**Appium Interiew Question Main**

* **What is Appium?**

Appium is an open source, free wire mobile application testing tool. With appium you can automate both Android and Ios app as well. It’s a cross platform and cross language and an extension to selenium.

Robbotium is an open source but for only Android.

Calabash is for both android and ios which you need to learn a new language from scratch.

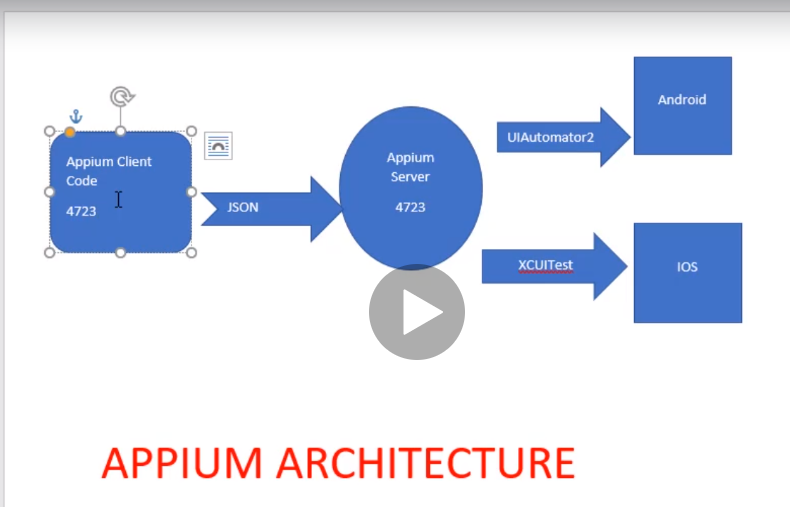
* **What are the platform?**

1. IOS
2. Android
3. Firefox OS

* **What are the languages supported by appium?**

Java, c#, Javascript, Python, Ruby

* **Appium working architecture?**



* 1. Download and start the Appium server in a specific port.
  2. Write your code in IDE and use the port.
  3. Use the same port in your code to redirect to Appium server.
  4. Appium server interprets those code and execute the commands on devices.
  5. When you will run the program your program will convert into json format by the help of Appium server as Appium only except json format.
  6. The direction that we traverse same direction we get the Reponses and java converted it to the user.
  7. If you want run your script in android then we have to use the android framework UIautomator to perform android testing.
  8. For IOS we have the ios frame work xcultest which Appium will use for ios testing.
* **How appium works?**

MOBILE

app

APPIUM

SERVER

DRIVER

CLIENT

DRIVER CLIENT 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.

Code follows Json wire protocol while sending the client script to the and Appium server receives code as the form of session object or headless browser. The server interpret and execute commands on the mobile app. Appium server work as media.

* **What are the types of mobile app?**

There are 3 types of app: 1.Native app 2.Web App 3. Hybrid app.

* **What is Native app?**

A native application is a software program that is developed for use on a particular platform or device. This applications are developed with single mobile operating systems like ios,android or windows this applications cannot be used on a platform other than their own in which application is been developed and design. User will be not able to use ios application in android or android application in windows. Advantages of these applications are that they are highly responsive with accurate UI. Native applications can be downloaded from their web stores with meeting users end.

* **What is Web App?**

A web app works via a web browser on the mobile but requires either a cell signal or wi-fi to function. **Web apps** as responsive versions of website to work on any **mobile device**.

* **What is Hybrid App?**

A hybrid application (hybrid app) is one that combines elements of both native and Web applications. Native applications are developed for a specific platform and installed on a computing device. ... Hybrid apps are often mentioned in the context of mobile computing.

* **What are Android app and ios app format?**
* Android-.apk
* IOS-.app
* **Which app is better native or hybrid?**

**Native apps** are built especially for the platform they're to be used on (iOS,Android etc). React **Native** allows a proportion of the code to be shared between platforms and empowers developers to create **apps** which feel less clunky and perform a**better** than **hybrid apps**.

* **What is emulator and simulator?**

**Emulator is a virtual mobile device that runs on computer for Android application testing with** without using a physical device. It duplicates the inner works of real device. Emulator is slower.

