**[How to Quickly Set Up Android Testing on Calabash](http://www.optimusinfo.com/set-up-android-testing-calabash/" \o "Permanent Link: How to Quickly Set Up Android Testing on Calabash)**

April 10, 2013/[0 Comments](http://www.optimusinfo.com/set-up-android-testing-calabash/#respond)/in [Uncategorized](http://www.optimusinfo.com/category/uncategorized/) [Mobile Application Testing](http://www.optimusinfo.com/service/mobile-application-testing/) /

Calabash-android is a great little test automation tool for [functional testing](http://optimusqa.staging.wpengine.com/services/functional-testing/) of Android applications that uses the simple English test script creation of Cucumber to automate your application. It lets you automate testing with little or no coding.

Here is what you need to get started with calabash-android.

**Install Android, Ruby, Cucumber and Calabash**

1. Install the android-sdk to run the emulator.
2. Install Ruby on your system.
3. Run the command “gem install cucumber” in cmd to install Cucumber on your system. After successful extraction check its version and update if necessary.
4. Run the command “gem install calabash-android” in cmd to install calabash-android on your system.

**Configure your System**

1. Set the Android path to the following in system variables:  
   [code language=”text”]ANDROID\_HOME=C:Program FilesAndroidandroid-sdk[/code]
2. Set the JAVA\_Home path to the following in system variables:  
   [code language=”text”]JAVA\_HOME=C:/Program Files/Java/jdk1.7.0\_17[/code]
3. Set the ANT\_HOME path to the following in system variables:  
   [code language=”text”]ANT\_HOME=D:apache-ant-1.8.4[/code]
4. Set the path to the following in system variables:  
   [code language=”text”]path=D:apache-ant-1.8.4bin;[/code]

**Open your cmd and Install the Mobile Application**

1. Navigate to platform-tools directory.
2. Copy and paste your application inside this directory.
3. Open the android-avd manager and create a new emulator.
4. Start your emulator and wait till it completely loads.
5. Open cmd and navigate to platform-tools. In cmd enter the following command:  
   [code language=”text”]&gt;adb install your\_file\_name.apk[/code]
6. After your app is installed, verify that it is present in menu screen of your emulator.

**Create a New Directory**

1. Create a new directory using the following command:  
   [code language=”text”]&gt; mkdir D:FirstProject[/code]
2. Now generate the folder structure with feature file, step-definitions, support and lib using the following command:  
   [code language=”text”]D:FirstProject&gt; calabash-android gen[/code]
3. Now sign your apk using its keystore path, password and alias:  
   [code language=”text”]D:FirstProject&gt; calabash-android setup[/code]

Be sure to note the complete path to debug.keystore. When you are asked to enter the keystore location, enter the full path including the name: i.e**D:Calabashkeystore\_name**.

The system will ask for debug.keystore of the application, the password and the alias to sign your apk. .calabash-settings file. The folder structure will be created upon completion.

1. Now write following code in your feature file-> my\_first:  
   [code language=”text”]  
   Feature: General functionality for InfantAudit   
   Scenario: Create New Store   
   Given I am on the Splash Screen   
   Then I press the “Log in” button[/code]
2. Write the following code in step definitions:  
   [code language=”ruby”]$LOAD\_PATH.unshift(File.dirname(\_\_FILE\_\_) + ‘/../../lib’) unless $LOAD\_PATH.include?(File.dirname(\_\_FILE\_\_) + ‘/../../lib’)  
   require ‘calabash-android/calabash\_steps’  
   require ‘utilities.rb’

Given /^I am on the Splash Screen$/ do  
wait\_for(180) {element\_exists("button id:’btnLogin’")}  
end[/code]

1. Open cmd and run your code using following command:  
   [code language=”text”]D:FirstProject&gt; calabash-android run your\_file\_name.apk[/code]
2. If you will be automating the next sprint of this application then you have to re-build and re-sign application to create a new test server for each sprint using following command:  
   [code language=”text”]  
   D:FirstProject&gt; calabash-android build your\_file\_name.apk  
   D:FirstProject&gt; calabash-android resign your\_file\_name.apk[/code]

**No Test Server Error**

You sometimes get the following error while running apk through calabash which you can resolve using the method described below.

No test server found for this combination of app and calabash version. Recreating test server. /Library/Ruby/Gems/1.8/gems/calabash-android-0.4.3.pre4/lib/calabash-android/helpers.rb:148:in `extract\_md5\_fingerprint': No MD5 fingerprint found: (RuntimeError)

**Recommended Solution**

1. Copy your “Calabash” folder having “feature” file.
2. Paste it into your android workspace having gen, res, src folder.
3. Navigate to new path and again run apk file.
4. Now, Test server will create inside your calabash folder.

**Alternate Solution:**

1. Re-sign your apk file using command:  
   [code language=”text”]-D:FirstProject&gt; calabash-android build your\_file\_name.apk[/code]
2. Build your apk file using command:  
   [code language=”text”]D:FirstProject&gt; calabash-android resign your\_file\_name.apk[/code]