**Android Automation Learning**

Learning Apium

Steps followed:

Step1: Started following the steps provided in <https://www.guru99.com/introduction-to-appium.html> and Java need to be installed

Step2: Downloaded Android Studio <https://developer.android.com/studio> and installed just by following click next

Step3: Make sure that Java path and Android path is set properly.

Go to Advance settings 🡪 Environment variable and set ANDROID\_HOME = C:\Users\Kadiahnakoppal\_Harip\AppData\Local\Android\Sdk

Path=%ANDROID\_HOME%\platform-tools

Step4: Downloaded Appium from <http://appium.io/> and install, just follow click next to install it.

Step5: You can start the appium server / you can start the new session also (File-> New Session Window)

Step6: Go to ‘C:\Users\Kadiahnakoppal\_Harip\AppData\Local\Android\Sdk\platform-tools’ in command prompt and run ‘adb devices’. Then You can see List of devices there

Step7: I can find Appium in windows screen , just by clicking I started the Appium server

Step8: Installed eclipse and created project named android Then Create Create a new Project >> Package >> Class

Class name ‘Calculator’

import java.net.MalformedURLException;

import java.net.URL;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

//import org.openqa.selenium.remote.CapabilityType;

import org.openqa.selenium.remote.DesiredCapabilities;

import org.openqa.selenium.remote.RemoteWebDriver;

import org.testng.annotations.\*;

public class Calculator {

WebDriver driver;

@BeforeClass

public void setUp() throws MalformedURLException{

//Set up desired capabilities and pass the Android app-activity and app-package to Appium

DesiredCapabilities capabilities = new DesiredCapabilities();

capabilities.setCapability("BROWSER\_NAME", "Android");

capabilities.setCapability("VERSION", "4.4.2");

capabilities.setCapability("deviceName","Emulator");

capabilities.setCapability("platformName","Android");

capabilities.setCapability("appPackage", "com.android.calculator2");

// This package name of your app (you can get it from apk info app)

capabilities.setCapability("appActivity","com.android.calculator2.Calculator"); // This is Launcher activity of your app (you can get it from apk info app)

//Create RemoteWebDriver instance and connect to the Appium server

//It will launch the Calculator App in Android Device using the configurations specified in Desired Capabilities

driver = new RemoteWebDriver(new URL("http://127.0.0.1:4723/wd/hub"), capabilities);

}

@Test

public void testCal() throws Exception {

//locate the Text on the calculator by using By.name()

WebElement two=driver.findElement(By.id("digit\_2"));

two.click();

WebElement plus=driver.findElement(By.id("op\_add"));

plus.click();

WebElement four=driver.findElement(By.id("digit\_4"));

four.click();

WebElement equalTo=driver.findElement(By.id("eq"));

equalTo.click();

//locate the edit box of the calculator by using By.tagName()

WebElement results=driver.findElement(By.id("result"));

//Check the calculated value on the edit box

assert results.getText().equals("6"):"Actual value is : "+results.getText()+" did not match with expected value: 6";

}

@AfterClass

public void teardown(){

//close the app

driver.quit();