A **simulator** is an environment for testing IOS Application. It duplicates the behavior of the real device. Simulator is faster.

* **What is Android browser and IOS browser?**

Android default browser is chrome. IOS is safari.

* **How long have you worked in mobile app testing and web application testing in Nike?**

I have joined in Nike as a QA Automation engineer and first three months I was involved in a web automation project. Then I started working in a Mobile automation project for Android version name HDP Project which I was doing almost 14 months for a complete mobile project. From last 12 months I have worked on the web app of the HDP project on the other hand our ios version work is also going on. Explain your last project.

Based on the client requirement we chose android version first.

* **Which tools, devices and version you used did you use?**

I have used Appium for mobile automation tool, and we tested mainly on the Android real devices.

* Appium version 1.6.4
* **Android 7.0 Nougat has finally become the most-used** version of the mobile operating system, running on 28.5 percent of devices (across both versions 7.0 and 7.1),
* Devices : Emulator, Real Devices (Samsung Galaxy S7/S7 Edge 5.1”, samsung galaxy s8(most selling 5.8” screen), Google Pixel-5” screen.
* **What are the pre requisites?**

1. Java 1.8
2. Android SDK
3. Node.js

* Appium comes through node modules.
* Npm->to download all node modules.

1. Download Appium Server

* npm install –g appium (install appium)
* appium –v (to check appiumverison)
* appium (to start the server)

1. Download Appium-client jar file
2. Install appium-doctor

install appium-doctor (install appium doctor

appium-doctor (to check all the configuration are okay or not for appium?)

1. Real devices

* **How did you configure appium android testing?**

Open cmd type appium-doctor to see all the configurations are ok, Start the Appium server, Open Android UI Automator Viewer to inspect element, Open emulator from cmd and Connect and setup real android device. First you have to set the android .apk(app) path. Then by the help of **Desired Capabilities you need to configure** PlatformName, PlatformVersion, Device Name and NEW\_COMMAND\_TIMEOUT, "100") means How long (in seconds) Appium will wait for a new command. In the driver client, You can write your test steps or commands. Then it follows json wire protocol to send the data in Appium server is an HTTP SERVER. The server interpret those commands and execute this commands on mobile devices. You can see the server response on your appium log. Then you have to create and object of android driver setup the host,port,wd and hub. By using the reference you can perform.

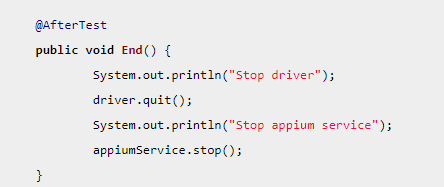
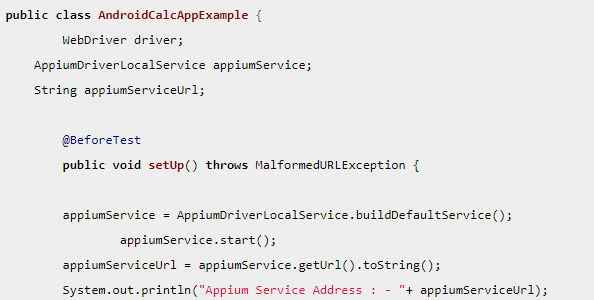
|  |  |
| --- | --- |
|  | File appDir=new File("src");  File app=new File(appDir,"testApp.apk"); // this code is just for understanding; testApp.apk is udemy app; |
|  | DesiredCapabilities cap=new DesiredCapabilities(); |
|  | cap.setCapability(MobileCapabilityType.PLATFORM\_NAME, MobilePlatform.ANDROID); |
|  | cap.setCapability(MobileCapabilityType.DEVICE\_NAME, "Android device"); |
|  |  |
|  | cap.setCapability(MobileCapabilityType.NEW\_COMMAND\_TIMEOUT, "100"); |
|  | cap.setCapability(MobileCapabilityType.APP, app.getAbsolutePath()); |
|  |  |
|  |  |
|  | AndroidDriver<AndroidElement> driver=new AndroidDriver<>(new URL("http://127.0.0.1:4723/wd/hub"),cap); |
|  | **For driver you have provide the HOST 127.0.0.1 and the PORT:4723, wd is the short form of WebDriver, hub refers to Selenium Grid configuration where the two components are: Hub and Nodes.** |

* **How to run Appium programtically?**

There are 3 ways you can run programmatically:

1. Using AppiumDriverLocalService class
2. Using Appium.js with Node.exe
3. Start Appium server using Command Prompt ( to create a bat file and give the file in before test)

We can achieve this with the help of Appium java Client ['AppiumDriverLocalService'](http://appium.github.io/java-client/io/appium/java_client/service/local/AppiumDriverLocalService.html)class. With the help of 'buildDefaultService()' we can easily start and stop appium service.



