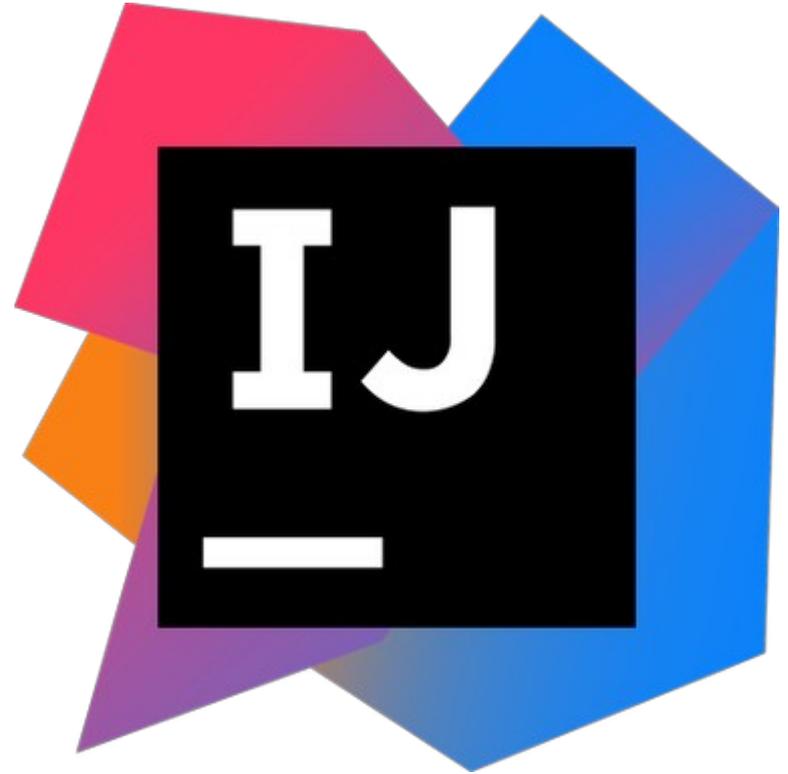


Android Studio, IDE

Création d'un nouveau Projet



JetBrains

Entreprise



JetBrains est une entreprise informatique éditant des logiciels pour développeurs de logiciels. JetBrains a des bureaux à Prague, Saint-Petersbourg, Boston, Moscou et Munich. [Wikipédia](#)

Siège social : Prague 4, Prague, Tchéquie

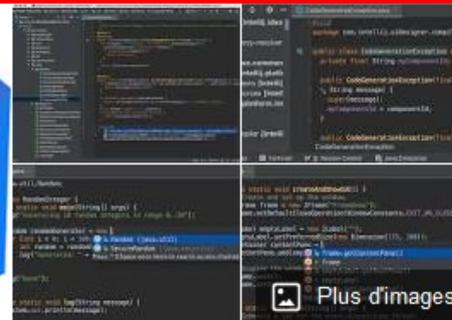
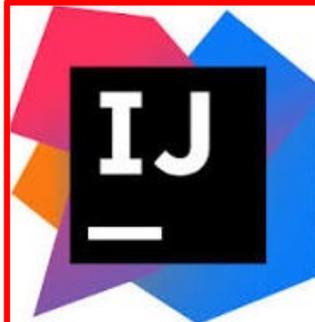
Nombre d'employés : 1 200

Création : 14 août 2000, Tchéquie

PDG : Oleg Stepanov (2012–), Maxim Shafirov (2012–)

Filiale : Code Complete Software, Inc.

Type d'activité : Société à capitaux privés



IntelliJ IDEA

Logiciel



IntelliJ IDEA également appelé « IntelliJ », « IDEA » ou « IDJ » est un environnement de développement intégré de technologie Java destiné au développement de logiciels informatiques. [Wikipédia](#)

Date de sortie initiale : janvier 2001

Dernière version : 2019.3.1 (18 décembre 2019)

Développé par : JetBrains

Système d'exploitation : Microsoft Windows, Linux et macOS

Version avancée : 2017.1 EAP (8 février 2017); 2018.1 EAP (7 mars 2018); 2018.2 EAP (6 juin 2018)

Programmé en : Java, Kotlin



The integrated team environment

Explore Space

Built with Kotlin



Family of capable and ergonomic development environments

Find your IDE

Free for education and open source projects



Essential productivity tool set for .NET developers

Learn more

Free for students and teachers



Multiplatform programming language

Start developing with Kotlin



Choose your project

Phone and Tablet

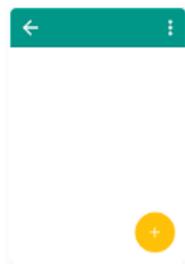
Wear OS

TV

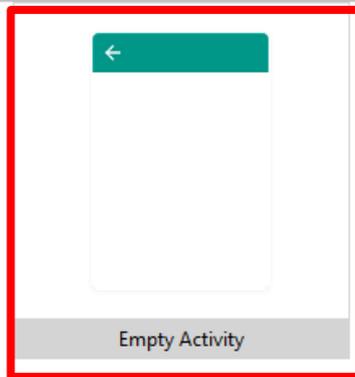
Android Auto

Android Things

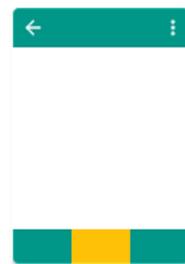
Add No Activity



Basic Activity



Empty Activity



Bottom Navigation Activity



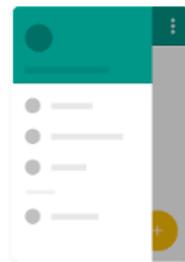
Fragment + ViewModel



Fullscreen Activity



Master/Detail Flow



Navigation Drawer Activity

Empty Activity

Creates a new empty activity

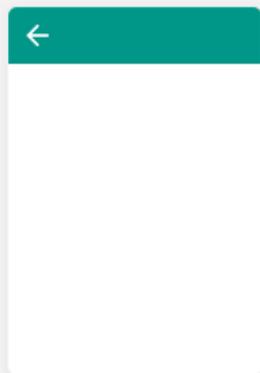
Previous

Next

Cancel

Finish

Configure your project



Empty Activity

Creates a new empty activity

Name

My Application

Package name

com.example.myapplication

Save location

C:\Users\xi06\AndroidStudioProjects\MyApplication3

Language

(Select a language)

Minimum API level

API 19: Android 4.4 (KitKat)

i Your app will run on approximately **95,3%** of devices.

