**Appium Overview**

**Appium**: Mobile software application's craze is increasing day by day. Also peoples are converting there current running software web sites in mobile version and creating **apps for mobile application software**. For testing point of view, It is very important for us to learn mobile's software application automation testing to stay connected with new technology. **Appium** is mobile application software testing tool which is currently trending in **mobile automation testing industry**.

**Apium** is very close to **selenium webdriver** software testing tool. It will be very easy to learn appium software testing tool for you if you already knows selenium webdriver. If you do not have knowledge of selenium then please refer selenium webdriver tutorials first.

**Appium Tutorial**

## Appium Tutorial: Set up Appium with Eclipse on Windows

***Introduction***

***Step 1: Install the Java Development Kit (JDK)***

***Step 2: Set Up Java Environment Variable Path***

***Step 3: Install Android SDK / ADB on Windows***

***Step 4: Install Android SDK Packages***

***Step 5: Set up Android Environment Variable***

***Step 6: Download and Install NodeJs***

***Step 7: Install Microsoft .net Framework***

***Step 8: Download And Install Appium Desktop Client***

***Step 9: Enabling Developer Mode Options on Android Phone or Tablet***

***Step 10: Install PdaNet to Connect with Android Device***

***Step 11: Install Eclipse IDE And Set up a Project***

***Step 12: Set Up Appium Project in Eclipse***

***Step 13: First Appium Test to Launch Amazon App***

## Appium On Mac

***Step 1: Install Eclipse on Mac OS***

***Step 2: Deploy Application to Real IOS Device***

***Step 3: SetUp Appium on Mac***

***Step 4: Build & Run Xcode Project***

***Step 5: Appium Test on Real IOS Device***

## Appium Tutorial: Basics

***Chapter 1: How to use Appium UIAutomatorViewer***

***Chapter 2: How to use Locators in Appium***

***Chapter 3: Inspect elements of Mobile Web Application***

***Chapter 4: How to Inspect and Automate WebView in Hybrid App***

## Appium Tutorial: Intermediate

***Topic 1: Execute Appium Test with TestNG***

***Topic 2 : How to perform Parallel Execution in Appium with TestNG***

## Appium Tutorial: Things to Know

**Topic 1: How to find aapPackage & aapActivity**

**Topic 2: How to transfer APK file from Phone to PC**

***Topic 3: ADB Commands***

***Topic 4: How to Connect Android Device to PC using Wifi***

## Appium Tutorial: Setting Up Virtual Devices

***Topic 1: How to Create an Android Virtual Device using Android Emulator***

***Topic 2: Creating Custom Device Definition Using AVD Manager***

# APPIUM Tutorial

APPIUM is a freely distributed open source mobile application UI testing framework.

Appium allows native, hybrid and web application testing and supports automation test on physical devices as well as on emulator or simulator both.

It offers cross-platform application testing i.e. single API works for both Android and iOS platform test scripts.

It has **NO** dependency on Mobile device OS. Because, APPIUM has framework or wrapper that translate Selenium Webdriver commands into UIAutomation (iOS) or UIAutomator (Android) commands depending on the device type not any OS type.

Appium supports all languages that have Selenium client libraries like- Java, Objective-C, JavaScript with node.js, PHP, Ruby, Python, C# etc.

* APPIUM Design Concepts
* Prerequisite to use APPIUM
* APPIUM Inspector
* APPIUM Installation on Windows
* Your First APPIUM Test Case for Native Android App
* Limitations using APPIUM
* Common Encountered Errors and Troubleshooting Steps in Appium

#### APPIUM Design Concepts

* Appium is an 'HTTP Server' written using Node.js platform and drives iOS and Android session using Webdriver JSON wire protocol. Hence, before initializing the Appium Server, Node.js must be pre-installed on the system.
* When Appium is downloaded and installed, then a server is setup on our machine that exposes a REST API.
* It receives connection and command request from the client and execute that command on mobile devices (Android / iOS).
* It responds back with HTTP responses. Again, to execute this request, it uses the mobile test automation frameworks to drive the user interface of the apps. Framework like:-
  + Apple Instruments for iOS (Instruments are available only in Xcode 3.0 or later with OS X v10.5 and later)
  + Google UIAutomator for Android API level 16 or higher
  + Selendroid for Android API level 15 or less

