How to use QTium Framework

Contents

[INTRODUCTION 1](#_Toc394933195)

[*What is QTium Framework?* 1](#_Toc394933196)

[*QTium Framework platform support* 2](#_Toc394933197)

[*What you should already know* 2](#_Toc394933198)

[SETUP DEVELOPMENT ENVIRONMENT 2](#_Toc394933199)

[*Software requirements* 2](#_Toc394933200)

[*Browser requirements for testing desktop web-based application* 2](#_Toc394933201)

[*Software requirements for testing web mobile and native mobile application* 2](#_Toc394933202)

[*Appium installation on Windows* 3](#_Toc394933203)

[*Running Appium on Windows* 8](#_Toc394933204)

[*Appium installation on Mac OS X – automate native app on iOS* 9](#_Toc394933205)

[*Running Appium on Mac OS X – automate native app on iOS* 11](#_Toc394933206)

[*Appium installation on Mac OS X – automate native app on OS X* 12](#_Toc394933207)

[*Eclipse workspace* 12](#_Toc394933208)

[WORKING WITH QTIUM FRAMEWORK 16](#_Toc394933209)

[*Project structure* 16](#_Toc394933210)

[*Popular classes* 16](#_Toc394933211)

[*Test suite file* 17](#_Toc394933212)

# INTRODUCTION

This document guides how to implement test script using QTium Framework with TestNG is test management.

## What is QTium Framework?

QTium Framework is a test automation framework. It is a wrapper for Selenium WebDriver, Appium.

## QTium Framework platform support

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **FireFox** | **Chrome** | **Internet Explorer** | **Built-in Browser** | **Safari** | **Native app** |
| Windows |  |  |  | **N/A** |  |  |
| Android |  |  | **N/A** |  | **N/A** |  |
| iOS |  |  | **N/A** | **N/A** |  |  |
| OS X |  |  | **N/A** | **N/A** |  |  |

## What you should already know

* Java programming language
* Eclipse
* Using TestNG basically

# SETUP DEVELOPMENT ENVIRONMENT

## Software requirements

1. OS: Windows 7 or above or Mac OS X 10.7 or above
2. JDK 1.6 or above
3. Eclipse 3.4 or above
4. TestNG Eclipse plug-in
   1. Following this [link](http://testng.org/doc/download.html) to install TestNG for Eclipse

## Browser requirements for testing desktop web-based application

**Need a correct version**

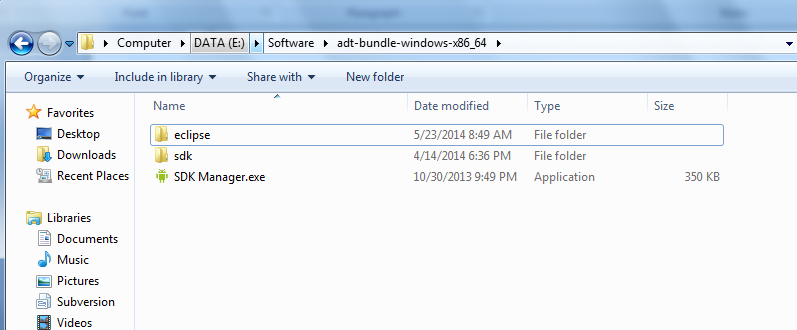
1. FireFox
2. Or Safari
3. Or Chrome
4. Or IE