[Help me choose](#)

This project will support instant apps

Use androidx.* artifacts

i Please select a language

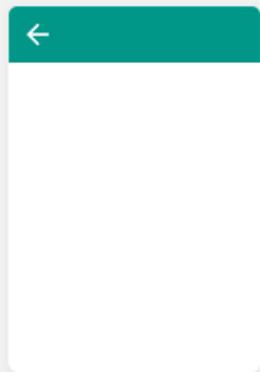
Previous

Next

Cancel

Finish

Configure your project



Empty Activity

Creates a new empty activity

Name

My Application

Package name

com.example.myapplication

Save location

C:\Users\xi06\AndroidStudioProjects\MyApplication3

Language

(Select a language)

Minimum API level API 19: Android 4.4 (KitKat)

- Your app will run on the minimum API level.
[Help me choose](#)
 - This project will use the minimum API level.
 - Use androidx.* as the implementation of the AndroidX classes.
- API 22: Android 5.1 (Lollipop)
 - API 23: Android 6.0 (Marshmallow)
 - API 24: Android 7.0 (Nougat)
 - API 25: Android 7.1.1 (Nougat)
 - API 26: Android 8.0 (Oreo)
 - API 27: Android 8.1 (Oreo)
 - API 28: Android 9.0 (Pie)
 - API 29: Android 10.0 (Q)

! Please select a language

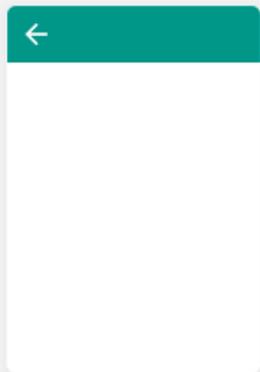
Previous

Next

Cancel

Finish

Configure your project



Empty Activity

Creates a new empty activity

Name

My Application

Package name

com.example.myapplication

Save location

C:\Users\xi06\AndroidStudioProjects\MyApplication3

Language

(Select a language)

Minimum API level API 29: Android 10.0 (Q)

Your app will run on < 1% of devices.

[Help me choose](#)

This project will support instant apps

Use androidx.* artifacts

 Please select a language



Previous

Next

Cancel

Finish

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.0 Ice Cream Sandwich	15	
4.1 Jelly Bean	16	99,6%
4.2 Jelly Bean	17	98,1%
4.3 Jelly Bean	18	95,9%
4.4 KitKat	19	95,3%
5.0 Lollipop	21	85,0%
5.1 Lollipop	22	80,2%
6.0 Marshmallow	23	62,6%
7.0 Nougat	24	37,1%
7.1 Nougat	25	14,2%
8.0 Oreo	26	6,0%
8.1 Oreo	27	1,1%

The minimum SDK version determines the lowest level of Android that your app will run on.

You typically want to target as many users as possible, so you would ideally want to support everyone -- with a minimum SDK version of 1. However, that has some disadvantages, such as lack of features, and very few people use devices that old anymore.

Your choice of minimum SDK level should be a tradeoff between the distribution of users you wish to target and the features that your application will need.

Click each Android Version/API level for more information.

OK

Cancel

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.0 Ice Cream Sandwich	15	
4.1 Jelly Bean	16	99,6%
4.2 Jelly Bean	17	98,1%
4.3 Jelly Bean	18	95,9%
4.4 KitKat	19	95,3%
5.0 Lollipop	21	85,0%
5.1 Lollipop	22	80,2%
6.0 Marshmallow	23	62,6%
7.0 Nougat	24	37,1%
7.1 Nougat	25	14,2%
8.0 Oreo	26	6,0%
8.1 Oreo	27	1,1%

KitKat

Printing Framework

Print generic content
Print images
OEM print services

SMS Provider

Read and write SMS and MMS messages
Select default SMS app

Wireless and Connectivity

Host NFC card emulation
NFC reader mode
Infrared support

Multimedia

Adaptive video playback
On-demand audio timestamps
Surface image reader
Peak and RMS audio measurements
Loudness enhancer
Remote controllers
Closed captions

Animation and Graphics

Scenes and transitions
Animator pausing
Reusable bitmaps

User Content

Storage access framework
External storage access
Sync adapters

User Input

New sensor types, including step detector
Batched sensor events
Controller identities

User Interface

Immersive full-screen mode
Translucent system bars
Enhanced notification listener
Live regions for accessibility

<https://developer.android.com/about/versions/android-4.4.html>

OK

Cancel

Android 4.4 APIs

developer.android.com/about/versions/android-4.4.html

API Level: 19

Android 4.4 ([KITKAT](#)) is a new release for the Android platform that offers new features for users and app developers. This document provides an introduction to the most notable new APIs.

As an app developer, you should download the Android 4.4 system image and SDK platform from the [SDK Manager](#) as soon as possible. If you don't have a device running Android 4.4 on which to test your app, use the Android 4.4 system image to test your app on the [Android emulator](#). Then build your apps against the Android 4.4 platform to begin using the latest APIs.

Update your target API level

To better optimize your app for devices running Android 4.4, you should set your `targetSdkVersion` to "19", install it on an Android 4.4 system image, test it, then publish an update with this change.

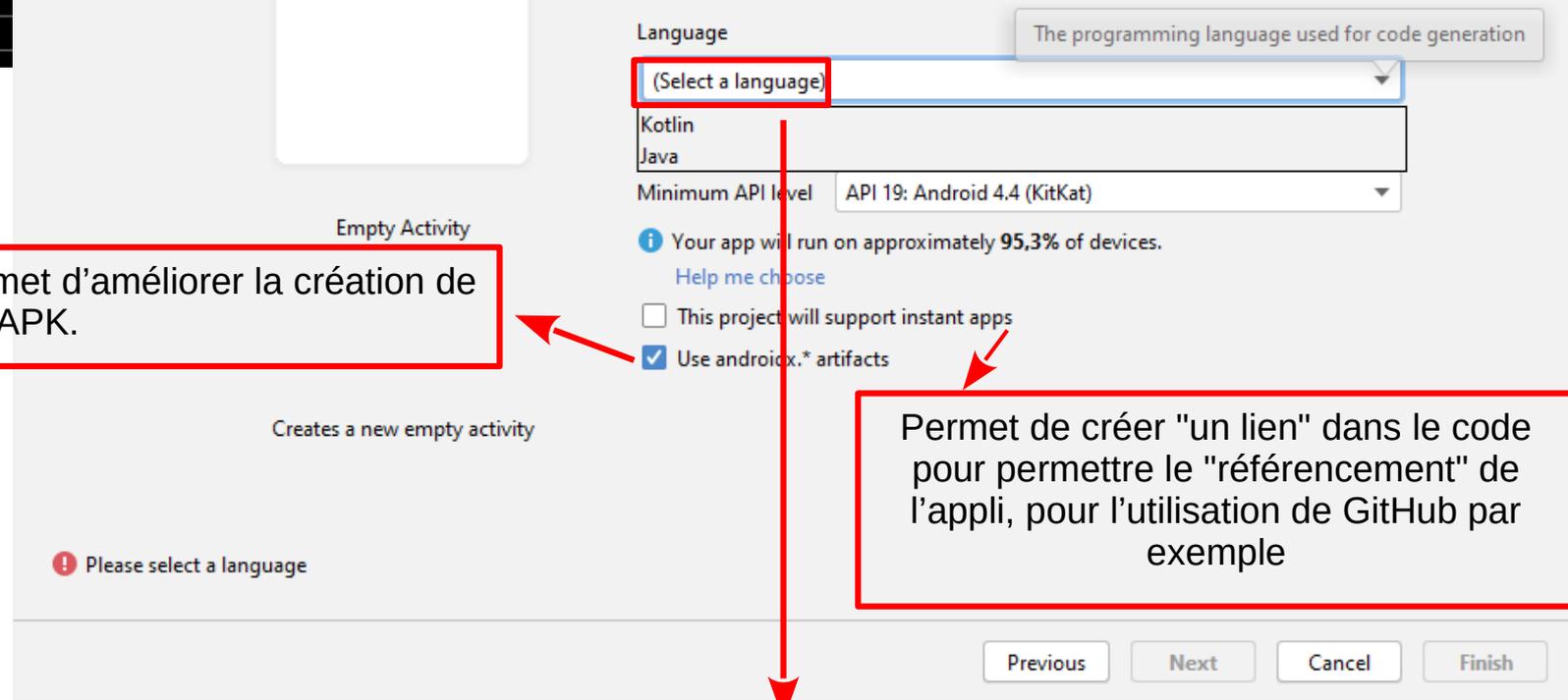
You can use APIs in Android 4.4 while also supporting older versions by adding conditions to your code that check for the system API level before executing APIs not supported by your `minSdkVersion`. To learn more about maintaining backward compatibility, read [Supporting Different Platform Versions](#).