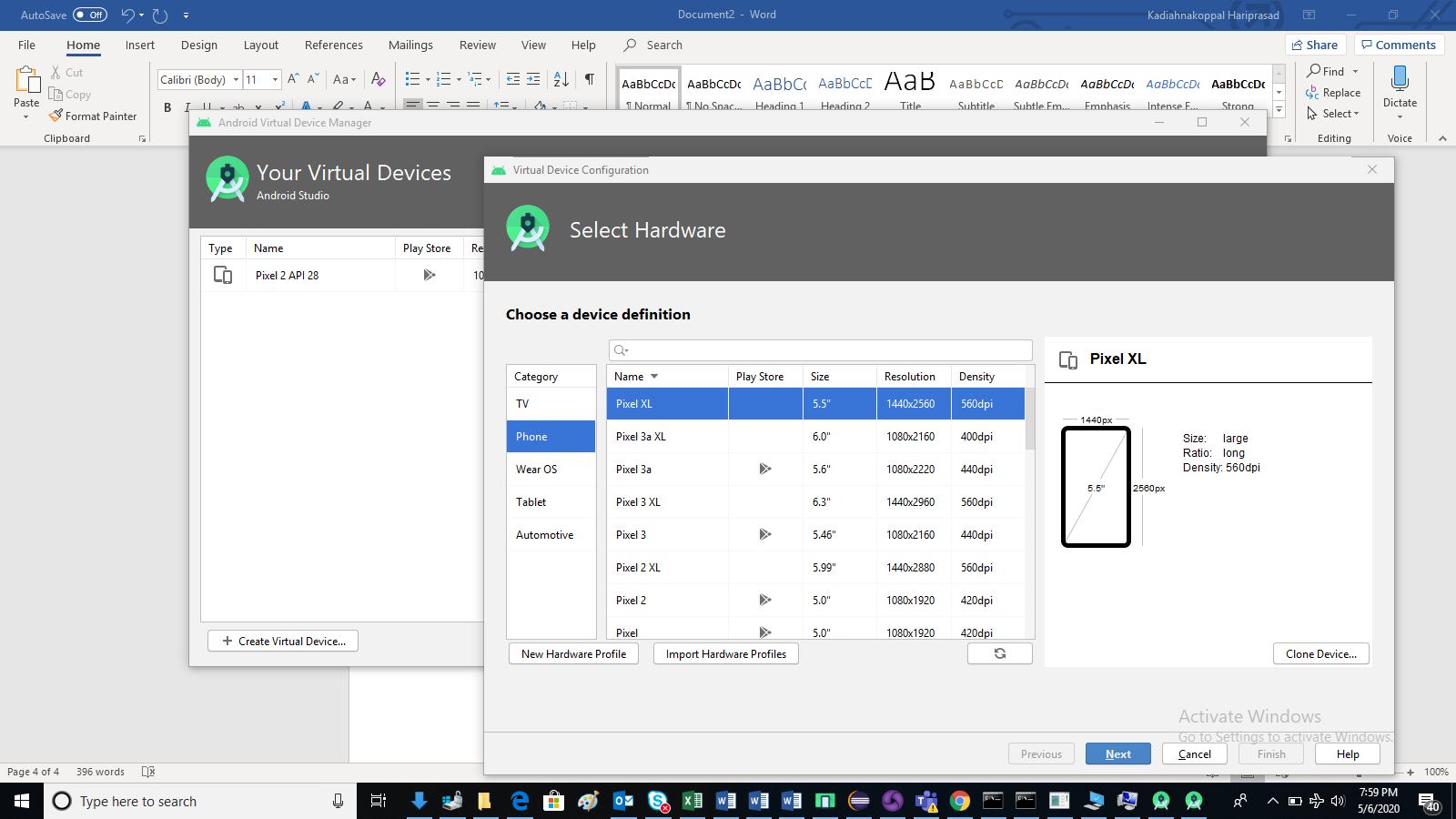
}

}

Step8: Download ‘selenium-server-standalone-2.53.1.jar’ and ‘testng.jar’ and place it in eclipse lib folder and build path. I think for testNG have install the plugin in eclipse and resolve error if it is.

Step9: Actually I opened the android studio and created the emulator device(test device). How I did it -🡪

I have opened android virtual device manager -> Click on Create virtual Device -> Downloaded one device and configured it then clicked on start button the mobile got opened



Step10: Just run the calculator.java file in eclipse (right click and choose testNg) The calculator should open and run the program.

Step11: There will be a chance of test may fail because of locator issue. We don’t have any option(support) to identify the locator using the Appium in windows so we have to use uiautomatorviewer for that click on “uiautomatorviewer.bat” file(This comes as part of android Sdk tool located in ‘C:\Users\Kadiahnakoppal\_Harip\AppData\Local\Android\Sdk\tools\bin’) or anywhere inside the “…sdk\tools\” folder. This opens UI Automator Viewer UI Window. Once we click on device screenshot icon which is in top left screen you can see whatever is there in emulator same screen will appear. And Just click on any of the item we can see the screen got replicated over there.

**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](https://www.guru99.com/xpath-selenium.html)in a mobile application? | 4. Use 'UIAutomatorviewer' to find DOM element for Android application. |

**Interview Questions related to Appium**

**1) Explain what is Appium?**

Appium is a freely distributed open source mobile application UI[Testing](https://www.guru99.com/software-testing.html)framework.

