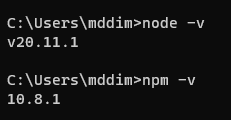
# Appium: Installations and Guidelines

## Appium Basic Setup on Windows

1. **Check if Node.js is installed**

node -v **AND** npm -v

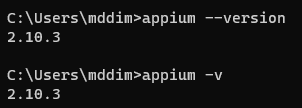


1. **If not, download & install Node.js**

[**https://nodejs.org/en/download/**](https://nodejs.org/en/download/)

1. **Check if appium is available on the system**

appium --version **OR** appium -v



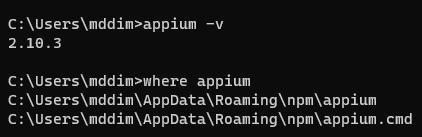
1. **If not, install Appium via NPM**

npm install -g appium@latest

Installing Appium 2.0 only installs the Appium server, but not the drivers since all drivers have been isolated from the Appium server app. We will Install the Appium UiAutomator2 Driver for Android separately

1. **Verify appium is available on the system and its location**

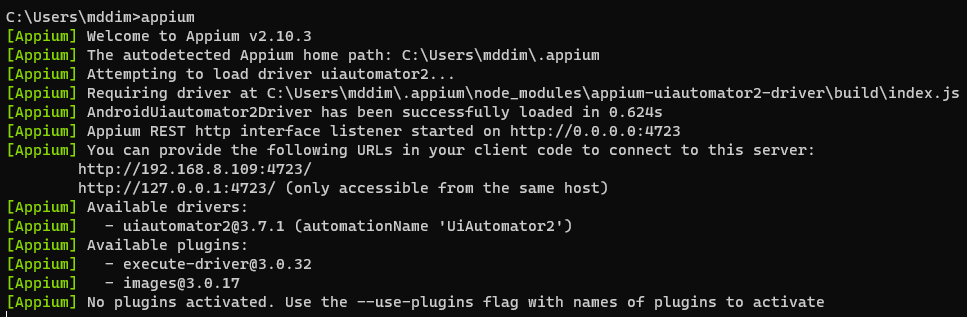
appium -v  **AND** where appium



1. **Start Appium to get information about it (CTRL + C to quit)**

appium

Your output will be slightly different from the example shown, because the uiautomator2 driver is not installed on your system.

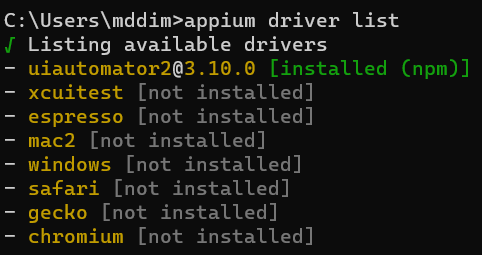


1. **Install required Appium driver**

appium driver install uiautomator2

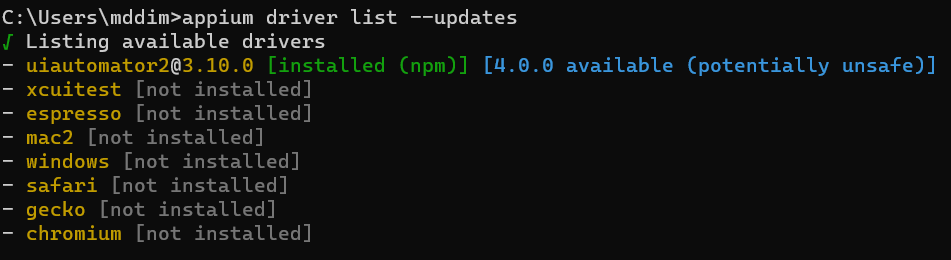
1. **Check installed drivers**

appium driver list



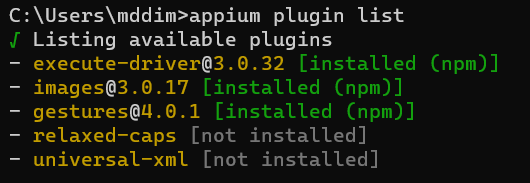
1. **Check for available updates (an update might not be needed).**

appium driver list --updates



1. **Check plugins (you don't have any installed yet, but just to be aware that they exist)**

appium plugin list



1. **Run Appium server**

appium **OR** appium --allow-cors

\* **Press Ctrl+C to stop appium server.**

**--allow-cors**: This is a flag that tells the Appium server to enable CORS - Cross-Origin Resource Sharing (CORS).

You might need to use this command if you're using a testing framework that runs on a different domain than the Appium server.

For example, if you're using the Appium Inspector Web Version to record and edit test scripts, you'll need to start the Appium server with CORS enabled.

**Pretty easy, right? Wrong! We're just getting started. 😊**

Even though Appium is installed and running, it has a bunch of dependencies, which means it cannot automate anything yet. We have to **set up Android automation requirements**.

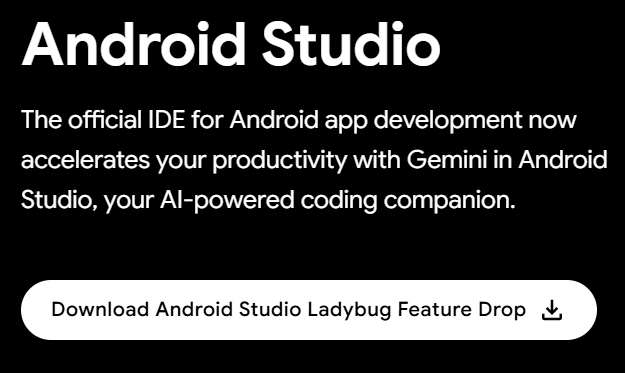
According to the driver's documentation, in addition to a working Appium server, we also need to set up   
the following:

* Android SDK
* Android SDK Tools
* Set up the ANDROID\_HOME environment and Path Variables
* Check if Java JDK is installed
* Set up the JAVA\_HOME environment and Path Variables
* Create and launch an Android Virtual Device (AVD)

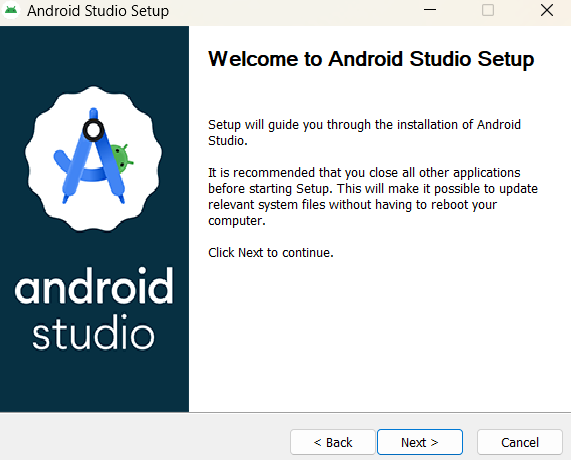
\* If you have Appium **Server open, close it.**

