**Introduction to Appium**

**Appium**

* Appium is an open-source Mobile automation tool for testing native app, hybrid app as well as mobile browser apps.
* Appium internally uses Web-Driver W3C protocol (which selenium uses) to test the apps, so it is just like selenium but for mobile.
* It is the only tool which supports cross platform testing.
* Supports WebDriver API Selenium Family.
* It is developed using sauce lab.
* There are other tools to automate mobile appln such as selendroid, robotium etc. which supports only android devices whereas appium supports multi-platform.

**Supported Platforms**

* Android
* iOS
* Firefox OS

We can write code in Any Language Supported by WebDriver

Java, C#, Javascript, Python, Ruby

**Appium Architecture**

HTTP Req

Appium Client Code (Eclipse)

4723

Android

UIAutomator2

W3C WebDriver

iOS

XCUITest

HTTP Res

* Appium is a node.js HTTP server that exposes a REST API.
* Appium client initiates session using Desired Capabilities.
* Drives Android session using UiAutomator2 driver.
* Drives iOS session using XCUITest driver.

**Difference between Web Application & Mobile Application**

|  |  |
| --- | --- |
| Web Application | Mobile Application |
| * Internet Reqd. | * Internet may or may not Reqd. |
| * Browse Reqd. | * Browser may or may not Reqd. |
| * URL Reqd. | * URL may or may not Reqd. |
| * Can’t access Device Recourses. | * Can access Device Recourses. |
| * Doesn’t Utilize System Memory. | * Utilizes System Memory. |
| * Platform Independent (FF, Chrome, Safari etc). | * Platform Dependent (Android or iOS). |

**Different Types of Mobile App**

1. Native Apps: Native Apps are the Apps which has been developed & compiled using specific programming Language for specific platform. They are also developed by using specific IDE for the given operating systems, such as Android studio for Android apps, XCode for iOS apps & Visual Studio for Windows apps.

Let’s see what Language they use to develop Native Application for Android, iOS & Windows

|  |  |  |  |
| --- | --- | --- | --- |
| Operating System | Programming Language Used | IDE (Integrated Development Environment) | Extension & Expansion |
| Android | Java & Kotlin | Android Studio | .APK (Android Package Kit) |
| Ios | Objective. C and Swift since iOS 8 | Xcode | .iPA (iPhone Application Archive) |
| Windows | .Net, XAML, Visual Basic & C# (Windows Phone apps that use XAML for UI & C# or Visual Basic for code) | Visual Studio | .APPX (Application Package Ready for distribution & Installation) |

* Native Application’s doesn’t have that much attractive UI (Ex- If we download True Caller instead of inbuilt Phone app, it will have more attractive UI).
* Native applications are fast because they developed using only one programming Language.
* Most of the inbuild applications are Native, they have more command over Mobile’s Hardware like Camera, Microphone, Volume button etc.
* Content of the Native apps is Constant.

Ex: Calculator, Calendar, Camera etc.

1. Hybrid Apps: Hybrid App are the App which are wrapped inside a Native Container.

* We use more than one Programming language for the development of these applications (Java+JSP+css5+HTML) That’s why these applications are slow.
* UI of these applications are attractive.
* Content of Application is not Constant.

Ex: YouTube, Gmail, Amazon etc.

1. Mobile Web App: The App which we can access through our Mobile browser is known as Mobile Web App.

Ex: <https://m.facebook.com>

**Q. How to know an app is Native or Hybrid?**

Ans. We have to talk to the Product Owner/Developers or we have to follow Req. Documents/User Stories.

**Types of mobile testing**

* Mobile Device testing: can be both hardware testing and software testing
* Mobile software testing: testing the applns in mobile phones

**Q. What are the different types of testing you will perform on mobile app?**

* Functional Testing
* E2E/System Testing
* Integration Testing
* Smoke Testing
* Compatibility Testing
* Adhoc Testing
* Acceptance Testing
* Performance Testing
* Usability Testing
* Globalisation Testing
* Regression Testing

**Mobile device testing**

* Interruption Testing 🡪 Interfering the flow of the application is called as Interference Testing.

The 3 Categories of Interruption Testing in Mobile Applications

* Device Dependent

This category covers all the interruptions which are related to the built-in feature of the device which is being tested. This does not involve other applications on the mobile phone. Some of the scenarios which are device dependent are the device getting locked, change in volume, device getting shutdown or restarted, and so on.

* Functionality Dependent

It covers a combination of both other application and device functionalities that results in interrupting scenarios such as incoming calls, messages, and notifications from other mobile apps in the form of pop-up or push notifications.

* Network Dependent

Includes Interruptions that arise when switching from one type of network to another. For example, switching to a mobile network connection from Wi-fi when out of range, or switching between 5G, 4G, or 3G based on the network availability in a particular location.

Causes of interruptions

* Battery low
* Battery full- when charging
* Incoming phone call
* Incoming SMS
* Incoming Alert from another mobile application
* Plugged in for charging
* Plugged out from charging
* Device shut off
* Application Update reminders
* Alarm
* Network connection loss
* Network connection restoration
* Orientation Testing 🡪 Testing the application in different different orientation (like Landscape or Portrait) is called as Orientation Testing.
* Network Testing 🡪 Testing the behaviour of an application in various network condition (like Data ON, Data Off, WIFI, Aeroplane mode, Strong/Low Network) is called as Network Testing.
* Installation Testing 🡪 Checking if the application is getting installed properly or not
* Uninstall Testing🡪 To test if you can uninstall the application
* Battery Testing 🡪 Testing the battery consumption of any application from the last time it’s fully charged and also testing the behaviour of an application under different battery levels is called as Battery Testing.
* GPS (Geo Location) 🡪 Testing the application Language Location wise and testing if application’s features are according to the geographical location is called as GPS Testing. 🡪 you can do it using browser stack
* UI Testing 🡪 Testing the User Interface of any Application is called as UI Testing.
* Gesture Testing 🡪 Testing the gesture actions in mobile applications like click, swipe, scroll etc.
* Field Testing
* Security Testing
* UI(SFX/VFX) 🡪 Testing the interruption in any kind of High-End games which leads to Blue Screen is called as VFX Testing. Testing the sounds in different kind of application is called as SFX Testing.