For more information about how API levels work, read [What is API Level?](#)

Important Behavior Changes

If you have previously published an app for Android, be aware that your app might be affected by changes in Android 4.4.

If your app reads from external storage...



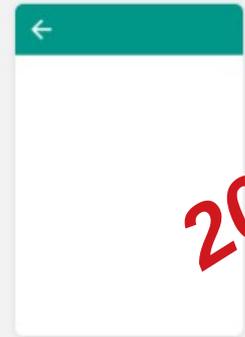
C'est une librairie qui permet d'améliorer la création de l'APK.

Permet de créer "un lien" dans le code pour permettre le "référencement" de l'appli, pour l'utilisation de GitHub par exemple

Kotlin est un langage de programmation orienté objet et fonctionnel, avec un typage statique qui permet de compiler pour la machine virtuelle Java et JavaScript. Son développement provient principalement d'une équipe de programmeurs chez JetBrains basée à Saint-Petersbourg en Russie (son nom vient de l'île de Kotlin, près de St. Pétersbourg).

Google annonce pendant la conférence Google I/O 2017 que Kotlin devient le second langage de programmation supporté officiellement sur Android après Java. En 2020 Kotlin devient le langage plébiscité par Google, il n'est pas encore le langage par défaut, mais se retrouve en haut de liste.

Configure your project



Empty Activity

2019

Name: My Application

Package name: com.example.myapplication

Save location: C:\Users\xi06\AndroidStudioProjects\MyApplication

Language: **Java**

Minimum API level: API 15: Android 4.0.3 (IceCreamSandwich)

i Your app will run on approximately **100%** of devices.

Configure your project



Empty Activity

2020

Name: My Application

Package name: com.example.myapplication

Save location: C:\Users\xi06\AndroidStudioProjects\MyApplication3

Language: **(Select a language)**

Kotlin

Java

Minimum API level: API 19: Android 4.4 (KitKat)

i Your app will run on approximately **95,3%** of devices.
[Help me choose](#)

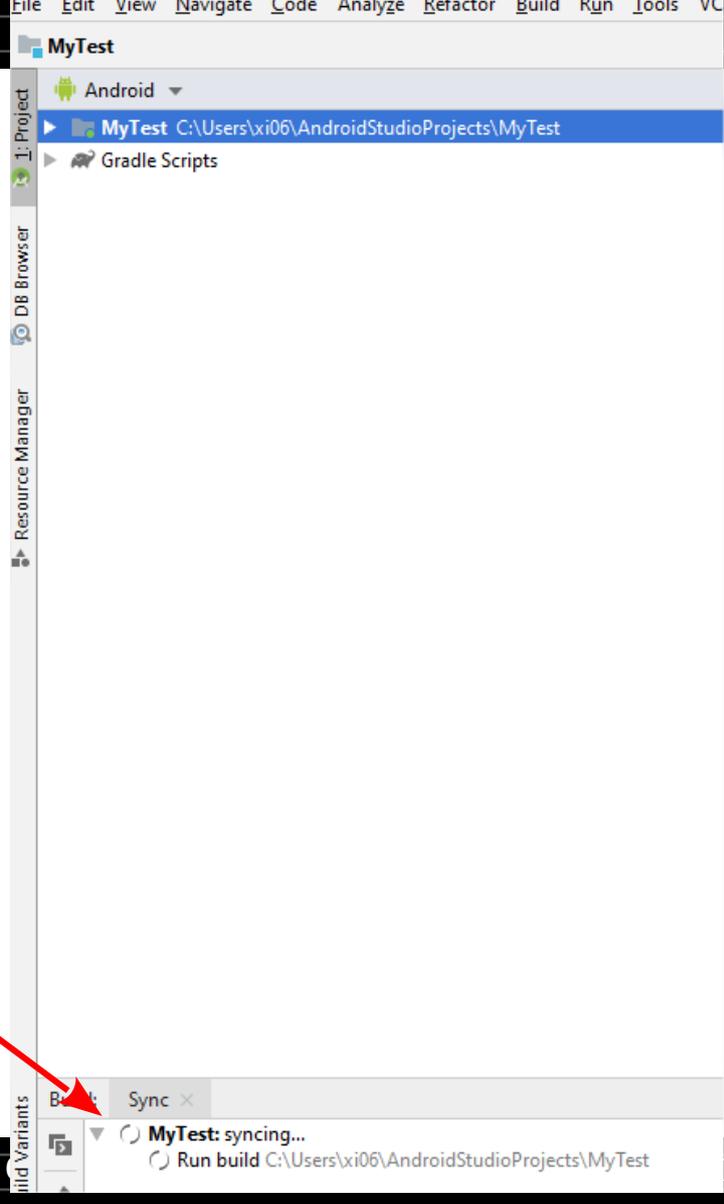
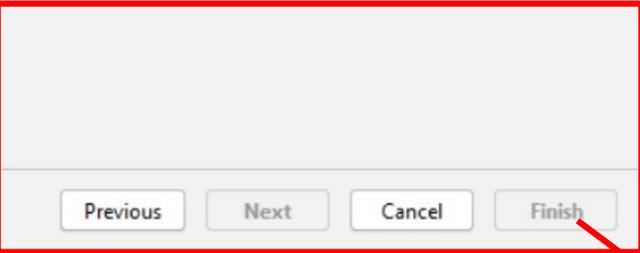
This project will support instant apps

Use androidx.* artifacts

Creates a new empty activity

i Please select a language

Previous Next C



The screenshot displays the Android Studio interface. On the left, the 'Resource Manager' tab is active, showing a project structure with folders for 'com.example.mytest', 'MainActivity', 'com.example.mytest (androidTest)', 'com.example.mytest (test)', 'java (generated)', 'res', and 'Gradle Scripts'. The 'MainActivity' folder is selected. In the center, the Java code for 'MainActivity' is visible, showing it extends 'AppCompatActivity' and overrides the 'onCreate' method to call 'super.onCreate()' and 'setContentView(R.layout.activity_main)'. On the right, a 'Build' window is open, showing a successful sync operation at 16/02/2020 09:35. The build log includes the following steps:

```
Build: Sync x
  ✓ MyTest: synced successfully at 16/02/2020 09:35
    ✓ Run build C:\Users\xi06\AndroidStudioProjects\MyTest
      ✓ Load build
      ✓ Configure build
      ✓ Calculate task graph
      ✓ Run tasks
```

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS DB Navigator Help

MyTest > app > src > main > res > layout > activity_main.xml

app

- manifests
 - AndroidManifest.xml
- java
 - com.example.mytest
 - MainActivity
 - com.example.mytest (androidTest)
 - com.example.mytest (test)
- java (generated)
- res
 - drawable
 - ic_launcher_background.xml
 - ic_launcher_foreground.xml (v24)
 - layout
 - activity_main.xml
- mipmap
 - ic_launcher (6)
 - ic_launcher_round (6)
- values
 - colors.xml
 - strings.xml
 - styles.xml

Gradle Scripts

- build.gradle (Project: MyTest)
- build.gradle (Module: app)
- gradle-wrapper.properties (Gradle Version)
- proguard-rules.pro (ProGuard Rules for app)
- gradle.properties (Project Properties)
- settings.gradle (Project Settings)
- local.properties (SDK Location)

Build: Sync

- MyTest: synced successfully at 16/02/2020 09:35
- Run build C:\Users\xi06\AndroidStudioProjects\MyTest
 - Load build
 - Configure build
 - Calculate task graph
 - Run tasks

Palette

- Common
 - Ab TextView
- Text
 - Button
- Buttons
 - ImageView
- Widgets
 - RecyclerView
- Layouts
 - << fragment >
- Containers
 - ScrollView
- Google
 - Switch
- Legacy

Component Tree

- ConstraintLayout
 - Ab TextView- "Hello World!"