#### Prerequisite to use APPIUM

1. ANDROID SDK [[Link](http://developer.android.com/sdk/index.html)]-
2. JDK (Java Development Kit) [[Link](http://www.oracle.com/technetwork/java/javase/downloads/index.html)]
3. TestNG [[Link](http://testng.org/doc/download.html)]
4. Eclipse [[Link](http://www.eclipse.org/downloads/)]
5. Selenium Server JAR [[Link](http://docs.seleniumhq.org/download/)]
6. Webdriver Language Binding Library [[Link](http://docs.seleniumhq.org/download/)]
7. APPIUM For Windows [[Link](https://bitbucket.org/appium/appium.app/downloads/)]
8. APK App Info On Google Play [[Link](https://play.google.com/store/apps/details?id=de.migali.soft.apkinfo&hl=en)]
9. Node.js (Not Required - Whenever Appium server is installed, it by default comes with "Node.exe" & NPM. It's included in Current version of Appium.)

#### APPIUM Inspector

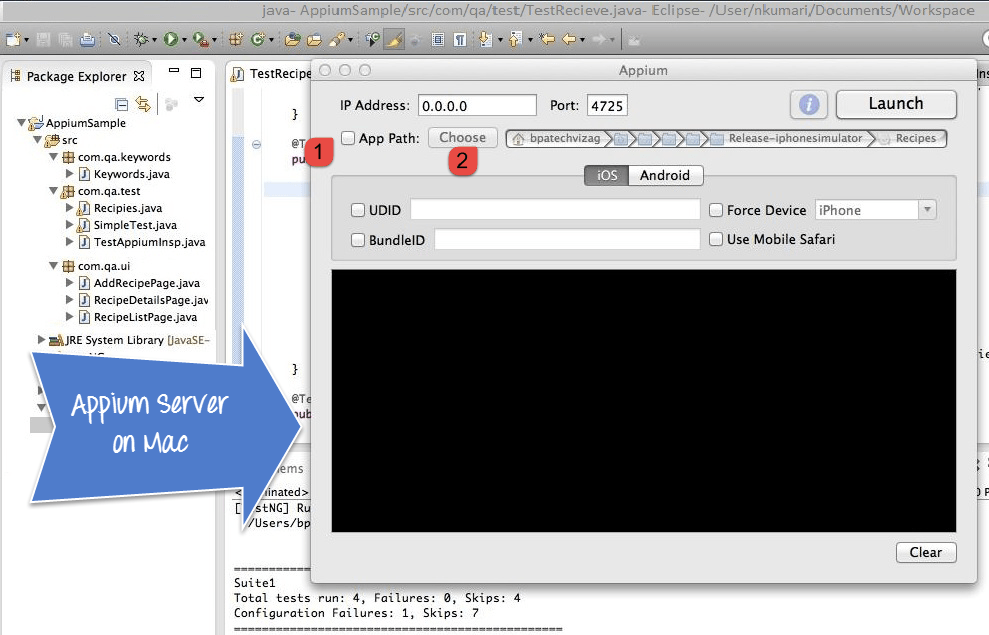
Similar to Selenium IDE record and playback tool, Appium has an 'Inspector' to record and Playback. It records and plays native application behavior by inspecting DOM and generates the test scripts in any desired language. However, currently there is no support for Appium Inspector for Microsoft Windows. In Windows, it launches the Appium Server but fails to inspect elements. However, UIAutomator viewer can be used as an option for Inspecting elements.