## Software requirements for testing web mobile and native mobile application

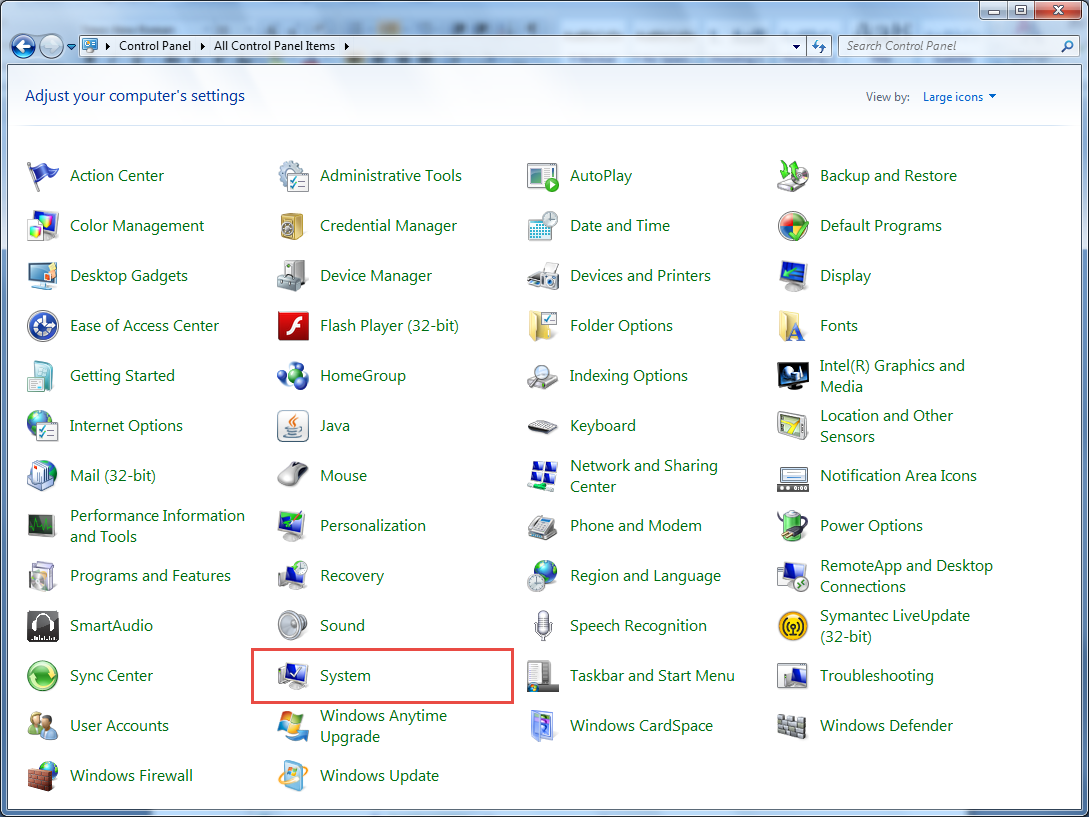
1. Appium
2. Android
   1. Browser:
      * Chrome
      * Built-in Android browser
3. iOS
   1. Browser
      * Safari

## Appium installation on Windows

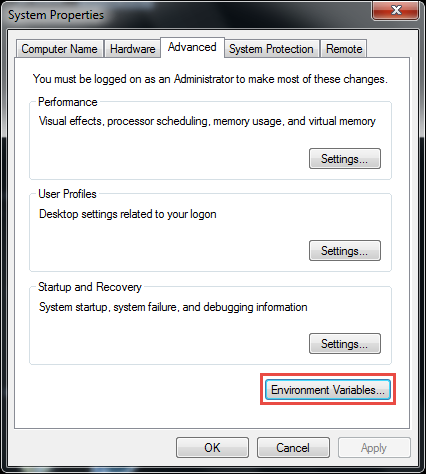
1. Download [Android SDK](http://developer.android.com/sdk/index.html)
2. Extract it in anyplace



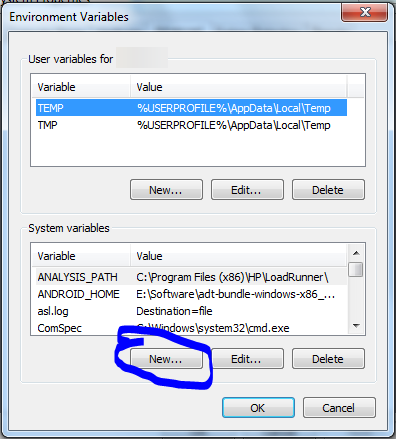
1. Android Environment Variable setup in Windows
   1. Go to Control panel, select System