Design Text

Attributes

- Ab <unnamed> TextView
- id
- Declared Attributes
- Layout
 - Constraint Widget
- Constraints

```
<?xml version="1.0" encoding="utf-8" ?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Logcat

NDK Resolution Outcome: Project settings: Gradle model version=5.4.1, NDK version is UNKNOWN (11 minutes ago)

17

Se repérer dans le projet

La figure précédente montre les principaux éléments de l'interface Android Studio.

Tout projet Android doit respecter une hiérarchie bien précise qui permettra au compilateur de retrouver les différents éléments et ressources lors de la génération de l'application.

Cette hiérarchie favorise la modularité des applications Android. À la création du projet, Android Studio crée automatiquement des dossiers pour contenir les fichiers de code Java, les fichiers XML, et les fichiers multimédias.

L'explorateur de projet vous permettra de naviguer dans ces dossiers.

Les dossiers que nous utiliserons le plus sont **java** et **res**. Le premier contient le code Java qui définit le comportement de l'application (situé dans le répertoire de votre projet sous **app src main** et le second comporte des sous-dossiers dans **app src main res** où sont stockés les ressources qui définissent l'interface de l'application. (l'apparence)

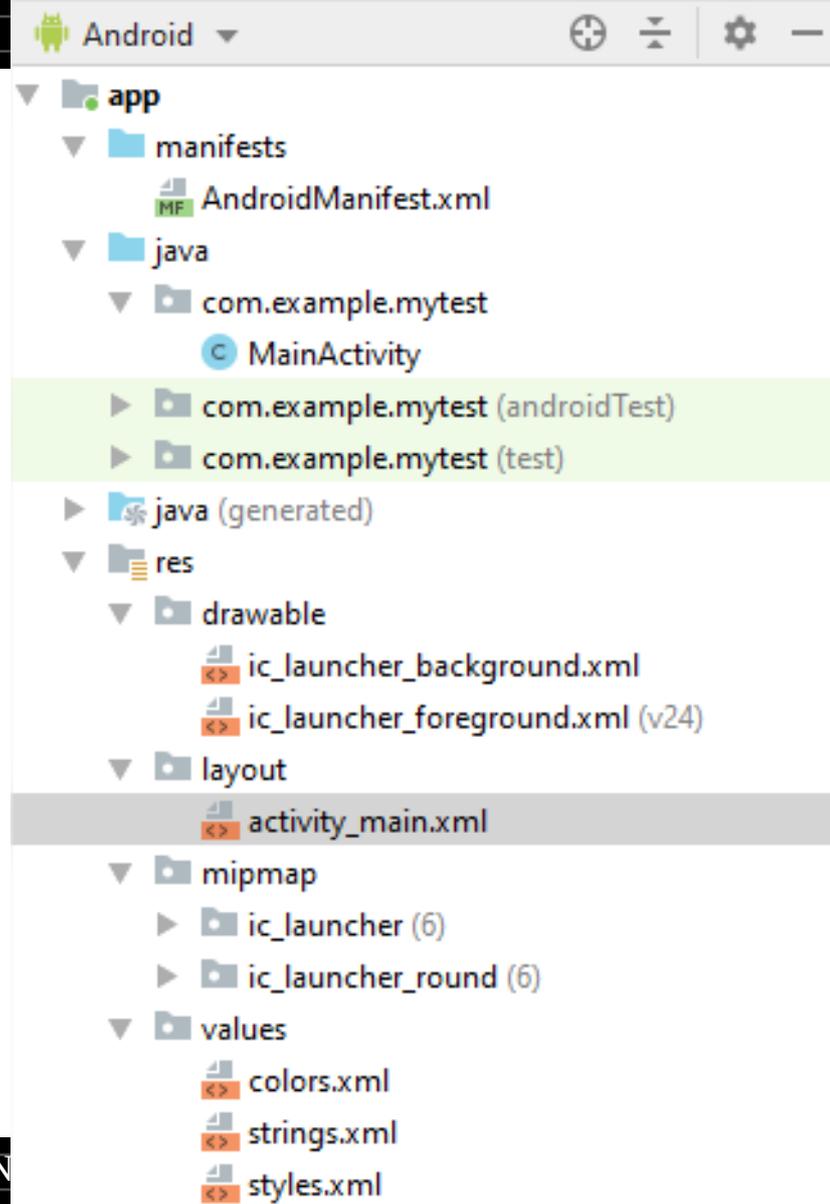
La séparation entre fonctionnalité et apparence est un point essentiel de la philosophie Android

Le code de la classe principale de l'application (**MainActivity.java**) est situé dans le sous dossier **com.example.mytest** de **java**.

Nous verrons après l'explication du code qui y est généré par défaut.

C'est dans le dossier **java** que seront enregistrées toutes les classes que nous allons créer dans nos projets.

Par ailleurs, tout ce qui touche à l'interface utilisateur sera intégré dans les sous dossiers de **res** dont voici une brève description :



Layout

layout regroupe les fichiers XML qui définissent la disposition des composants sur l'écran.

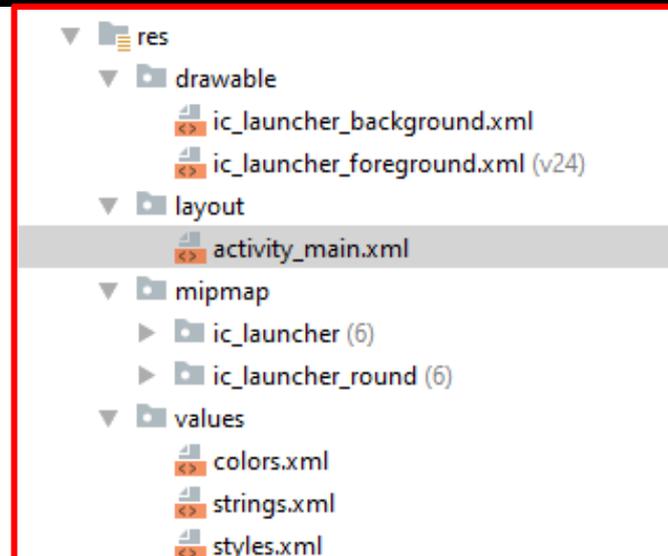
Il contient déjà, dès la création du projet, le **layout** de l'activité principale que nous avons créée.

Un fichier XML est créé par Android Studio **activity_main.xml**

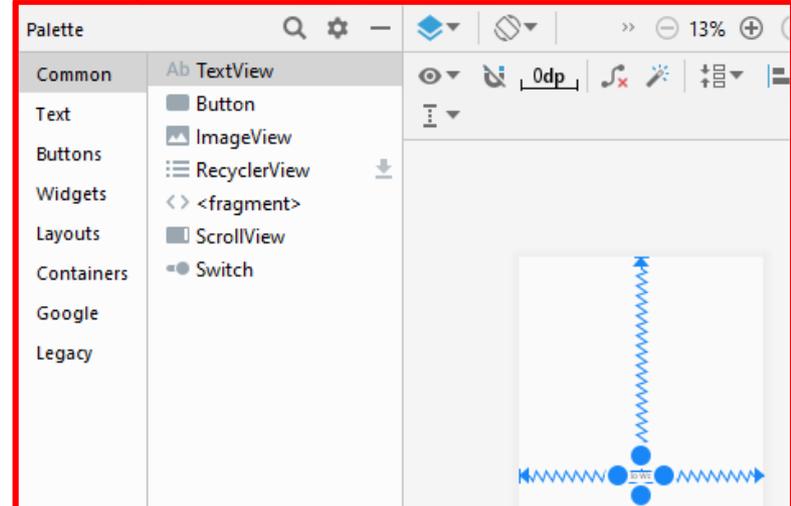
Il définit l'apparence générale de l'activité, sa disposition, sa taille, sa barre d'outil, éventuellement des boutons d'action flottant.

Ce sera donc ce dernier que nous manipulerons pour disposer les composants de l'activité et créer notre interface graphique.

Nous pourrons le faire via la manipulation d'objets ou via le code XML



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MainActivity">
8
9     <TextView
10         android:layout_width="wrap_content"
11         android:layout_height="wrap_content"
12         android:text="Hello World!"
13         app:layout_constraintBottom_toBottomOf="parent"
14         app:layout_constraintLeft_toLeftOf="parent"
15         app:layout_constraintRight_toRightOf="parent"
16         app:layout_constraintTop_toTopOf="parent" />
17
18 </androidx.constraintlayout.widget.ConstraintLayout>
```

