TP02: Run our first app on different devices

1. Before we begin

In this **`Practice`**, we'll take the **`firstapp`** we built in **TP01** and, firstly, set up an Android Virtual Device (AVD), and see our code in action on an **Android Emulator**. Secondly, we'll configure a physical device to execute our app.

2. What we'll learn

- ⊡ How to create an AVD (Android Virtual Device) and run an app on the Android Emulator.
- ⊡ How to configure a physical/real device of which we can install and execute our app.

3. Run the app on the Emulator[i]

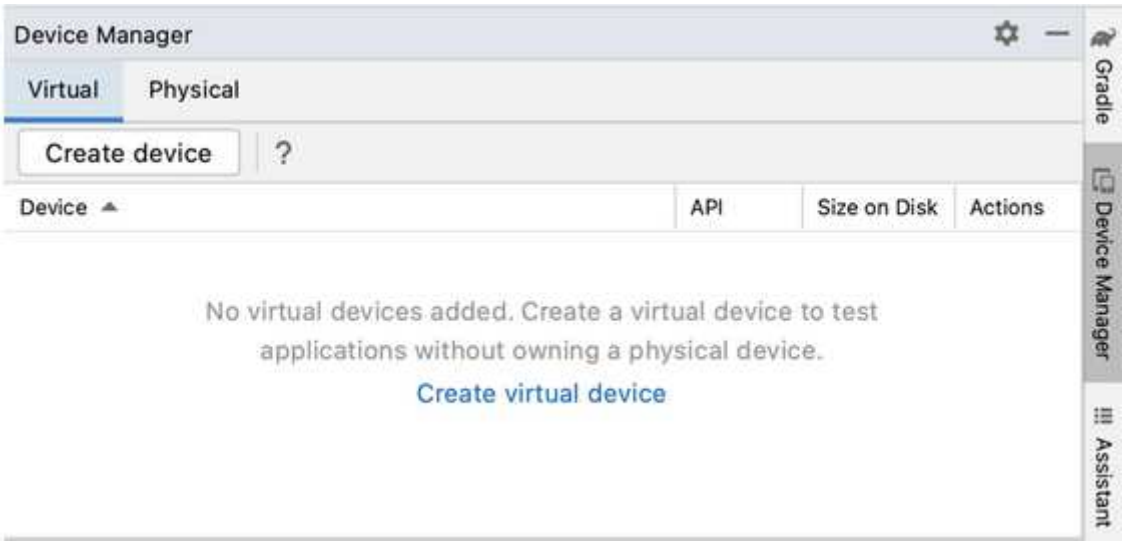
In this task, we'll use the Device Manager to create an Android Virtual Device (AVD). An AVD is a software version, also called an emulator, of a mobile device that runs on your computer and mimics the configuration of a particular type of Android device. This could be any phone, tablet, TV, watch, or Android auto device.

3.1 Create an AVD

To run an Android app in an emulator on your computer, you first create a virtual device.

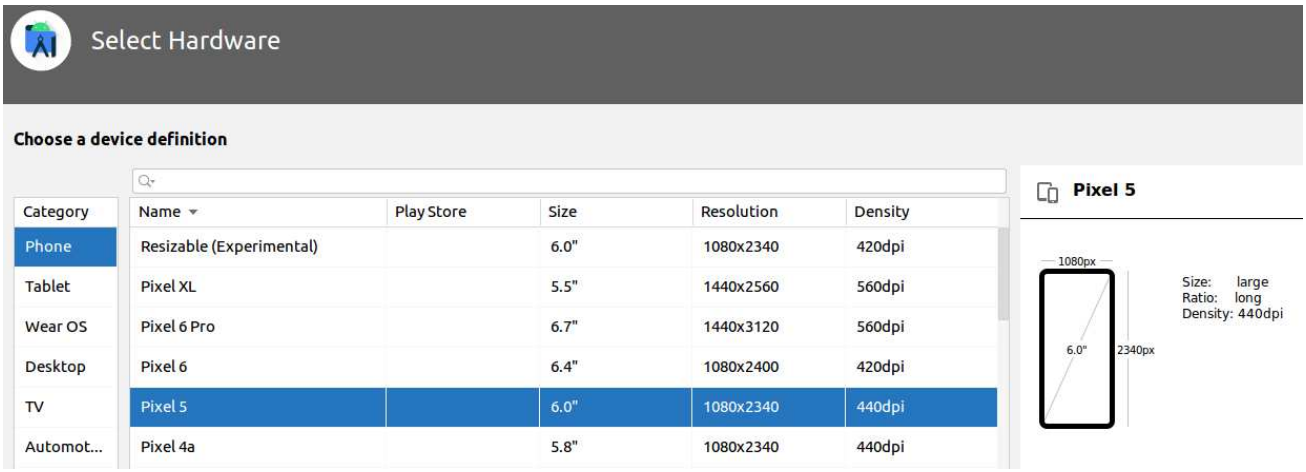
1. In Android Studio, select Tools > Device Manager.

