Integrate the Perfecto plugin with your Android Studio project

On this page:

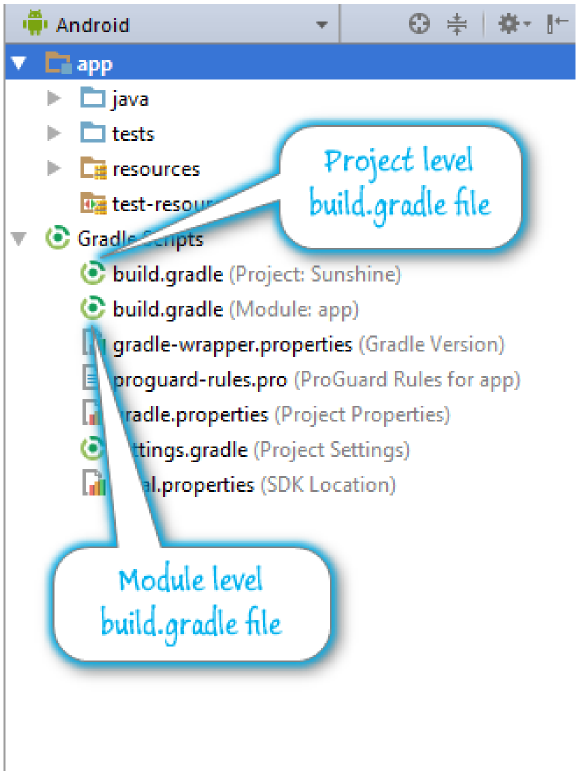
The Perfecto Gradle plugin allows the tester to:

- Select a device or multiple devices from the Perfecto Lab to run Instrumentation Tests for Android applications.
- Install the application and test *apk* files onto the selected devices.
- Run the test methods on the devices.
- See the progress of the test set on the console.
- Access a DigitalZoom Report Library grid that presents the results of the tests.

The procedure presented here is used when the application and test *apk* files are available on the local disc storage and the plugin *jar* file is not installed.

Work within an Android Studio project

Android Studio projects are integrated with Gradle and the project will include several **build.gradle** files - one for the project level and one for each module of the project, See below in this screenshot:


Android
app
java
tests
resources
test-resou
Gradle Scripts
build.gradle (Project: Sunshine)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
radle.properties (Project Properties)
ttings.gradle (Project Settings)
l.properties (SDK Location)
Project level build.gradle file
Module level build.gradle file

To execute the plugin with an Android Studio project:

1. Open the *project* **build.gradle** file.
2. Add the lines to the build gradle file that define the location of the plugin library and the dependency on the plugin:

   2a. To configure Gradle to automatically locate and download the plugin library add the following lines to ***build.gradle*** file:

```
buildscript {

repositories {

jcenter()

maven {

url "https://repo1.perfectomobile.com/public/repositories/maven"
        }
    }


dependencies {

classpath "com.perfectomobile.instrumentedtest.gradleplugin:plugin:+"
    }
}
```

2b. If the plugin library is already downloaded to a folder (for example: **libs** sub-folder) add the following lines to ***build.gradle*** file:

```
buildscript {

repositories
{

jcenter()


flatDir
dirs: 'libs'
    }


dependencies
{

classpath
"com.
perfectomobil
e.
instrumentedt
est.
gradleplugin:
plugin:+"
    }
}
```

3. Open the *module* **build.gradle** file.
4. Add the line that defines the plugin task

```
apply plugin: 'com.perfectomobile.
instrumentedtest.gradleplugin'
```

5. Add the plugin configuration settings - Lab authentication parameters
    1. Include the URL for the Perfecto Lab
    2. Your personal Security Token, generated for the Perfecto Lab

    ```
    perfectoGradleSettings {
        cloudURL "mobilecloud.
    perfectomobile.com"
        securityToken
    "AAABNg0ODAoPeNqtkT1PwzAQhnf/CkssM...
    JxQ3HEI8NsX02ff"
    }
    ```

