**Appium - Selenium for Mobile Automation Testing 2017**

Workshop Details:

|  |  |
| --- | --- |
| **Duration :** | 3 Days |
| **Objectives and Take away :** | This workshop is designed to teach participants how the functionality of Appium and how to begin the process of building an automated framework.  **Take away**: In-depth understanding of Appium and its related tools and framework and their usage. |
| **Participants’ Entry Profile :** | Participants should have a basic understanding of Java programming language. In addition, you should be well-versed with the fundamentals of Selenium Web driver, Junit and TestNG Framework. Some basic Knowledge of Mobile. |
| **Training Methodology :** | The workshop will follow Synergetics methodology of  **Concept Visualization**  **Active Experimentation**  **Application Development.**  The workshop will be 100% Hands-On with each participant having access to system during the session |

Setup Requirements

|  |  |
| --- | --- |
| **Hardware and Software Requirements :** | Participant’s as well as Trainer’s Machine are required to have :  **Hardware**   * Intel Core i3 2.2 GHz * 80 GB HDD * LCD Color Monitor * 2 GB RAM * LAN Connectivity   **iOS Requirements**   * Mac OS X 10.12 recommended * XCode 8 recommended * Apple Developer Tools (iPhone simulator SDK, command line tools) * [Ensure you read our documentation on setting yourself up for iOS testing!](http://appium.io/slate/en/master/#running-on-osx.md)   **Android Requirements**   * [Android SDK](http://developer.android.com/) API >= 17 (Additional features require 18/19) * Appium supports Android on OS X, Linux and Windows. Make sure you follow the directions for setting up your environment properly for testing on different OSes: * [linux](http://appium.io/slate/en/master/#running-on-linux.md) * [osx](http://appium.io/slate/en/master/#running-on-osx.md) * [windows](http://appium.io/slate/en/master/#running-on-windows.md)  **Windows Requirements**  * Windows 10 * [Documentation](http://appium.io/slate/en/master/#running-on-windows.md)  **FirefoxOS Requirements**  * [Firefox OS Simulator](https://developer.mozilla.org/en/docs/Tools/Firefox_OS_Simulator)   **Software**   * JDK (Java Development Kit) [[Link](http://www.oracle.com/technetwork/java/javase/downloads/index.html)] * TestNG [[Link](http://testng.org/doc/download.html)] * Eclipse [[Link](http://www.eclipse.org/downloads/)] (Must match with bit architecture of machine) * Selenium Server JAR [[Link](http://docs.seleniumhq.org/download/)] * Webdriver Language Binding Library [[Link](http://docs.seleniumhq.org/download/)] * APPIUM For Windows [[Link](https://bitbucket.org/appium/appium.app/downloads/)] * APK App Info On Google Play [[Link](https://play.google.com/store/apps/details?id=de.migali.soft.apkinfo&hl=en)] * 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.) * MS Office 2007 onwards * The installable must match in bits architecture of the machine. * A shared disk-space among participants and trainer for trainer to share training stuff with participants. * Internet connection to fetch maven dependencies. |
| **Training Lab Requirements:** | Whiteboard 6 feet by 4 feet (minimum) and Whiteboard markers – Red, Blue, Green, Black  Video Projector (1024 X 768 resolutions) |

Course Content

|  |  |
| --- | --- |
| **Day 1** | * **Module 1: Introduction to Appium & Mobile automation Testing**   + Introduction to Mobile automation testing   + Selenium 3.0 and APIs   + Automation for IOS and Android devices   + Running the scripts on Emulators   + Running the scripts on Real devices   + Various tools available for Mobile automation testing   + Understanding the API Levels and Appium Concepts   + Client/Server Architecture, Session, Desired Capabilities, Appium Server and Appium Clients   + Taking a look at available Android Virtual Devices * **Module 2: Basic installation for Appium on Windows**   + Visiting Appium.exe   + Downloading Appium for Windows   + Downloading Appium jars and other dependencies   + Configuring SDK Manager   + Downloading different Android API levels   + Setting up Android environment variables   + Downloading Appium GUI Tool   + Understanding Application path, Package & Launch Activity   + Starting Appium Server   + Running a sample test to make sure everything works fine * **Module 3: Downloading Appium Dependencies, Desired Capabilities, Packages & Activities information**   + Understanding Desired Capabilities class   + Extracting Packages and Activities information of pre and post installed apps   + Downloading Appium Maven Dependencies   + Adding Device Name   + Connecting real device to USB and running ADB commands * **Module 4: Starting the Appium server programmatically on Windows OS**   + Starting the Appium server through code / programmatically on Windows   + Downloading Appium Maven Dependencies   + Stopping the Appium server   + Kill all nodes * **Module 5: UIAutomator Viewer**   + Opening UIAutomator Viewer through SDK tools kit   + Understanding different locator strategies   + Taking the Device screenshot   + Device screenshot with Compressed hierarchy   + Understanding the different locator tags |
| **Day 2** | * **Module 6: Appium Inspector**   + What is Appium Inspector?   + How and when to run it?   + Record and Play using Appium Inspector   + Locating elements with the help of Appium Inspector   + Inspector will only run with active session * **Module 7: Object Repository**   + What is object repository?   + How to use object repository in framework   + What is the use of maintain Object repository   + Types of creating object repository * **Module 8: Object Locators and  Techniques**   + Locating Elements on the App   + What all can be locatable   + Locating elements within elements   + Locating multiple elements   + Find Element By Android UIAutomator   + Exploring UIAutomator API   + Finding Elements by ClassName   + Finding Elements by ids   + Finding Elements by Xpaths   + Using different functions inside UIAutomator class   + Exercises |
| **Day 3** | * **Module 9: Testing Android Native, Hybrid & Web Apps on Windows**    + Native App Test – Making a call with Internal Dialer Application   + Native App Test – Adding a new contact to Phonebook   + Native App Test – Sending SMS from a real device   + Native App Test – Automating a standard App   + TouchActions and Events – Handling user gestures   + WebApp Test – Automating Web based testing on Chrome Browser   + Hybrid App Test – Automating the mobile messaging app * **Module 10: Framework Design**   + Page Object Model (Appium community best practice)     - Writing and executing suite of tests for your sample application using POM     - Controlling test flows with TestNG’s xml capabilities and Maven projects   + Understanding TestNG and tagging   + TestNG for capturing results & reports   + Client Libraries:     - Java Exercise (this will be the main example throughout the class)   + Python Exercise (example to demonstrate language-of-choice flexibilities)   + Developing repeatable functions:   + Testing the calculator application   + Testing a simple text message   + Test adding a new contact   + Reorganizing POM into a Keyword framework * **Module 11: Framework**    + Appium - Design Pattern   + Page Factory   + Appium - Page Object Model with Page Factory and Testing   + Appium - Integrating Cucumber BDD with Page Object Model and Page Factories * **Module 12: Touch Actions and Events- Handling user gestures with Appium**    + Drag and Drop   + Android Input Key Events   + Swipe Test   + Scroll Test   + Tapping an Element |