

QA for Android OS

Course Summary

Description

QA for Android OS is a three-day training course focusing specifically on tools and techniques to effectively test the entire Android Platform.

Topics

- Introduction to Android OS Testing
- System Testing
- QA Engineering

Audience

This course is designed for those wanting to learn the tools and techniques to effectively test the entire Android Platform.

Prerequisites

There are no prerequisites for this course.

Duration

Three days

QA for Android OS

Course Outline

I. Introduction to Android OS Testing

A. Intro

1. About the trainer
2. Intro to the course
3. Students questionnaire to understand the needs and experience level of the target audience

B. Android OS Overview

1. Versions and Updates
2. Security Updates
3. Open Source
4. Google Services vs Clean AOSP
5. Android Applications
6. System Services and System Applications
7. Permissions based OS

C. Application Layer Architecture

1. SDK Platform
 - a) Install
 - b) d.android.com
 - (1) Resources and Videos
 - (2) APIs and Versions
 - c) source.android.com
 - (1) Differences between the Android Source code to the SDK
 - (2) How to download the source
 - (3) CDD
 - (4) CTS Overview
 - d) Android Studio
 - (1) Key shortcuts
 - (2) Downloading other APIs
 - (3) Android Emulators
 - (4) Importing an android github demo project and running it
- e) ADB

Lab: Apk install, uninstall, update; Adb pull/push files; su

2. App Architecture

- a) Activity
- b) Service
- c) Broadcast Receiver
- d) Content Provider

- e) Manifests
- f) Gradle build system
 - (1) Android Specific
 - (2) Maven

Lab: Run Yamba

D. Application Testing - 3 hrs

1. Functional UI Testing (Covered in depth on day 3)
 - a) Black box testing: UIAutomator
 - b) White box testing: Espresso
 - c) Testing Platform: Appium
 - d) Continuous Integration
 2. Tools:
 - a) UIAutomatorViewer
 - b) Monkey
 - c) Input key
- (Quick Lab)**
3. Others:
 - a) JUnit Tests
 - b) Selenium
 - c) Robotium
 - d) Cloud testing
 4. Emulators vs Real Device:
 - a) Versions
 - b) Memory
 - c) CPU
 - d) Real Device, Vanilla and AOSP devices
 - e) Emulator Control
 - (1) Network throttling
 - (2) GPS Location

II. System Testing

A. System Tools

1. Settings app
 - a) Developer control
 - b) Application overview - storage and permissions
 - c) Data metering
2. Logcat
3. Linux shell overview
4. Process ids and Thread Ids
5. top and ps:
 - a) Overview
 - b) Thread watcher

QA for Android OS

Course Outline (cont'd)

- 6. Diskspace and application database size
 - 7. Android Studio Profiler
 - 8. Dumpsys - resource and performance testing:
 - a) Meminfo
 - b) Wakelocks
 - c) Alarms
 - d) Power
 - e) Battery
 - f) Battery-historian + Lab
 - 9. dumpstate and bugreport
 - B. Crashes
 - 1. Understanding a crash
 - 2. ANR and Service ANRs
 - 3. Native Crashes and Tombstone files

Lab: Security Permissions crash
 - C. App Tools
 - 1. Drozer

Lab: Drozer
 - 2. Activity Manager (AM)

Lab: AM Quick

 - 3. Package Manager (PM)

Lab: PM Quick
- ### III. QA Engineering
- A. Java Overview
 - 1. Packages, Imports and Folders
 - 2. Primitives vs Objects
 - 3. Classes and Functions
 - 4. Abstract, Inheritance and interfaces

Lab
 - B. UIAutomator - 3 hrs
 - 1. Setup
 - 2. Test Class:
 - a) Annotations:
 - (1) RunWith
 - (2) SdkSupress
 - (3) Before/After/Test
 - (4) Small/Medium/Large Test
 - (5) Test Suite
 - b) Run your first simple JUnit Test
 - 3. Accessing UI Components using UiDevice:
 - a) Actions
 - (1) UiObject - individual object actions
 - (2) UiCollection - actions on a collection of items
 - (3) UiScrollable - auto scrolling a ListView to get UiObject
 - b) Test state and verify results
 - 4. Overview of the full API

Lab
 - C. Espresso Overview - 2 hrs
 - 1. Setup
 - 2. ViewMatcher - Finding a view
 - 3. ViewActions - Actions on views
 - 4. ViewAssertion - Assertions on views
 - 5. Debugging and Logging

Lab

 - 6. Espresso Test Recorder
 - D. Advanced concepts (Jenkins and CI) and summary