Appium

Android and IOS

Contents

[Appium Tutorial 2](#_Toc58429757)

[What is Appium? 2](#_Toc58429758)

[Features of Appium 3](#_Toc58429759)

[Advantages of Appium 4](#_Toc58429760)

[Disadvantages of Appium 4](#_Toc58429761)

[Appium Architecture 4](#_Toc58429762)

[How Appium work? 5](#_Toc58429763)

[Appium in Android 5](#_Toc58429764)

[Appium in iOS 6](#_Toc58429765)

[Types of Mobile Applications 7](#_Toc58429766)

[Native Applications 7](#_Toc58429767)

[Web Applications 7](#_Toc58429768)

[Hybrid Applications 7](#_Toc58429769)

[Prerequisites to use Appium 8](#_Toc58429770)

[Appium Installation on windows 8](#_Toc58429771)

[Install and setup Java 8](#_Toc58429772)

[Install and setup Android Studio 8](#_Toc58429773)

[Step by step installation of Appium Desktop Client 11](#_Toc58429774)

[Appium Doctor 16](#_Toc58429775)

[Limitation of Appium 17](#_Toc58429776)

[Competitors of Appium 17](#_Toc58429777)

[Appium vs Robotium 17](#_Toc58429778)

[Appium vs Selendroid 17](#_Toc58429779)

Appium Tutorial



Appium tutorial provides basic and advanced concepts of the Appium Automation testing tool. Our Appium tutorial is designed for beginners and professionals.

In this tutorial, we are going to discuss the following topic:

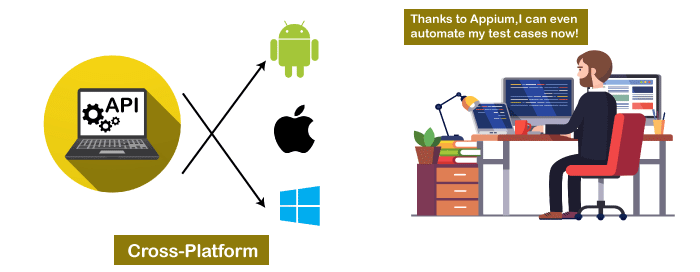
* What is Appium?
* Features of Appium
* Appium Architecture
* Advantages of Appium
* Disadvantages of Appium
* How Appium works?
* Types of Mobile Application
* Prerequisite to use Appium
* Installation of Appium on Windows OS
* Limitation of Appium
* Competitors of Appium
  + Appium vs Robotium
  + Appium vs Selendroid

What is Appium?

**Appium** is an open-source automation mobile testing tool, which is used to test the application. It is developed and supported by **Sauce Labs** to automate native and hybrid mobile apps. It is a cross-platform mobile automation tool, which means that it allows the same test to be run on multiple platforms. Multiple devices can be easily tested by Appium in parallel.

In today's development area, the demand for mobile applications is high. Currently, people are converting their websites into mobile apps. Therefore, it is very important to know about mobile software automation testing technology and also stay connected with new technology. **Appium** is a mobile application testing tool that is currently trending in **Mobile Automation Testing Technology**.

Appium is used for automated testing of **native**, **hybrid**, and **web** applications. It supports automation test on the simulators (iOS) and emulators (Android) as well as physical devices (Android and iOS both). Previously, this tool mainly focused on IOS and Android applications that were limited to mobile application testing only. Few updates back, Appium declared that it would now support desktop application testing for windows as well.



Appium is very much similar to the **Selenium Webdriver** testing tool. So, if you already know Selenium Webdriver, Appium becomes very easy to learn. Appium has **NO dependency** on mobile device OS because it has a framework that converts the Selenium Webdriver commands to UIAutomator and UIAutomation commands for Android and iOS respectively, that depends on the device type rather than the OS type.

It supports several languages such as Java, PHP, Objective C, C#, Python, JavaScript with node.js, and Ruby, and many more that have Selenium client libraries. Selenium is the backend of Appium that provides control over the functionality of Selenium for testing needs.