Steps to start with Appium Inspector on Mac machine:-

**Step-1** Download and start your Appium server with the default IP Address 0.0.0.0 and the port 4725.

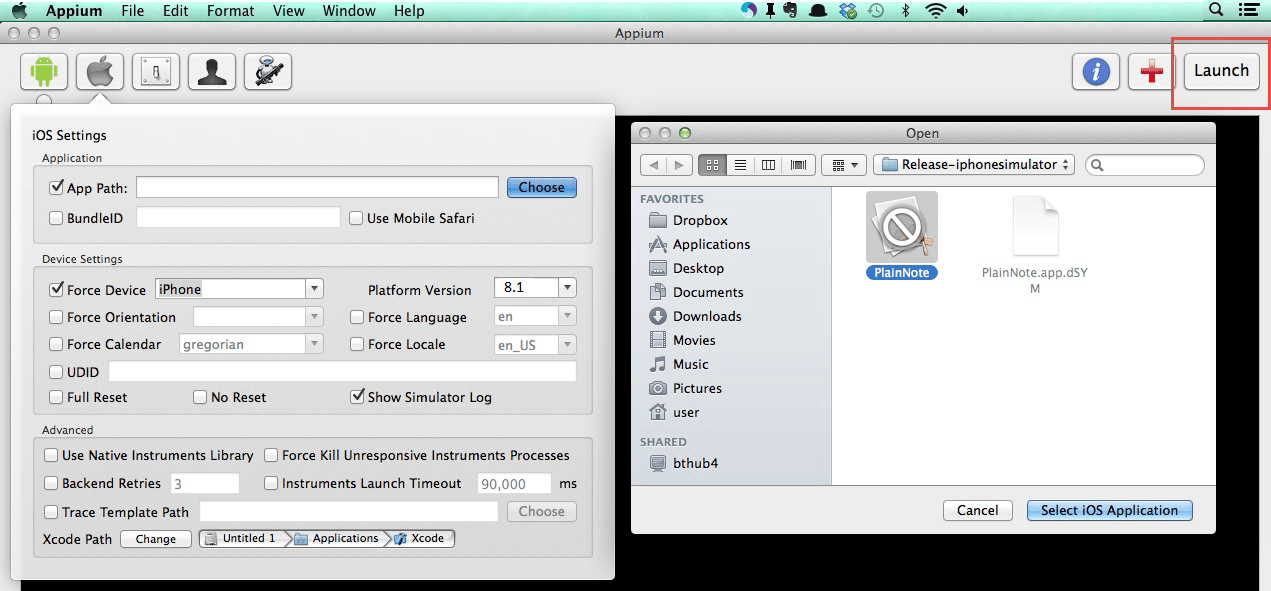
1. Select the source file or .app files from local to test.
2. Check the 'App Path' Checkbox to enable 'Choose' button.

Now, clicking on 'Choose' button will give the option to browse and select test file from the local drive.

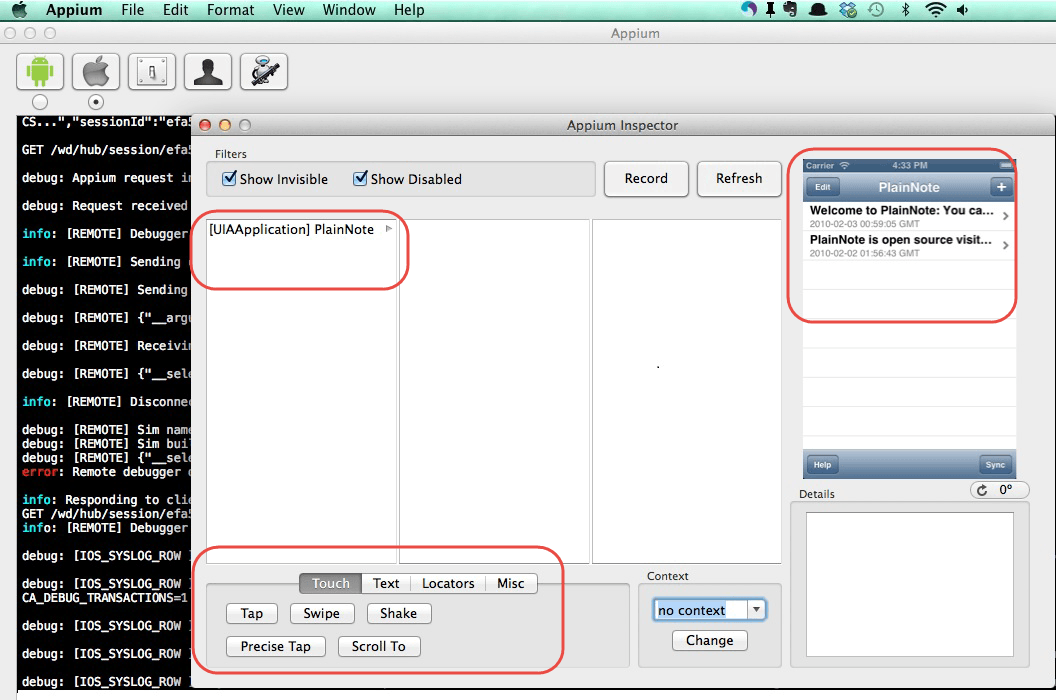


**Step 3-** Start Simulator on Mac machine.

**Step 4-** Click 'Launch' button from top right corner, which enable a blue color icon. Again, click on this blue color icon, it will open the Appium inspector and Simulator with pre-selected application.