The **Device Manager** dialog opens. If you created a virtual device previously, it's listed in this dialog.



2. Click Create Virtual Device.

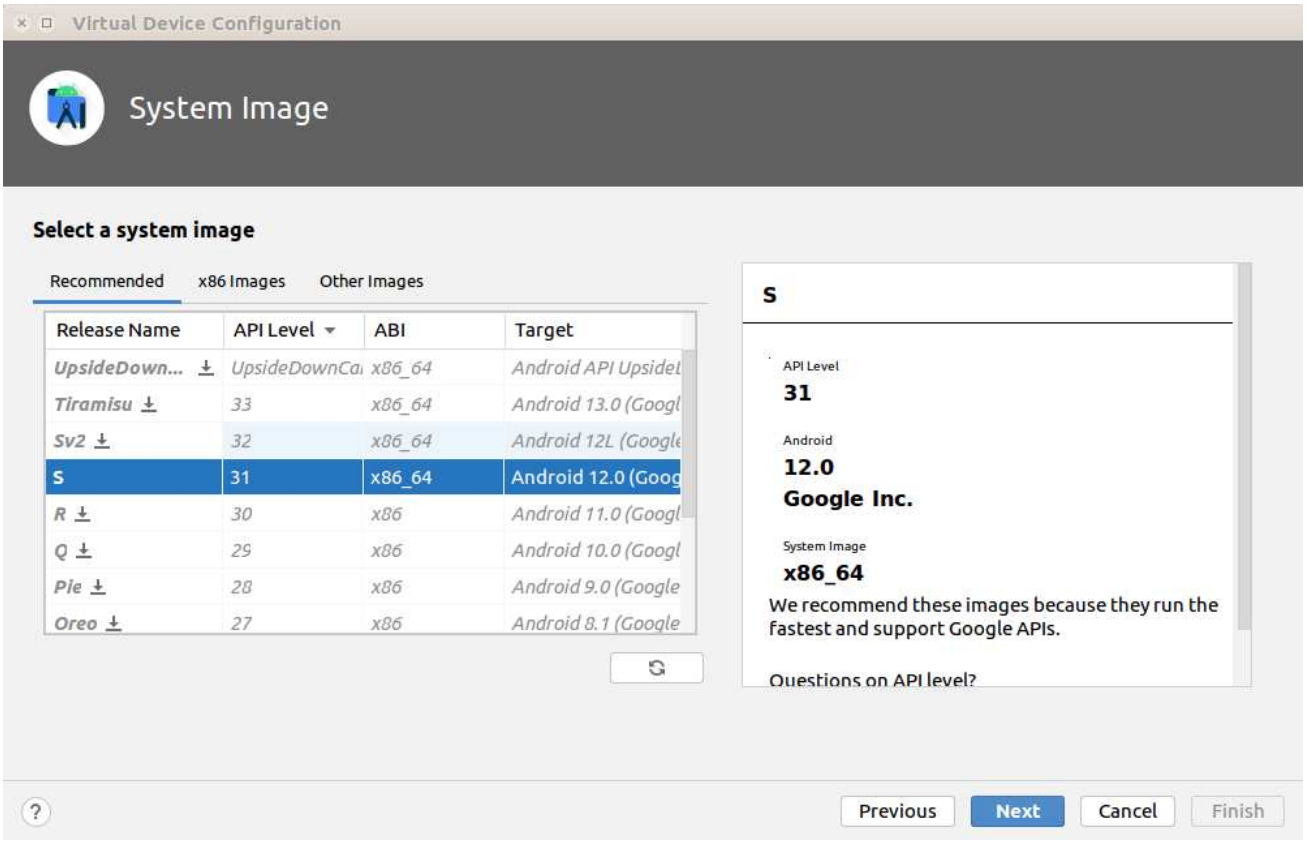
The Virtual Device Configuration dialog appears.



