**Download and install Java**

Set Java\_Home in environment variable

-Java\_Home : Java installed path

**Install appium-doctor**

npm install appium-doctor -g

To run: Appium-doctor

**Appium:**

1) Node.js - JRE for java same as Node.js for Javascript

-Install node.js in and check the version using

node -v

Note: close the command prompt ad reopen after install

2) Install the appium using npm:

npm install -g appium

Need to download Android SDK which is two methods

1. Install android studio then the SDK file will be downloaded along with that

2. Else download SDK manager packages manually

**3)a) Download Android studio**:

Find the Android SDK file in below mentioned path after Android Studio installation

C:\Users\<username>\AppData\Local\Android\SDK

After installaton set Android\_Home path in Environment variable

IN environment variable:

-Android\_Home : C:\Users\<username>\AppData\Local\Android\SDK ====>>IF SDK installed through Android studio

In path variable:

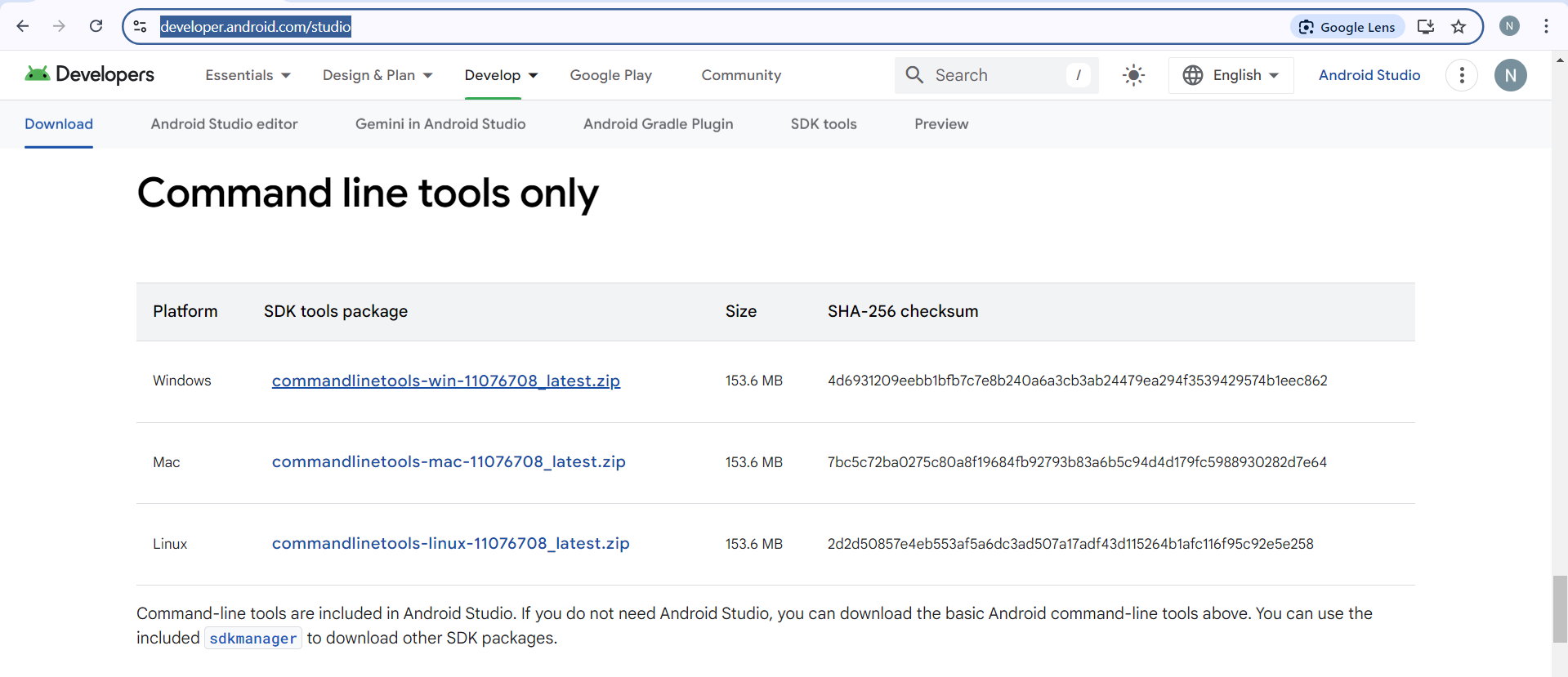
- %Android\_Home%

- %Android\_Home%\tools

- %Android\_Home%\platform-tools

**b) for installing Android command line tools:**

Install the command line tools for windows

****

-Android\_Home : respective path where the sdk command line tools download ===>If SDK installed through manually

-Once download create a latest folder and move the bin and lib files over there

The path would be C:/users/command-line\_tools/latest/bin

**Note: It will only install the sdk manager command line, we have install the platform and platform tools separately**

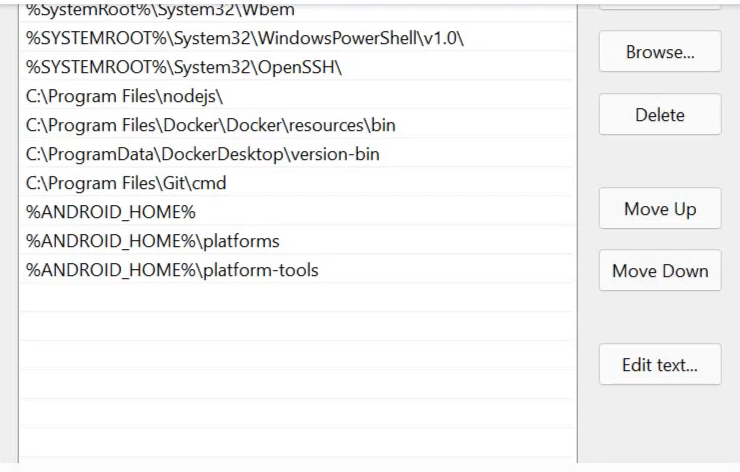
* Open the command line in sdk manager location and run the below command to download the platforms tools,

