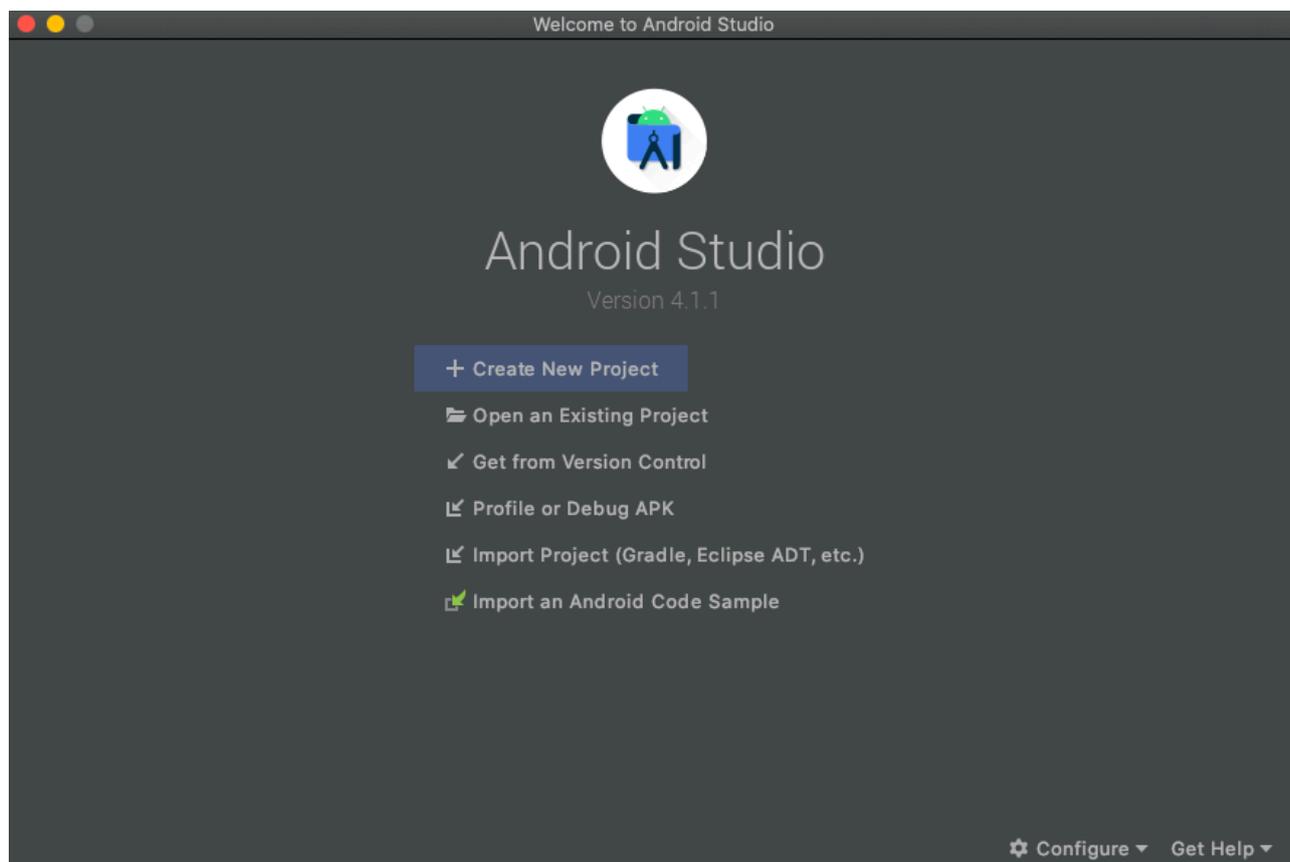


Android Studio



Android Studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS, Chrome OS et Linux15.

Page d'accueil



Création d'une application simple

La création d'application avec Android Studio est très simple, il faut préciser les informations relatives au projet : son nom, le package par défaut et la localisation du projet sur votre disque dur.