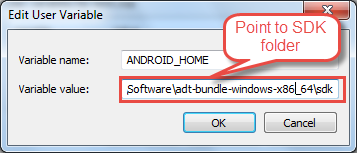


* 1. Switch to Advanced tab, select Environment Variable

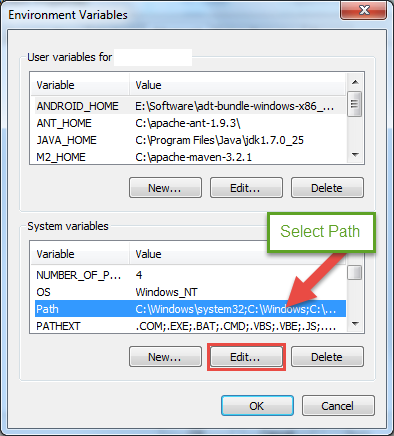


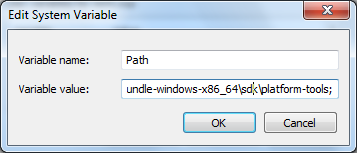
* 1. Create new System Variable





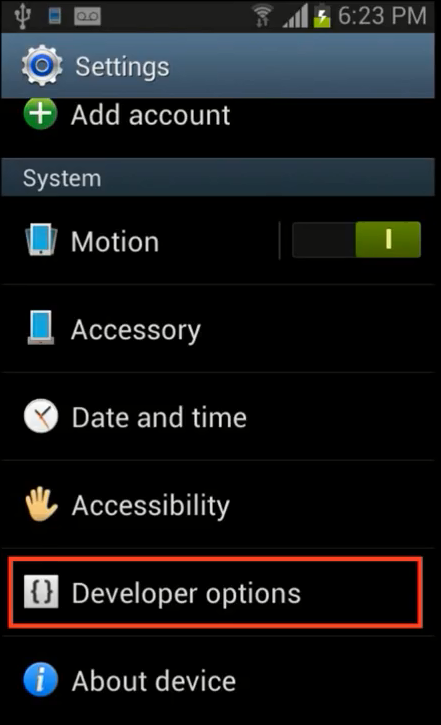
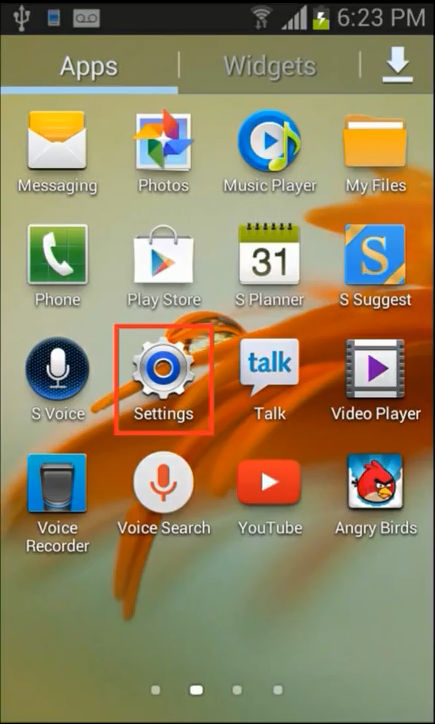
* 1. Edit system Path





Add more 2 directories to your path, depends on your Android SDK location, you can modify the path, example in my case (remember separate folders by ‘;’):

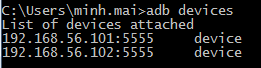
* %ANDROID\_HOME%\tools;
* %ANDROID\_HOME%\platform-tools;
  1. Enable “*Developer*” option in Android mobile



* 1. Test adb command
* Install USB driver for your android device, you can find and download driver for your device at this address : <http://developer.android.com/tools/extras/oem-usb.html>
* After installing driver for your device, connect it to your PC via USB port
* Open Command window and type : adb devices