Features of Appium

* Appium does not require application source code or library.
* Appium provides a strong and active community.
* Appium has multi-platform support i.e., it can run the same test cases on multiple platforms.
* Appium allows the parallel execution of test scripts.
* In Appium, a small change does not require re-installation of the application.
* Appium supports various languages like C#, Python, Java, Ruby, PHP, JavaScript with node.js, and many others that have Selenium client library.

Advantages of Appium

* Appium is an open-source tool, which means it is freely available. It is easy to install.
* It allows the automated testing of hybrid, native, and web applications.
* Unlike other testing tools, you do not need to include any additional agents in your app to make Appium compatible with automation. It tests the same app, which is going to upload in App Store.
* An additional feature added to Appium. Now it would support desktop application testing for windows as well along with mobile application testing.
* Appium is a cross-platform, freely available mobile testing tool, which allows us the cross-platform mobile testing. This means you can test on multiple platforms (single API for both Android and IOS platforms).

Disadvantages of Appium

Along with some features and advantages, Appium has some drawbacks too, which are as follows-

* Lack of detailed reports.
* Since the tests depend on the remote web driver, so it is a bit slow.
* It is not a limitation, but an overhead that Appium uses **UIAutomator** for Android that only supports Android SDK, API 16, or higher. However, Appium supports older APIs, but not directly. It uses another open-source library **Selendroid** to support older APIs.
* In iOS, only one instance (iOS Script) can run on one Mac OS device, which means one test can be executed at a time per Mac. If you want to run your tests on multiple iOS devices at the same time, you need to arrange the same number of Mac machines. But it would be expensive to arrange various Mac machines.

**Solution:** This problem can be resolved if you will run your script in the **mobile cloud of Sauce Lab**. Currently, it allows scripts to run on multiple iOS simulators at the same time.

Appium Architecture

Appium is an HTTP server that is written in **node.js**. It starts a "**test case**" on the device that gives rise to a server and listens for proxied commands from the main Appium server. Tester writes the Test scripts to execute on device or Emulator. Several webdriver sessions for different platforms like Android and IOS are created and handled by the Appium.



Test Scripts written by the tester executes on the Emulator or device by sending them as requests to the Appium server. Each vendor, such as IOS or Android, has a different method and mechanism to execute test cases on the device. So, the test case executes after listening commands from the Appium server. Appium uses JSON wire protocols to send command requests to Appium server.

How Appium work?

* When we install the Appium, a server is also installed with it on our machine that exposes the REST API.
* It receives command and connection requests from the client and executes that command on devices like iOS or Android.
* It replies with the HTTP responses.
* To execute requests, it uses a mobile test automation framework to run the user interface of the app. For Example -
  + **Apple** instruments used for iOS
  + **Selendroid** used for Android API 15 or less
  + **UIAutomator** used for Android API 16 or higher

Remember that Appium uses another open-source library Selendroid to support older APIs because UIAutomator only supports API 16 or higher.

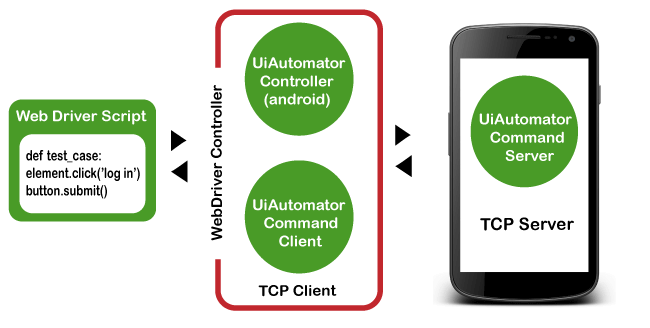
Now we will discuss how Appium works on different platforms -

Appium in Android

On Android, Appium proxies the command to a **UIAutomator** script running on the device. UIAutomator is a native UI automation framework of Android that allows you to run **Junit** test cases directly into the device using command line. Although it uses Java programming language, but Appium allows to run it from any WebDriver supported language.

