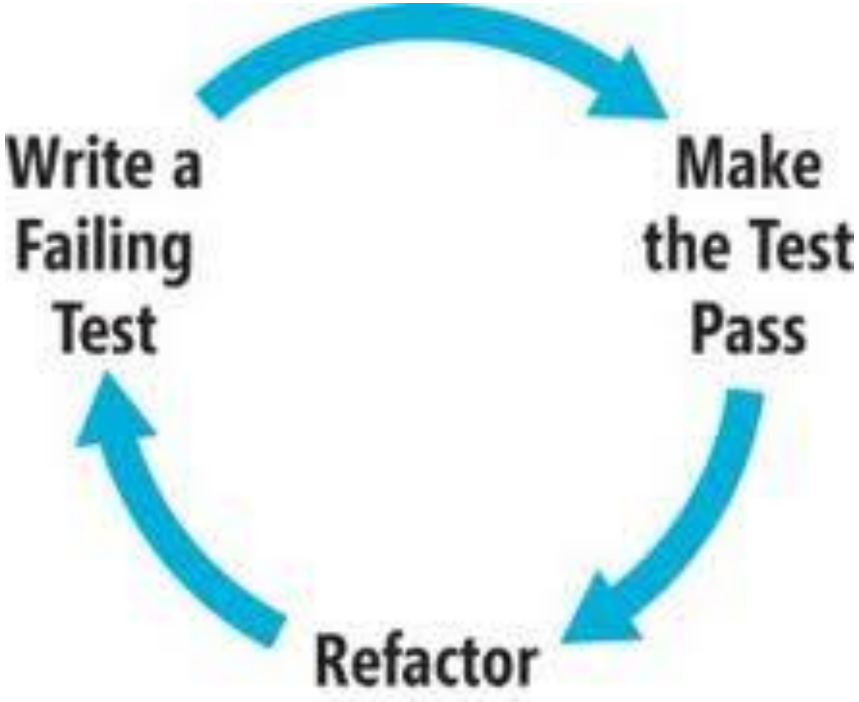
TDD

- ○ Intro
- ○ Difference between unit test and TDD
- ○ Sample Projects
- ○ Applying TDD to an existing project
- ○ Lessons Learned



TDD
Intro

- ○ Write test first
- ○ See it fail
- ○ Write simplest possible solution to get test to pass
- ○ Refactor
- ○ Wash, Rinse, Repeat
- ○ Built in regression testing
- ○ Longer life for your codebase
- ○ YAGNI feature development



TDD

Value of TDD

- Tests help us catch mistakes
- Tests help us shape our design to actual use, avoids gold-plating
- Code becomes usable
- Code becomes lean
- Red/Green/Refactor helps kill procrastination

TDD
Unit Test vs TDD

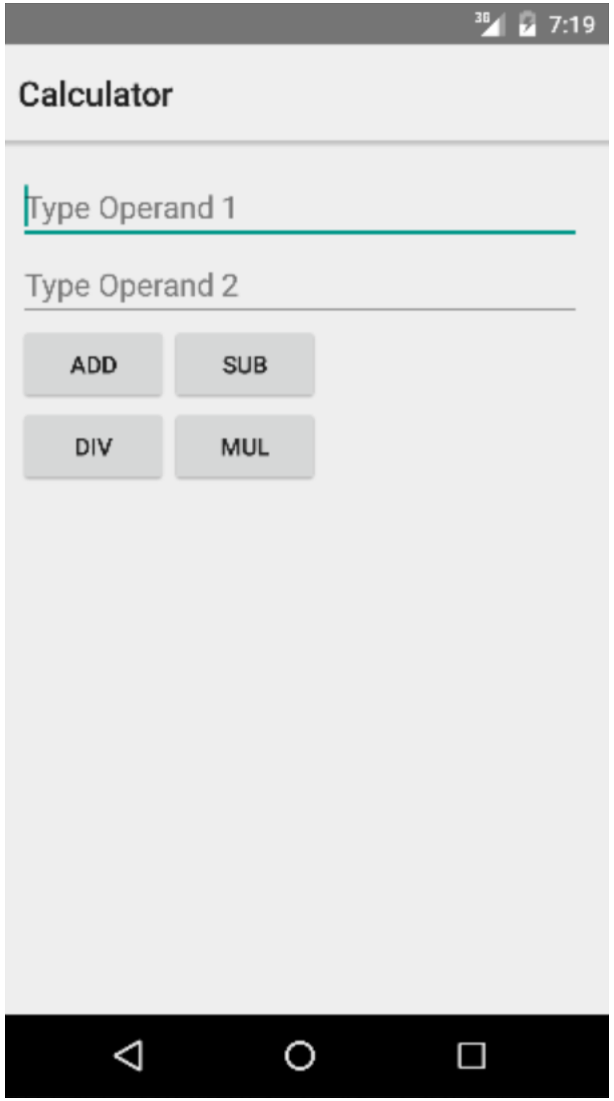
- You can do unit testing with TDD
- You can't do TDD without unit testing