* The result of above command is a list of connected Android device to your PC

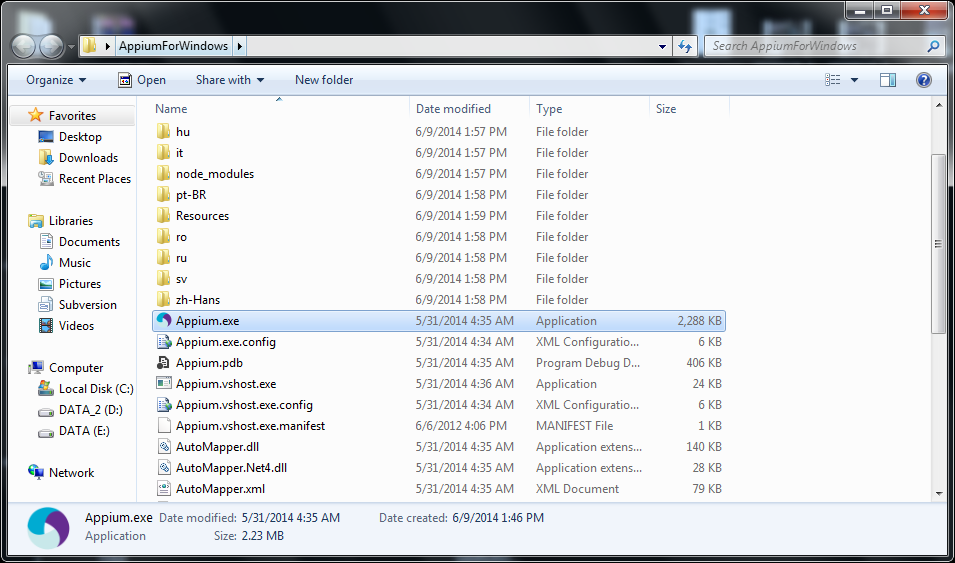


If your devices it not listed in the list, try to re-install newer version of driver for your device.

1. Download Appium for Windows
   1. Go to <http://appium.io/> and click Download Appium

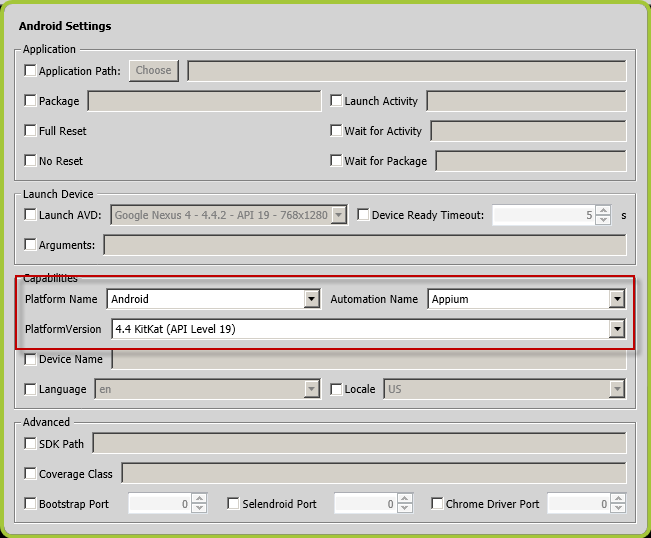


* 1. Extract the compressed files to any place, we have the portable Appium application



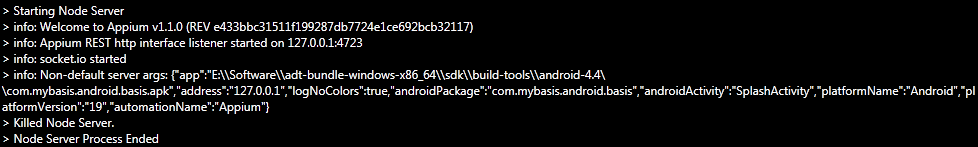
## Running Appium on Windows

1. Start Appium
2. Click on Android Logo () and setting like bellow



1. Leave other setting default
2. Click Launch ()

If the server starts successfully the screen looks like this:



## Appium installation on Mac OS X – automate native app on iOS

1. Software requirements:
   1. Mac OS X 10.7 or higher
   2. Xcode and Xcode command line tools (Xcode 5.0 or higher)
2. Installation:
   1. Install NodeJS with HomeBrew
      * Install Homebrew using command line

ruby -e "$(curl -fsSL https://raw.github.com/Homebrew/homebrew/go/install)"

* + - Brew update to ensure your Homebrew is up to date

brew update

* + - Add the Homebrew location to your $PATH and source your bash or zsh profile file after adding/saving this

export PATH="/usr/local/bin:$PATH"