Android uses **bootstrap.jar**, which works as a TCP server. It is used to send the test commands to perform the actions on Android device using UIAutomator.

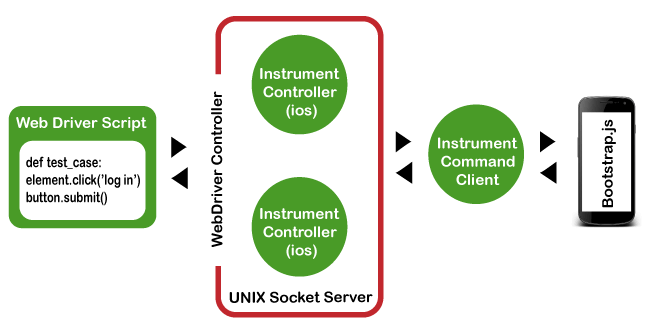
In the below figure, see the Appium architecture in respect to Android automation -



Appium in iOS

As Android uses UIAutomator, iOS uses UIAutomation. Similar to the Android, Appium proxies the command to a **UIAutomation** test case running on the Mac instruments environment. Apple provides this application "**instrument**" that performs various activities like building, profiling, and controlling iOS apps. On the other hand, it also has an automation component where you can write commands in JavaScript. It uses UIAutomation API to interact with Application UI. Appium uses same libraries to automate iOS Apps.

In the below figure, see the Appium architecture in respect to iOS automation -



Note in the above diagram that bootstrap.js is used in place of bootstrap.jar. It performs the actions on our AUT (Application Under Test).

Types of Mobile Applications

As we discussed earlier, Appium has the ability to deal with all sorts of applications, i.e., native, hybrid, and web. Let's understand them in details -

Native Applications

Native applications are software programs that are developed by keeping a certain platform in mind. These applications are developed using a specific software development kit. Native apps are developed for use on a specific device and can be installed from the App Store, such as Google Play Store or Apple's App Store. They can work offline and can also use the device notification system.

Some native application examples are - Pinterest, Skype, Snapchat, etc.

Web Applications

Web applications are not real applications, they are websites that run on browsers. These applications are developed using HTML, CSS, and JavaScript at a very low price. Unlike Android and iOS apps, they do not require a Software Development Kit (SDK) for developers to work with. Web applications are not developed for a particular platform. Since the web applications run on web browsers, they don't require any installation.

Some web application examples are - Flipkart, Ali Express, twitter, etc.

Hybrid Applications

Hybrid application is a combination of native and web applications. Like native applications, these applications can be downloaded from the App Store and also can take advantage of device features, but actually they are web applications inside. They are developed using web development languages - HTML, CSS, and JavaScript like the hybrid app, which allows them to run on any platform.

Some hybrid application examples are - OLA, Instagram, Basecamp, etc.

Prerequisites to use Appium

* Install **Java** (JDK)
* Install **Android Studio**
* Install additional Android **SDK tools**
* Install **Appium jar** file
* js (Not required - It by default comes with "node.js" and "NPM", whenever the Appium server is installed. Therefore, it is not required to install node.js and NPM separately. It is already included in current version of Appium.)
* Install **Appium Desktop Client**
* Install **Eclipse IDE** for Java

Appium Installation on windows

Setting up Appium is not only the installation of a single tool or setting up a few variables. A complete setup of Appium requires the installation of many other tools. Then Appium starts testing of an application. This tutorial mainly focuses on Appium setup for Android on Windows Operating system.