- TDD means writing the unit test before writing the code
- Unit tests don't mandate when you write tests

- TDD is more painless than classic unit testing
- Unit tests are often written at the end of a coding cycle to improve code coverage metrics

TDD
Sample Project - Calculator

- User interface has two fields to enter numbers
- User interface has four buttons to add, subtract, multiply and divide
- Fields should start with Type Operand 1 and 2
- When add button is clicked show addition of Op 1 and Op 2
- When sub button is clicked show subtraction of Op 1 and Op 2
- When div button is clicked show division of Op 1 and Op 2
- When mul button is clicked show multiplication of Op 1 and Op 2
- Left align values



Applying TDD to an Existing Project

- TDD any new code
- Ring Fence existing code
- Refactor existing code

Applying TDD to an Existing Project

- Introduce automated Continuous Integration to build the code
- Configure android projects for TDD
- Add minimal unit tests based on existing tests, add to CI
- Show team how to create unit tests
- Add testing code coverage metrics to CI, initially expected to be 5-10%
- Add example functional tests, in given/when/then format
- Add a unit test for any new features or sprouts, using mock objects etc.
- Wrap or ring fence existing code
- Remove any old code that is no longer used
- Refactor wrapped code over time to increase the code coverage to 60-70%

Applying TDD to an Existing Project
Lessons Learned

- What worked
  - Take baby steps
  - Metrics should evolve….
    - Maintainability Index, Depth of Inheritance, Cyclomatic Complexity, Class Coupling, Lines of Code etc.
- Not so much ☹
  - Don't ship the tests
  - Don't be driven by metrics
    - Keep an eye on performance metrics
- Remember….
  - You don't need anyone's permission to start unit testing/TDD

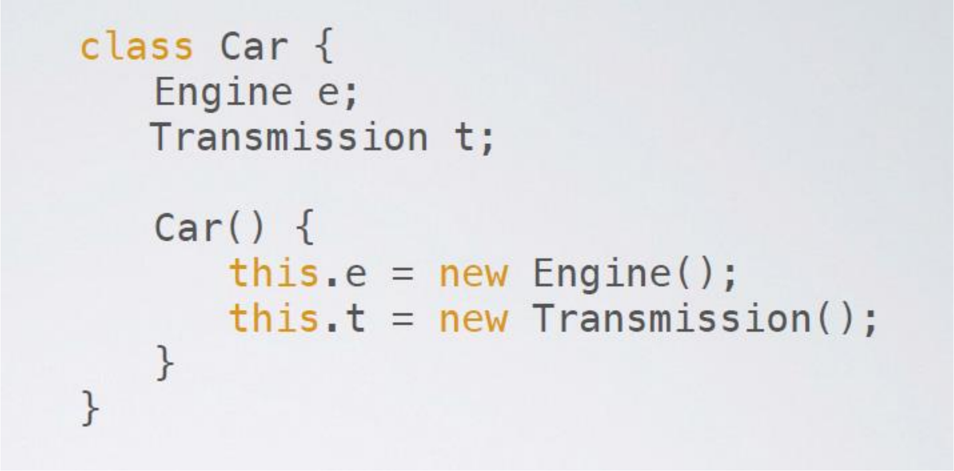
Applying TDD to an Existing Project
How to Write Untestable Code

- Make Your Own Dependencies
- Heavy Duty Constructors
- Depend on Concrete classes
- Conditional Slalom
- Depend on Large Context Objects
- Use Statics
- Use Global Flags
- Use Singletons Everywhere
- Be Defensive - They're out to Get Your Code!
- Use Primitives Wherever Possible
- Look for Everything You Need
- Use static initializes
- Couple functional code directly to the external system it depends on
- Mix Object Lifecycles
- Side Effects are the Way to Go
- Create Utility Classes and Functions/Methods
- Create Managers and Controllers
- Do Compilicated Creation Work in Objects
- Greenlight if-branches and switch statements
- Utils, Utils, Utils!
- Use "Refactoring" whenever you need to get away with something
- Final Methods
- Handcuff your users to Specific Types