## Setup Android SDK and SDK tools

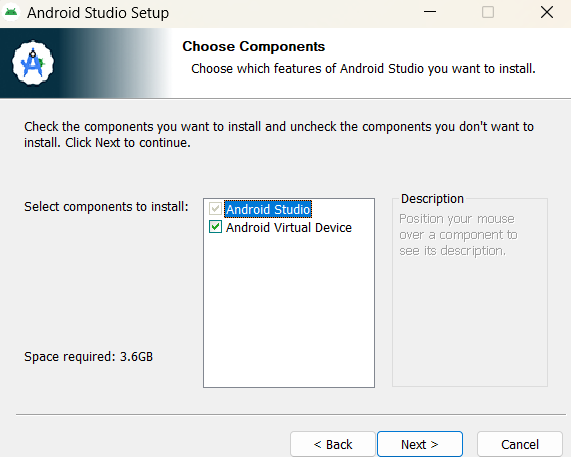
The easiest way to set up the Android SDK requirements is by downloading Android Studio. Download it from here: <https://developer.android.com/studio>.

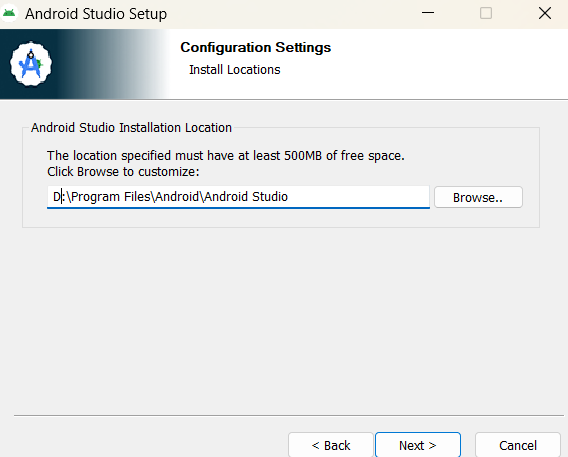


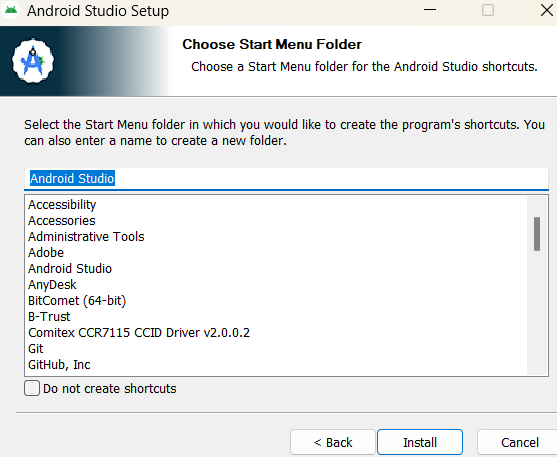
**Open the file** and follow the **installation guide**.

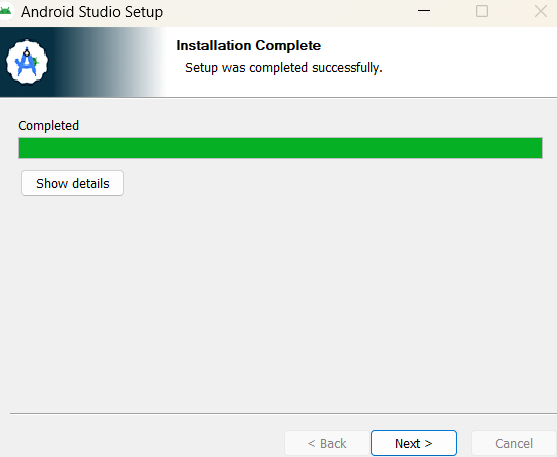


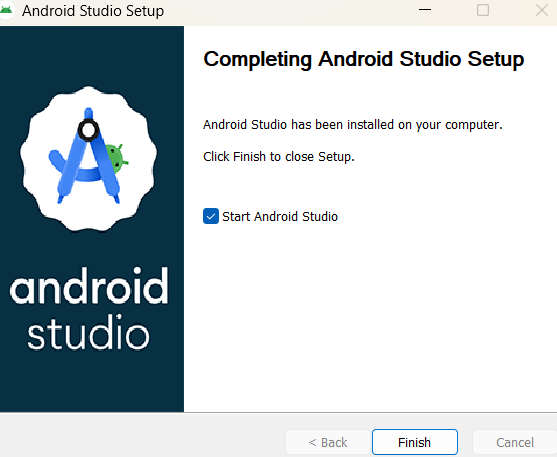
**We will use the emulator, so check "Android Virtual Device"**