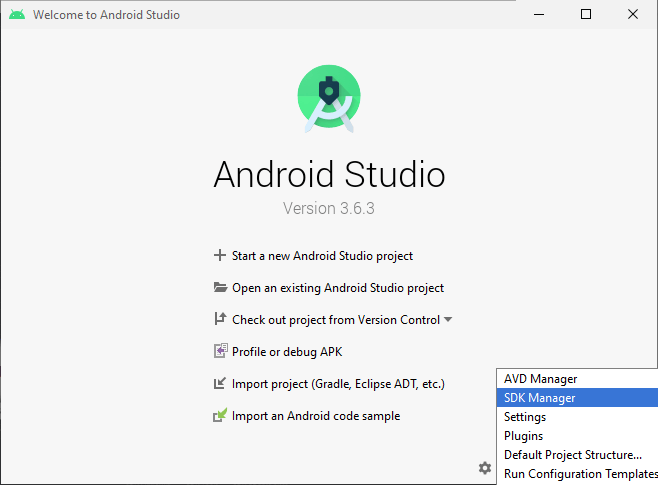
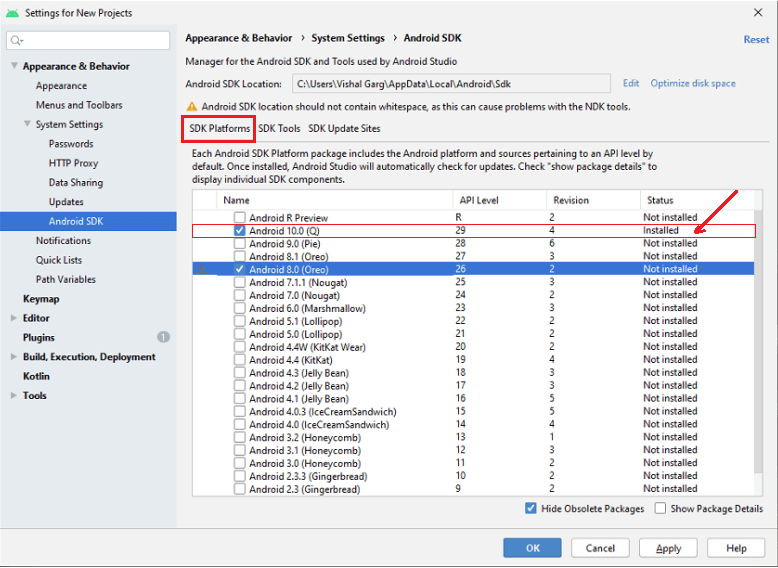
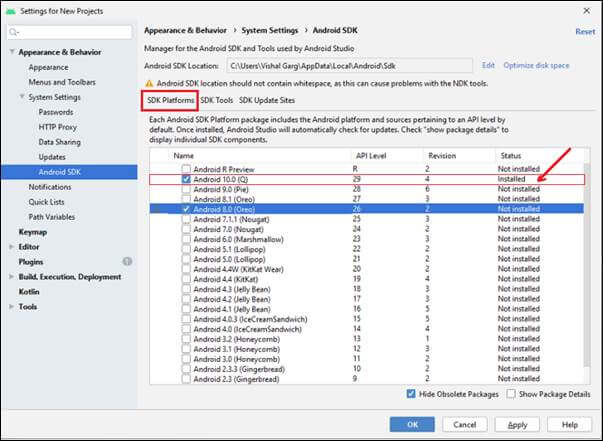
Install and setup Java

With Appium, you need to write automation test scripts that can interact with your mobile screen and control fields (such as text boxes, buttons, etc.) in your mobile application, and also can perform actions on them. Java is used to write these automation test scripts. So, download the latest version of JDK and setup the environment variable.

* [Download the latest version of Java](https://www.oracle.com/in/java/technologies/javase-downloads.html)
* Setup Java Environment Variable.

