

# Appium: Installations and Guidelines

## 1. Appium Basic Setup on Windows

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

```
node -v      AND      npm -v
```

```
C:\Users\mddim>node -v  
v20.11.1
```

```
C:\Users\mddim>npm -v  
10.8.1
```

### 2. If not, download & install Node.js

<https://nodejs.org/en/download/>

### 3. Check if appium is available on the system

```
appium --version OR appium -v
```

```
C:\Users\mddim>appium --version  
2.10.3
```

```
C:\Users\mddim>appium -v  
2.10.3
```

### 4. 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

### 5. Verify appium is available on the system and its location

```
appium -v      AND      where appium
```

```
C:\Users\mddim>appium -v  
2.10.3  
  
C:\Users\mddim>where appium  
C:\Users\mddim\AppData\Roaming\npm\appium  
C:\Users\mddim\AppData\Roaming\npm\appium.cmd
```

### 6. 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.

```
C:\Users\mddim>appium
[Appium] Welcome to Appium v2.10.3
[Appium] The autodetected Appium home path: C:\Users\mddim\.appium
[Appium] Attempting to load driver uiautomator2...
[Appium] Requiring driver at C:\Users\mddim\.appium\node_modules\appium-uiautomator2-driver\build\index.js
[Appium] AndroidUiAutomator2Driver has been successfully loaded in 0.624s
[Appium] Appium REST http interface listener started on http://0.0.0.0:4723
[Appium] You can provide the following URLs in your client code to connect to this server:
      http://192.168.8.109:4723/
      http://127.0.0.1:4723/ (only accessible from the same host)
[Appium] Available drivers:
[Appium]   - uiautomator2@3.7.1 (automationName 'UiAutomator2')
[Appium] Available plugins:
[Appium]   - execute-driver@3.0.32
[Appium]   - images@3.0.17
[Appium] No plugins activated. Use the --use-plugins flag with names of plugins to activate
```

## 7. Install required Appium driver

```
appium driver install uiautomator2
```

## 8. Check installed drivers

```
appium driver list
```

```
C:\Users\mddim>appium driver list
✓ Listing available drivers
- uiautomator2@3.10.0 [installed (npm)]
- xcuitest [not installed]
- espresso [not installed]
- mac2 [not installed]
- windows [not installed]
- safari [not installed]
- gecko [not installed]
- chromium [not installed]
```

## 9. Check for available updates (an update might not be needed).

```
appium driver list --updates
```

```
C:\Users\mddim>appium driver list --updates
✓ Listing available drivers
- uiautomator2@3.10.0 [installed (npm)] [4.0.0 available (potentially unsafe)]
- xcuitest [not installed]
- espresso [not installed]
- mac2 [not installed]
- windows [not installed]
- safari [not installed]
- gecko [not installed]
- chromium [not installed]
```

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

```
appium plugin list
```

```
C:\Users\mddim>appium plugin list
✓ Listing available plugins
- execute-driver@3.0.32 [installed (npm)]
- images@3.0.17 [installed (npm)]
- gestures@4.0.1 [installed (npm)]
- relaxed-caps [not installed]
- universal-xml [not installed]
```

## 11. 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)

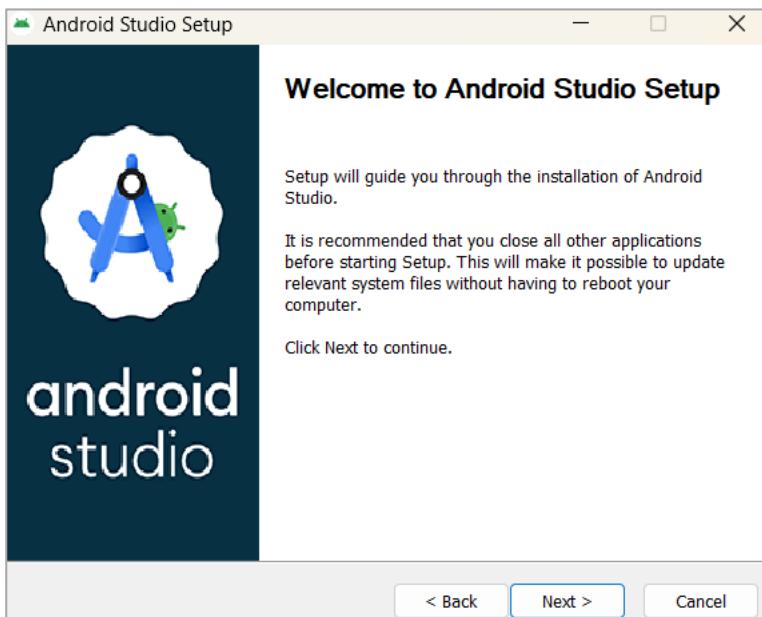
\* If you have **Appium Server open, close it**.

## 2. Setup Android SDK and SDK tools

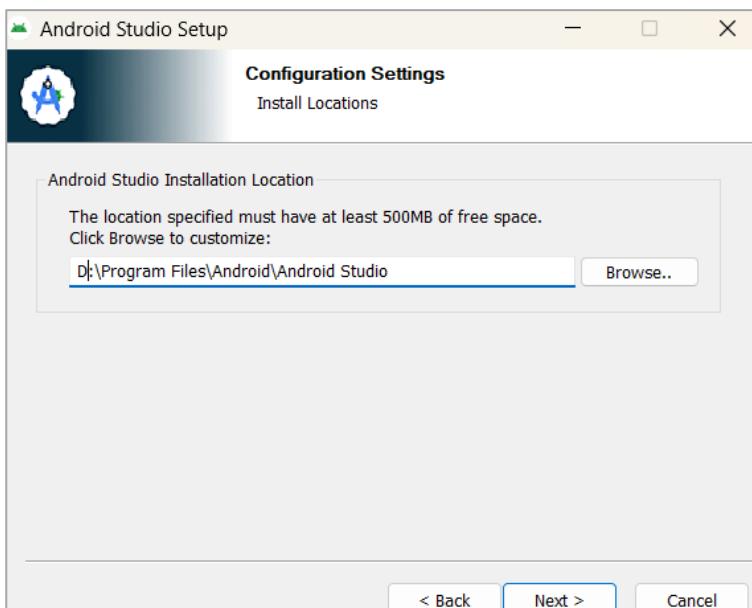
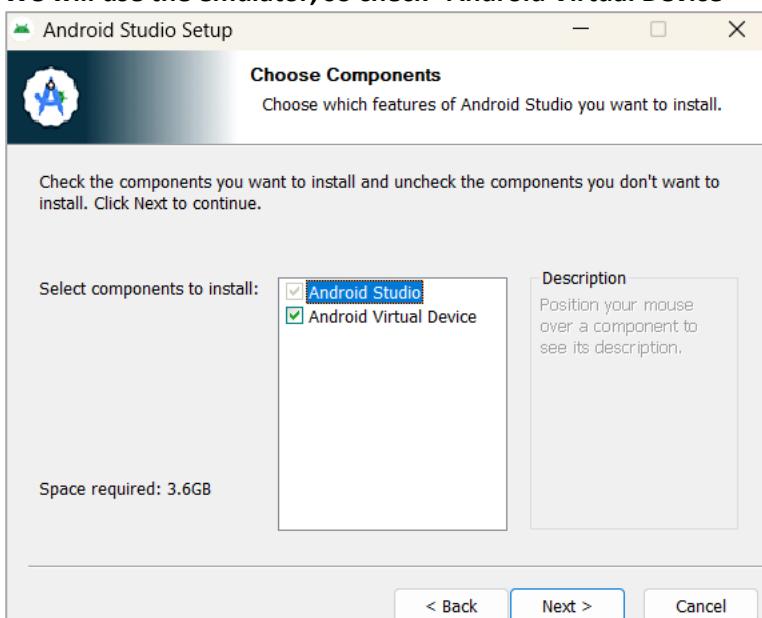
The easiest way to set up the Android SDK requirements is by downloading Android Studio. Download it from here: <https://developer.android.com/studio>.