Le package correspond à l'identifiant technique de votre application. Si vous souhaitez déployer plus tard votre application sur Android Store, c'est le package qui permet d'identifier de manière

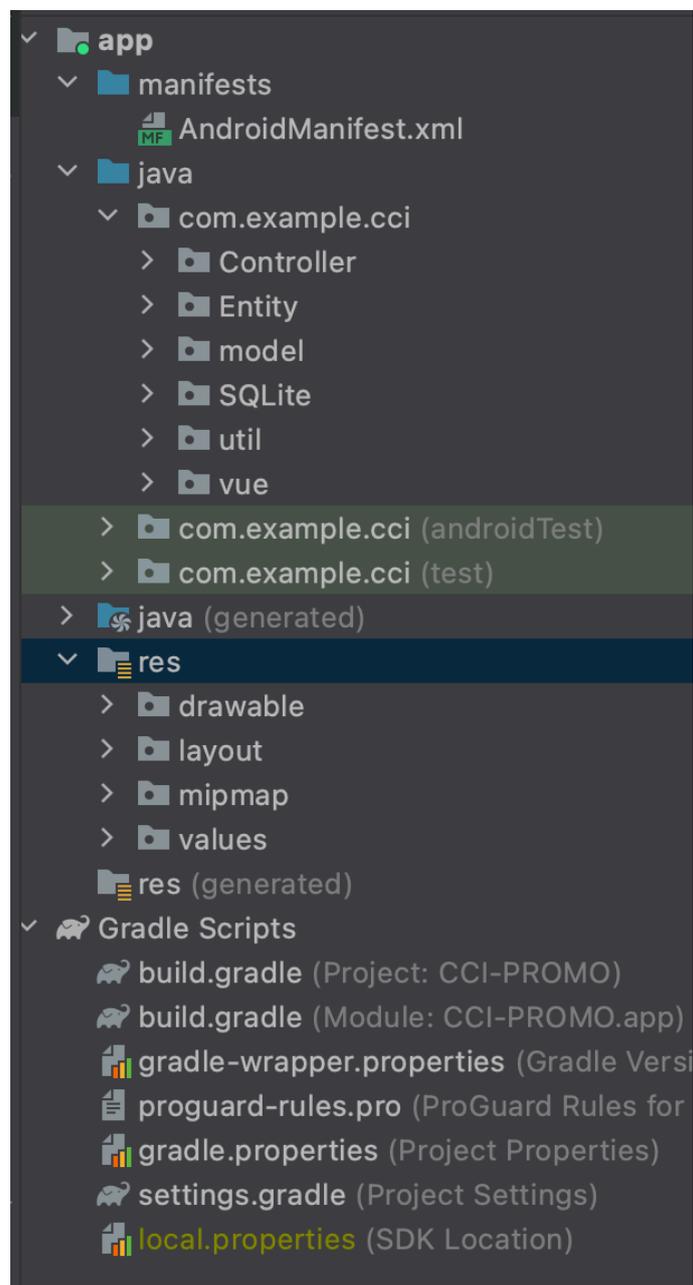
unique votre application et l'organisation à qui elle appartient. Ainsi, l'éditeur d'Android Studio s'ouvre une fois les fichiers du projet créés. Le projet est automatiquement construit et il ne doit pas y avoir d'erreur.

Organisation du projet

Un projet Android Studio est géré par l'outil de build Gradle qui est un outil dédié aux projets Java. Un projet géré par Gradle possède un ou plusieurs fichiers build.gradle pour configurer sa construction.

Les projets Android sont composés de modules. Les projets les plus simples n'ont besoin que d'un seul module qui contient l'ensemble des fichiers sources. Par défaut, un projet Android contient un seul module nommé app. Dans le répertoire de votre projet, vous pouvez trouver un répertoire app qui contient l'ensemble des fichiers pour ce module.

