



I n t e r a E s c h

How to launch the Project

INTERA-ESCH

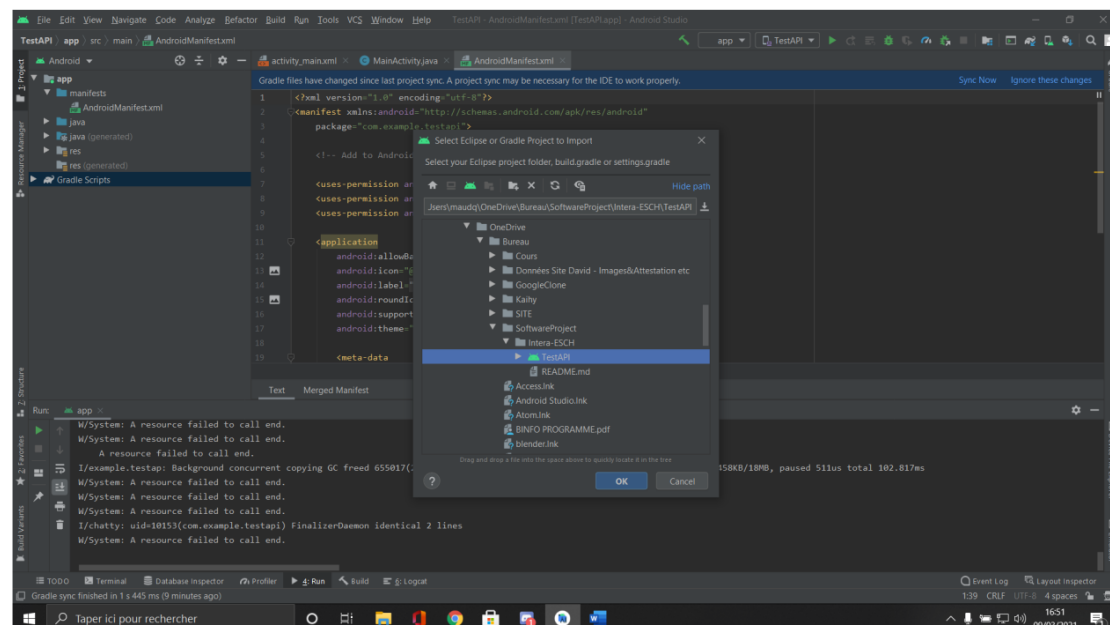
University Of Luxembourg – Group 2 | Software Engineering Project | Years 2020/2021

Before all

Before anything chosen, you must have Android Studio on your machine. It could not be easier! Follow this link and download the software by following the Android installation guide -> <https://developer.android.com/studio>

First Step – Import the Project

After cloning the project in one of your folders, go to Android studio and click on File -> New -> Import Project.



Then find the project you just took and click on it.
Confirm with OK.

Second Step – Configure Google API

Our app uses Google libraries. Therefore, the app uses a Google key which gives it access to libraries. It may be that when you download the project, everything is already ready for you! But you may miss it if we decide to remove it (the key may become chargeable), if so, refer to our tutorial or to the many other tutorials available on the Internet to have an API Key :