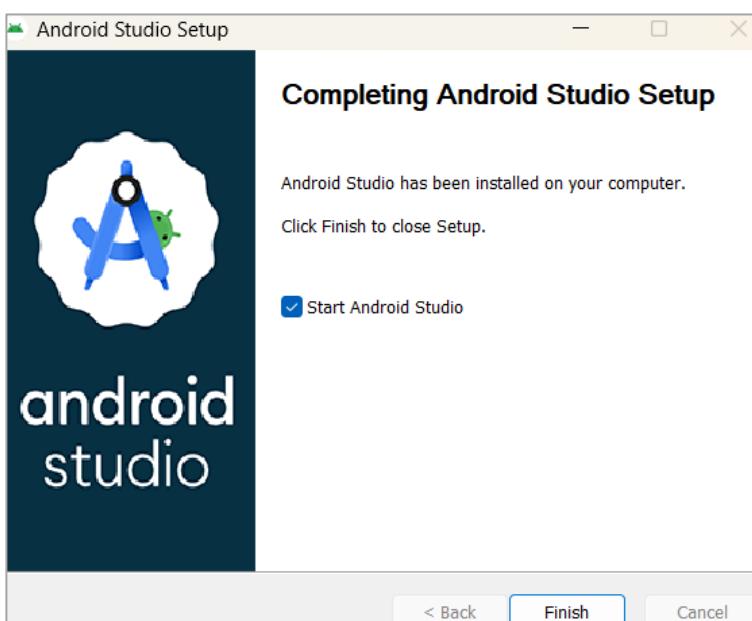
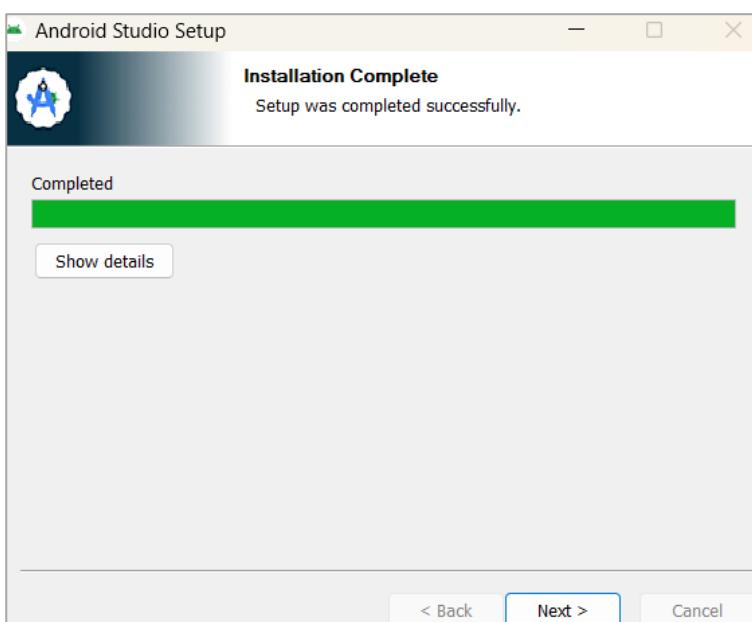
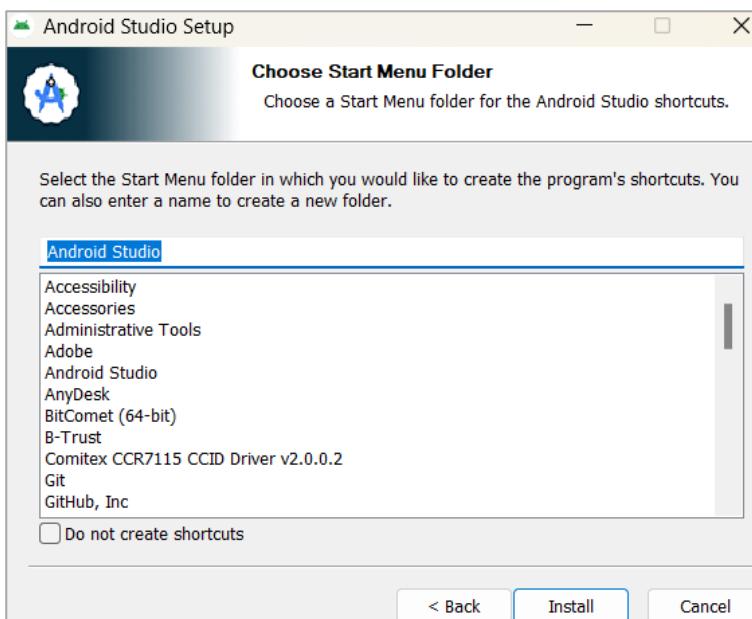


Open the file and follow the installation guide.