**Step 5**- Launching your Appium Inspector will show the element hierarchy in column-wise structure. Also, user can apply actions using buttons like Tap, Swipe etc.

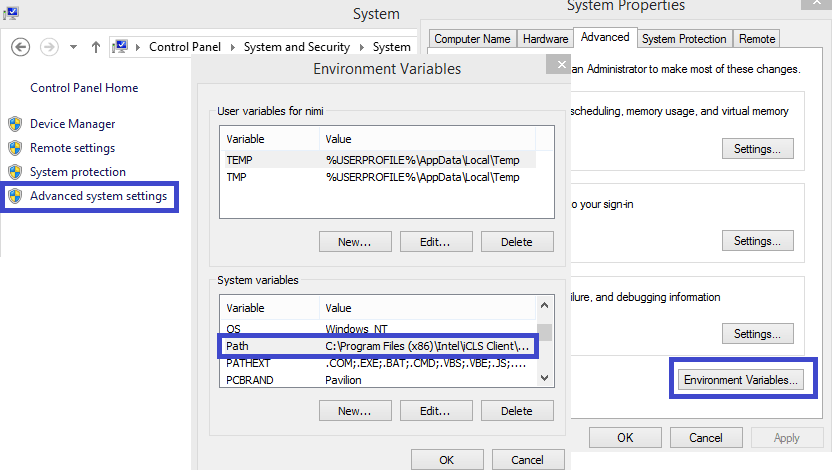


**Step 6-** Click on 'Stop' button to stop recording.

#### APPIUM Installation on Windows

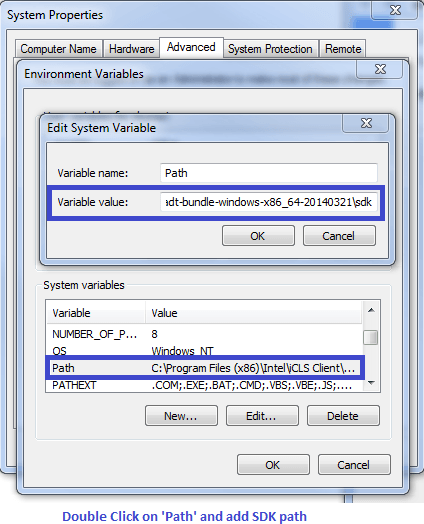
**Step 1-** Install Android SDK in your system.

a. Go to Control panel >> System and Security >> System and from the left panel click on 'Advance System Settings'. From 'System Properties' pop up, click on 'Advance' tab and then click on "Environment Variables" button.



b. Now, from 'Environment variables' pop up, 'double click on 'Path' and set ANDROID\_HOME variable that point to your SDK directory. In the path append the whole SDK folder path.

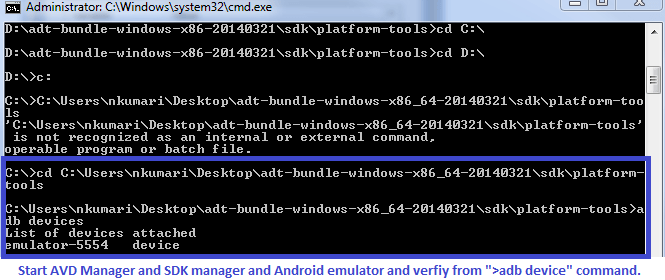
e.g. - C:\User\ABC\Desktop\adt-bundled-windows-x86\_64-20140321\sdk



**Step 2-** Start your Android emulator or any attach any Android device to your system (Make sure you have Android Debugging option enabled in your Android device. To check Debugging Option. Go to Device Settings >> Developer Options >> Check "Debugging Option").

**Step 3-** Open Command Prompt and navigate to your Android SDK's \platform-tools\ directory (E.g. D:\adt-bundle-windows-x86\_64-20130514\sdk\platform-tools).

