System Requirements for Appium

The basic requirements for the Appium server are:

1. A macOS, Linux, or Windows operating system
2. [Node.js](https://nodejs.org/) version in the [SemVer](https://semver.org/) range ^14.17.0 || ^16.13.0 || >=18.0.0
3. [npm](https://npmjs.com/) version >=8 (npm is usually bundled with Node.js, but can be upgraded independently).

Note: Appium is relatively lightweight and doesn't have significant disk space or RAM requirements. It can even be run in resource-constrained environments like Raspberry Pi.

Driver Requirements[¶](https://appium.io/docs/en/2.3/quickstart/requirements/#driver-requirements)

Drivers for automating specific platforms will likely have other requirements. Refer to the documentation of the [Appium driver(s)](https://appium.io/docs/en/2.3/ecosystem/drivers/) for that platform for additional dependencies. It is almost universally the case that Appium drivers for a given platform will require the developer toolchain and SDKs for that platform to be installed.

Install Appium Framework

Step 1: Open CMD in (Run as Administrator).

Step 2: Run this command to install the Appium (npm I -g appium) (You can install Appium globally using npm).

Step 3: After successfully run the install command verify the Appium via this command (Appium -v or Appium –version).

Step 4: To start the Appium Server use this command (Appium).

Step 5: To Stop the Appium server in window press (Ctrl + C).

Install of Appium Driver

You can't do much with Appium unless you have a [driver](https://appium.io/docs/en/2.3/intro/drivers/), which is an interface that allows Appium to automate a particular platform.

Set up Android automation requirements

Android SDK

According to the driver, in addition to a working Appium server, we also need to set up the following:

* The easiest way to set up the Android SDK requirements is by downloading [Android Studio](https://developer.android.com/studio). We need to use its SDK manager (*Settings -> Appearance & Behavior -> System Settings -> Android SDK*) to download the following items:
  + Android SDK Platform (select whichever Android platform we want to automate, for example, API level 30)
  + Android SDK Platform-Tools
* If you wish, you can also download these items without Android Studio:
  + Android SDK Platform can be downloaded using sdkmanager included in [Android command-line tools](https://developer.android.com/studio#command-line-tools-only)
  + [Android SDK Platform-Tools](https://developer.android.com/tools/releases/platform-tools)
* Set up the ANDROID\_HOME environment variable to point to the directory where the Android SDK is installed. You can usually find the path to this directory in the Android Studio SDK manager. It will contain the platform-tools and other directories.

Java JDK[¶](https://appium.io/docs/en/2.3/quickstart/uiauto2-driver/#java-jdk)

* Install the Java JDK (for the most recent Android API levels, JDK 9 is required, otherwise JDK 8 is required). You can download this from [Oracle](https://jdk.java.net/) or [Adoptium](https://adoptium.net/en-GB/temurin/releases/). Make sure you get the JDK and not the JRE.
* Set up the JAVA\_HOME environment variable to point to the JDK home directory. It will contain the bin, include, and other directories.

Why Java JDK is requires in Appium Setup?

1. Appium is built on Java and relies on it for functionality. Key components, including the Appium server, use Java to implement their features. The JAVA\_HOME variable points to the JDK installation, enabling these components to find necessary Java executables and libraries for proper execution.
2. This is crucial for running Appium commands, especially for starting the Appium server and managing Android emulators or devices.
3. **Path Configuration**: Setting JAVA\_HOME allows easy access to Java binaries, enabling Appium commands, like starting the server and managing devices, to run smoothly.
4. **Compatibility and Version Control**: JAVA\_HOME simplifies managing and switching between JDK versions, ensuring the correct one is used for Appium.
5. **Other Build Tools**: Tools like Gradle and Maven rely on JAVA\_HOME to find Java, avoiding build/runtime issues with Appium.
6. **Error Reduction**: Incorrect JAVA\_HOME setup can cause errors, like Java not being found, disrupting Appium and Android SDK operations.

Prepare the Devices

1. If using an emulator, use Android Studio to create and launch an Android Virtual Device (AVD).
2. If using a real device, you should [set it up for development and enable USB Debugging](https://developer.android.com/studio/debug/dev-options).
3. Verify the devices weather your real device or emulator is connected or not and it also give a number which you need to use in test script.

Appium Driver Install Steps

Since we are trying to atomate test flutter App so there are multiple Appium driver and among these our requirement is Appium Flutter Driver.

What is Appium Flutter Driver?

Appium Flutter Driver is a tool for automating tests on Flutter applications across various platforms and operating systems. It is a part of the Appium framework, which is an open-source mobile test automation tool maintained by the community.

Install Appium Flutter Driver

Command: npm install -g appium-flutter-driver

Verify the Install Appium Flutter Driver

Command: Appium driver list