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



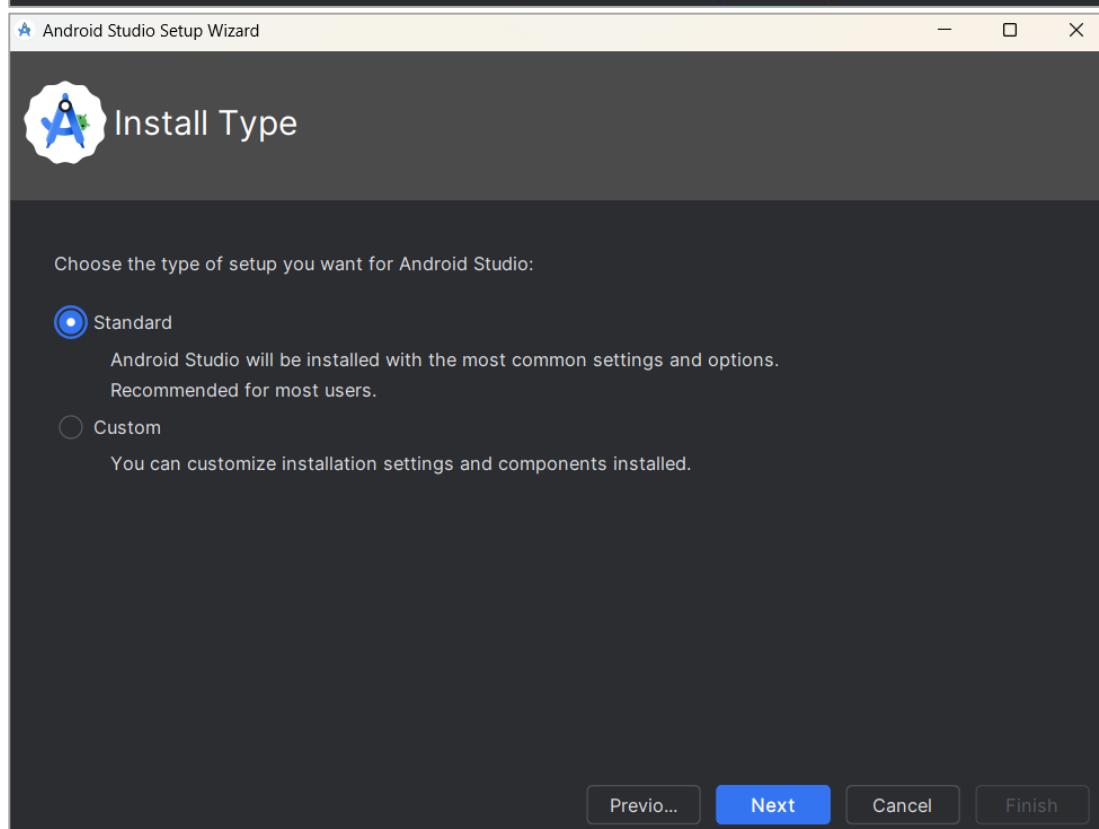
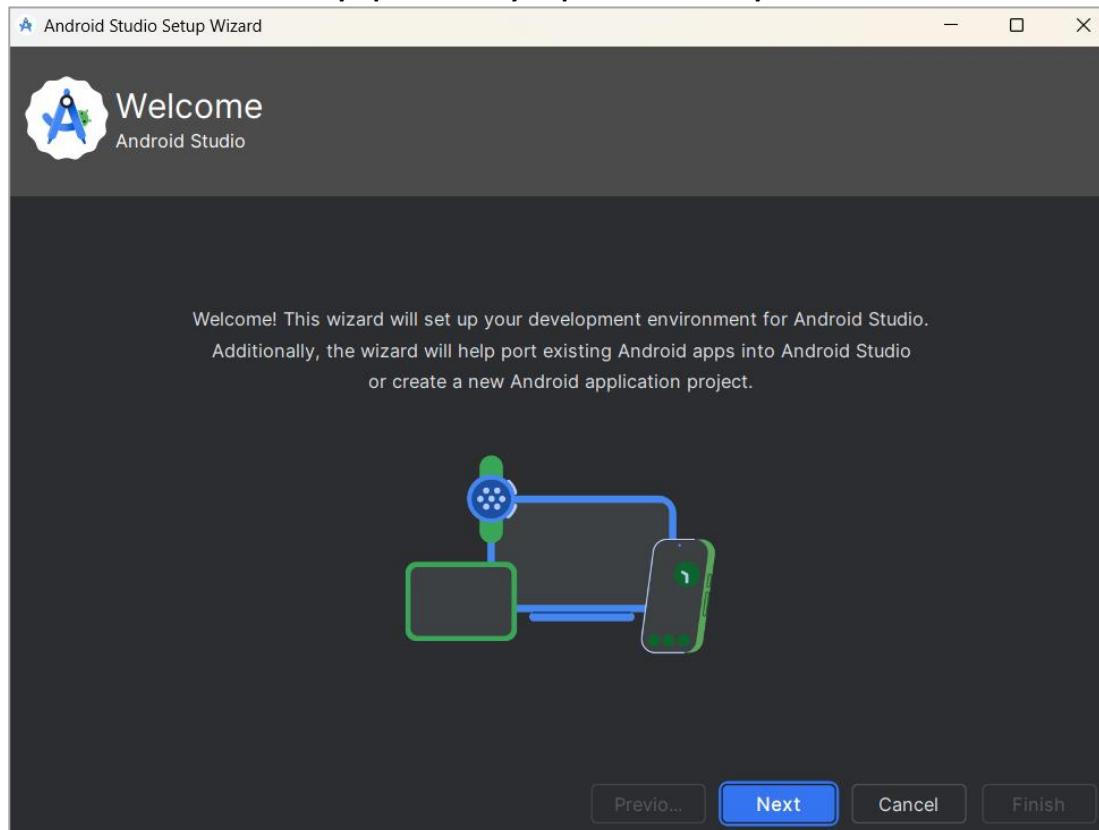