**2) List out the Appium abilities?**

Appium abilities are

* Test Web
* Provides cross-platform for Native and Hybrid mobile automation
* Support JSON wire protocol
* It does not require recompilation of App
* Support automation test on physical device as well as similar or emulator both
* It has no dependency on mobile device

**3) List out the pre-requisite to use APPIUM?**

Pre-requisite to use APPIUM is

* ANDROID SDK
* JDK
* TestNG
* Eclipse
* Selenium Server JAR
* Webdriver Language Binding Library
* APPIUM for Windows
* APK App Info On Google Play
* js

**4) List out the limitations of using Appium?**

* Appium does not support testing of Android Version lower than 4.2
* Limited support for hybrid app testing. E.g., not possible to test the switching action of application from the web app to native and vice-versa
* **No support to run Appium Inspector on Microsoft Windows – We can use *UIAutomateviewer*** **for this in Windows, this comes as part of the SDK**

**6) Explain the design concept of Appium?**

* Appium is an "HTTP Server" written using[Node.js](https://www.guru99.com/node-js-tutorial.html)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)

**7) What language does Appium support?**

Appium support any language that support HTTP request like Java,[JavaScript](https://www.guru99.com/interactive-javascript-tutorials.html)with Node.js, Python, Ruby, PHP, Perl, etc.

**8) Explain the pros and cons of Appium?**

**Pros**:

* For programmer irrespective of the platform, he is automating ( Android or iOS) all the complexities will remain under single Appium server
* It opens the door to cross-platform mobile testing which means the same test would work on multiple platforms
* Appium does not require extra components in your App to make it automation friendly
* It can automate Hybrid, Web and Native mobile applications

**Cons**:

* Running scripts on multiple iOS simulators at the same time is possible with Appium
* It uses UIAutomator for Android Automation which supports only Android SDK platform, API 16 or higher and to support the older API's they have used another open source library called Selendroid

**10) Mention what are the basic requirement for writing Appium tests?**

For writing Appium tests you require,

* **Driver Client**: Appium drives mobile applications as though it were a user. Using a client library you write your Appium tests which wrap your test steps and sends to the Appium server over HTTP.
* **Appium Session**: You have to first initialize a session, as such Appium test takes place in the session. Once the Automation is done for one session, it can be ended and wait for another session
* **Desired Capabilities**: To initialize an Appium session you need to define certain parameters known as "desired capabilities" like PlatformName, PlatformVersion, Device Name and so on. It specifies the kind of automation one requires from the Appium server.
* **Driver Commands**: You can write your test steps using a large and expressive vocabulary of commands.

**12) Do you need a server machine to run tests on Appium**?

No, you don't need server machine to run tests on Appium. Appium facilitates a 2-tier architecture where a test machine connects to a test server running Appium and automating the whole thing. **You can have Appium running on the same machine where your test runs.**

#### **What are Emulators?**

An emulator is a software that mimics the hardware and software of the target device on your computer. They do this by translating the ISA (Instruction Set Architecture) of the target device to the one used by the computer you are using to conduct testing using binary translation.

#### **What are Simulators?**

A simulator is a software that helps your computer run certain programs built for a different Operating System. They are mostly meant for iPhone and iPad devices, unlike Android devices that can be emulated easily.

### Question 1. What Is Mobile Application Testing And How Is It Different From Mobile Testing?

[Mobile Application Testing](https://www.edureka.co/blog/mobile-application-testing/) (MAT) is the testing of an application on mobile devices and it is different from Mobile Testing (MT) in the term that, in MT we focus on the native application features of the Mobile devices like Call, SMS, Media Player, etc. Meanwhile, in MAT we focus only on the functionality & features of the application under Test.

### Question 9. What Is Appium’s Strongest Point in your opinion?

Appium is based on [Selenium](https://www.edureka.co/blog/selenium-tutorial) which is an HTTP protocol by Google designed to automate browsers. The idea is actually very nice as automating an app (especially a webview-based one) is not so different (in terms of required APIs) from automating a browser.