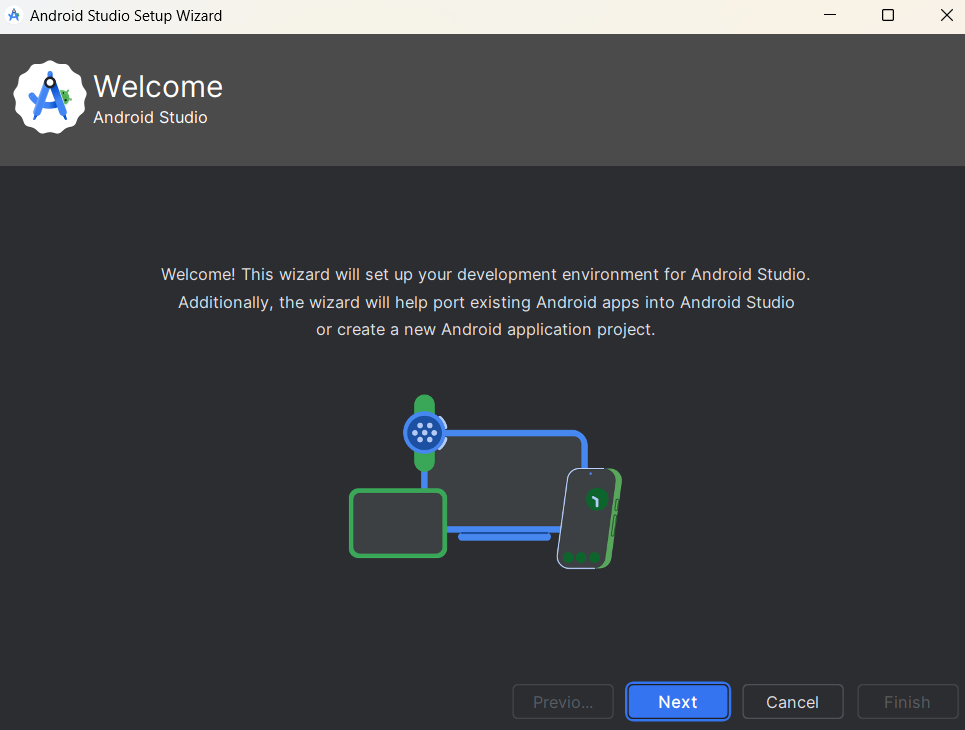


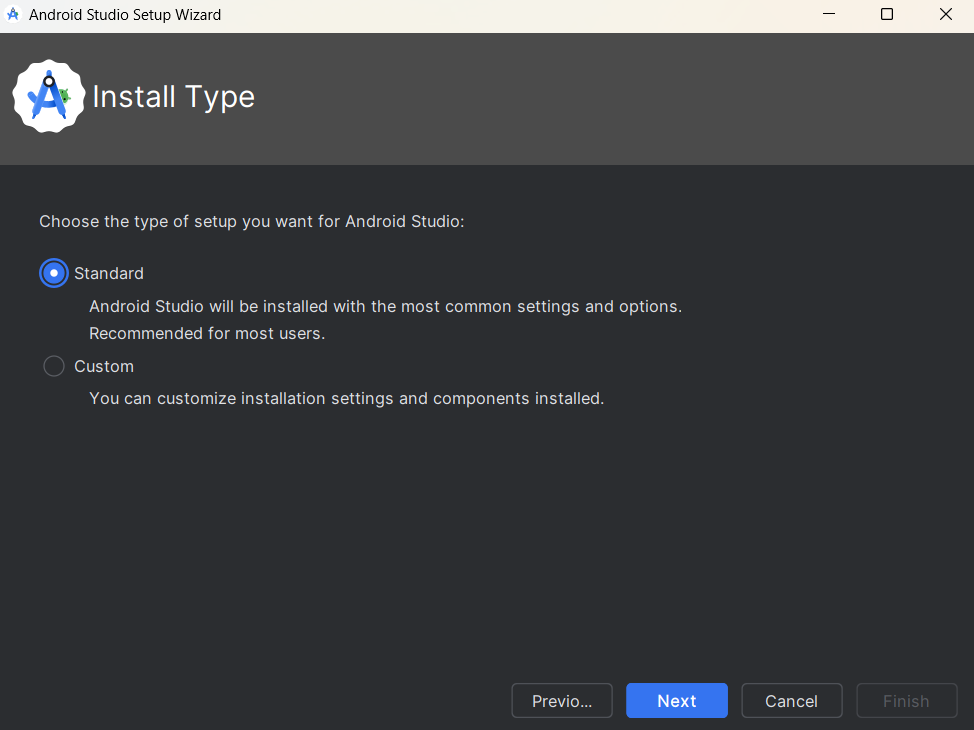
If you **check** "**Start Android Studio**" this will automatically open **Android Studio Setup Wizard** when you **click Finish.**

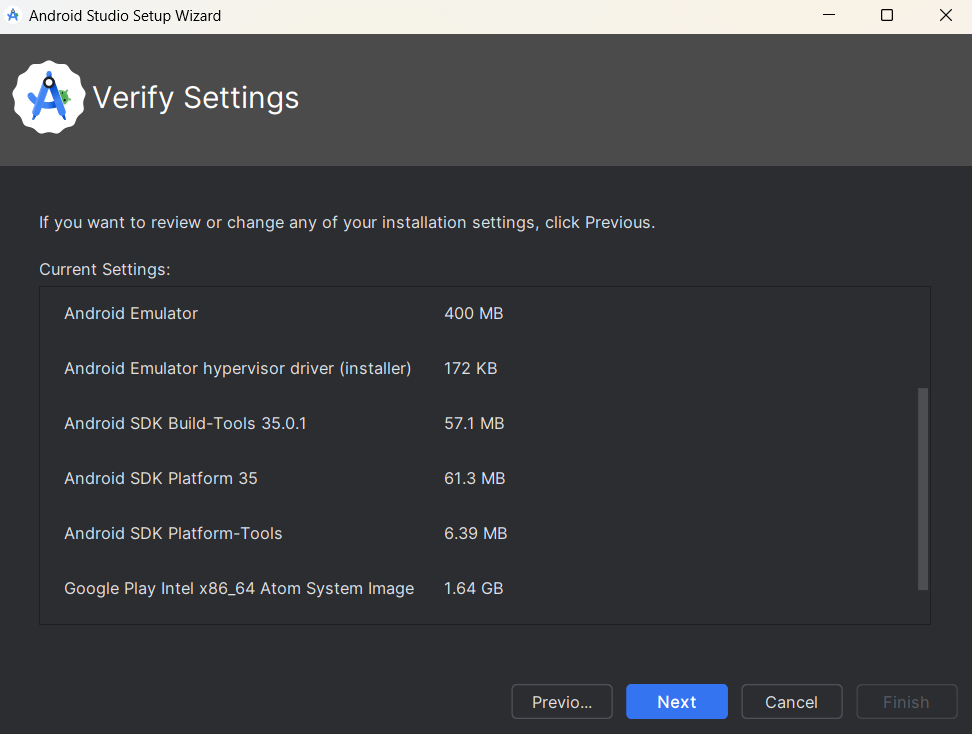
## Installing Android SDK

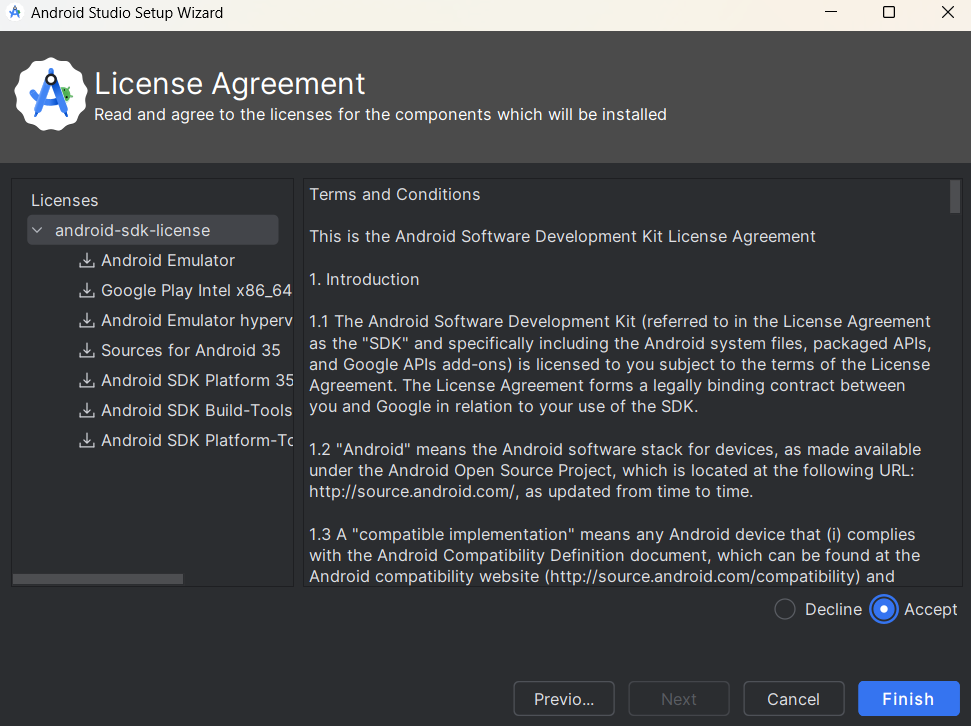
The Android SDK is an essential toolkit for any developer looking to create, test, and debug Android applications. It includes various tools, libraries, and documentation to streamline the development process and ensure that apps are compatible with the wide range of Android devices available in the market.

