

# Emulator & Simulator FAQs

## Table of Contents

- [What type of keyboard and buttons do the Android emulators have?](#)
- [How can I run Android tests without Appium?](#)
- [What mobile web browsers can I automate in the Android emulator?](#)
- [How can I test with mobile real devices instead of using the simulators or emulators?](#)
- [How do I use XCUITest-driver on Appium 1.6.0 with iOS 9.x tests?](#)

## What type of keyboard and buttons do the Android emulators have?

Android Emulators have software buttons and a hardware keyboard. In a regular Android emulator the device buttons are software buttons displayed on the right side of the emulator. For the Android emulators with different skins (e.g Google Nexus 7 HD, LG Nexus 4, Samsung Galaxy Nexus, Samsung Galaxy S3, etc) the device buttons are also software buttons that are overlaid on top of the skin. For instance, if you hover the mouse around the edges of any of our Android emulators with an specified skin, a hover icon will appear and you should be able to find whatever buttons actually exist on the device that the skinned emulator is trying to emulate (e.g power button along the top, volume buttons along the edge, back/home buttons right below the screen, etc).

## How can I run Android tests without Appium?

If you wish to use the native Android testing framework (Espresso), please refer to the following documentation for [real device testing](#) or [emulator and simulator testing](#) .

## What mobile web browsers can I automate in the Android emulator?

Currently the only browser that can be automated in our Android emulators is the stock browser (i.e Chrome).

## How can I test with mobile real devices instead of using the simulators or emulators?

The mobile real device cloud is a new feature that Sauce Labs is currently working on. For more information about this feature please directly email your account representative or email ([sales@sauce labs.com](mailto:sales@sauce labs.com)).

## How do I use XCUITest-driver on Appium 1.6.0 with iOS 9.x tests?

Appium 1.6.0 will default to using `automationName: XCUITest` for iOS 10.x tests, and `automationName: UIAutomation` for iOS 9.x tests. However, if you specify `automationName: XCUITest` in the device capabilities for the test, you can get XCUITest for iOS 9.x tests.