The dialog shows a list of pre-configured devices, organized by category, from which you can choose. For each device, the table provides a column for its display size (in inches), screen resolution (in pixels), and pixel density (number of pixels per inch).

3. Select Phone as the category.
4. Select a phone, such as the Pixel 5, and then click Next.

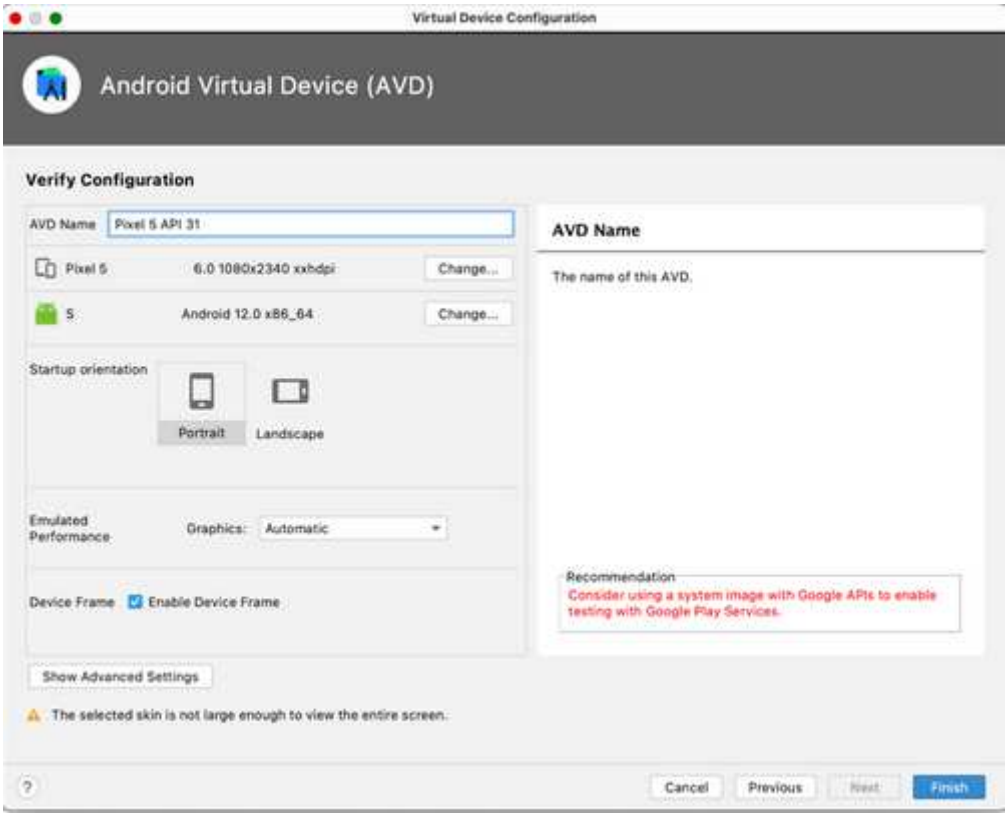
This step opens another screen where you can choose the version of Android to run on your virtual device. This lets you test your app on different versions of Android.