* **What app version did you use?**

Native app;

* **Where did you test the app?**

In real android device; because tha’s how you can test while calling, receiving text that time how the app is working. Need to see the app behavior in different circumstances.

* **What challenge did you face in appium?**

1. Screen size and scrolling challenge. For example when you want to test small phone like Samsung duos mini and Samsung galaxy 7 the number options will not displayed same in both screen, for mini you have to scroll and click. We have to write our code in try block follwing catch block where we mention the the mobile elemnent for scrolling.
2. For **Android 7** (**Nougat**) version our appium 1.7 was not working. Then we had to discuss and research and came up with a solution **Appium** version (1.6.4). As of today, this **is** the only version which supports **Android 7** (**Nougat**).
3. Also in some of the new appium jar some Classes deprecated.
4. Sometime real device connection lost while receiving any call during the testing.

* **Imagine you have 4 phones and you want to run your test in a single phone how would you achieve this?**

Every phone has  Unique Device Identifier or **UDID**. By configuring this UDID you can achieve this.

* **How do you handle multiple devices on parallel testing?**

Cloud testing or Sauce lab. Registering Sauce Labs Account for running IOS Apps on cloud.you have to select the mobile version.First you need to send the app to sauce lab using username and access key then you will pass the instruction to automate the app using desiredcapailities.

|  |
| --- |
|  |
|  |  |
|  | DesiredCapabilities dc=DesiredCapabilities.iphone();  dc.setCapability("platformName", "iOS"); |
|  |  |
|  | //cap.setCapability(MobileCapabilityType.PLATFORM\_NAME,MobilePlatform.IOS); |
|  |  |
|  | dc.setCapability("platformVersion", "7.1"); |
|  |  |
|  | //cap.setCapability(MobileCapabilityType.PLATFORM\_VERSION,"7.1"); |
|  |  |
|  | dc.setCapability("appiumVersion", "1.3.4"); |
|  |  |
|  | dc.setCapability("deviceName", "iPhone Simulator"); |
|  |  |
|  | dc.setCapability("browserName", "safari"); |
|  |  |
|  | IOSDriver driver=new IOSDriver(new URL("http://rahul1:e950d779-1817-4c3c-b122-06715b814dfd@ondemand.saucelabs.com:80/wd/hub"),dc); |

* **Did you automate finger print or face recognition?**

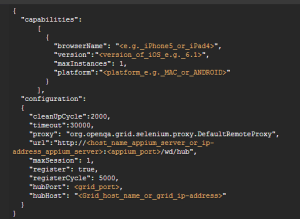
I did manually.

* **How is your appium framework designed?**

Explain the Same framework as web app but for each screen we created separated class. Mentiond about server start and end issue.

* **What is appium grid and how to use it?**

Appium grid is used for parallel testing. First you have to register the hub and then you have to register the nodes means real devices. The configuration of Appium Node is usually maintained with Jason configuration files as shown in the following diagram:

[](https://i0.wp.com/blogs.perficient.com/files/2017/03/2-3-1.png?ssl=1)

* **What is the difference between mobile app and web app?**

There are many **differences between** both of them. The most basic **difference** is that the **Web App** runs on browser and **mobile app** runs on the device itself. ... **Web Application** needs active internet for it to work on your devices. **Mobile App** works without an active internet.

* **What underlying API does Appium follow?**

It is an extension to selenium. All the selenium api’s are present in appium also. But apppium has some of his own methods for mobile based testing. You need to install appium jar.

* **What are Desired capabilities?**

It is the overall configuration like which app, which device, which platform, which version. All the configuration is send to appium servers and it will invoke those particular operation respectively.