Applying TDD to an Existing Project

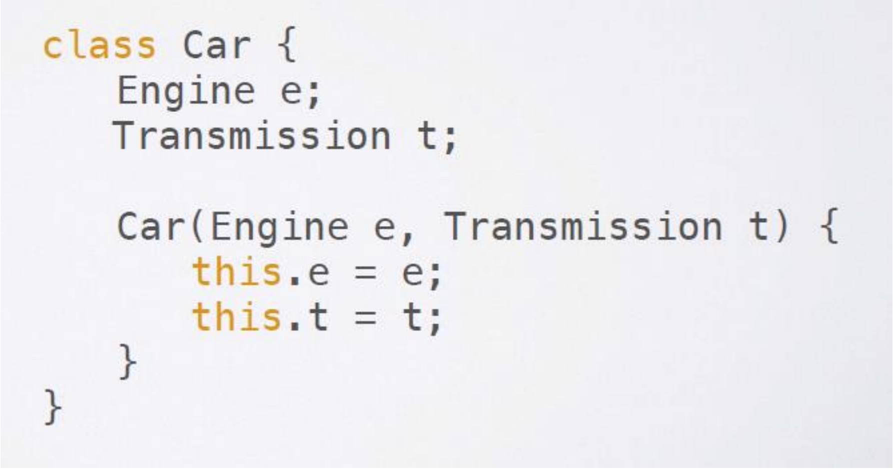
How to Write Untestable Code – Tightly Coupled

NO



```
class Car {
    Engine e;
    Transmission t;

    Car() {
        this.e = new Engine();
        this.t = new Transmission();
    }
}
```

YES



```
class Car {
    Engine e;
    Transmission t;

    Car(Engine e, Transmission t) {
        this.e = e;
        this.t = t;
    }
}
```

Resources
URLs

https://developer.android.com/training/testing/unit-testing/local-unit-tests.html

http://tools.android.com/tech-docs/unit-testing-support

https://github.com/googlesamples/android-testing/tree/master/unit/BasicSample

https://code.google.com/p/hamcrest/wiki/Tutorial

https://github.com/calabash/calabash-android

http://agile2009.agilealliance.org/node/403/

http://david.heinemeierhansson.com/2014/tdd-is-dead-long-live-testing.html

http://martinfowler.com/articles/is-tdd-dead/

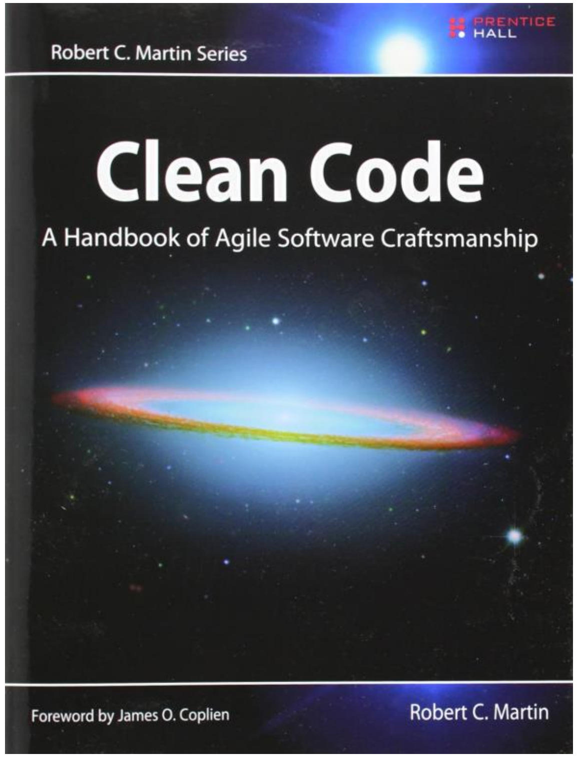
Dagger http://square.github.io/dagger/

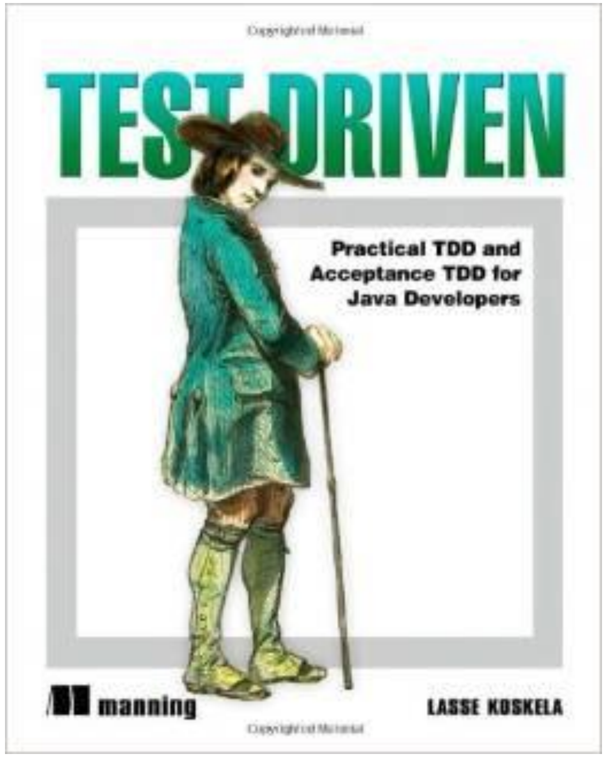
http://www.infoq.com/presentations/Dagger