    Note: You could also supply the configuration file location parameter in this gradle-file clause.
6. Create a configuration file.
7. Open a *command-line* (or terminal) window at the project folder.
8. Execute the plugin using the following command in the command-line window:

Important Note: User should verify that the "*assembleDebug*" and "*assembleAndroidTest*" gradle tasks execute before executing the ***perfecto-android-inst*** task. The plugin will search for the generated *apk* files in the standard gradle output folders or in the package Debug and AndroidTest folders. If the apk files are located in other locations, user should supply the ***apkPath*** and ***testApkPath*** configuration parameters.

Note: When testing an Android (*aar* or *jar*) library, set both the **apkPath** and **testApkPath** fields to the location specified for the ***assembleAndroidTest*** gradle task. By default, this location is: *<project folder>/app/build/outputs/apk/androidTest/debug/app-debug-androidTest.apk*

Here we also supply the full path to the Configuration File created in steps 6-10. Other configuration parameters (except for device selection) could be added to the command line.

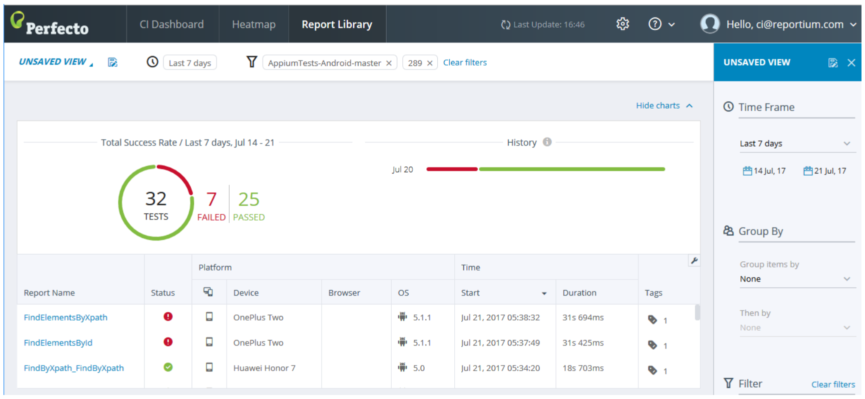
```
gradle perfecto-android-inst -
PconfigFileLocation="C:
\temp\Espresso\ConfigFile.json"
```

This will:
- Select the devices as specified in the Device selection parameters of the configuration file [or a random device if no specification provided],
- Install the application and test *apk* files onto the device
- Run the test methods (based on the configuration parameters)
- Send output to the console window
- Generate an execution report that can be viewed in the Test analysis with Smart Reporting interface.

9. During the execution, the plugin will report on the progress of the execution, and the completion of each test method to the command-line window.
10. At the end of the execution, a high-level summary report of the completion status for each device used will be presented in the command-line window
11. Copy the report URL from the summary report on your console:

```
View the detailed report at: https://demo.
reporting.perfectomobile.com/library?
startExecutionTime[1]=lastMonth&tags[0]
=5af27a82-54cc-405a-8c6e-fa46fcae874b
Finished flow execution
```

12. Open the URL in your browser to access the execution report:



		Platform				Time		
Report Name	Status		Device	Browser	OS	Start	Duration	Tags
FindElementsByXpath	●		OnePlus Two		5.1.1	Jul 21, 2017 05:38:32	31s 694ms	1
FindElementsById	●		OnePlus Two		5.1.1	Jul 21, 2017 05:37:49	31s 425ms	1
FindByXpath_FindByXpath	✓		Huawei Honor 7		5.0	Jul 21, 2017 05:34:20	18s 703ms	1

Demo video

The following video demostrates these steps.

Related articles

- Configuration Parameters for the Gradle Plugin
- Android configuration parameters for the Gradle Plugin
- iOS configuration parameters for the Gradle Plugin
- Run XCUITest on Perfecto devices
- Appium Desktop