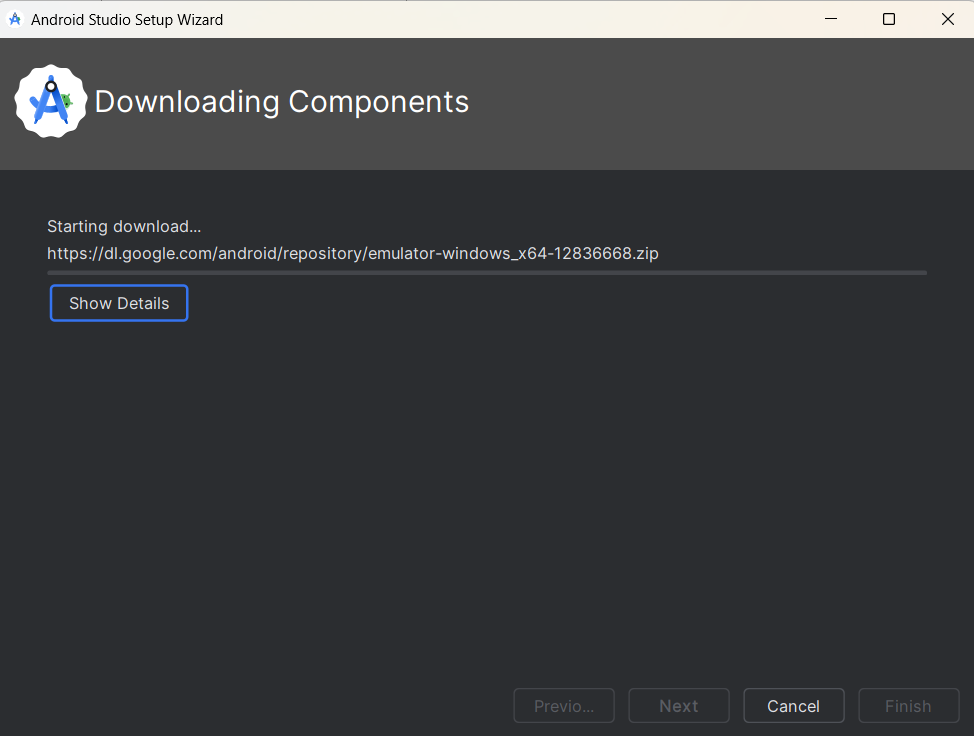
**You don't have to select any options. They're preselected for you.**