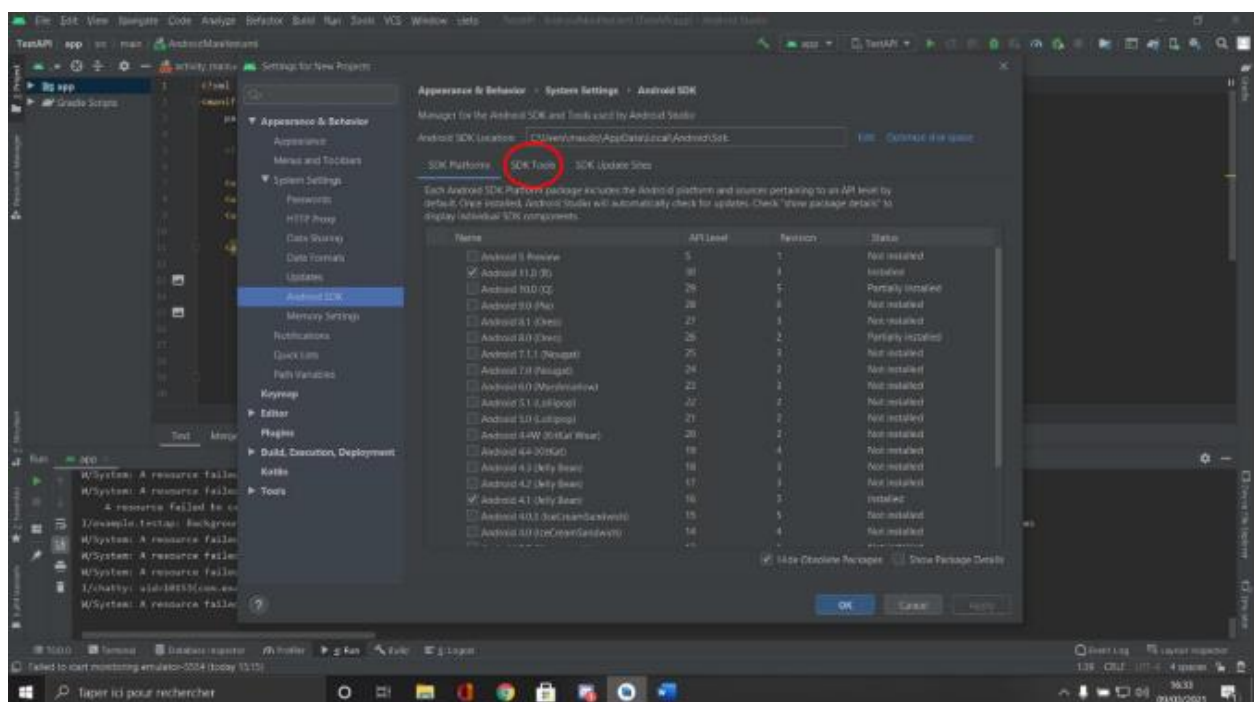
“How to had a GOOGLE API KEY.pdf”

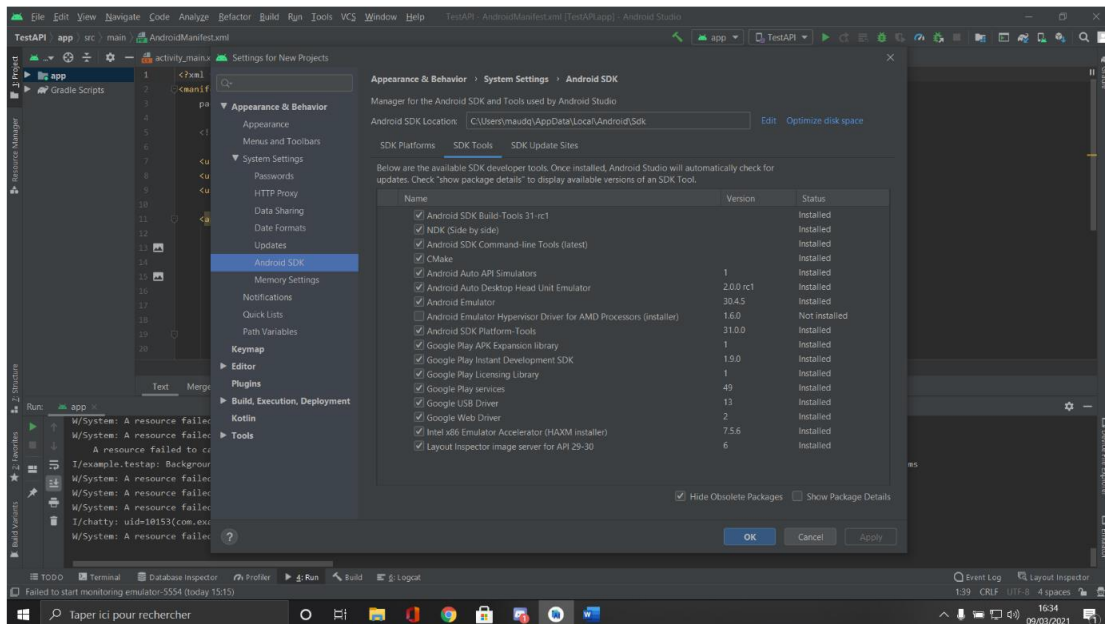
Third Step – Configure Android Studio

If it isn't, open Android Studio.