5. If there's a download link next to S, click > Accept > Next > Finish. The presence of the download link indicates that the image[1] isn't installed on your computer, in which case you must install the image before you can configure the virtual device. Expect the download to take some time to complete.

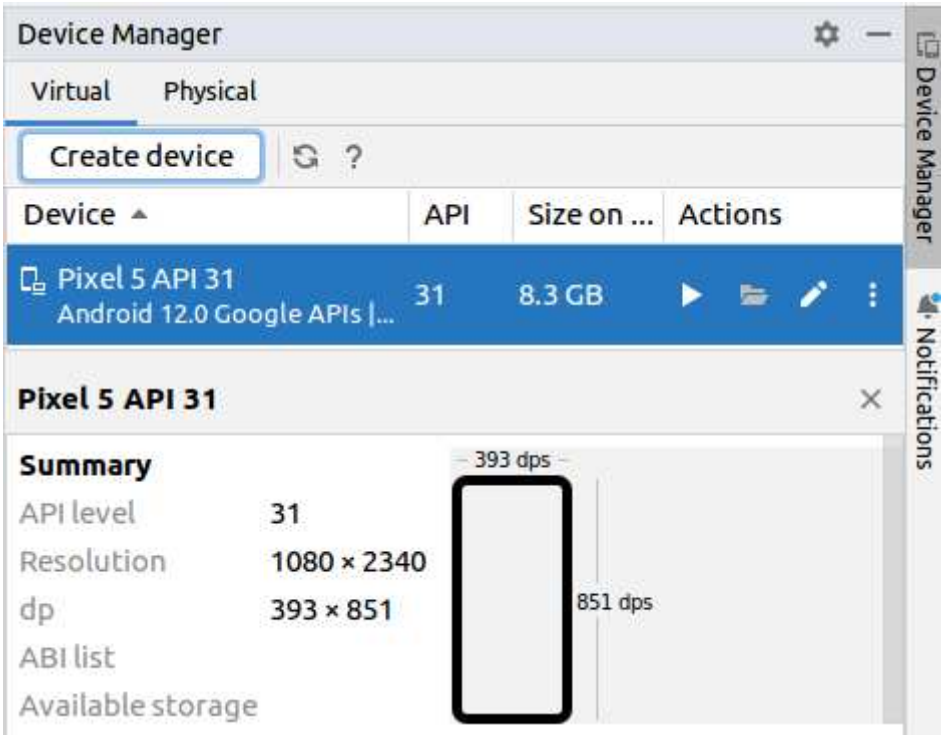
This action opens another screen, where you can choose additional configuration details for your device

[1] These Android system images use a lot of disk space, so only a few are part of your original installation. Many more versions of the Android system are available than are shown in the **Recommended** tab. To see them, look under? the **x86 Images** and **Other Images** tabs in the **Virtual Device Configuration** dialog.



7. In the AVD Name field, enter a name for your AVD or use the default. Leave the rest of the fields unchanged.
8. Click Finish.

This action returns to the Android Virtual Device Manager pane.