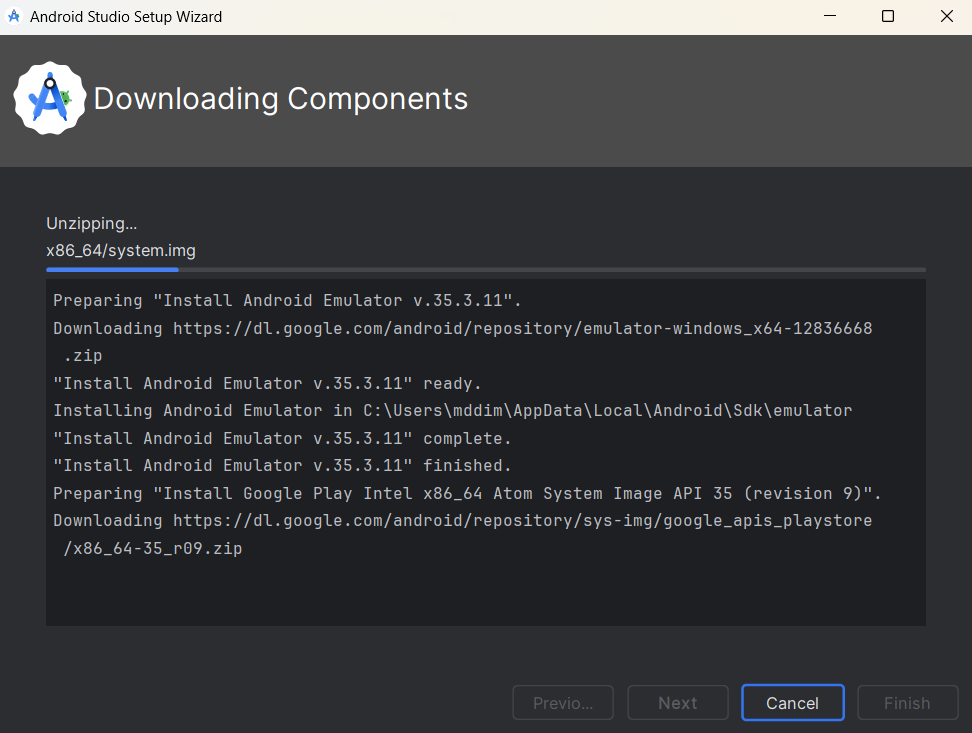


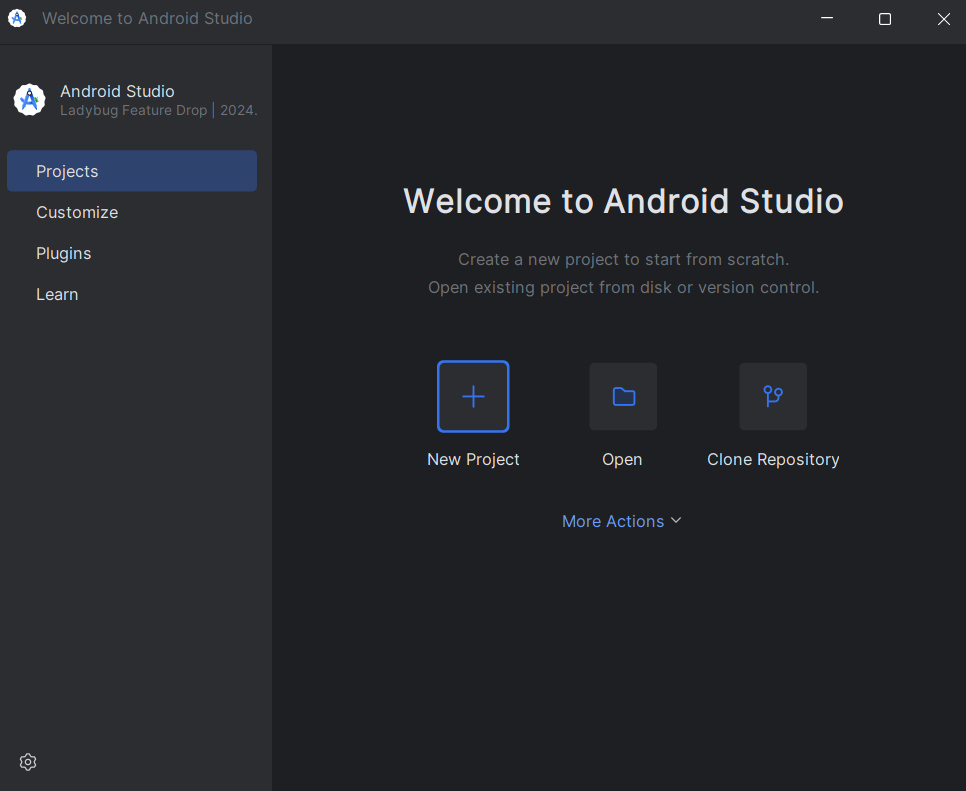












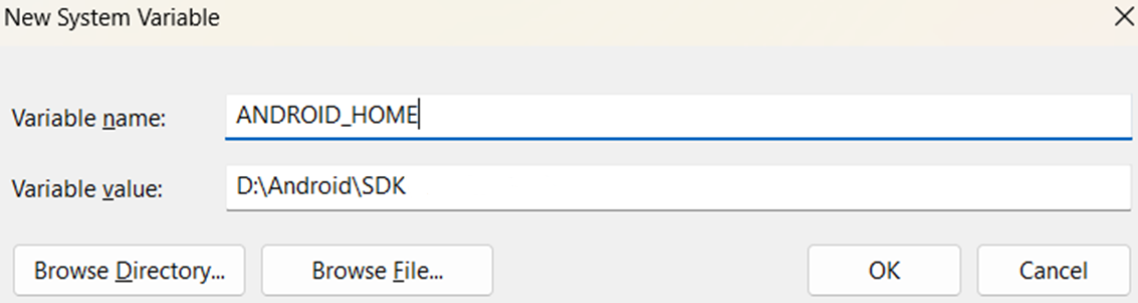
**You have successfully installed Android Studio, as well as Android SDK. Close it for now.**

## Set Environment Variables ANDROID\_HOME and Path

### Set ANDROID\_HOME

The environment variable ANDROID\_HOME is necessary for Appium to connect to Android Studio. Type **[variables]** in Windows search bar and open [Edit the system environment variables]

* Click "**Environment Variables**" button
* Under "**System variables**", click "**New**".
* Enter the variable name as **ANDROID\_HOME**.
* Set the **variable value** to the path where your **Android SDK is located**.

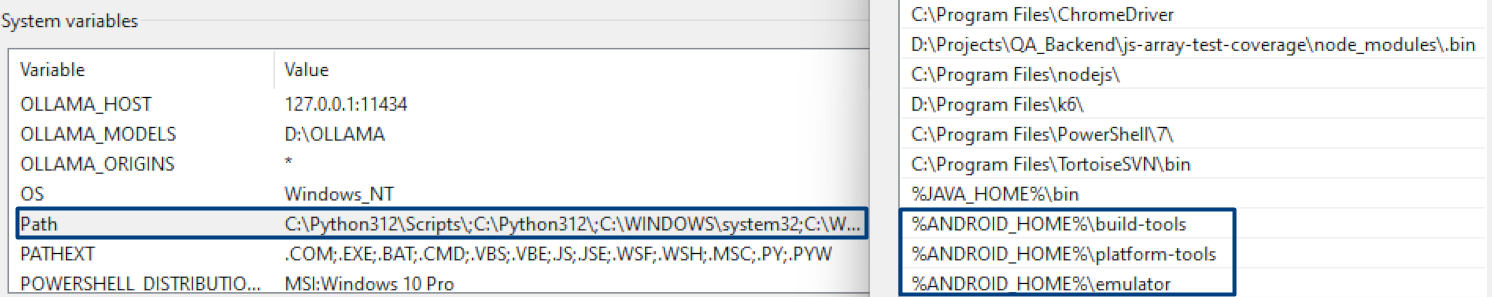


### Editing the Path Variable:

Select Path from the list of System variables and click "**Edit**". Click "**New**" to add each new path:

* **%ANDROID\_HOME%\build-tools**
* **%ANDROID\_HOME%\platform-tools**
* **%ANDROID\_HOME%\emulator**

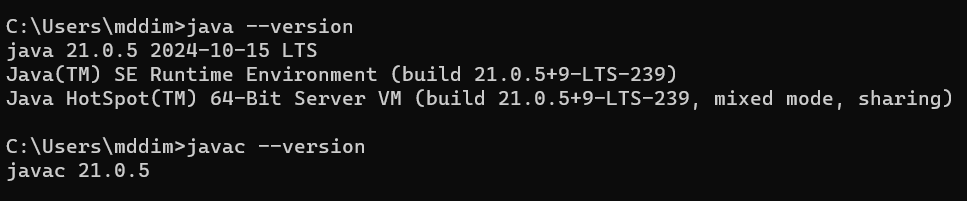
**Ensure each path is correctly entered** and then **click "OK"** to close **each dialog**.