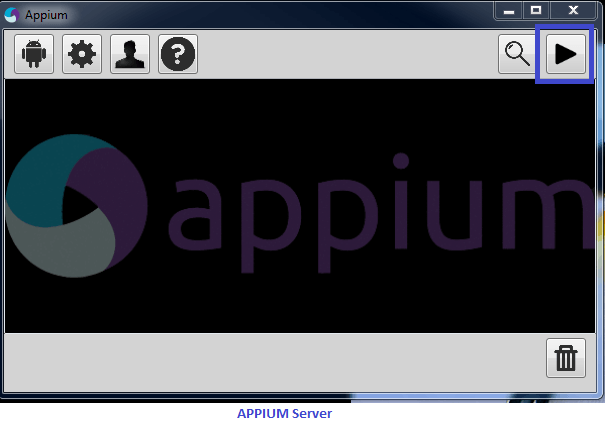
**Step 4**- Run 'adb devices' command. You can see your connected device listed in Command Prompt window. (In CMD write '>adb devices'- This command will list the connected emulator instances. E.g.: adb –s emulator-5554 install <Location of .apk file>)



**Step 5**- Run 'adb start-server' command. It will start ADB server that will be used by Appium to send commands to your Android device.

**Step 6-** Now, navigate to Appium directory in your system and start Appium by clicking Appium.exe file.

**Step 7-** Do not alter the IP address or port number and click 'Launch' button. Your Appium console start at 127.0.0.1:4723 as shown in below.



Steps 8- Click on 'Start' button, Appium server started running on your system.

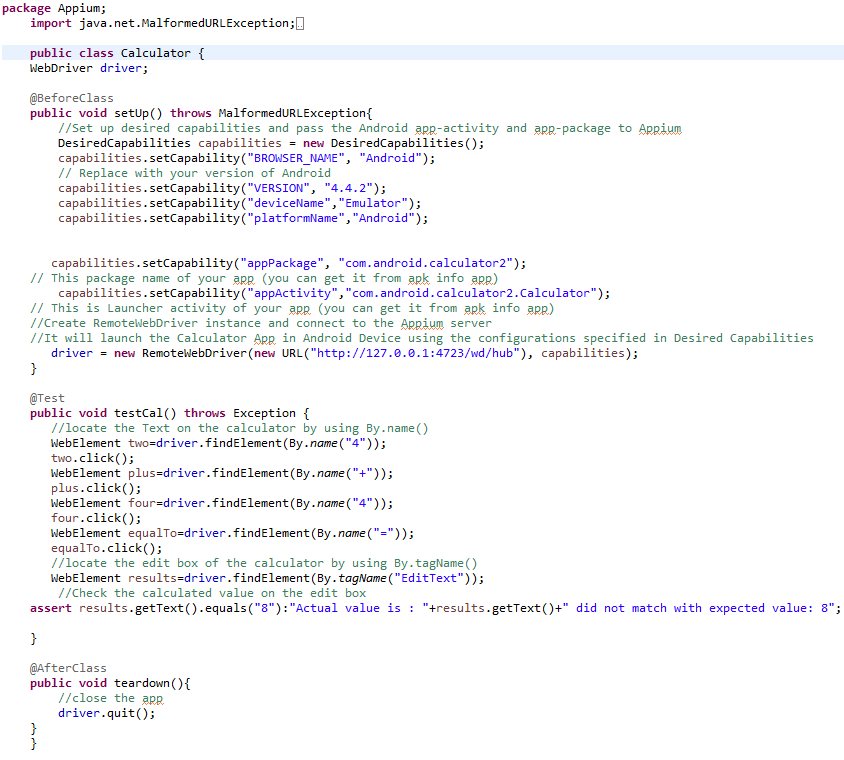
#### Your First APPIUM Test Case for Native Android App

**Step 1**) Download ADT eclipse plugin or download ADT bundled separately here

**Step 2)** Open Eclipse and Create a new Project >> Package >> Class

**Step 3)** Import Selenium library and TestNG inside that new project.

**Step 4)** Now Create a small test Program for 'Calculator.app' to sum two numbers.

[](http://cdn.guru99.com/images/1-2015/012315_1021_Introductio8.png)

Download the above Appium Code

Appium Server and Android Emulator from 'AVD Manager' and Click Run >> TestNG. Above program will run the 'Calculator.app' on selected emulator and Result displayed under Eclipse console using TestNG framework.

#### Limitations using APPIUM

1. Appium does not support testing of Android Version lower than 4.2
2. Limited support for hybrid app testing. eg: not possible to test the switching action of application from the web app to native and vice-versa.
3. No support to run Appium Inspector on Microsoft Windows.