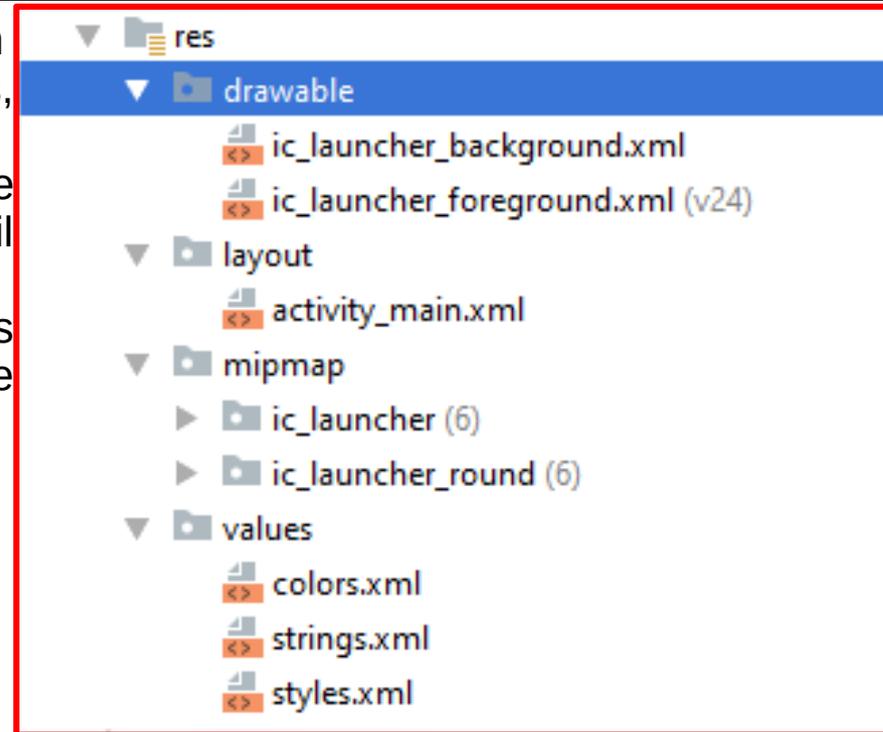


drawable

drawable contient tout élément qui peut être dessiné sur l'écran : images (en PNG de préférence), formes, animations, transitions, etc

Les dossiers **drawable** permettent aux développeurs de proposer des éléments graphiques pour tout genre d'appareil Android en fonction de sa résolution.

En peuplant correctement ces dossiers, on peut ainsi créer des applications avec une interface qui s'adapte à chaque résolution d'écran avec un seul fichier apk



Menu :

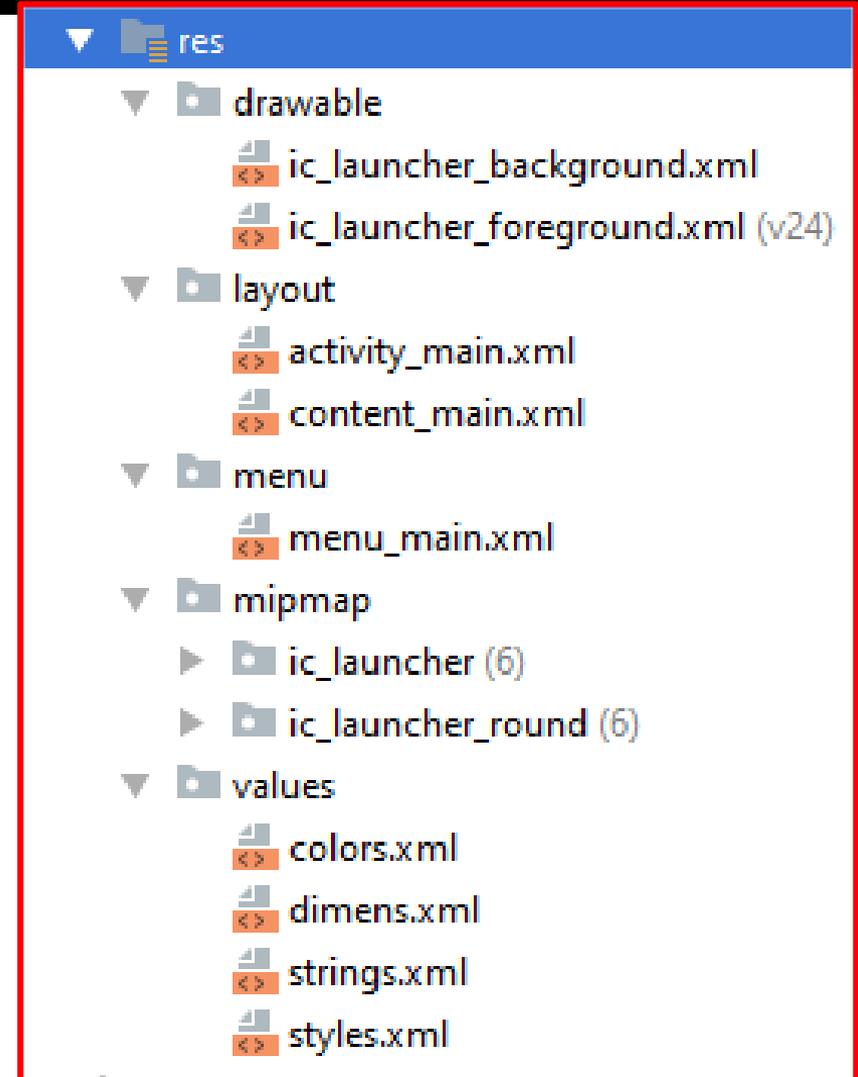
contient les fichiers XML définissant les menus.