Install and setup Android Studio

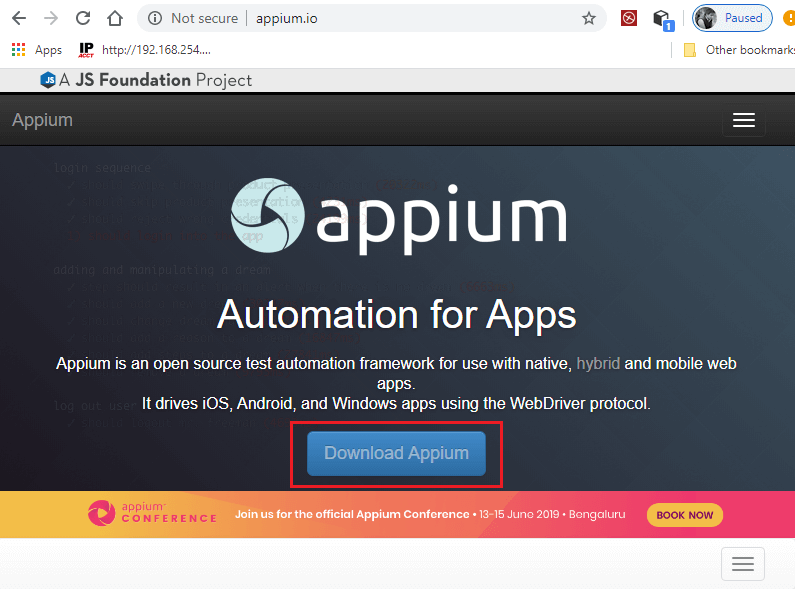
This is one of the most time taking step to download and install the Android Studio. It may cause of errors in Appium if not done properly. With downloading the Android Studio, there are many other changes need to be setup. Here, we would recommend you to download the latest version of Android Studio.

* [Download the Android Studio](http://www.automationtestinghub.com/download-and-install-android/)  
  Download the**latest version of Android Studio** by clicking the above link. Double click on downloaded **.exe** file to run. To complete the installation process, continue with all default options and finish the installation process. Android studio takes time to install.
* **Install additional Android SDK tools**  
  After the complete installation of Android Studio, you will see the main panel of Android Studio, as given in the screenshot below. Click on the **Configure** drop-down menu at the bottom right corner and select the **SDK Manager**.  
    
  The default screen will open up with Android SDK, where under the SDK Platform, check the latest version of Android is installed. You can also use another android version if you are working with a real android device.  
    
  On the same page, switch to **SDK Tools** and check that some of the marked components are installed/updated. If not, mark the checkbox and install them.  
  
* **Setup Android Environment Variable**

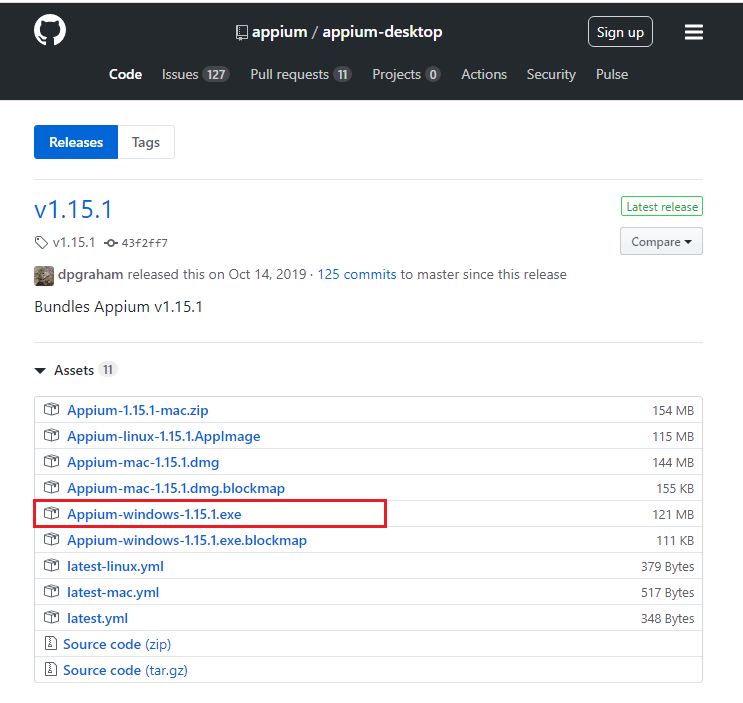
Step by step installation of Appium Desktop Client