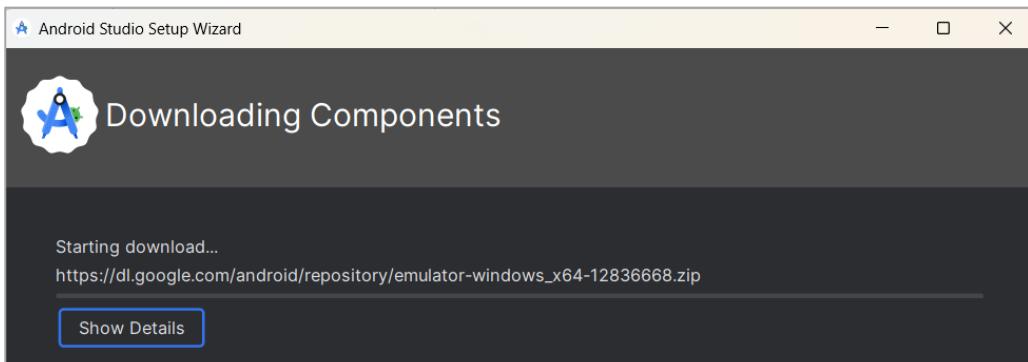
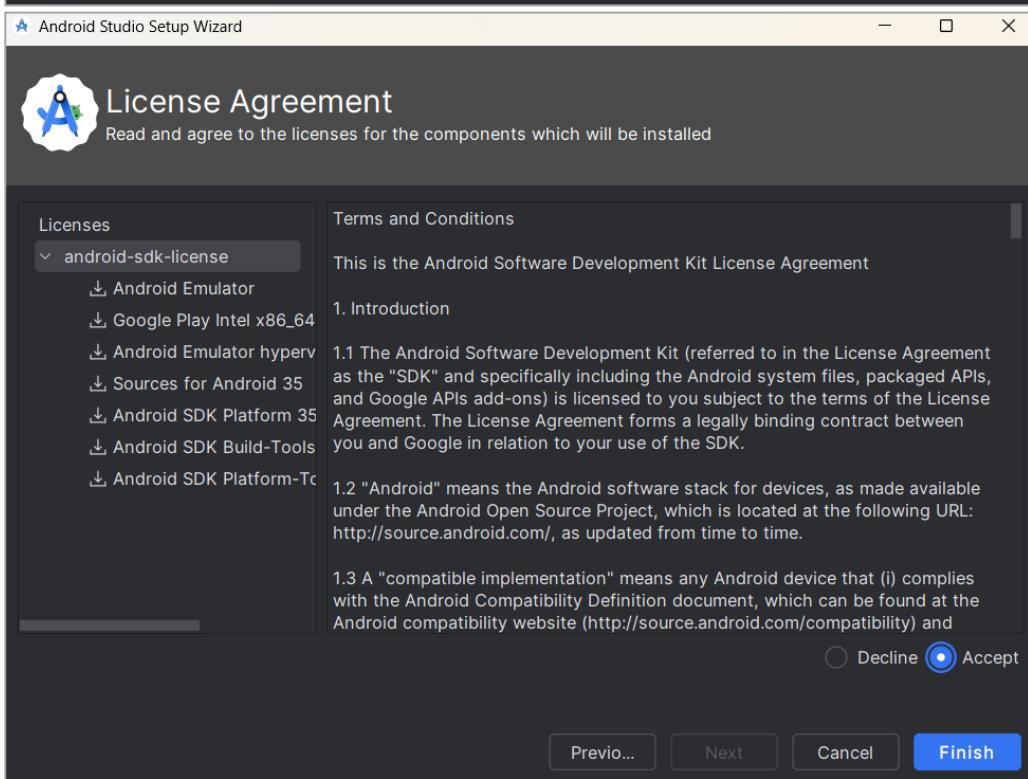
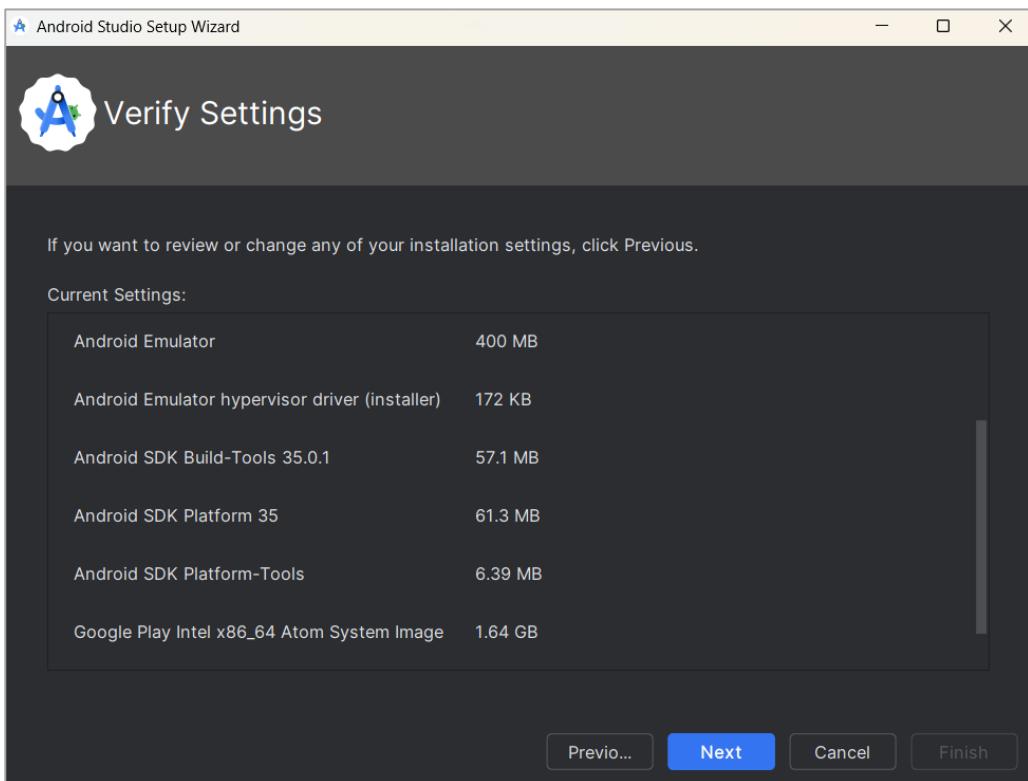
If you check "Start Android Studio" this will automatically open **Android Studio Setup Wizard** when you click **Finish**.