Mipmap :

contient les images de l'icône de votre applications sous différentes résolutions.

Values :

contient les fichiers XML qui définissent des valeurs constantes (des chaînes de caractères, des dimensions, des couleurs, des styles etc).



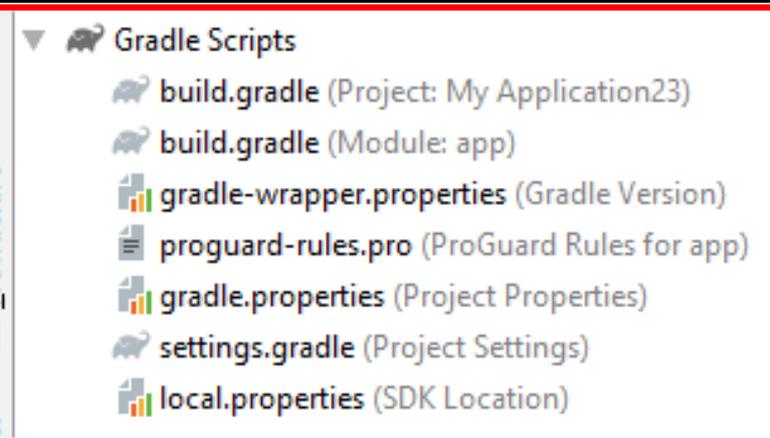
Gradle

Android Studio utilise un système qu'on appelle **Gradle** pour compiler et générer les applications.

Pour fonctionner le Gradle a besoin d'un script qui définit les règles de compilation et génération (configuration et dépendances Android Studio crée ainsi un script gradle pour chaque module (**build.gradle (Module :app)**) du projet ainsi qu'un script pour le projet entier. (**build.gradle (Project :MonAppli)**)

Dans le build gradle de l'application on définit entre autre la version du SDK utilisée pour la compilation, la version minimale du SDK nécessaire pour faire tourner l'application rétro compatibilité l'identifiant de l'application (le nom du package), etc

Obligation d'être connecté à Internet pour utiliser Android Studio. Gradle se synchronise à chaque compilation.



Android manifests

Vous trouverez également dans le dossier **manifests** du projet un fichier nommé **AndroidManifest.xml**. Ce fichier est obligatoire dans tout projet Android, et doit toujours avoir ce même nom.

Ce fichier permet au système de reconnaître l'application.

On peut accéder à une description rapide des classes ou des attributs XML en sélectionnant leur nom dans l'éditeur et appuyant sur Ctrl+Q



```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.mytest">
4
5     <application
6         android:allowBackup="true"
7         android:icon="@mipmap/ic_launcher"
8         android:label="MyTest"
9         android:roundIcon="@mipmap/ic_launcher_round"
10        android:supportsRtl="true"
11        android:theme="@style/AppTheme">
12        <activity android:name=".MainActivity">
13            <intent-filter>
14                <action android:name="android.intent.action.MAIN" />
15
16                <category android:name="android.intent.category.LAUNCHER" />
17            </intent-filter>
18        </activity>
19    </application>
20
21 </manifest>
```

Activity :

Une activité représente un écran contenant une interface utilisateur Une application est composée d'un ensemble d'activités. Les activités peuvent interagir entre elles, intra-application ou inter-applications.

Apk files:

C'est un fichier en format application package produit à l'issu de la compilation d'une application Android. C'est le fichier destiné aux utilisateurs. Ce fichier inclue le code de l'application sous forme d'un executable DVM (.dex), les ressources multimédias, et le AndroidManifest.xml.

DVM Dalvik Virtual Machine (Ou ART) :

Machine virtuelle disponible sur tout appareil Android. Elle exécute des fichiers en format .dex (format de pseudo code optimisé)

Intent :

C'est un message qui permet d'activer un composant (une activité par exemple) C'est un messenger qui demande une action à un autre composant. (de la même application ou d'une autre)

Fragment :

Un fragment représente un comportement ou une portion de l'interface utilisateur dans une activité On peut combiner plusieurs fragments dans une même activité pour créer une interface à volets multiples et réutiliser un fragment dans plusieurs activités.

Layout :

Le layout définit la structure visuelle d'une interface utilisateur. Les éléments d'un layout peuvent être définis soit dans un fichier XML ou bien à l'exécution à partir d'instructions dans le code. L'utilisation XML permet de mieux séparer l'apparence de l'application d'une part et le code qui définit son comportement d'autre part. Ceci permet de s'adapter plus facilement aux différentes cibles, il suffit de définir plusieurs fichiers XML pour un même code.

Manifest :

Toute application Android doit posséder un fichier XML nommé AndroidManifest.xml. Ce fichier contient des informations essentielles sur l'application que le système doit connaître pour pouvoir la lancer. Ce fichier définit le nom du package de l'application, décrit les composants de l'application, détermine le processus qui accueillera ces composants. Il déclare les permissions dont l'application a besoin et celles que les autres doivent avoir pour interagir avec l'application. Il déclare aussi la version Android minimale pour l'application et liste les bibliothèques dont l'application a besoin.

Service :

Un service est un composant qui tourne en tâche de fond, pour effectuer de longues opérations ou exécuter des tâches pour un autre processus. Un service ne possède pas une interface utilisateur.

Ressources :

Ce sont les ressources en relation avec la présentation de l'application (images fichiers audio etc. Pour chaque fichier de ressource qu'on inclue dans le projet, un identifiant unique est créé pour référencer cette ressource Le fait d'avoir des ressources séparées du code nous donne la possibilité de proposer des versions différentes en fonction de la configuration de la cible. En utilisant par exemple des images différentes selon la résolution de l'appareil, une mise en forme adaptée en fonction de l'orientation de l'affichage (portrait ou paysage), des fichiers de chaînes de caractères en plusieurs langues pour que l'interface s'affiche dans la langue de l'utilisateur, etc.

XML :

Extensible Markup Language (langage de balisage extensible), est un langage informatique conçu pour faciliter les échanges de données entre les systèmes d'informations. Il définit un ensemble de règles pour encoder les informations en format texte d'une façon indépendante de la machine Il est basé sur des balises, cependant celles ci ne sont pas définies par le standard XML, c'est l'utilisateur/programmeur qui définit ses propres balises Le langage ne fait que définir les règles d'écriture.

Pour terminer...

Pour le développement web, il existe 3 unités principales : les px (pixels), les em et les %.