#### ****Common Encountered Errors and Troubleshooting Steps in Appium****

|  |  |
| --- | --- |
| **Error** | **Troubleshooting Steps** |
| 1. error:- The following desired capabilities are required, but were not provided:device Name, platformName | 1. Add desired capabilities: device Name, platformName in APPIUM script. e.g:capabilities.setCapability("deviceName","Emulator"); capabilities.setCapability("platformName","Android"); |
| 2. error: Could not find adb. Please set the ANDROID\_HOME environment variable with the Android SDK root directory path. | 2. You probably need to set up SDK root directory path in system 'Environment Variables' in 'Path' column |
| 3.error:org.openqa.selenium.SessionNotCreatedException: A new session could not be created. | 3. You need to set a correct App path and restart the Appium server. |
| 4. How to find DOM element or xPath in mobile application? | 4. Use 'UIAutomatorviewer' to find DOM element for Android application. |

# Uiautomatorviewer Tutorial

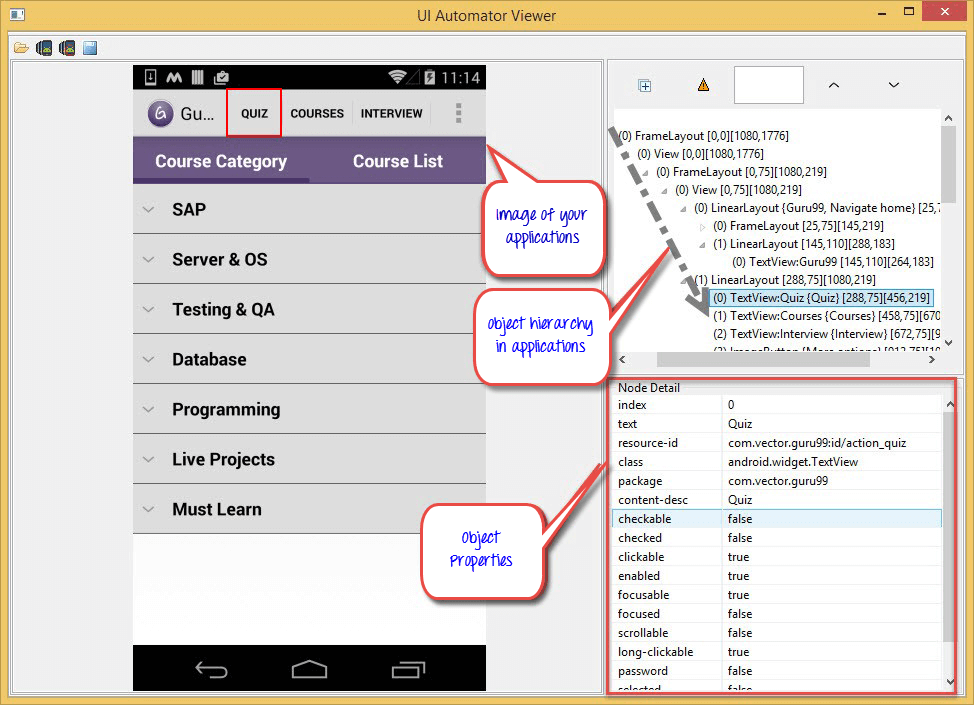
Identifying Android application (AUT) objects using Uiautomatorviewer to write automation scripts

To automate any android application using Appium, user needs to identify the objects in AUT (Application under test).

While executing automation scripts, Appium uses "Uiautomatorviewer" to identify different properties of the object and use the properties to identify the required object.

### What is it?

"UIautomatorviewer" is a GUI tool to scan and analyze the UI components of an android application. With "UIautomatorviewer", you can inspect the UI of an android application in order to find out the hierarchy and view different properties (id, text…) of the element.