Click on **Tools**, then in **SDK Manager**.

You should have this: (click on **SDK Tools**)

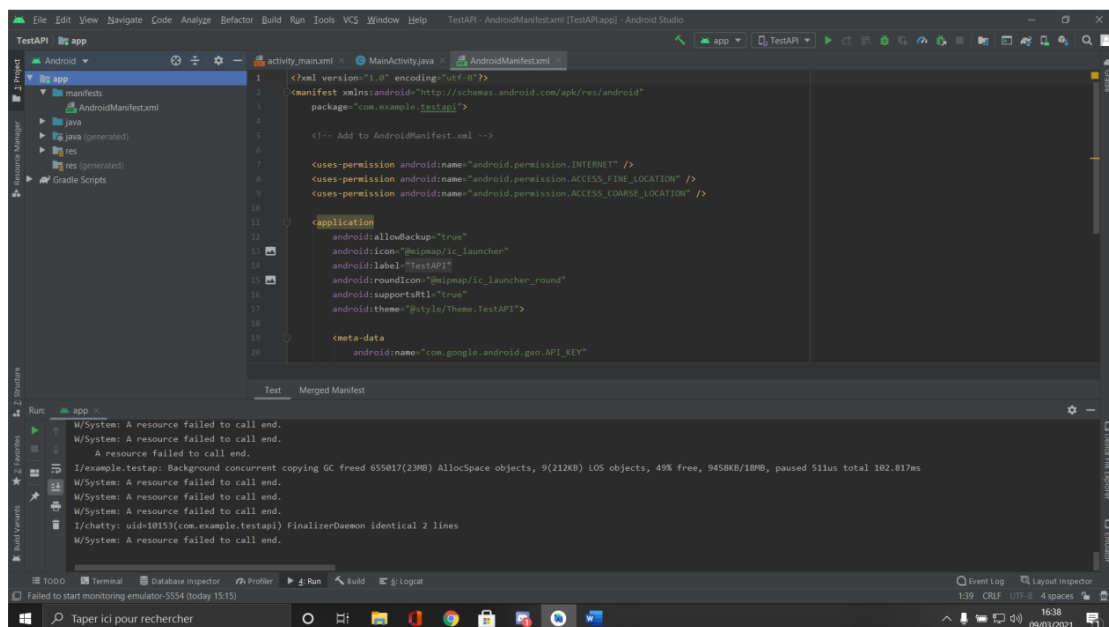




Here, check all that I checked. In truth, all you need is Google services. But anyway, the other libraries will surely serve you one day. The download can take several tens of minutes.

Once the download is complete, click on "Ok".

Search your tree for the AndroidManifest.XML file



The top lines allow us to give access to the Internet and the Location of our device to Android studio.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

Finally, below you should see these lines:

```
<meta-data android:name="com.google.android.geo.API_KEY"
    android:value="YOUR_API_KEY" />
```

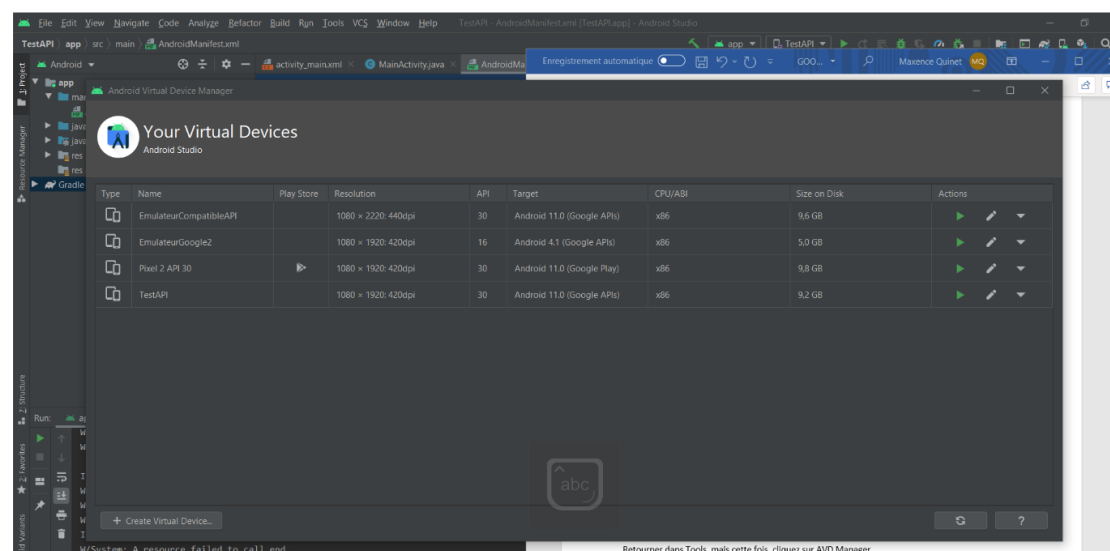
If there is no API key already entered in the correct row, be sure to create one, and add it yourself.