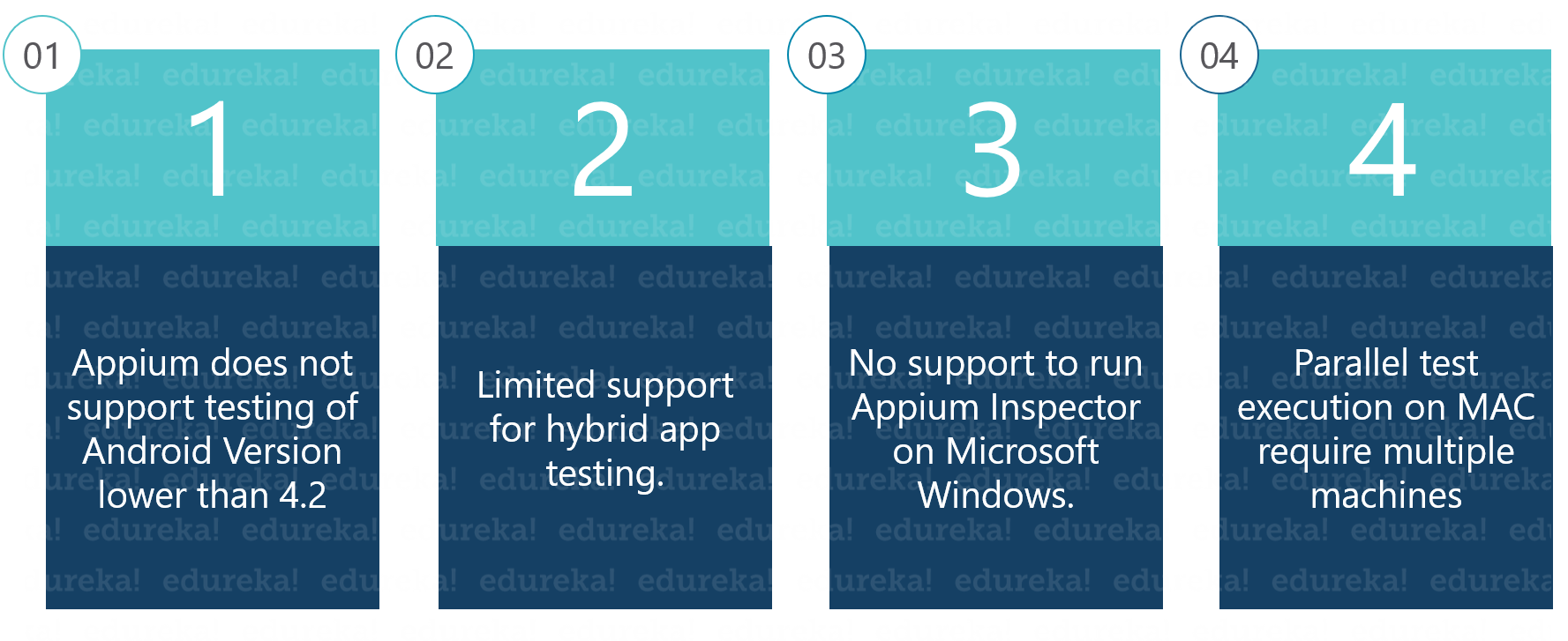
### Question 10. What Is Appium?

Appium is an open source, cross-platform [automation](https://www.edureka.co/blog/rpa-tutorial/) testing tool. It is used for automating test cases for **native, hybrid and web applications**. The tool has a major focus on both Android and [iOS apps](https://www.edureka.co/blog/swift-tutorial) and was only restricted to the mobile application testing domain.

Recently, a few updates back**, Appium also announced that they will support the testing of desktop applications for windows.**

**Appium is developed and maintained by Sauce Labs.**

### Question 14. List Out The Limitations Of Using Appium?



**Question 40: List The Selenium Commands That Work With Appium.**

There are a number of Selenium commands that work with Appium tool.

* Locate commands using ID or class names.
* Raise events on elements **e.g. Click()**.
* Text commands like **type()**.
* **Get/Set** element properties.
* Commands to run JavaScript.
* Switch context between different web views like switching **<*iFrames*>** in Selenium Webdriver.
* Commands to manage alert boxes

**Question 41: List OneThing Which You Cannot Do With Emulators But You Can Do With A Real Device.**

You can test the interrupts like

* Phone calls & Messages
* Battery drains out while using the application under test
* Low battery scenarios
* Memory card mount/unmount scenarios
* Actual performance of your application
* Bluetooth related testing.

Sample xpaths:

//input[@id=’email’]

//\*[contains(@name, ‘sub’)]

//img[contains(@src, ‘spirit’)]

//img[starts-with(@src, ‘spirit’)]

//a[contains(text(),’order’)]

//input[@name=’username’] | //input[@id=’login-username’]

//input[@type='submit']

//input[@value='Search' and @type='submit']