* **How would you inspect elements of Native Android APP?**

Android UI Automator tools.

* **How would you identify object uniquely when there are multiple objects with same class name?**

By using driver.findElement/li[1] listing concept and indexing. Or using get(index#).

* **How would you Inspect Elements on IOS Apps?**

Appium Inspector only.

* **How would you identify Mobile Browser objects?**

User agent will change my browser as mobile layout the get the object. Then we can also using chrome casting technique.

* **Is Appium Server different for both Android and IOS?**

**Appium** is “cross-platform”: it allows you to write tests against multiple platforms (**iOS**, **Android**), using the same API. This enables code reuse between **iOS** and **Android** testsuites.

* **How do you detect that device got connected to the PC?**

Adb devices from terminal;

* **Will Appium work for Mobile Browser Automation?**

Yes. Appium will inject selenium.

* **Mention what are the possible errors one might encounter using Appium**?

The possible errors one might face in Appium includes

* Error 1: The following desired capabilities are needed but not provided: Device Name, platformName
* Error 2: Could not find adb. Please set the ANDROID\_HOME environment variable with the Android SDK root directory path
* Error 3: openqa.selenium.SessionNotCreatedException: A new session could not be created
* Error 4: How to find DOM element or[XPath](https://www.guru99.com/xpath-selenium.html)in a mobile application?
* For ios each time you run the test you have stop the server for inspecting.
* **Mention what are the most difficult scenarios to test with Appium?**

The most difficult scenario to test with Appium is data exchange.

* **In Android, do you need an app's .apk to automate using Appium or you also need app in my workspace?**

In Android, you only need .apk file to automate using Appium.

* **Explain the design of Appium?**

Appium is considered as a HTTP server that is written using Node.js platform. It runs on both android and IOS session with the help of web driver JSON wire protocol. After the download and installation of the Appium is completed the server is then setup on the machine which exposes a REST API. Then the Appium also receives connections and command request from clients. These commands are then executed on mobile devices. Generallyappium respond backs with a HTTP response. To execute the request, Appium uses mobile test automation frame works so that it can drive the user interface of the apps. Some of the mobile automation frame works are

1. Google UIAutomator for Android API of level 16 or more than that  
2. Apple instruments for IOS platform  
3. For android API level 15 or lesser than that, Selendroid is used.

* **Name the types of errors that you can face in Appium.**

**1. Error type one –** These types of error occurs when there is the need of desired capabilities but they are not provided. Missing of Device name or platform name is considered to be part of this error type.

**2. Error type two –** These types of error occurs when you cannot find adb. To avoid this type of error can be avoided by setting Android Home environment variable with Android SDK root directory path.

**3. Errors type three –**This falls under the category of penqa.selenium.SessionNotCreatedException which will not allow you to create a new session.

* **What is Appium’s most considerable limitation?**

Hand down my chin starting thinking and mumbling. If I had to provide one single thing you should be aware of about Appium before starting using it, it would surely be: **multiple session handling**. Since Appium is a server, it serves HTTP requests; you might have two different computers running a test each against the same Appium server: what happens? As for now, Appium does not support.

* **How can I exchange data between my test and the app I am testing?**

Appium, actually the WebDriver specification, is not made for exchanging data with your app, it is made to automate it. For this reason, you will probably be surprised in finding data exchange not so easy. Actually it is not impossible to exchange data with your app , however it will require you to build more layers of testability.

* **What are Testability layers?**

In order to make things better, as a developer, what you can do is adding testability layers to your app. The logic behind this approach is simply having some test-related objects in your app which are activated only when your tests run. I learned about this strategy from one of my colleagues Lukasz and such a technique can be really powerful. Enable your testability layers when testing in order to make data exchange easy.

* **I need to debug Appium, is it difficult?**

**No really!** Appium is a NODE.JS application, so it is Javascript in the essence. The code is available on GITHUB and can be downloaded in few seconds as it is small and not so complex. Depending on what you have to debug, you will probably need to go deeper in your debugging experience, however there are some key points where setting a breakpoint is always worth: the proxy component is worth a mention. In appium/lib/server/proxy.js you can set a breakpoint in function doProxy(req,res), that will be hit everytime commands are sent to platform-specific components to be translated into automation commands.