-sdkmanager "platform-tools" "platforms;android-33"

**Problem**: While run the above command, CLI throws error as JDK version and SDK version mismatch

**Solution**: Install the Java JDK version same as SDK manager version

* Set the path variables for platforms and platform tools



Here we Good to go…………

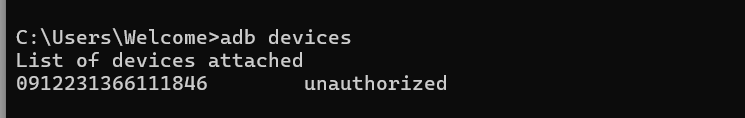
Turn on the Developer options in mobile.

* About Device/BuildNumber (click 7 times) . Developer options enabaled
* Turn on the USB debugging in Developer options

**adb – Android debug bridge**

Purpose: make the connection between system and mobile for debugging

To get the Device id, run the “adb devices” command



Create the first Project:

* Create maven project and download the below dependencies
* Selenium java
* Appium java client

Command to list the connected devices

* adb devices

To start the apm server, execute the below command

* appium

Lets code!!!!!!

* We need to set the below desired capabilities
* platform (OS)
* platform version (OS version)
* device id (udid)
* device name
* app package
* app activity

To find the app package and activity use below third party website

* apk info
* apk analyzer

or use the below command to find without using external apps

* adb shell

once the shell is opened, execute the below command

* **dumpsys activity | grep -E 'mCurrentFocus|mFocusedApp'**

Note: above command is only execute and return the package name when the respective application is opened

**Note: UiAutomator2 driver is necessary to connect with Appium.**

**To install UiAutomator2 driver, enter the below command**

**>appium driver install uiautomator2**

**>appium driver list --installed**

**Problem 1** : System throws session not created exception

Solution: downgrade appium java client to 8 version and seleninum java to 4

**Compatibility Matrix :** [https://github.com/appium/java-client?tab=readme-ov-file#compatibility-matrix](https://github.com/appium/java-client?tab=readme-ov-file%23compatibility-matrix)

**Problem 2**: Cannot find the Ui automator2 driver

Solution : install the uiautomator 2 driver 🡺 **appium driver install uiautomator2**

After installation restart the appium server

**Problem 3**: Could not find 'adb.exe' in ["C:\\Users\\Welcome\\Sdkmanager\\cmdline-tools\\latest\\platform-tools\\adb.exe","C:\\Users\\Welcome\\Sdkmanager\\cmdline-tools\\latest\\emulator\\adb.exe","C:\\Users\\Welcome\\Sdkmanager\\cmdline-tools\\latest\\cmdline-tools\\latest\\bin\\adb.exe","C:\\Users\\Welcome\\Sdkmanager\\cmdline-tools\\latest\\tools\\adb.exe","C:\\Users\\Welcome\\Sdkmanager\\cmdline-tools\\latest\\tools\\bin\\adb.exe","C:\\Users\\Welcome\\Sdkmanager\\cmdline-tools\\latest\\adb.exe"]. Do you have Android Build Tools installed at 'C:\Users\Welcome\Sdkmanager\cmdline-tools\latest'?

Solution: place the Platform-tools and platform under Latest folder.

Problem 4: After clearing all the error, still the app is not launching and the console is also clean without any error

Solution: Check all the capability name carefully, it doesn’t throw error if the capabilities name is wrong….

I have faced error because of appPackage capability , mistakenly typed as packageName

**Now How to inspect elements in app?**

If you are testing web page on a mobile browser of the app, you can connect the device and enable the debugging and hit “chrome://inspect”. You can inspect the elements through this on your web browser. But this works only with web browser not with installed app

* Install Appium Inspector
* Set the desired capabilities for the app
* Start the session

Difference between RemoteWebDriver, AppiumDriver, AndroidDriver and iOSDriver

**RemoteWebDriver**: Remotewebdriver used for Selenium project. Since appium operates on the client-server model, appium uses this to initialize the session. However directly using RemoteWebdriver is not recommended since there are other drives available that offer additional feature or convenience function

**AppiumDriver**: This driver class inherits from RemoteWebDriver, but it adds in additional functions that are useful in the context of a mobile automation test through the appium server.

**AndroidDriver**: Only use this driver class, if you want to start a test on Android device or android emulator

**iOSDriver**: : Only use this driver class, if you want to start a test on iOS device or iOS emulator

**How to connect device without USB cable:**

Kill the adb server Initially using **adb kill-server**

Turn on the Wirless debugging in Developer options

Select pair device with pairing code option, it will display ip address with pairing code. Pair the ip adrdress with adb device using below command

* adb pair 192.168.x.x:portname

ex: adb pair 192.168.1.3:33455

Then system ask for pairing code, enter the pairing code shown in mobile device and then connect the device with the static port number displayed over the pair device option,

* adb connect 192.168.1.3:36319

Mobile Locators:

* id
* accessibility id
* classname
* Xpath

**Appium-gesture Plugin: (for scroll and swipe action)**

* Install the plugin

**appium plugin install --source=npm appium-gestures-plugin**

* Activate the plugin

**appium --use-plugins=gestures**

**Note:** Do not activate the plugin in different port, activate the plugin along with the appium server