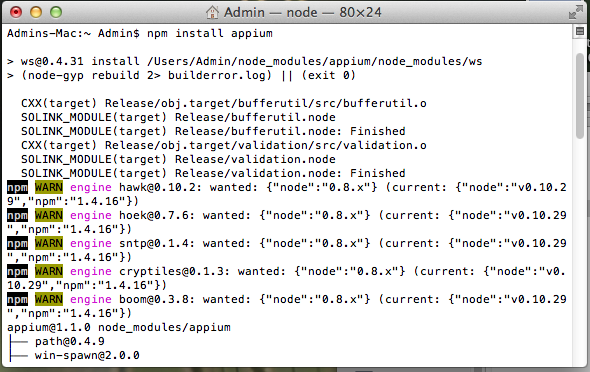
* + - Install Node

sudo brew install node

If this command failed, try this one:

sudo brew postinstall node

Check npm existed on /usr/local/bin. If not try:

ln –sf /usr/local/lib/node\_modules/npm/bin/npm-cli.js /usr/local/bin/npm

* 1. Install Appium by npm

npm install -g appium

Check existence of files: appium, appium-doctor, authorize-ios on /urs/local/bin

If not, try:

ln -sf /Users/<user>/node-modules/appium/bin/appium.js /usr/local/bin/appium

ln -sf /Users/<user>/node-modules/appium/bin/appium-doctor.js /usr/local/bin/appium-doctor

ln -sf /Users/<user>/node-modules/appium/bin/authorize-ios.js /usr/local/bin/authorize-ios

## Running Appium on Mac OS X – automate native app on iOS

Running this command line:

appium –a <address> -p <port>

Example: appium -a 127.0.0.1 -p 4725

If you want to running automation on real device, please follow these steps:

1. Install *ios webkit debug proxy*

cd /usr/local

git checkout 7e209f0 Library/Formula/libimobiledevice.rb

brew unlink libimobiledevice

brew install libimobiledevice

1. Launch *ios webkit debug proxy* and *appium*
   * Open terminal to run command

ios\_webkit\_debug\_proxy –c <udid> -d

* + Open new other terminal to launch appium

appium –a <address> -p <port> --app com.apple.mobilesafari –U <udid>

<address>: 127.0.0.1 or your machine’s IP

<port>: port number

<udid>: device UDID. Refer this [link](http://whatsmyudid.com/) to find your UDID

1. Make sure real device enable safari *Web Inspector*



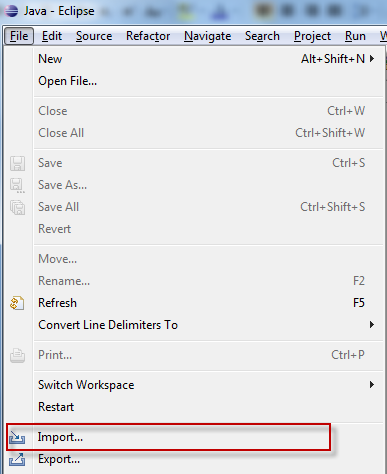
1. Install launcher: Refer to file [***Safari Launcher installation.docx***](Safari%20Launcher%20installation.docx)

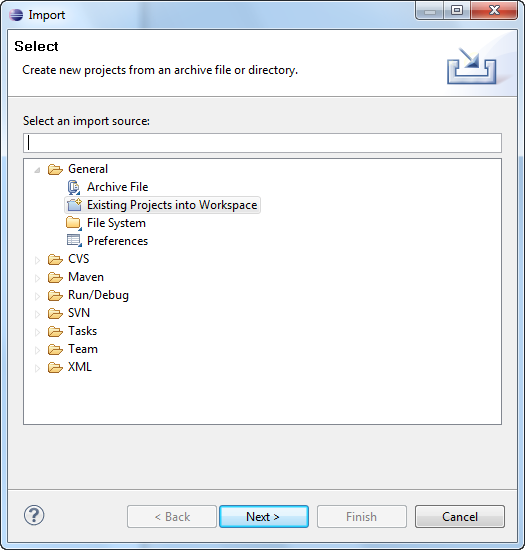
## Appium installation on Mac OS X – automate native app on OS X

Open AppiumforMac.app, this app will automatically configure server and start HTTP server on port 4622

