



EBook Gratis

APRENDIZAJE android-testing

Free unaffiliated eBook created from
Stack Overflow contributors.

#android-
testing

Tabla de contenido

Acerca de.....	1
Capítulo 1: Comenzando con las pruebas de Android.....	2
Observaciones.....	2
Examples.....	2
Tipos de prueba.....	2
Creditos.....	4

Acerca de

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

Capítulo 1: Comenzando con las pruebas de Android

Observaciones

Esta sección proporciona una descripción general de qué es la prueba de Android y por qué un desarrollador puede querer usarla.

También debe mencionar los temas importantes dentro de las pruebas de Android y vincular a los temas relacionados. Dado que la Documentación para pruebas de Android es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

Tipos de prueba

Las pruebas de Android se basan en JUnit, y puede ejecutarlas como pruebas de unidad local en la JVM o como pruebas instrumentadas en un dispositivo Android. Esta página proporciona una introducción a los conceptos y herramientas para construir pruebas de Android

- Pruebas de unidades locales (ubicadas en nombre-módulo / src / test / java /)
- Pruebas instrumentadas (ubicadas en nombre-módulo / src / androidTest / java /)

Pruebas unitarias locales JUnit

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

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

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

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

Pruebas instrumentadas Ejemplo dependencias {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' // Opcional - Hamcrest library androidTestCompile 'org.hamcrest: hamcrest-library: 1.3' // Opcional - Pruebas de interfaz de usuario con Espresso androidTestCompile 'com.android.support.test.espresso: espresso-core: 2.2.2' // Opcional - Pruebas de UI con UI Automator androidTestCompile

```
'com.android.support.test.uiautomator: uiautomator-v18: 2.1.2'}
```

```
android {defaultConfig {testInstrumentationRunner  
"android.support.test.runner.AndroidJUnitRunner"}}
```

Clase de prueba unitaria instrumentada

```
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));  
    }  
}
```

Lea Comenzando con las pruebas de Android en línea: <https://riptutorial.com/es/android-testing/topic/8959/comenzando-con-las-pruebas-de-android>

Creditos

S. No	Capítulos	Contributors
1	Comenzando con las pruebas de Android	Community , Umar