Now after installing Java and Android Studio and setting up environment variables for both, install Appium Desktop Client. Appium Studio is a GUI app to install Appium server. It comes with all pre-requirements for installing and using Appium server. It has an inspector, which is used to get basic information on your mobile app. Steps for installing Appium Desktop Client on windows are as follow:

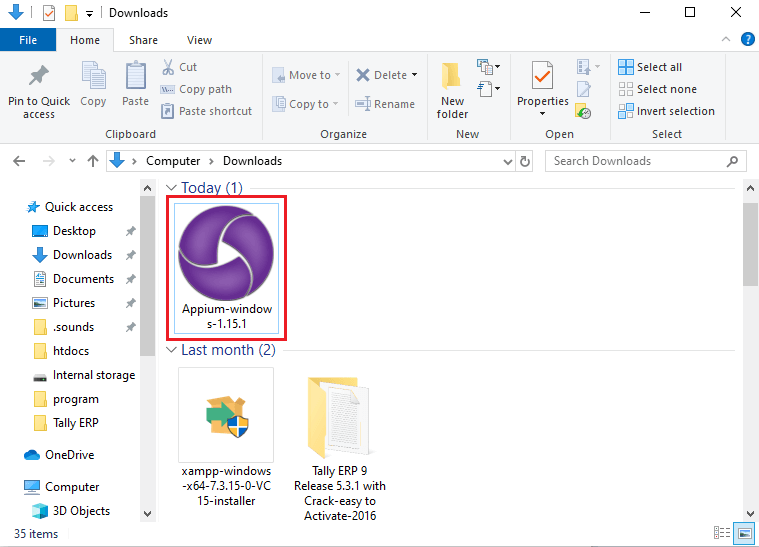
**Step 1:** Go to the following link <http://appium.io/> and click on the **Download Appium** button.



**Step 2:** For Windows OS, download the **exe** file of **Appium desktop setup**. So, click on the file as shown in below screenshot.



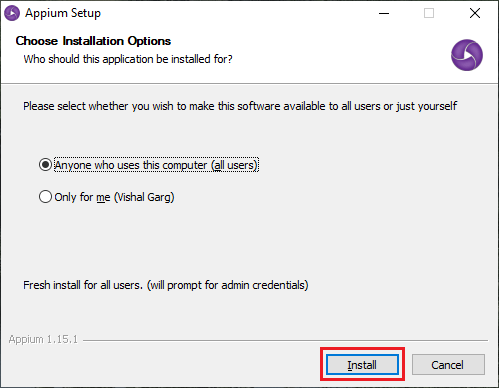
**Step 3:** After downloading the exe file, double click on the downloaded file to run.



Note: On window OS, you don't need to install Appium. It runs from .exe file directly, whereas you need to install the dmg for Mac machine.

**Step 4:** Here, you need to choose the installation options. You can select all users, if you have multiple users on your system, otherwise select **only me** for the administrator of the system.

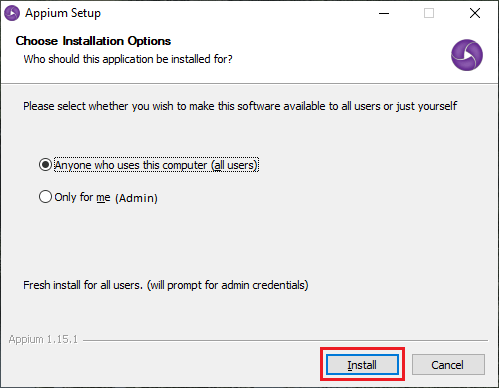
After choosing the installation option, click on the **install** button.