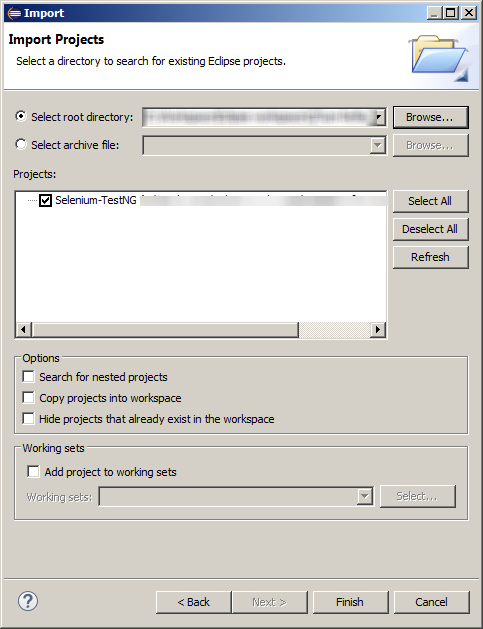
## Eclipse workspace

1. Import project **Selenium-TestNG** to Eclipse workspace
   1. File -> Import -> Existing Projects Into Workspace



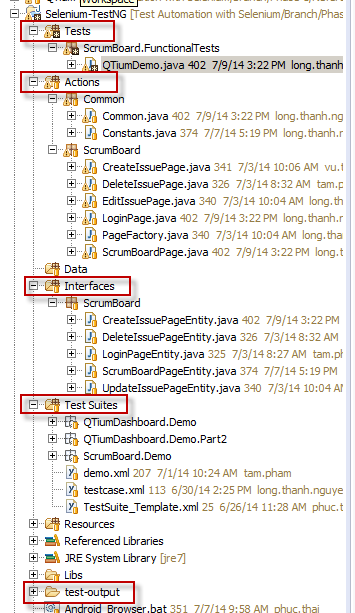


* 1. Browse folder that contain project **Selenium-TestNG** and select it to finish import



# WORKING WITH QTIUM FRAMEWORK

## Project structure



* **Tests**: test modules
* **Actions**: test action (business level actions, page level actions)
* **Interfaces**: interface of AUT
* **Test Suites**: TestNG test suite
* **test-output**: the test result generated by TestNG

## Popular classes

* **QTiumAutomation**: instance of this class is a automation object that has some built-in actions
* **QTiumAssert**: assertion tool class helper
* **QTiumSetting**: just only support for setting “object wait” only

## Test suite file

Using QTium Framework requires a test suite file that has a needed TestNG *listener* or other elements.

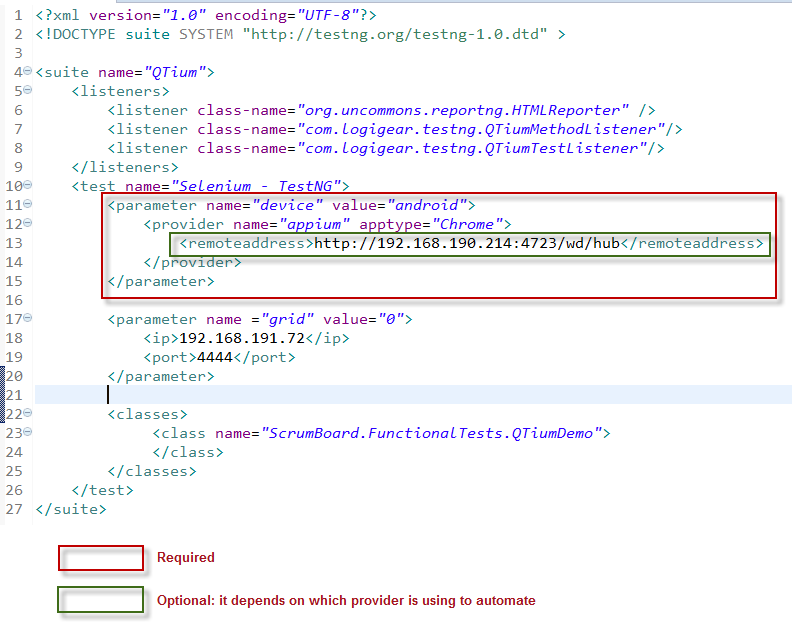
The following is an example of test suite that running Chrome browser on Android



1. Required listener



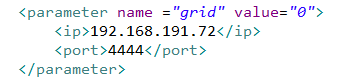
1. Required parameters:



Please refer to QTium provider matrix for more detailed cases.

1. Optional parameters

Use this parameter if you want to run on Selenium Grid mode



1. Required classes



**name**: full class name that is a test module

1. QTium provider matrix

This matrix helps you define a right test suite