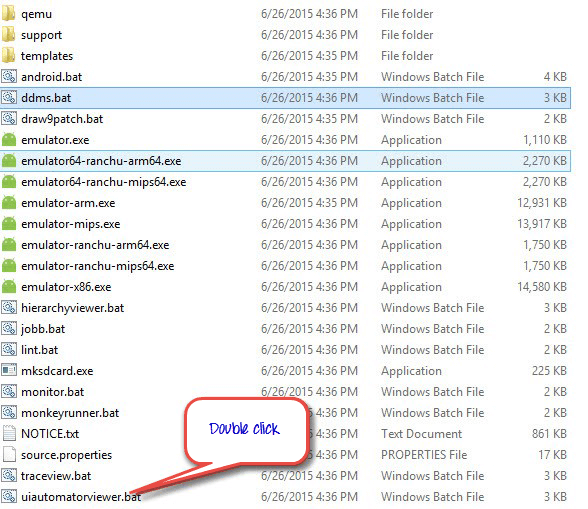


### Where to get it?

"Uiautomatorviewer" is a part of the Android SDK manager and will be accessible once you install the sdk manager. Download and install Android SDK manger from here

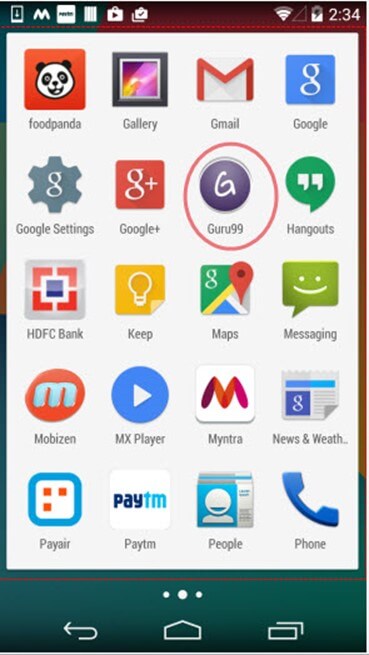
### How to get started?

Once Android SDK installed, navigate to link **c:\users\<username>\AppData\Local\Android\sdk\tools** you'll notice a batch file with name '**uiautomatorviewer.bat**'. Double click on it to launch "Uiautomatorviewer" GUI

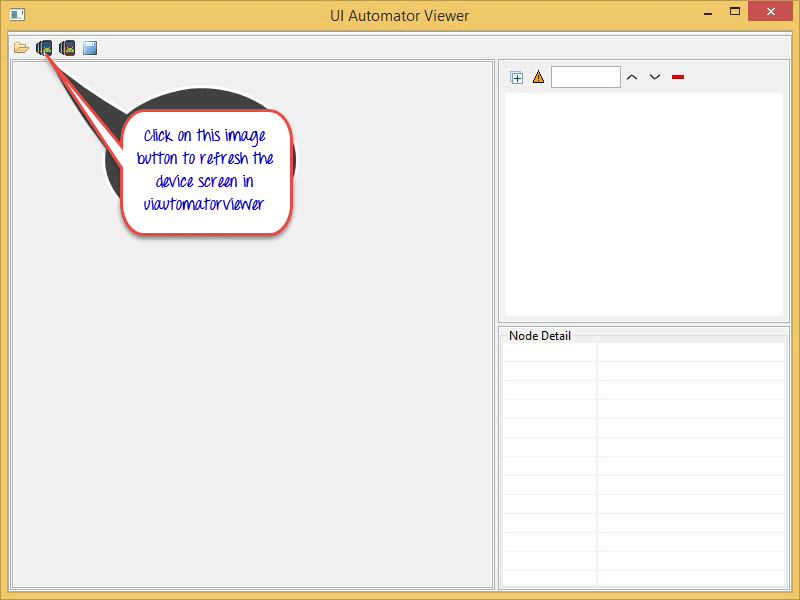


### How to use Uiautomatorviewer to find objects in my application

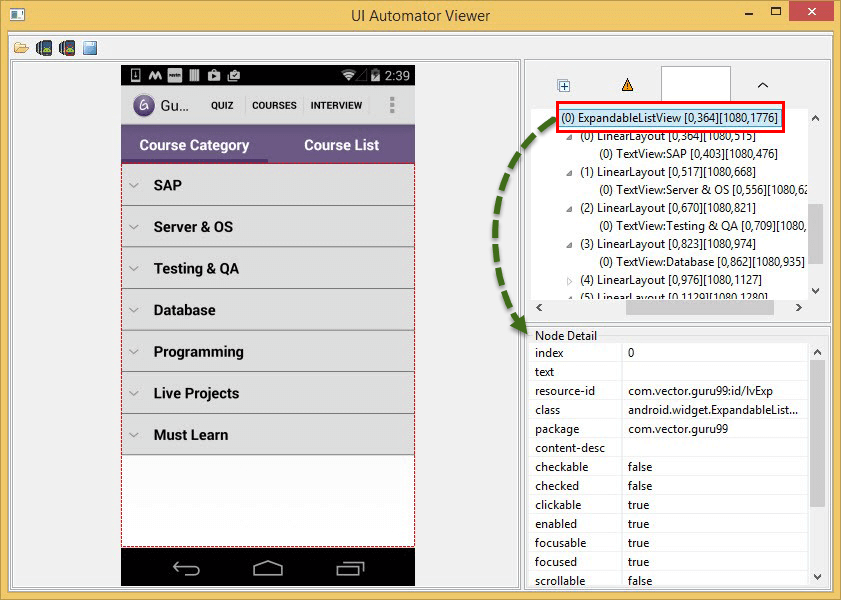
1. Enable "developer" options on your device. Click here to know how to enable developer options on Android devices
2. Connect your android device to PC via USB cable
3. Select "Guru99" app from applications