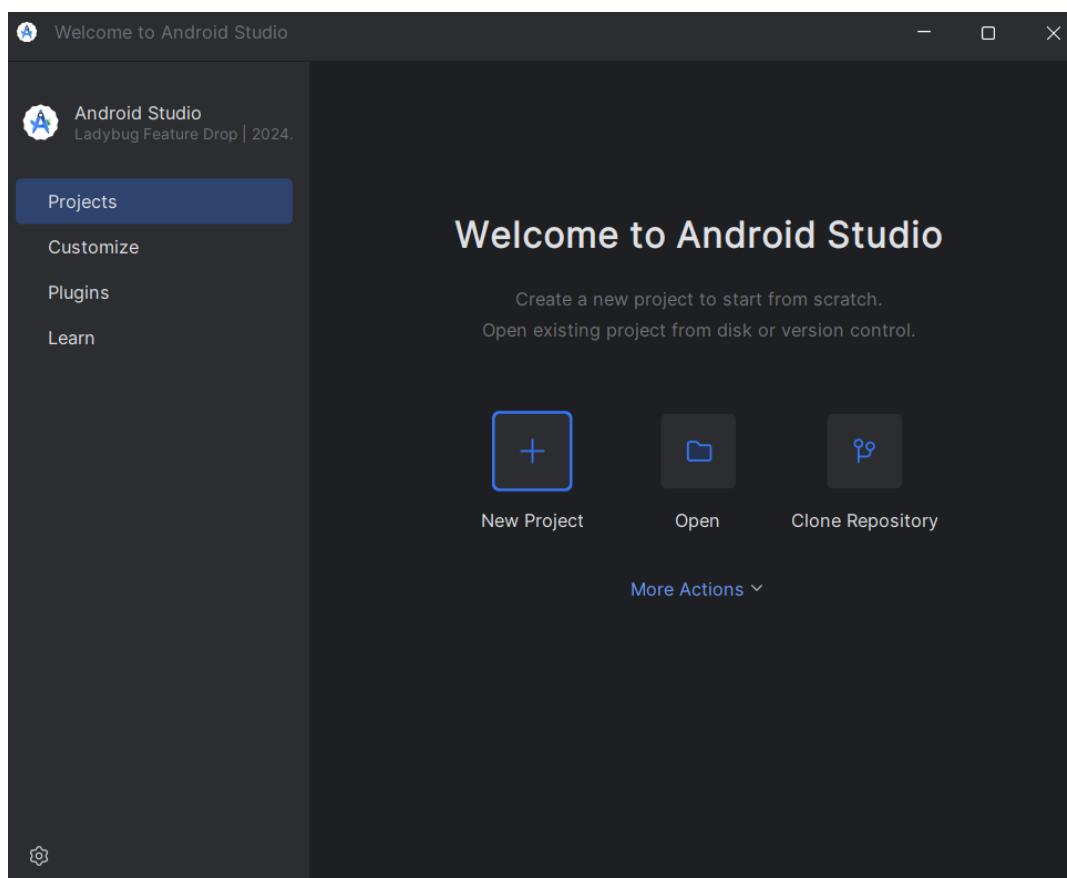
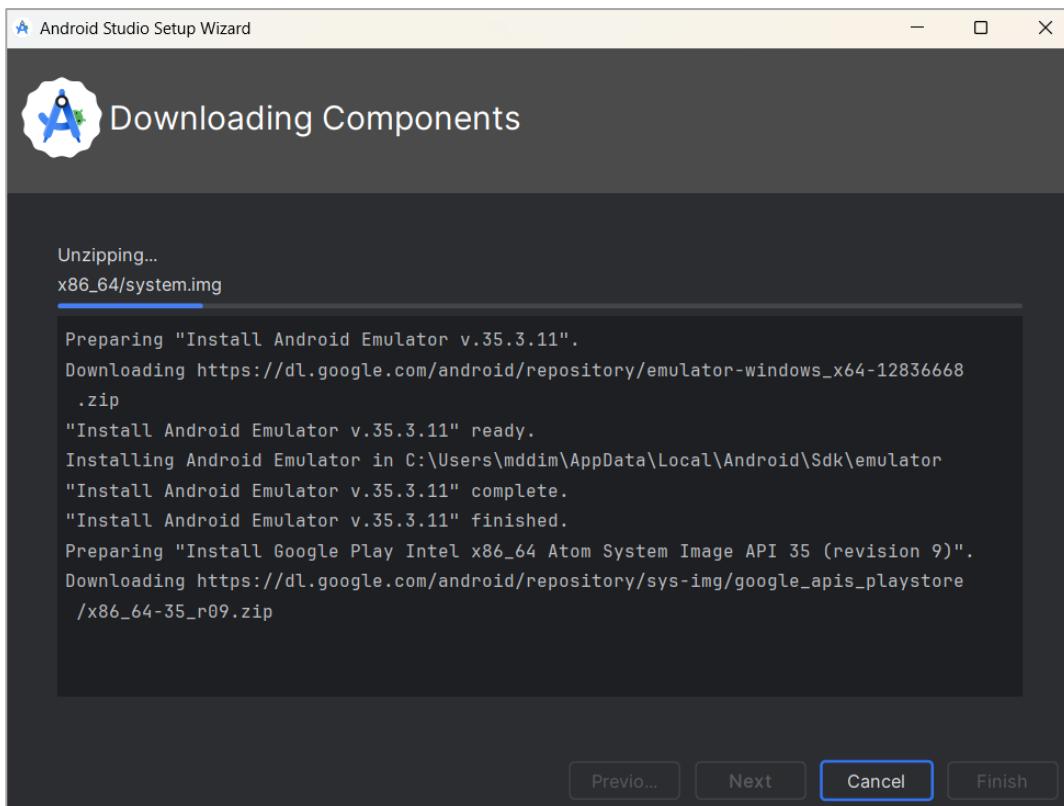
### 3. Installing Android SDK

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.







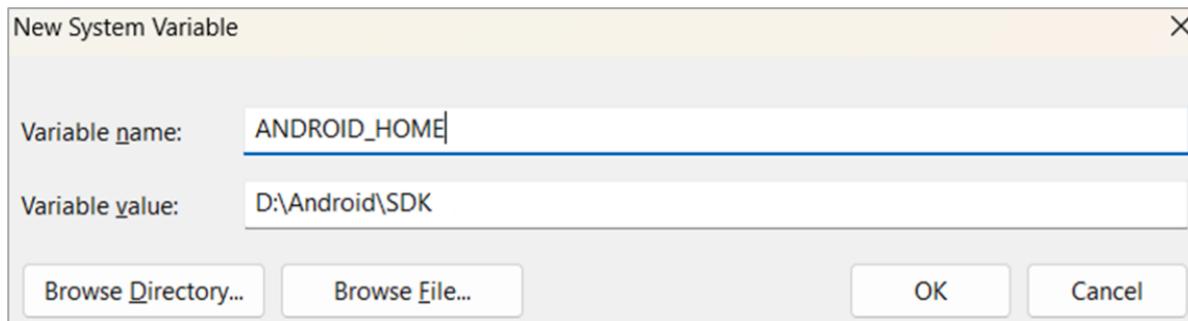
You have successfully installed Android Studio, as well as Android SDK. Close it for now.