Anything with respect to Java to get a console to debug

Must read the next document: ***Android Mobile Automation.docx***

**What is Android SDK used for?**

The Android SDK includes sample projects with source code, development tools, an emulator, and required libraries to build and test Android **applications**.

**Whats is Xcode used for?**

**Xcode** is an integrated development environment (IDE) for macOS containing a suite of software development tools developed by Apple for developing software for macOS, iOS, watchOS, and tvOS. ... Registered developers can download preview releases and prior versions of the suite through the Apple Developer website.

**What is port testing?**

**Port testing** is a type of **testing** in which one will install the app into original customer environment(part) and check when suitable to that environment or not is known as **port testing**.

The most commonly **blocked ports** are **port** 80 and **port** 25.

**What are the common exceptions you faced in appium?**

* **How to debug mobile server?**
* **Mention android and ios native and web app?**

**What is app package and app activity in Appium?**

Android App package is the destination of your apk file or installed file in your Android device, whereas Android App Activity is the destination path of the screen which will be launched by Appium server.

For Android there is a app call APK Info which you can download from the play store and you will be able to see any of the apps package and activity name.

**Please note:**

* 1. For web view Appium injects selenium api to process the browser scripts.
  2. Appium can automate chrome driver from minimum 71 version.
* **How to switch one native app to another native app in ios?**

static void launchSecurityXxxApp(AppiumDriver<AndroidElement> driver) throws MalformedURLException {

String appPackage="com.xxx.xxxx";

String appActivity="com.xxx.xxxx.Launchable";

Activity activity = new Activity(appPackage, appActivity);

activity.setStopApp(false);

((AndroidDriver<AndroidElement>) driver).startActivity(activity);

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

}

public class SwitchAppTest {

    public AppiumDriver<MobileElement> driver = null;

    String calculatorAppPackageName = "com.sec.android.app.popupcalculator";

    String calculatorAppActivityName = "Calculator";

    String settingsAppPackageName = "com.android.settings";

    String settingsAppActivityName = "com.android.settings.GridSettings";

    @BeforeTest

    public void setupstart() throws MalformedURLException {

        DesiredCapabilities capabilities = DesiredCapabilities.android();

        capabilities.setCapability(MobileCapabilityType.AUTOMATION\_NAME, "Appium");

        capabilities.setCapability(MobileCapabilityType.PLATFORM\_NAME, "Android");

        capabilities.setCapability(MobileCapabilityType.DEVICE\_NAME, "4100b79b459381f7");

        capabilities.setCapability("appPackage", calculatorAppPackageName);

        capabilities.setCapability("appActivity", calculatorAppActivityName);

        driver = new AndroidDriver<MobileElement>(new URL("<http://localhost:4723/wd/hub>"), capabilities);

    }

    @Test

    public void calcTest1() throws Exception {

        //Multiply 2 numbers in calculator app

        driver.findElement(By.xpath("//android.widget.Button[@text='4']")).click();

        driver.findElement(By.xpath("//android.widget.Button[@content-desc='Multiplication']")).click();

        driver.findElement(By.xpath("//android.widget.Button[@text='4']")).click();

        driver.findElement(By.xpath("//android.widget.Button[@content-desc='Equal']")).click();

**// launch settings App**

        Activity activity = new Activity(settingsAppPackageName, settingsAppActivityName);

        activity.setStopApp(false);

        ((AndroidDriver<MobileElement>) driver).startActivity(activity);

**// Switch OFF WIFI**

        driver.findElement(By.xpath("//android.widget.LinearLayout[@content-desc='Wi-Fi']")).click();

        driver.findElement(By.className("android.widget.Switch")).click();

**// Re launch calculator App**

        activity = new Activity(calculatorAppPackageName, calculatorAppActivityName);

        activity.setStopApp(false);

        ((AndroidDriver<MobileElement>) driver).startActivity(activity);

        String result = driver.findElement(By.className("android.widget.EditText")).getText();

        System.out.println("Result : " + result);

    }

    @AfterTest

    public void tearDown() {

        driver.quit();

    }

}