Fourth Step – Emulator Configuration

The choice of emulator for our project is important, as some do not have a version compatible with our Gradle, while others are not compatible with Google APIs. To be sure that the project works 100% bug-free, use the same device as us.

Click on Tools, and this time click on AVD Manager.

You arrive on this page:



The screenshot shows the 'Virtual Device Configuration' window in Android Studio. The 'Phone' category is selected in the left sidebar. The main list shows 'Pixel 2' as the selected device. The specifications for 'Pixel 2' are displayed on the right: 5.0" screen, 1080x1920 resolution, 420dpi density, and a large ratio. The 'Next' button is highlighted in blue.

Category	Name	Play Store	Size	Resolution	Density
TV	Pixel 3		5.46"	1080x2160	440dpi
TestAPI	TestAPI				
TestAPIv2	TestAPIv2				
Phone	Pixel 2 XL		5.99"	1440x2880	560dpi
Wear OS	Pixel 2		5.0"	1080x1920	420dpi
Tablet	Pixel		5.0"	1080x1920	420dpi
Automotive	Nexus S		4.0"	480x800	hdpi
	Nexus One		3.7"	480x800	hdpi
	Nexus 6P		5.7"	1440x2560	560dpi

Pixel 2 specifications:
 Size: large
 Ratio: long
 Density: 420dpi

The screenshot shows the Android Studio IDE with the 'Select a system image' dialog open. The dialog has a 'Recommended' tab and an 'Other Images' tab. Under 'Recommended', the 'x86 Images' sub-tab is selected, displaying a table of system images. The 'Android 11.0 (Google APIs)' image is highlighted. The 'System Image' section on the right shows the selected image details: API Level 30, Android 11.0, Google Inc., and System Image x86. The background shows the Android Studio IDE with the 'TestAPI' project open.

Release Name	API Level	ABI	Target
S Download	5	x86_64	Android API 5 (Google Play)
S Download	5	x86_64	Android API 5 (Google APIs)
R Download	30	x86_64	Android 11.0 (Google Play)
R Download	30	x86	Android 11.0 (Google APIs)
R Download	30	x86_64	Android 11.0 (Google APIs)
Q Download	29	x86_64	Android 10.0 (Google Play)
Q Download	29	x86	Android 10.0 (Google APIs)
Q Download	29	x86_64	Android 10.0 (Google APIs)
P Download	28	x86_64	Android 10.0

System Image
API Level
30
Android
11.0
Google Inc.
System Image
x86

Questions on API level?
See the API level distribution chart

Previous Next Cancel Refresh

PAGE 5

Finally, give your emulator a name if you wish, and click Finish.

FINISH! Your emulator will be created in a few minutes and will then be ready to use.

Last Step – Verification

Before you leave, once everything has finished downloading, please build your gradle.

To do this, do: Build -> Rebuild Project

It is important to Build the project so that all dependencies and changes made during this tutorial are applied!

Be sure to take your emulator before launching your application with the green arrow!

The 1st time, it may take a while because the project will start for the 1st time, so don't worry!

Conclusion

Thank you for following this tutorial to try our application.

Our INTERA-ESCH application is a student project to validate a subject at the University of Luxembourg. Hoping that you like it !

Sincerely, the Intera-ESCH team.

PS : Our mascot say to you "Bye-Bye"