## 4. Set Environment Variables ANDROID\_HOME and Path

### 4.1. 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.



### 4.2. 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.

System variables	
Variable	Value
OLLAMA_HOST	127.0.0.1:11434
OLLAMA_MODELS	D:\OLLAMA
OLLAMA_ORIGINS	*
OS	Windows_NT
Path	C:\Python312\Scripts\;C:\Python312\;C:\WINDOWS\system32;C:\W...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.PY;.PYW
POWERSHELL DISTRIBUTIO...	MSI:Windows 10 Pro

C:\Program Files\ChromeDriver
D:\Projects\QA_Backend\js-array-test-coverage\node_modules\.bin
C:\Program Files\nodejs\
D:\Program Files\k6\
C:\Program Files\PowerShell\7\
C:\Program Files\TortoiseSVN\bin
%JAVA_HOME%\bin
%ANDROID_HOME%\build-tools
%ANDROID_HOME%\platform-tools
%ANDROID_HOME%\emulator

## 5. Setup Java JDK

### 1. Check Java JDK is present

```
java --version AND javac --version
```

```
C:\Users\mddim>java --version
java 21.0.5 2024-10-15 LTS
Java(TM) SE Runtime Environment (build 21.0.5+9-LTS-239)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.5+9-LTS-239, mixed mode, sharing)

C:\Users\mddim>javac --version
javac 21.0.5
```

### 2. If not, download & install Java JDK

<https://www.oracle.com/java/technologies/downloads/#java21>

**Java SE Development Kit 21.0.6 downloads**

JDK 21 binaries are free to use in production and free to redistribute, at no cost, under the [Oracle No-Fee Terms and Conditions](#) (NFTC).

JDK 21 will receive updates under the NFTC, until September 2026, a year after the release of the next LTS. Subsequent JDK 21 updates will be licensed under the [Java SE OTN License](#) (OTN) and production use beyond the [limited free grants](#) of the OTN license will [require a fee](#).

[Linux](#) [macOS](#) [Windows](#)

Product/file description	File size	Download
x64 Compressed Archive	185.92 MB	<a href="https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.zip">https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.zip (sha256)</a>
x64 Installer	164.31 MB	<a href="https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.exe">https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.exe (sha256)</a>
x64 MSI Installer	163.06 MB	<a href="https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.msi">https://download.oracle.com/java/21/latest/jdk-21_windows-x64_bin.msi (sha256)</a>

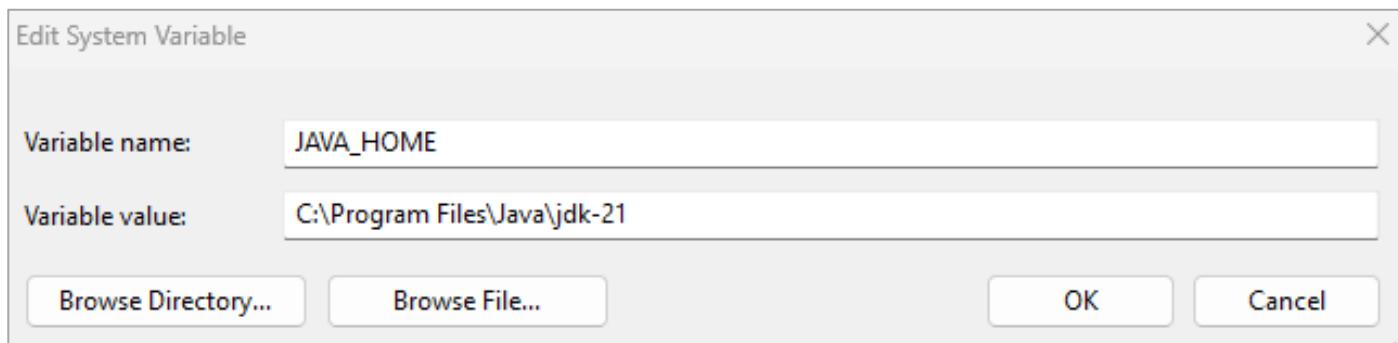
### 3. Using Installer (.exe) is the easiest way.