http://googletesting.blogspot.com/2008/07/how-to-write-3v1l-untestable-code.html

http://misko.hevery.com/attachments/Guide-Writing%20Testable%20Code.pdf

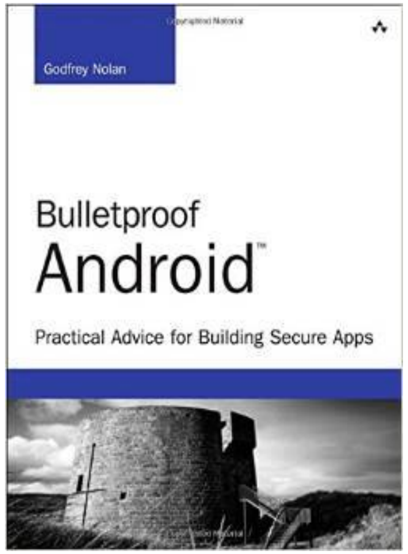
Resources
Books

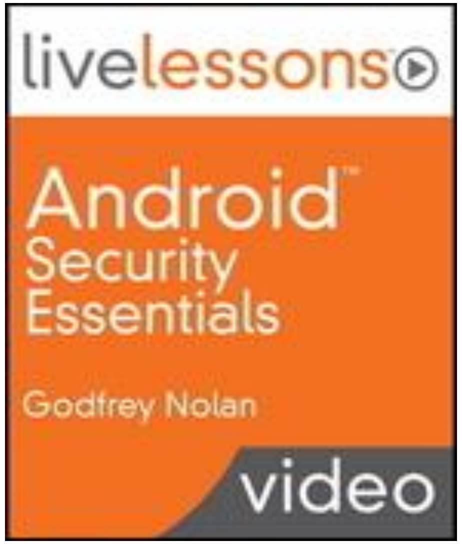

PRENTICE HALL
Robert C. Martin Series
Clean Code
A Handbook of Agile Software Craftsmanship
Foreword by James O. Coplien
Robert C. Martin

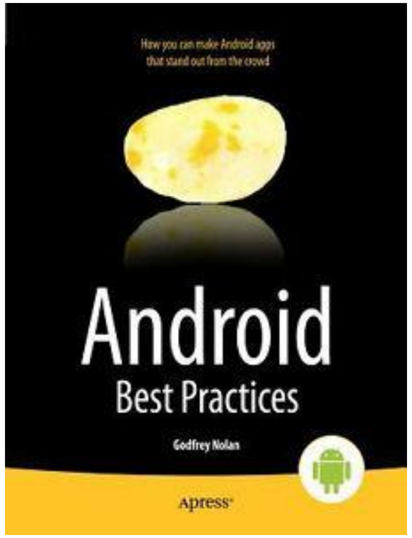


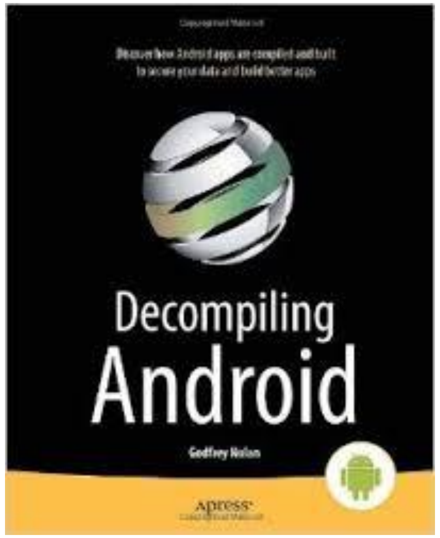
Contact Details

- godfrey@riis.com
- @riisllc, @godfreynolan








Discover how Android apps are compiled and built
to secure your data and build better apps
Decompiling
Android
Godfrey Nolan
Apress®