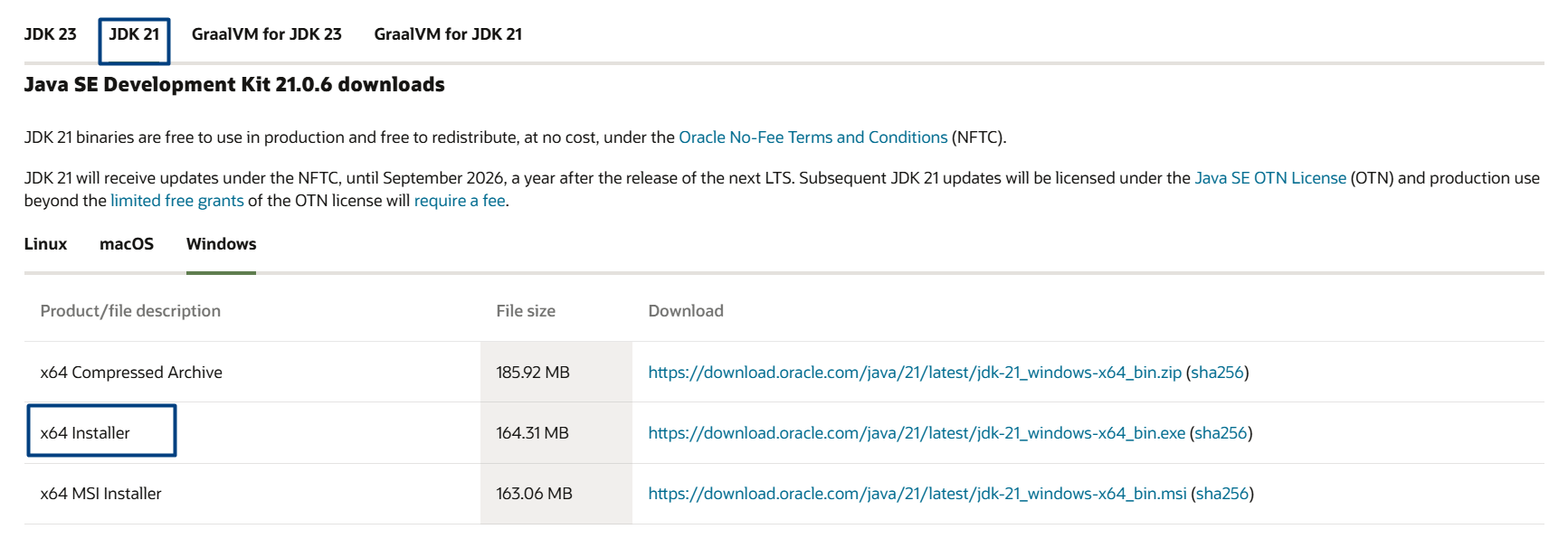
## Setup Java JDK

1. **Check Java JDK is present**

java --version **AND** javac --version



1. **If not, download & install Java JDK**[**https://www.oracle.com/java/technologies/downloads/#java21**](https://www.oracle.com/java/technologies/downloads/#java21)



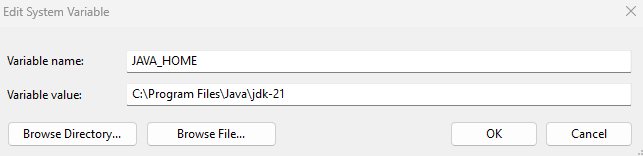
1. **Using Installer (.exe) is the easiest way.**

## Set Environment Variables JAVA\_HOME and Path

### 6.1. Set JAVA\_HOME

The **JAVA\_HOME** environment variable is necessary to specify the location of the Java Development Kit (JDK) installation directory. This is required by many development tools and applications, including Appium, to run   
Java-based commands and tools. Type **[variables]** in Windows search bar and open [Edit the system environment variables]

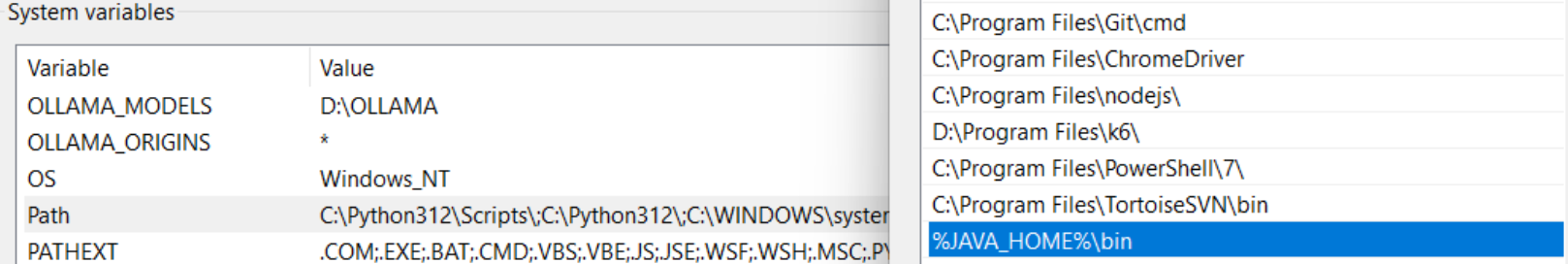
* Click "**Environment Variables**" button
* Under "**System variables**", click "**New**".
* Enter the variable name as **JAVA\_HOME**.
* Set the **variable value** to the path where your **Java JDK is located**.



### 6.2. Editing the Path Variable:

Select Path from the list of System variables and click "**Edit**". Click "**New**" to add the new path:

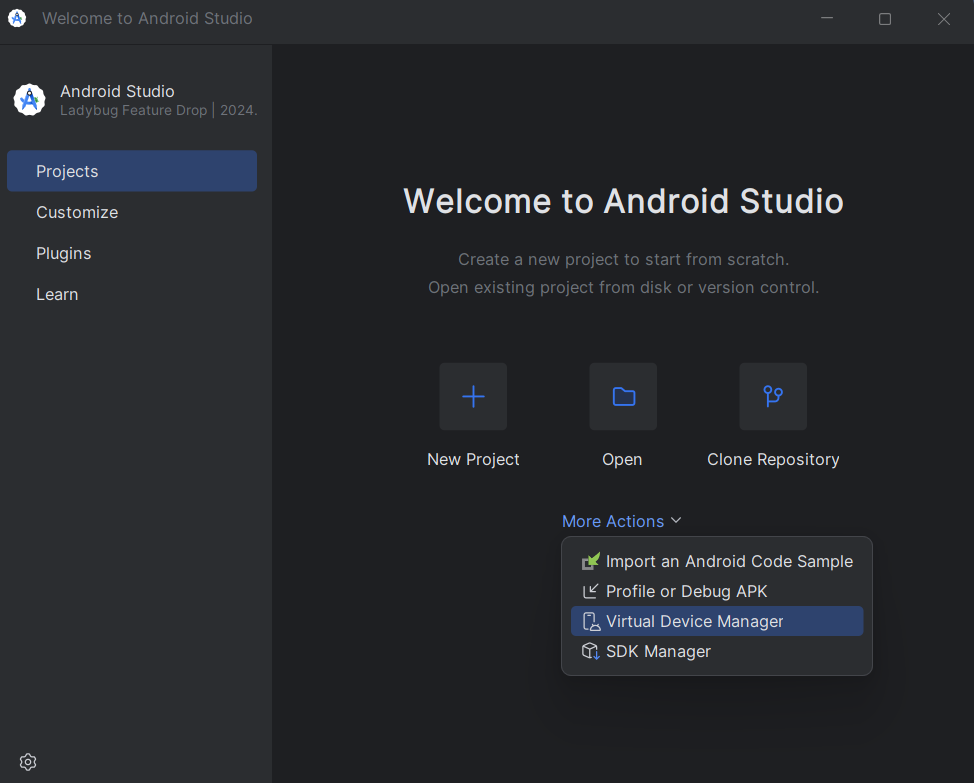
* **%JAVA\_HOME%\bin**



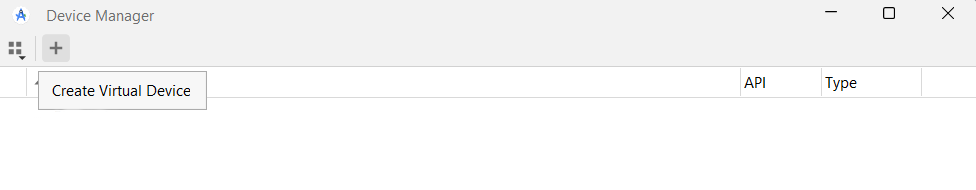
This path points to the **bin directory** within the JDK installation directory, which contains the Java executable files necessary for running Java applications.

## Create and Run an Android Virtual Device (AVD)

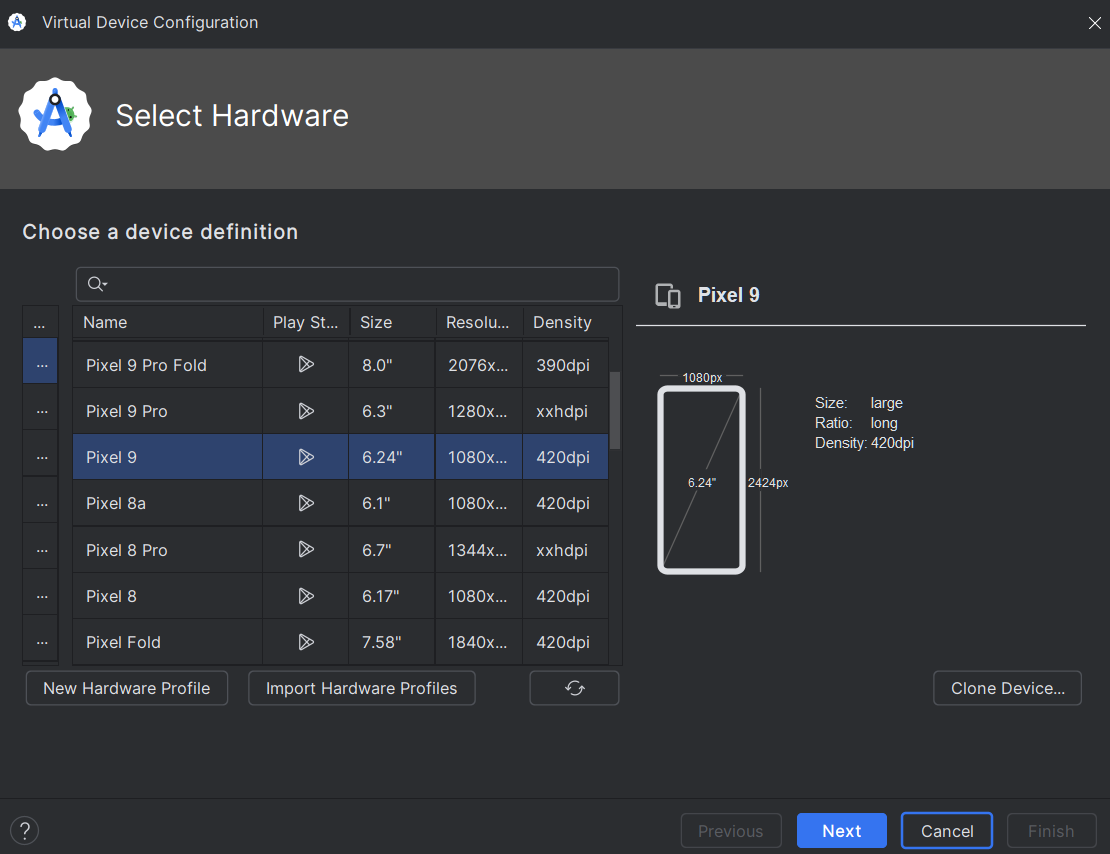
1. **Open** Android Studio 🡪 **More Actions** 🡪 **Virtual Device Manager**



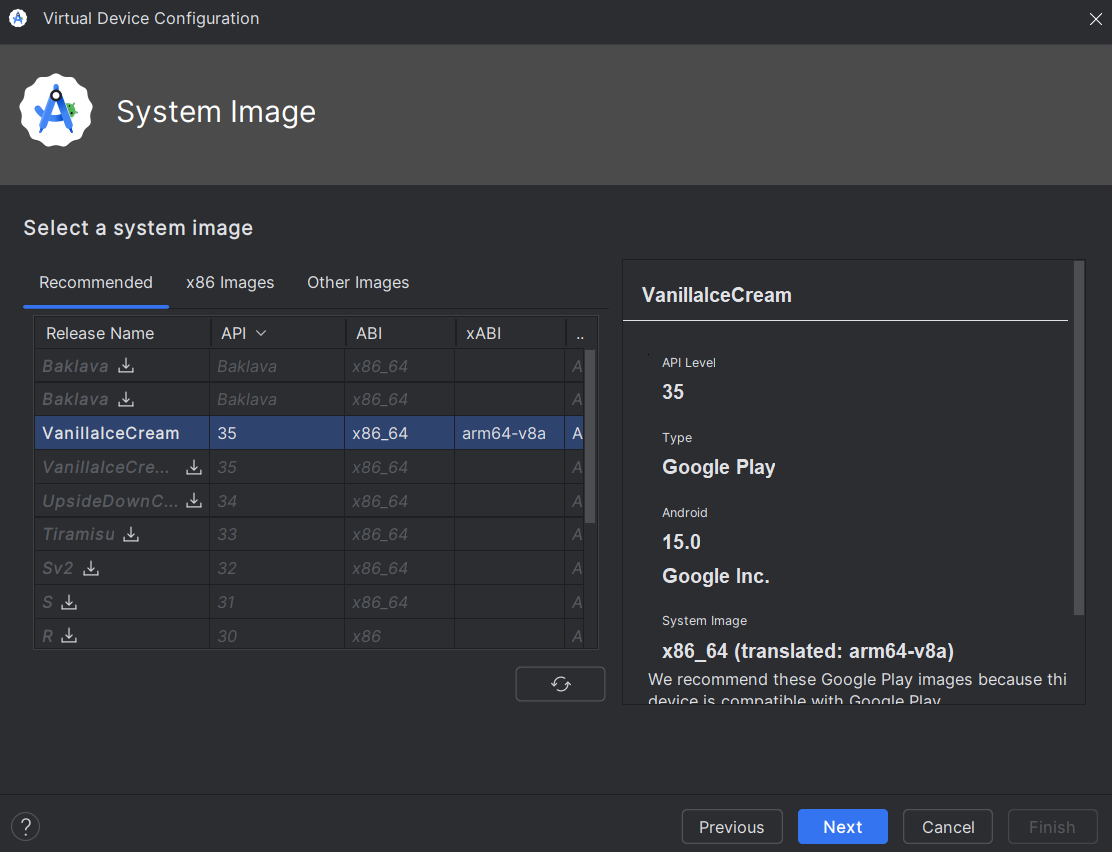
1. **Now we will create an** Android virtual device**, on which tests will be executed.**