This will start the installation, which may take time to install.

**Step 5:** Appium has been successfully installed on your system. For completing Appium setup, click on the **Finish** button.

Note that **Run Appium** is checked. This will start the Appium Desktop.



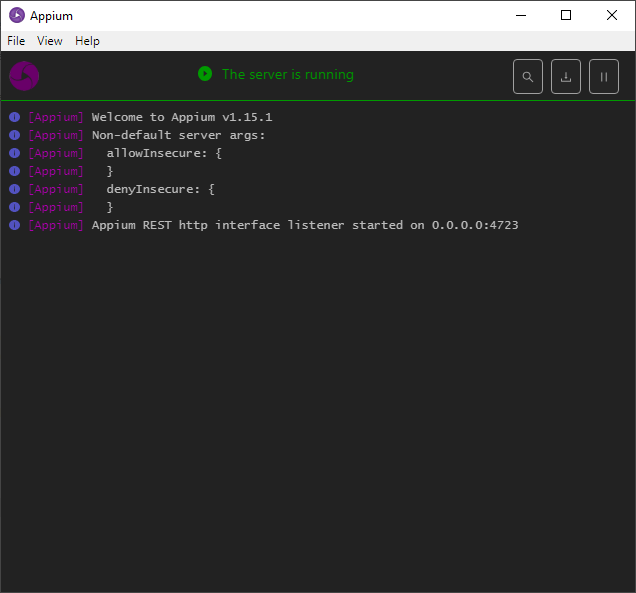
**Step 6:** Now, Appium Desktop UI will appear on the screen containing the default **Host** and **Port**, which can be changed.

Default Host - 0.0.0.0

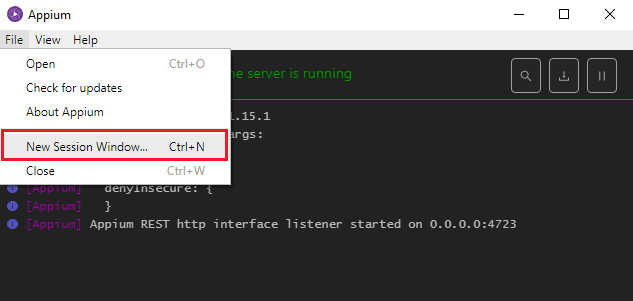
Default Port - 4723

Click on the **Start Server** button and start Appium Server.

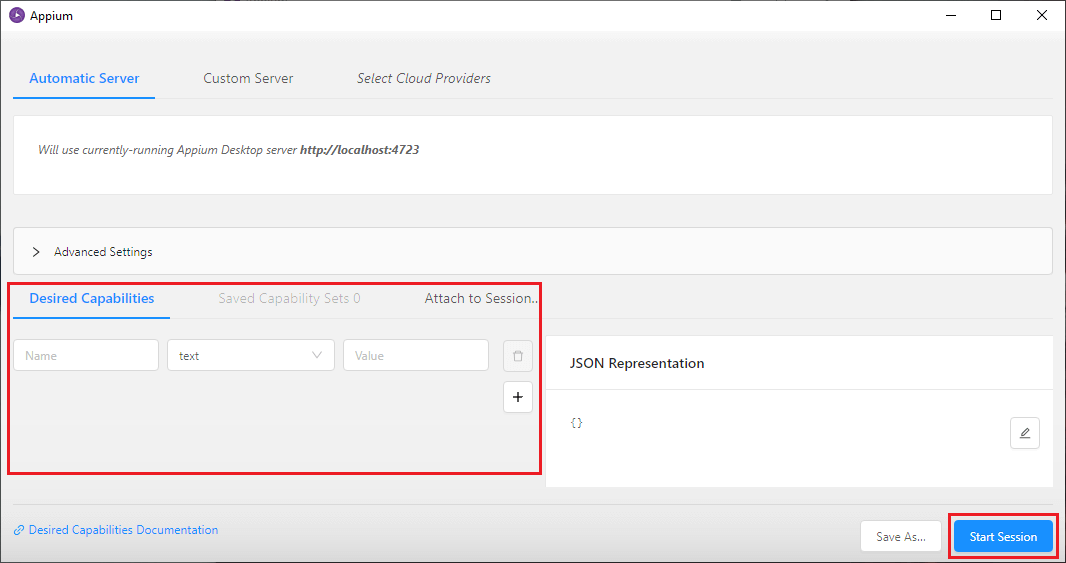
**Step 7:** A new server will launch on the Host and Port specified in previous step. You can see the message on the top of the screen that Server is running.



