



FREE eBook

LEARNING android-testing

Free unaffiliated eBook created from
Stack Overflow contributors.

#android-
testing

Table of Contents

About.....	1
Chapter 1: Getting started with android-testing	2
Remarks.....	2
Examples.....	2
Test Types.....	2
Credits	4

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [android-testing](#)

It is an unofficial and free android-testing ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official android-testing.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with android-testing

Remarks

This section provides an overview of what android-testing is, and why a developer might want to use it.

It should also mention any large subjects within android-testing, and link out to the related topics. Since the Documentation for android-testing is new, you may need to create initial versions of those related topics.

Examples

Test Types

Android tests are based on JUnit, and you can run them either as local unit tests on the JVM or as instrumented tests on an Android device. This page provides an introduction to the concepts and tools for building Android tests

- Local unit tests (Located at module-name/src/test/java/)
- Instrumented tests (Located at module-name/src/androidTest/java/)

Local unit tests JUnit

```
@RunWith (AndroidJUnit4.class)
@LargeTest
public class MainActivityInstrumentationTest {

    @Rule
    public ActivityTestRule mActivityTestRule = new ActivityTestRule<>(
        MainActivity.class);

    @Test
    public void sayHello(){
        onView(withText ("Say hello!")).perform(click());

        onView(withId(R.id.textView)).check(matches(withText ("Hello, World!")));
    }
}
```

Instrumented tests Example 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' // 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" } }
```

Instrumented Unit Test Class

```
import android.os.Parcel;
import android.support.test.runner.AndroidJUnit4;
import android.util.Pair;
import org.junit.Test;
import org.junit.runner.RunWith;
import java.util.List;
import static org.hamcrest.Matchers.is;
import static org.junit.Assert.assertThat;

@RunWith(AndroidJUnit4.class)
@SmallTest
public class LogHistoryAndroidUnitTest {

    public static final String TEST_STRING = "This is a string";
    public static final long TEST_LONG = 12345678L;
    private LogHistory mLogHistory;

    @Before
    public void createLogHistory() {
        mLogHistory = new LogHistory();
    }

    @Test
    public void logHistory_ParcelableWriteRead() {
        // Set up the Parcelable object to send and receive.
        mLogHistory.addEntry(TEST_STRING, TEST_LONG);

        // Write the data.
        Parcel parcel = Parcel.obtain();
        mLogHistory.writeToParcel(parcel, mLogHistory.describeContents());

        // After you're done with writing, you need to reset the parcel for reading.
        parcel.setDataPosition(0);

        // Read the data.
        LogHistory createdFromParcel = LogHistory.CREATOR.createFromParcel(parcel);
        List<Pair<String, Long>> createdFromParcelData = createdFromParcel.getData();

        // Verify that the received data is correct.
        assertThat(createdFromParcelData.size(), is(1));
        assertThat(createdFromParcelData.get(0).first, is(TEST_STRING));
        assertThat(createdFromParcelData.get(0).second, is(TEST_LONG));
    }
}
```

Read Getting started with android-testing online: <https://riptutorial.com/android-testing/topic/8959/getting-started-with-android-testing>

Credits

S. No	Chapters	Contributors
1	Getting started with android-testing	Community , Umar