Course Outline

Appium Training—Mobile Test Automation Workshop

Course Summary

Description

This workshop is designed to teach participants how the functionality of Appium and how to begin the process of building an automated framework.

Upon successful completion of the workshop, the candidate will receive a Certificate of Achievement.

Labs- his workshop is offered as a series of hands-on “projects” that require participants to understand, design, and implement components of an automated testing framework.

Topics

- Introduction to Appium
- Appium Structure
- Install and Configuration
- Appium GUI
- Identifying UI elements
- Framework Design
- Drivers and Capabilities
- Touch Actions and Events- Handling user gestures with Appium
- Setting up Appium Execution Environment
- Framework Enhancements

Audience

This course is designed for automation testers and automation developers.

Prerequisites

Before taking this course, students should have a background in programming (e.g. Java) and/or scripting and understanding of basic Appium features.

Duration

Four days

Course Outline

Appium Training—Mobile Test Automation Workshop

Course Outline

I. Introduction to Appium
- A. Why chose Appium?
- B. Pros/Cons
- C. Main Philosophy (4 principles that guided the Appium architecture)

II. Appium Structure
- A. Appium's HTTP/JSON request/response Protocol
- B. Client Libraries
- C. Server Architecture
- D. Rest API

III. Install and Configuration
- A. Eclipse or Intelij (we will focus on Appium/Java libraries)
- B. Appium components
- C. Android SDK and emulators
- D. Setting up client-libraries and Selenium (prerequisite for writing Appium test cases)
- E. Things you need to know before developing iOS scripts with Appium

IV. Appium GUI
- A. Difference between Windows & Mac (limitations you should know about)
- B. Object Inspector
- C. Android, iOS, General, Developer, and Robot settings
- D. Starting and Stopping Appium server from code

V. Identifying UI elements
- A. Developing a sample Android application
- B. Finding Elements with UIAutomator
- C. Understanding the difference between iOS and Android elements
- D. Understanding the UISelector Class
- E. Implicit and Explicit Wait

VI. Framework Design
- A. Page Object Model (Appium community best practice)
  1. Writing and executing suite of tests for your sample application using POM
  2. Controlling test flows with TestNG's xml capabilities and Maven projects
- B. Understanding TestNG and tagging
- C. TestNG for capturing results & reports
- D. Client Libraries:
  1. Java Exercise (this will be the main example throughout the class)
  2. Python Exercise (example to demonstrate language-of-choice flexibilities)
- E. Developing repeatable functions:
  1. Testing the calculator application
  2. Testing a simple text message
  3. Test adding a new contact
- F. Reorganizing POM into a Keyword framework

VII. Drivers and Capabilities
- A. How Appium leverages Selenium Webdriver
- B. Understanding Web, Appium, iOS, and Android Drivers
- C. How to decide which driver to instantiate
- D. How to control Appium's 'Desired Capabilities'

VIII. Touch Actions and Events- Handling user gestures with Appium
- A. Drag and Drop
- B. Android Input Key Events
- C. Swipe Test
- D. Scroll Test
- E. Tapping an Element

IX. Setting up Appium Execution Environment
- A. Selenium Grid for parallel execution on multiple Emulators
- B. Configuring devices & emulators
- C. Batch processing and Shell scripts

X. Framework Enhancements
- A. Continue adding repeatable functions
- B. Expanding test cases
- C. Advance logic, reusable functions
- D. Considerations for maintaining the framework for iOS and Android apps

XI. Additional Topics
- A. Deployment strategies
- B. Running on Emulators vs. real devices vs. a device cloud
- C. Appium Desktop
  1. Differences between Appium CLI and Desktop
  2. Pros and Cons of using Desktop