## 6. 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:

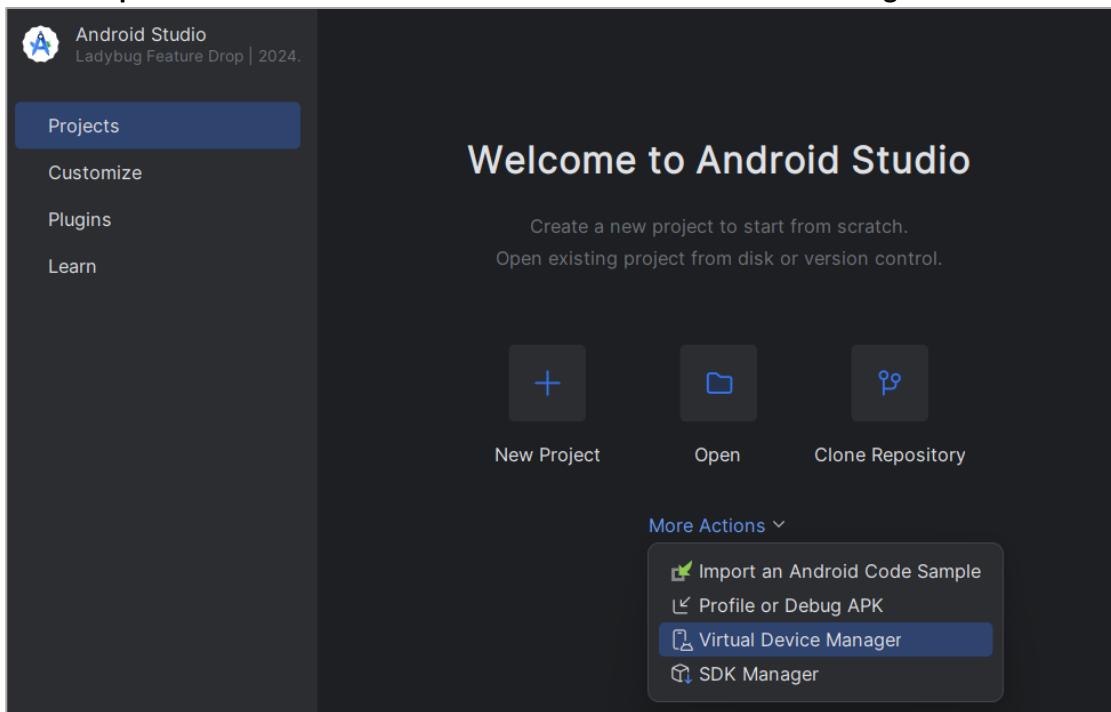
- **%JAVA\_HOME%\bin**

System variables		
Variable	Value	
OLLAMA_MODELS	D:\OLLAMA	C:\Program Files\Git\cmd
OLLAMA_ORIGINS	*	C:\Program Files\ChromeDriver
OS	Windows_NT	C:\Program Files\nodejs\
Path	C:\Python312\Scripts\C:\Python312;C:\WINDOWS\system32	D:\Program Files\k6\
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.P	C:\Program Files\PowerShell\7\
		C:\Program Files\TortoiseSVN\bin
		<b>%JAVA_HOME%\bin</b>

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

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

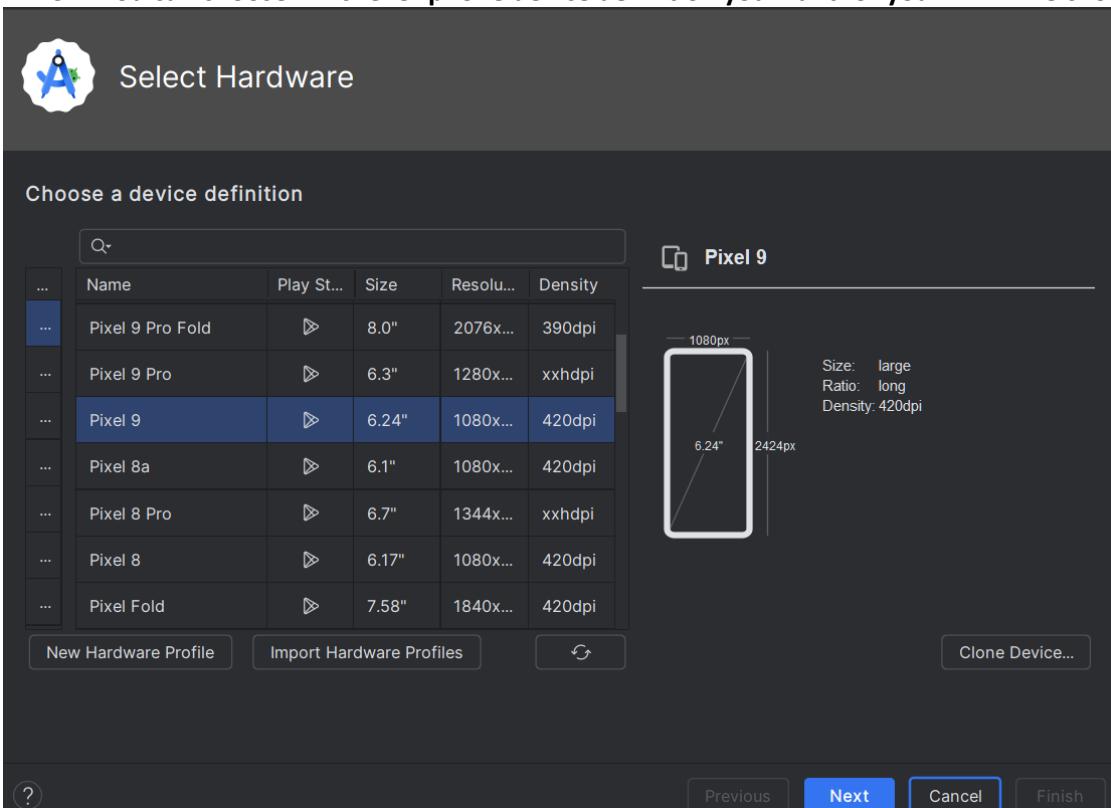
### 1. Open Android Studio → More Actions → Virtual Device Manager



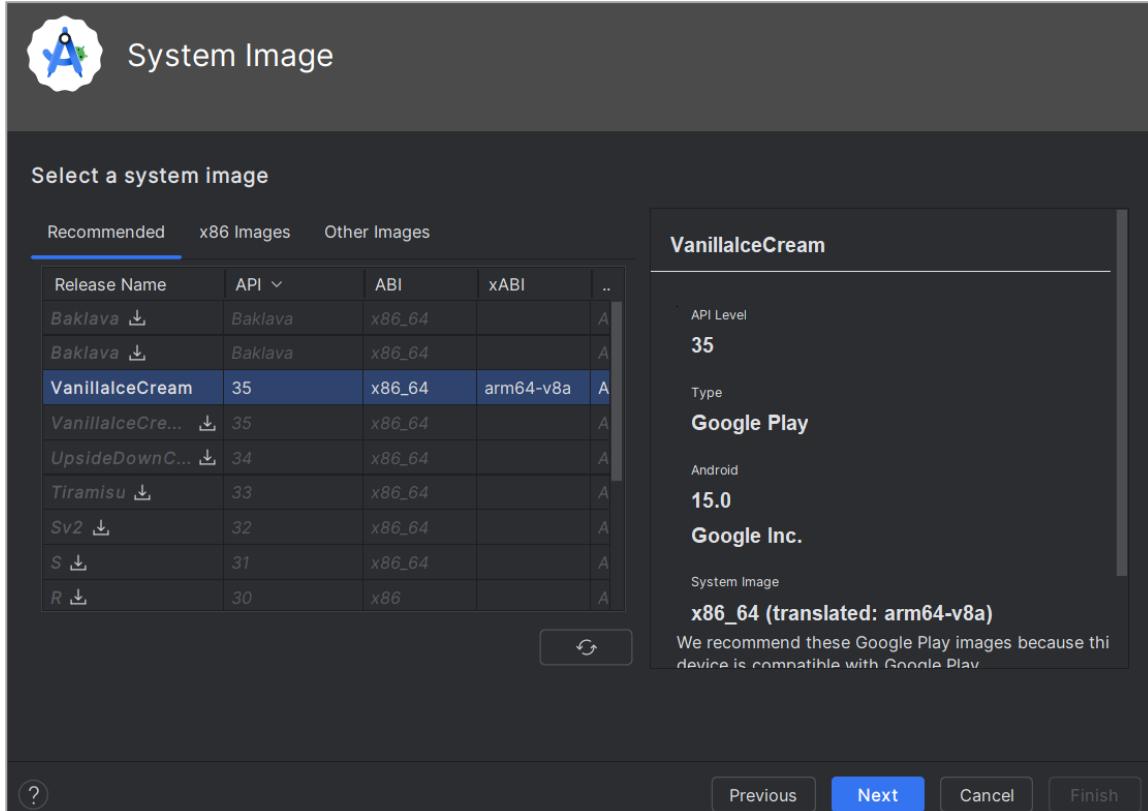
### 2. Now we will create an Android virtual device, on which tests will be executed.