**Step 8:** Go to **File** and click on the **New Session Window**.



**Step 9:** Provide all desired capabilities here and Click on the **Start Session** button to start a new session.



* [Download Eclipse IDE for Java](https://www.eclipse.org/downloads/eclipse-packages/)
* Turn on USB Debugging in Android Mobile Phone
* Download Appium Jar file for Eclipse
* Create a new Project for Appium in Eclipse
* Design test case to test application to launch the app on Android Device

To create a new AVD:

1. Open the **AVD Manager** by clicking Tools > **AVD Manager**.
2. Click Create Virtual Device, at the bottom of the **AVD Manager** dialog. ...
3. Select a hardware profile, and then click Next.
4. Select the system image for a particular API level, and then click Next.
5. Change **AVD** properties as needed, and then click Finish.

Please follow the steps in below link to setup AVD:

<https://developer.android.com/studio/run/managing-avds>

Note that Microsoft window does not support to run Appium Inspector.

Appium Doctor

To check the Appium installation and dependencies, you can install **appium-doctor** from [here](https://github.com/appium/appium-doctor).

**Appium-doctor** is an application tool to **verify Appium installation**. It shows all the missing things that you need to do. So, this will be very useful to run appium-doctor whenever you get any issue. It will install through **npm**.

Limitation of Appium

* Microsoft Windows does not support running Appium Inspector.
* Appium does not allow the testing for Android versions lower than 4.2.
* Appium provides limited support for testing Hybrid applications. E.g., Switching action of application is not possible to test i.e., web app to native app and vice versa.

Competitors of Appium

There are several tools available for automated testing mobile applications, such as Robotium, Appium, Experitest, Selendroid, Kobiton, and Testdroid, etc. They all are tough contestants for Appium. But Selendroid and Robotium are one of the top competitors of Appium. Let us know some differences and see how they differ from each other.

Appium vs Robotium

* **Appium** is a cross-platform tool that supports both iOS and Android. Whereas **Robotium** only supports Android.
* **Appium** supports various languages while **Robotium** only supports Java programming language.
* **Appium** does not require application source code/library, whereas **Robotium** tool requires application source code or library.
* **Appium** can be used to test native, web, and hybrid mobile applications, whereas **Robotium** can only test native and hybrid applications.
* **Appium** supports many frameworks like Selenium. But **Robotium** is not compatible with Selenium at all.
* **In Appium**, you don't have to reinstall the application for a small change. But **Robotium** code leads to complete rebuild for a small change.

Appium vs Selendroid

* **Appium** is an open-source automation tool that supports both iOS and Android, while **Selendroid** is a test automation framework that only supports Android.
* In **Appium**, a small change does not require reinstallation of the application. But **Selendroid** requires reinstallation of the application.
* **Appium** has a strong and active community, whereas **Selendroid** does not have a strong community like Appium.
* **Appium** supports many frameworks and languages. On the other hand, **Selendroid** is compatible with Jenkin and Selenium.
* **Appium** does not require application source code/library, while **Selendroid** requires application source code or library.
* **Appium** supports all Android APIs with a limitation. Appium uses UIAutomator for tests running on API>=17, while for older APIs, it runs tests using Selendroid.