Pour le développement d'application Android, il y en a 6 :

dp : Density independent Pixel (Densité de pixels indépendant) - Unité abstraite qui est basés sur la densité physique de l'écran. Cette unité est égale à 160 DPI (Points par pouce) par écran. Cette dimension sera utilisé pour la mise en page des éléments.

sp : Scale independent Pixel (Echelle de pixels indépendant) - Utilisé pour les tailles de polices. On pourrait comparé cette unité aux em du développement web. La police peut être plus ou moins grosse suivant les préférences utilisateurs

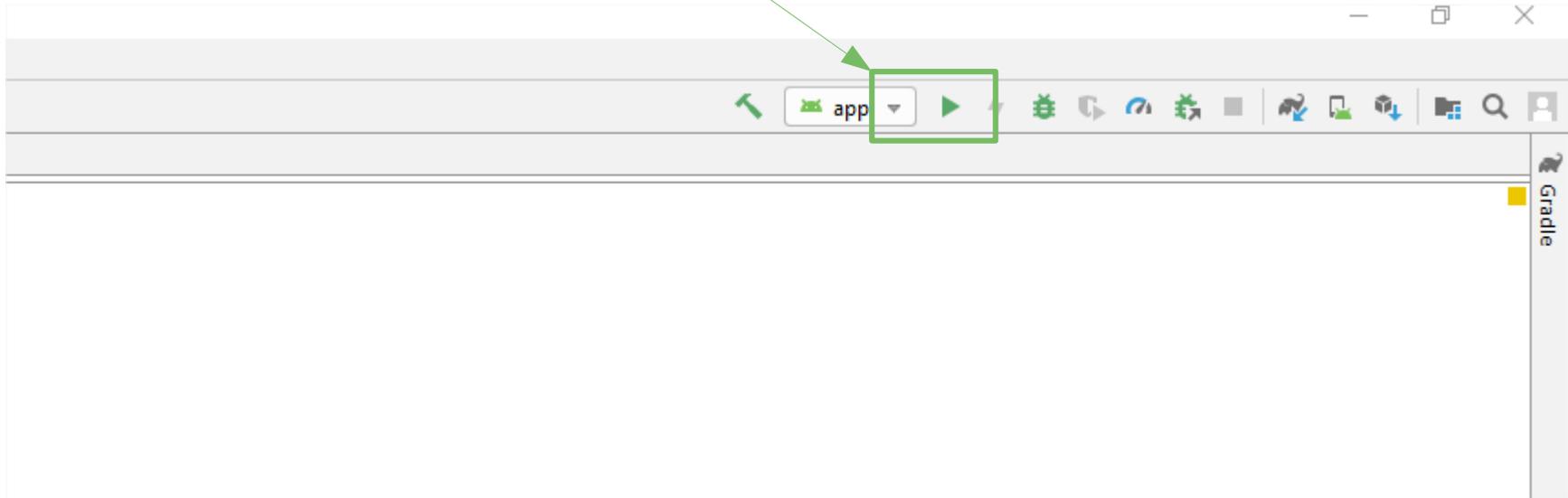
pt : Point - 72 points par pouces. basé sur la taille physique de l'écran.

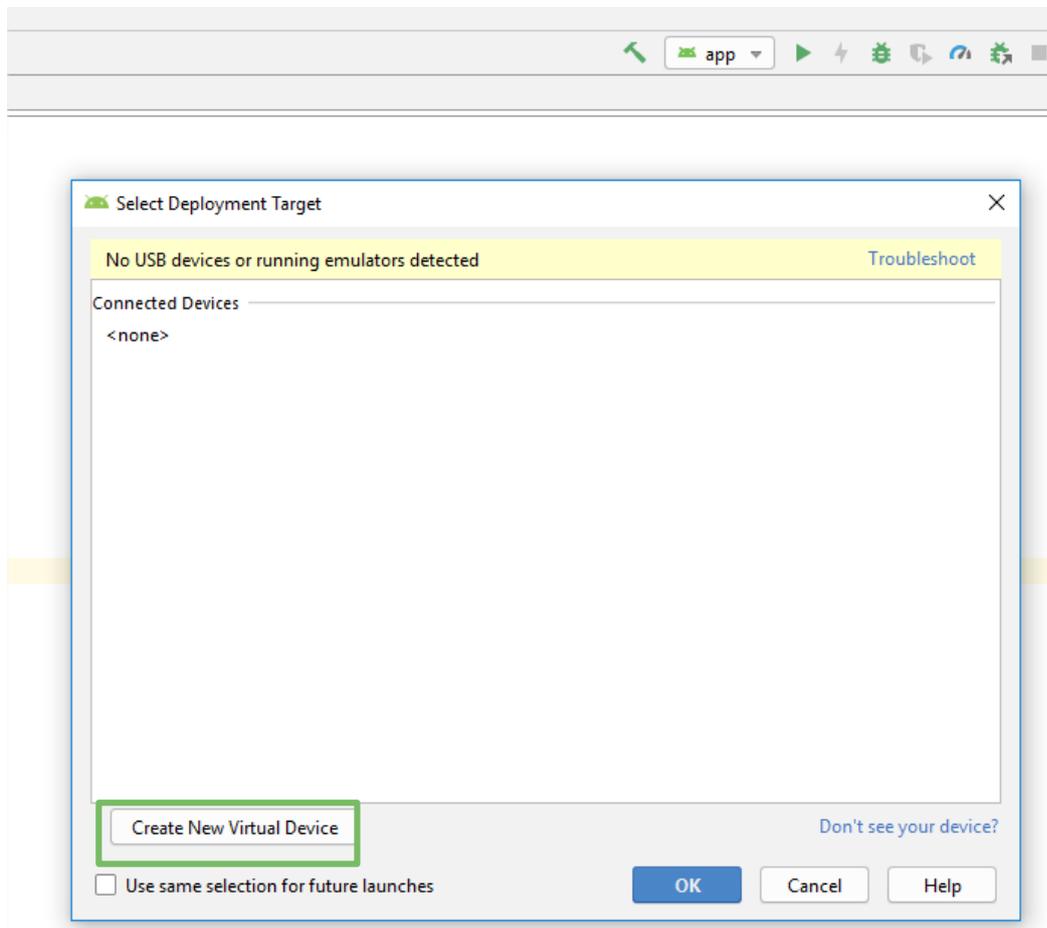
px : Pixels - Corresponds aux pixels réels de l'écran. Cette unité de mesure n'est pas recommandées car le rendu sur les différents types d'écran peut être différents. Le nombre de pixels par pouce peut varier suivant les appareils.

mm : Millimètre - basée sur la taille physique de l'écran

in : Inches (Pouces) - basée sur la taille physique de l'écran. Ces dimensions se définissent dans les layout (les fichiers XML).

Création de la machine virtuelle Android







Select Hardware

Android Studio

Choose a device definition



Category	Name	Play Store	Size	Resolution	Density
TV	Pixel XL		5,5"	1440x2560	560dpi
Phone	Pixel 2 XL		5,99"	1440x2880	560dpi
Wear OS	Pixel 2		5,0"	1080x1920	420dpi
Tablet	Pixel		5,0"	1080x1920	420dpi
	Nexus S		4,0"	480x800	hdpi
	Nexus One		3,7"	480x800	hdpi
	Nexus 6P		5,7"	1440x2560	560dpi
	Nexus 6		5,96"	1440x2560	560dpi
	Nexus 5X		5,2"	1080x1920	420dpi

New Hardware Profile

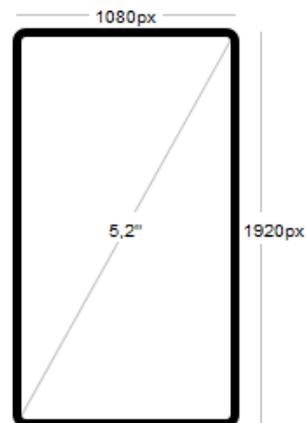
Import Hardware Profiles



Clone Device...