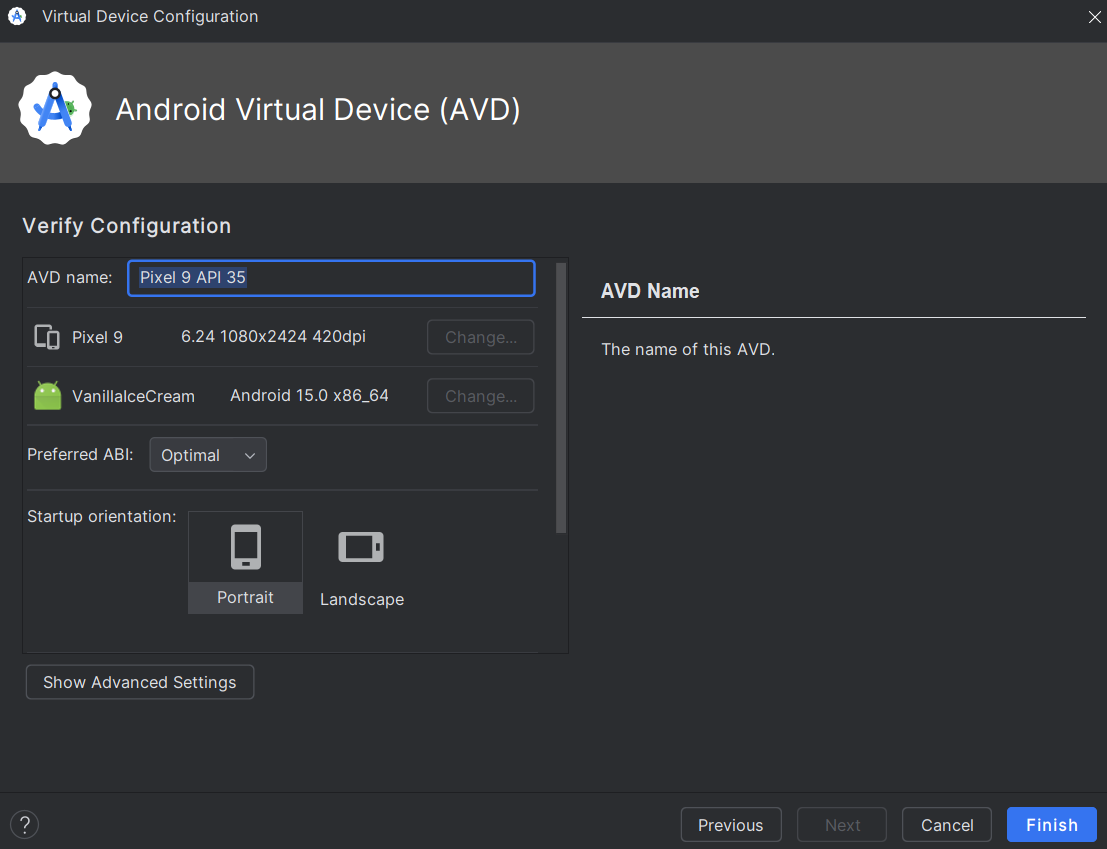
1. **You can choose whichever phone device definition you want for your AVD. We chose "Pixel 9".**



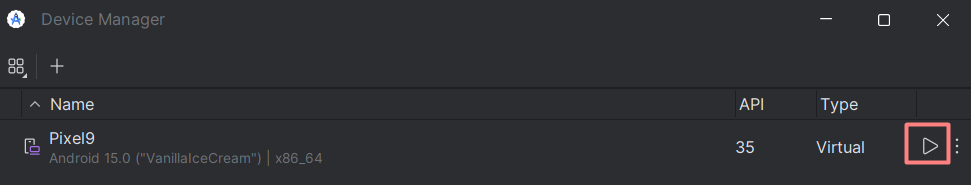
1. **Now you have to download the system image. Click down pointing arrow to do so. You can select a system image of your preference. We chose "VanillaIceCream". Wait for the downloading of the** SDK **to finish.**
2. **Press** [Finish] **at the end. Your installation is now complete.**
3. **Select the newly downloaded system image. Next.**



1. **You can change the name of your device.**



1. **Your AVD is ready for use. Start it by pressing its** [Play] **button in Device Manager:**



1. **Wait for the** AVD **to load:**



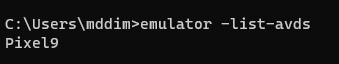
## Run an AVD via Command Prompt

Since you already have created an AVD, you don't have to open Android Studio and use the Device Manager to run it.

A much simple way to run an AVD is via CMD or terminal of your choosing.

* Open CMD or PowerShell
* Run the following command to see the available AVDs:

emulator -list-avds



* Run the following command to start the AVD (replace Pixel9 with the name of your AVD):

emulator -avd Pixel9

* You can also start AVD with a specific resolution (size) using **-skin** option (replace Pixel9 with the name of your AVD):

emulator -avd Pixel9 -skin 450x750

* To start AVD in a new state, without maintaining the old state (replace Pixel9 with the name of your AVD):

emulator -avd Pixel9 -no-snapshot-load

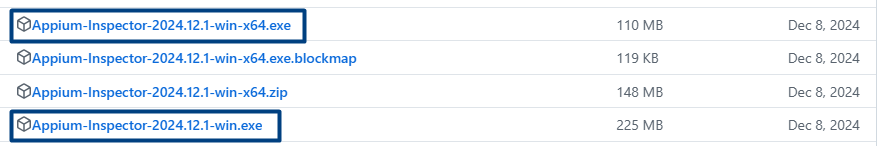
## Install Appium Inspector (Optional)

### 9.1. Use Web Version of Appium Inspector

* **If you tired of all the installations, use the web version here:**  
  <https://inspector.appiumpro.com/>

### Install Appium Inspector

* Install the corresponding for your system version of Appium Inspector here:  
  <https://github.com/appium/appium-inspector/releases>



* **When you open the .exe file you might get "Windows protected your PC" error. Click on "More info", and then "Run anyway".**