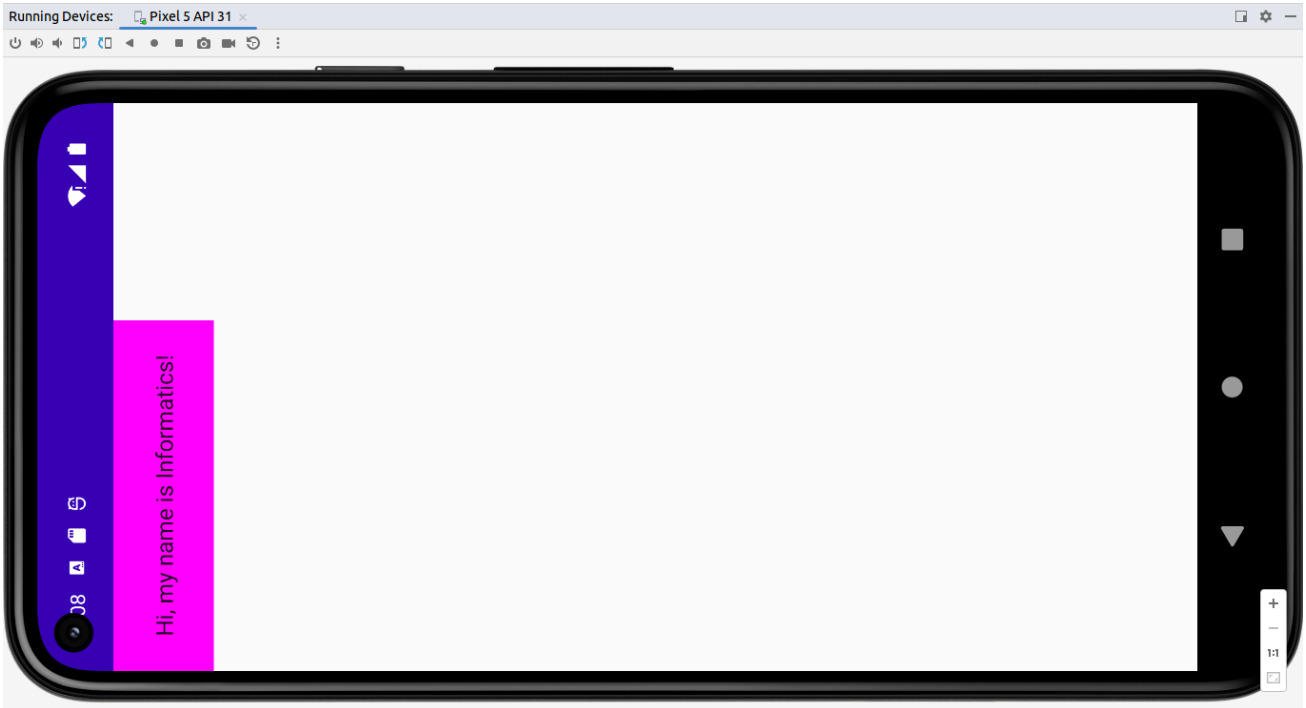
3.2 Run our app on the Android Emulator

1. Select the virtual device that you created from the dropdown menu at the top of the Android Studio

2. Click 

The virtual device starts just like a physical device. Expect this to take a while—potentially several minutes—for the emulator to start for the first time. The virtual device should open beside the code editor.

When our app is ready (i.e., Click Build & Refresh), it opens on the virtual device.



The virtual device is now up and running. The app launches, and we can see the background color and greeting text on the screen.

4. Run the app on a physical device[ii]:

Always we should test our Android apps on a real device before releasing it to users. In this section we'll learn how to configure our development environment and Android device for testing and debugging over an Android Debug Bridge (ADB) connection.

4.1 Set up a device for development

Before you can start debugging on your device, decide whether you want to connect to the device using a USB cable or Wi-Fi[2]. Then do the following:

Each **Ubuntu user** that wants to use ADB needs to be in the **plugdev** group. If you see an error message that says you're not in the **plugdev** group, add yourself to it using the following command:

```
sudo usermod -aG plugdev $LOGNAME
```

Groups only update on login, so you must log out for this change to take effect. When you log back in, you can use **id** to check that you're in the **plugdev** group.

- The system needs to have **udev** rules installed that cover the device. The **android-sdk-platform-tools-common** package contains a community-maintained default set of **udev** rules for Android devices. To install it, use the following command:

```
apt-get install android-sdk-platform-tools-common
```

4.2 Connect to your device using USB

When you're set up and plugged in over USB, click **Run**  in Android Studio to build and run your app on the device.

[i] https://developer.android.com/codelabs/basic-android-kotlin-compose-emulator
[ii] https://developer.android.com/studio/run/device

[2] Android 11 and higher supports deploying and debugging your app wirelessly from your workstation via Android Debug Bridge (ADB). For example, you can deploy your debuggable app to multiple remote devices without physically connecting your device via USB and contending with common USB connection issues, such as driver installation. To use wireless debugging refer to endnote (ii).