Nexus 5X



Size: large
Ratio: long
Density: 420dpi

Previous

Next

Cancel

Finish

Help



System Image

Android Studio

Select a system image

Recommended x86 Images Other Images

Release Name	API Level	ABI	Target
Pie Download	28	x86	Android 9.0 (Google Play)
Oreo Download	27	x86	Android 8.1 (Google Play)
Oreo Download	26	x86	Android 8.0 (Google Play)
Nougat Download	25	x86	Android 7.1.1 (Google Play)
Nougat Download	24	x86	Android 7.0 (Google Play)



! A system image must be selected to continue.

Pie



API Level

28

Android

9.0**Google Inc.**

System Image

x86

We recommend these Google Play images because this device is compatible with Google Play.

Questions on API level?

See the [API level distribution chart](#)

Previous

Next

Cancel

Finish

Help

System
Android Studio

License Agreement

Android Studio

Select a system image

Recommended x86 Image

Release Name
Pie Download
Oreo Download
Oreo Download
Nougat Download
Nougat Download

A system image must be

Licenses

- android-sdk-license
 - Google Play Intel x86 Atom System

Terms and Conditions

This is the Android Software Development Kit License Agreement

1. Introduction

1.1 The Android Software Development Kit (referred to in the License Agreement as the "SDK" and specifically including the Android system files, packaged APIs, and Google APIs add-ons) is licensed to you subject to the terms of the License Agreement. The License Agreement is a legally binding contract between you and Google in relation to your use of the SDK.

1.2 "Android" means the Android software stack for devices, as made available under the Android Open Source Project, which is located at the following URL: <http://source.android.com/>, and which may be updated from time to time.

1.3 A "compatible implementation" means any Android device that (i) complies with the Android Compatibility Definition document, which can be found at the Android compatibility website (<http://source.android.com/compatibility>) and which may be updated from time to time; and (ii) successfully passes the Android Compatibility Test Suite (CTS).

1.4 "Google" means Google Inc., a Delaware corporation with principal place of business at 1600 Amphitheatre Parkway, Mountain View, CA 94043, United States.

2. Accepting the License Agreement

2.1 In order to use the SDK, you must first agree to the License Agreement. You may not use the SDK if you do not accept the License Agreement.

 Decline Accept

Previous

Next

Cancel

Select a system image

Recommended x86 Image

Release Name
<i>Pie</i> Download
<i>Oreo</i> Download
<i>Oreo</i> Download
<i>Nougat</i> Download
<i>Nougat</i> Download

 A system image must be

Installing Requested Components

SDK Path: C:\Users\xi06\AppData\Local\Android\Sdk

Packages to install:

- Google Play Intel x86 Atom System Image (system-images;android-24;google_apis_playstore;x86)

Preparing "Install Google Play Intel x86 Atom System Image (revision: 19)".

Downloading https://dl.google.com/android/repository/sys-img/google_apis_playstore/x86-24_r19.zip

Downloading (3%): 23,3 / 775,1 MB ...

[://dl.google.com/android/repository/sys-img/google_apis_playstore/x86-24_r19.zip](https://dl.google.com/android/repository/sys-img/google_apis_playstore/x86-24_r19.zip)

 Please wait until the installation finishes

Previous

Next

Cancel

Finish

SDK Quickfix Installation

Component Installer

Android Studio

Installing Requested Components

SDK Path: C:\Users\xi06\AppData\Local\Android\Sdk

```
Packages to install:  
- Google Play Intel x86 Atom System Image (system-images;android-24;google_apis_playstore;x86)  
  
Preparing "Install Google Play Intel x86 Atom System Image (revision: 19)".  
Downloading https://dl.google.com/android/repository/sys-img/google_apis_playstore/x86-24_r19.zip  
"Install Google Play Intel x86 Atom System Image (revision: 19)" ready.  
Installing Google Play Intel x86 Atom System Image in  
C:\Users\xi06\AppData\Local\Android\Sdk\system-images\android-24\google_apis_playstore\x86  
"Install Google Play Intel x86 Atom System Image (revision: 19)" complete.  
"Install Google Play Intel x86 Atom System Image (revision: 19)" finished.
```

Done

Previous Next Cancel Finish

Select a system image

Recommended x86 Images Other Images

Release Name	API Level	ABI	Target
Pie Download	28	x86	Android 9.0 (Google Play)
Oreo Download	27	x86	Android 8.1 (Google Play)
Oreo Download	26	x86	Android 8.0 (Google Play)
Nougat Download	25	x86	Android 7.1.1 (Google Play)
Nougat	24	x86	Android 7.0 (Google Play)

Previous **Next** Cancel

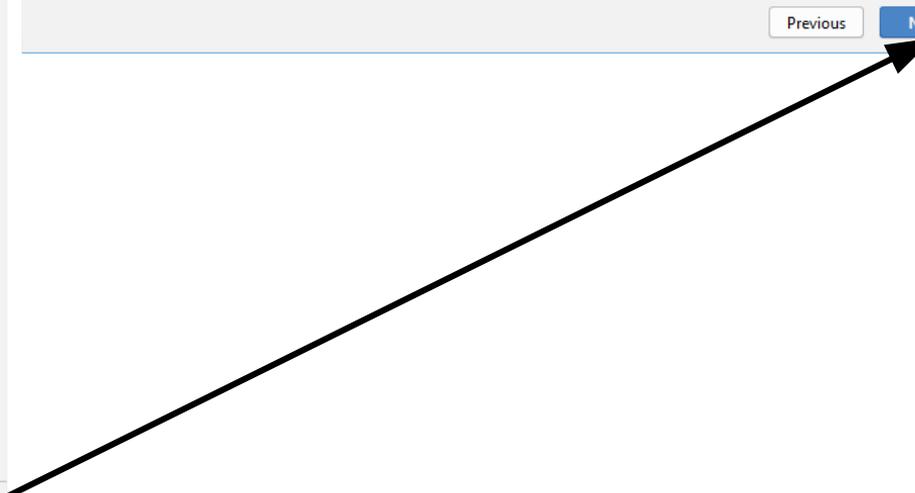
Nougat



API Level: **24**
Android: **7.0**
Google Play: **Go**
System: **x86**

We recommend these Google Play device is compatible with Google

Questions on API level?
See the [API level distribution chart](#)



Verify Configuration

AVD Name:

Nexus 5X	5.2 1080x1920 420dpi	<input type="button" value="Change..."/>
Nougat	Android 7.0 x86	<input type="button" value="Change..."/>

Startup orientation

Portrait Landscape

Emulated Performance: Graphics:

Device Frame Enable Device Frame

AVD Name

The name of this AVD.

Select Deployment Target

No USB devices or running emulators detected [Troubleshoot](#)

Connected Devices: <none>

Available Virtual Devices:

- Nexus SX API 24

[Don't see your device?](#)

Use same selection for future launches

Instant Run

Instant Run requires that the platform corresponding to your target device (Android 7.0 (Nougat)) is installed.

Instant Run



Instant Run requires that the platform corresponding to your target device (Android 7.0 (Nougat)) is installed.

Install and Continue

Proceed without Instant Run

SDK Quickfix Installation



Component Installer

Android Studio

Installing Requested Components

SDK Path: C:\Users\xi06\AppData\Local\Android\Sdk

Packages to install:

- Android SDK Platform 24 (platforms;android-24)

Preparing "Install Android SDK Platform 24 (revision: 2)".

Downloading https://dl.google.com/android/repository/platform-24_r02.zip

Downloading (8%): 6,3 / 78,8 MB ...

https://dl.google.com/android/repository/platform-24_r02.zip

Please wait until the installation finishes

Previous

Next

Cancel

Finish