1. Click the 'Device screenshot' button to refresh the "Uiautomatorviewer" and to load the guru99 application GUI on "Uiautomatorviewer"



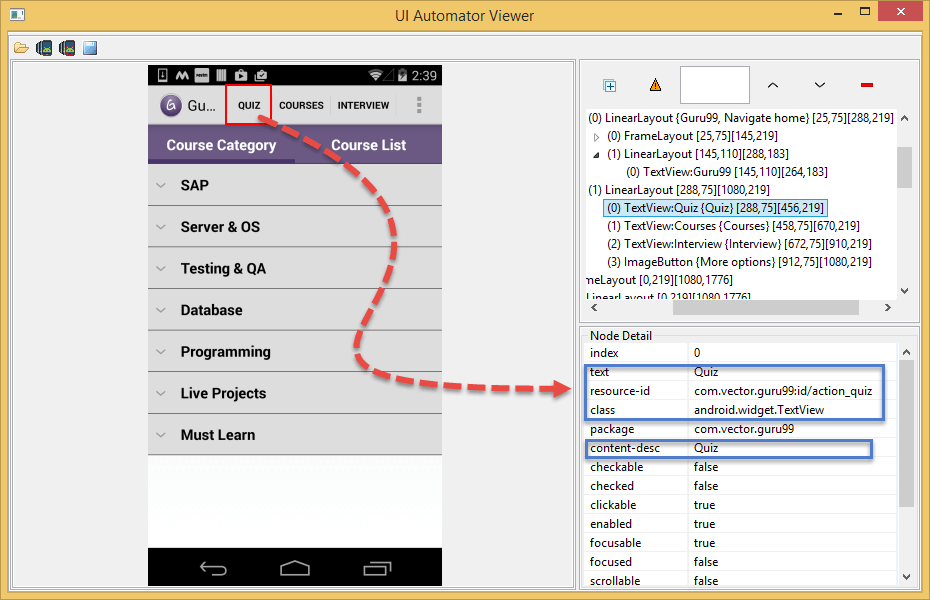
1. After refresh is completed a screenshot of Guru99 application opens



1. As you see in the above image, on the right side of the window there are 2 panels.

* Upper panel contains node hierarchy the way the UI components are arranged and contained, clicking on the each node gives properties of UI element in the lower panel

1. Select '**Quiz**' button in the above image to view different properties (text, resource-id...)



### How to use these properties to identify elements for automation

Well you cannot use the properties directly, each property has other names. Let's see how to use those properties values to work. Following attributes can be used to identify '**Quiz**' button in Guru99 app.

* **text** attribute can be used as "**name"**

[Uiautomatorviewer Tutorial](http://cdn.guru99.com/images/9-2015/082715_1132_Identifying7.png)

* **resource-id** attribute can be used as "**id"**

[Uiautomatorviewer Tutorial](http://cdn.guru99.com/images/9-2015/082715_1132_Identifying8.png)

* **class** attribute can be used as "**className"**

[Uiautomatorviewer Tutorial](http://cdn.guru99.com/images/9-2015/082715_1132_Identifying9.png)

* **content-desc** attribute can be used as "**AccessibilityId"**

[Uiautomatorviewer Tutorial](http://cdn.guru99.com/images/9-2015/082715_1132_Identifying10.png)

Along with above attributes we can write xpaths for object identification

### Error one might encounter while using Uiautomatorviewer

* I'm seeing the error- "No Android devices were detected by adb" as shown in below screen shot how I can resolve this