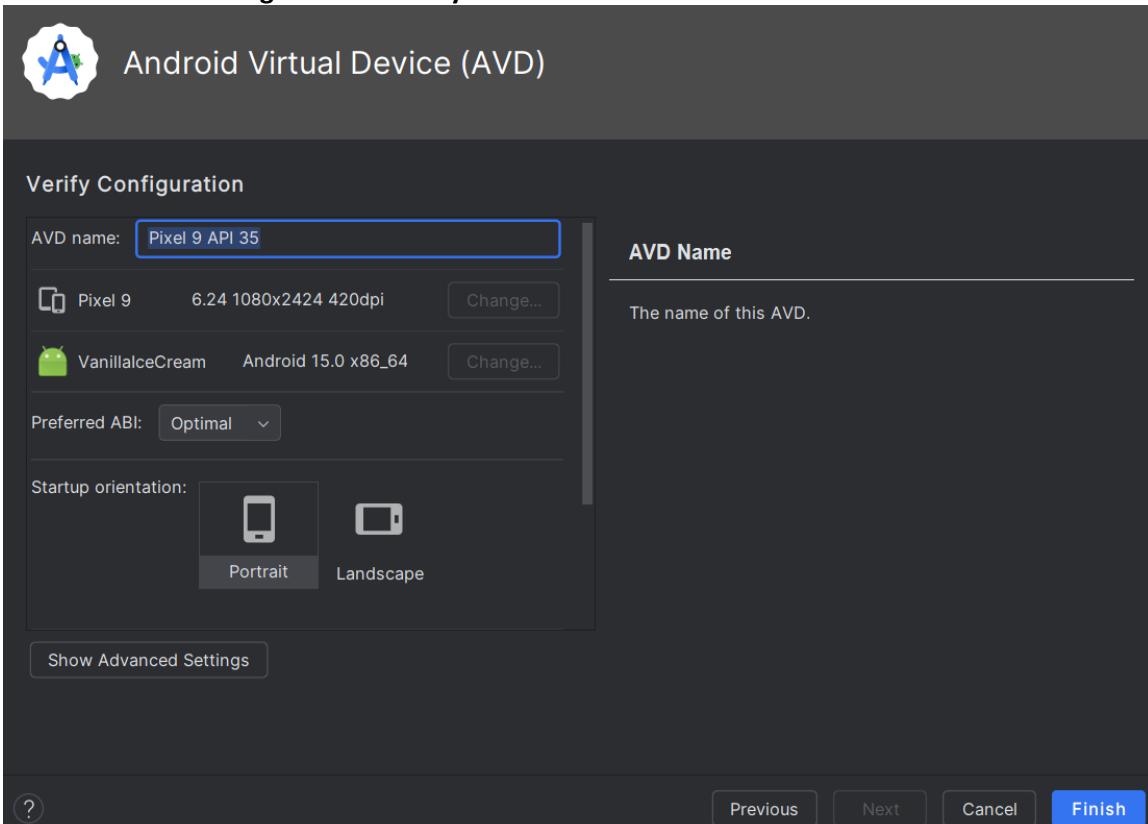
### 3. You can choose whichever phone device definition you want for your AVD. We chose "Pixel 9".



- 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.
- Press [Finish] at the end. Your installation is now complete.
- Select the newly downloaded system image. Next.



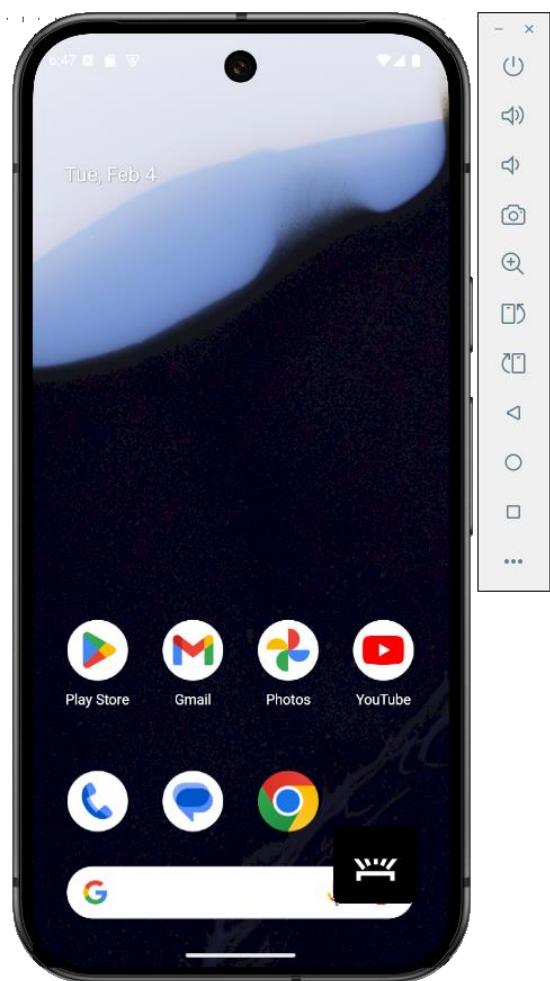
- You can change the name of your device.



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

Name	API	Type	
Pixel9 Android 15.0 ("VanillaIceCream")   x86_64	35	Virtual	

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



## 8. 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 simpler 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
```

```
C:\Users\mddim>emulator -list-avds
Pixel9
```

- 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
```

## 9. 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/>

### 9.2. Install Appium Inspector

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

 <a href="#">Appium-Inspector-2024.12.1-win-x64.exe</a>	110 MB	Dec 8, 2024
 <a href="#">Appium-Inspector-2024.12.1-win-x64.exe.blockmap</a>	119 KB	Dec 8, 2024
 <a href="#">Appium-Inspector-2024.12.1-win-x64.zip</a>	148 MB	Dec 8, 2024
 <a href="#">Appium-Inspector-2024.12.1-win.exe</a>	225 MB	Dec 8, 2024

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

