**Mobile Automation with Appium**

**PREREQUISITES**

* Must have knowledge of Manual Software Testing
* Android Mobile Devices and USB Cables are available for conduct of the testing on Real Devices
* In absence of Real Devices, Emulators shall be used
* iOS Mobile Devices, Developer License, USB Cable
* iOS requires participants to be registered as Developers at Apple Developers’ Site

**(It is mandatory to accomplish the training prerequisite conditions before nominating for the session)**

PROGRAM OVERVIEW

Workshop and lectures are designed to enable implementation and use Appium for Mobile automation testing

SKILLS GAINED BY THIS COURSE

* Test Automation of Mobile (Native, Web-based, Hybrid) Applications using Appium
* Test Automation using Selenium Desired Capabilities, TestNG, and, End-to-End Exercises

COURSE DESIGNED FOR

* Software Test Professionals, QA Managers and Architects, QA Specialists
* Manual Testers, Software Quality Control Engineers, Test Leads

**1.0 Introduction to Workshop**

Appium, Wire protocol, UI Automator, UI Automation, Desired Capabilities

**Mix:** Introduction to Mobile Automation using Appium

* Basic Terminology: Testing, Types of Testing, Manual/Automation
* What is Mobile Automation Testing
* Mobile Application Testing Tools
* Introduction to Appium
  + - Where Appium can be used
    - WebDriver Wire protocol
    - Hybrid Apps, Native Apps and WebApps
    - UIAutomator
    - iOS-driver
    - Drawbacks and Limitations

**Practicals:** Appium Setup & Configurations for Android & iOS

* Developer mode in Mobiles
* Debug level settings in Phone
* Connect & detect phones in DDMS with Pdanet software
* Connect & detect phone in DDMS without Pdanet software

**Practicals:** Setup & Installation for Appium

* Setup Java & Eclipse IDE
* Setup Selenium for Appium
* Install and setup Node JS
* Install Microsoft Dot Net framework
* Download & Setup Appium
  + - Configuring Various versioning API level requirements for Appium
    - Starting Appium Server from Console
* Starting Appium Server from command prompt
* Running Appium Program
* Create Android Virtual Machines (AVM) – Emulators
* Install Simulator
* Create, Build and Run first Script on Simulator`

**2.0 Appium Desired Capabilities, Packages and Activities**

* **Practicals:** Understanding Desired Capabilities class
* **Practicals:**  Build Test Execution Matrix to Use Emulator and/or Device

**3.0 Appium Android Locator Strategies – Identify Mobile Elements**

* **Practicals: Running a sample test to make sure everything works fine**
* **Practicals: Testing on Emulators and Real devices**
* Finding elements in android – UIAutomatorViewer
* Running Test Case with Appium – Android
* How to interact with Hybrid Elements – Android
* How to Run Mobile Web – Android
* Run Test Cases on Real Devices – Android

What is Appium Inspector?

* **Practicals: Locating elements with the help of UIAutomatorViewer**
* **Practicals: Testing Touch Actions for Android & iOS**
* Touch Events
* Build scripts to simulate events like home, backspace, delete
* Build event handlers to handle home, backspace, delete events
* Implement Wait Actions
* Implement handlers for Drag and drop elements
* Multitouch Actions
* Simulate
  + multiple gestures
  + Long press and delete elements
  + Toggle buttons
  + Pinch, Tap, Hold events

**4.0Practicals: Appium Complex Elements Locator Strategies**

* Locating Elements on the App
* What all can be locatable
* Locating elements within elements
* Locating multiple elements
* Finding Elements by ClassName
* Finding Elements by ids
* Finding Elements by XPath
* Understanding UISelector class

**5.0Practicals: Appium – Testing Android Native App**

* Testing Internal Dialer Application
  + - Extracting package and activity information
      * no need to install an app
    - Finding elements by class Name and id
      * Working with Android Phonebook
    - Extracting package and activity information
      * no need to install an app
    - Locating elements with same class Name
    - Finding Elements within Elements
* Use driver.quit to kill the Appium instance
* Benefits of using Ids
* Testing native apps Touch Actions and Events - Handling user gestures
* Drag and Drop
* Multitouch Actions
* Pinching and Tapping
* Android Key Events

**6.0Practicals: Appium – Testing Hybrid App**

* Testing Facebook App

**7.0Practicals: Appium – Web App Testing on Android’s default Browser**

* Opening browser on a Real Android Device
* Navigating to the URL
* Printing Title
* Capturing Screenshots
* Handling and Locating WebElements

**8.0Appium iOS Locator Strategies – Identify Mobile Elements**

What is Appium Inspector?

* Practicals: Record and Play using Appium Inspector
* Practicals: Locate elements of iOS App using Appium Inspector
* Practicals: Testing on IOS, Simulators and Real devices
* Testing on IOS Simulator
* Apple Developer id
* Testing Hybrid app on IOS / iPad simulator
* Provisioning profiles
* Desired Capabilities class
* Working with Xcode

**9.0Practicals: Appium – Testing Native / Hybrid Apps on IOS**

* How to interact with Hybrid Elements - iOS
* How to Run Mobile Web – iOS
* Working with Native app on iPhone real device
* Deploying the app using XCode
* Adding capabilities
* Running the Appium inspector
* Inspecting elements with appium inspector
* Generating XPath with the help of inspector
* Record and Play test
  + - clicking, tapping, pinching etc.
* Executing Native app test on the Real device

**10.0Practicals: Web Application Testing (iOS/Android)**

* Launching browser with Appium
* Firing various commands on browser
* Using UIAutomator to build Test Cases
* Identifying objects of mobile browser
* Build & Execute Scripts to use AppiumDriver commands on Mobile browser

**11.0Practicals: Exception Handling**

* Implement Exception Handling
* When launch fails
* Take Screen Shot of Mobile Screen when Exception occurs
* Implement Logger to log extra details of Cause of Exception

**12.0Practicals: Serial and Parallel Script Execution**

Deploy Appium scripts across Mobile Platforms

**13.0Appium Good Practices**

Appium Good Practices