Structure des fichiers d'un projet Android



Dans le répertoire d'un module, vous trouverez un fichier `build.gradle` pour configurer la construction du module et les répertoires suivants :

- **src/main/java** : ce répertoire contient les sources Java du module
- **src/main/res** : ce répertoire contient les ressources du module (Cf. plus bas)
- **src/test/java** : ce répertoire contient les sources Java des tests unitaires
- **src/androidTest/java** : ce répertoire contient les sources Java des tests instrumentés (c'est-à-dire les tests qui doivent être exécutés directement sur un périphérique Android).
- **libs** : ce répertoire contient éventuellement les bibliothèques nécessaires au module
- **build** : ce répertoire contient les fichiers de travail créés par Gradle.

On trouve également le fichier `src/main/AndroidManifest.xml` qui correspond au manifeste du module. Un fichier manifeste fournit au système l'ensemble des informations nécessaires à l'exécution du module.

Par défaut, Android Studio affiche l'explorateur de fichiers à gauche de l'éditeur. Cet explorateur est par défaut en mode Android, c'est-à-dire qu'il vous affiche une vue adaptée. Il ne vous affiche donc pas l'organisation réelle des fichiers sur votre disque dur. Vous pouvez choisir le mode Project pour voir l'organisation réelle mais le mode Android est généralement plus lisible et plus adapté pour réaliser les développements.

Lancement de l'application

Depuis Android Studio, vous pouvez lancer une application sur votre téléphone Android. Il vous suffit de brancher votre appareil en USB sur votre station de travail et de choisir le menu Run > Run "app". Android Studio utilise l'outil adb (Android Debug Bridge) pour installer l'application créée sur votre appareil et l'exécuter. Une fois installée, l'application est accessible depuis le menu des applications Android sur votre appareil.

Pour pouvoir utiliser le mode debug depuis Android Studio, vous devez activer le mode developer sur votre appareil. Pour plus d'information, consultez le guide utilisateur :

<https://developer.android.com/studio/debug/dev-options>

L'émulateur Android

Si vous ne disposez pas d'un appareil Android ou si vous souhaitez tester votre application dans des conditions particulières, vous pouvez utiliser l'émulateur Android. Il vous permet de choisir le type d'appareil que vous souhaitez émuler.

Pour créer un émulateur, cliquez sur le menu Tools > AVD Manager (AVD pour Android Virtual Device).

Cliquez sur le bouton « Create Virtual Device... » pour ajouter, depuis la liste fournie, le modèle d'appareil à émuler.

À partir de la fenêtre du AVD Manager, vous pouvez directement lancer l'émulateur. Lorsque vous lancez votre application depuis Android Studio, l'émulateur apparaît dans la liste des périphériques disponibles.

Notion d'activité

Le modèle de projet choisi génère le code pour une application avec une activité. Une activité est un composant d'une application Android qui représente un écran avec lequel l'utilisateur va

pouvoir interagir. Un composant Android est représenté par une classe Java. Cependant, un composant n'est pas instancié directement dans le code de l'application. C'est lors de l'exécution que le système prend en charge la création et le cycle de vie d'un objet de cette classe. Dans le cas d'une activité, un objet correspondant est instancié par le système quand il est nécessaire d'afficher l'écran à l'utilisateur (par exemple au lancement de l'application).

```
1 package dev.gayerie.premiereapplication;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.os.Bundle;
6
7 public class MainActivity extends AppCompatActivity {
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main);
13    }
14 }
```

Les ressources

- **drawable** : ce répertoire contient les images (PNG, JPEG...) ou des fichiers XML décrivant des images vectorielles (format SVG).
- **layout** : ce répertoire contient les fichiers XML décrivant la disposition des vues d'une activité.
- **mipmap** : ce répertoire contient les icônes de l'application. Comme une application Android doit pouvoir être installée sur différents périphériques, il est nécessaire de fournir plusieurs résolutions de la même icône. La technique du mipmap consiste précisément à fournir plusieurs résolutions d'une même image afin d'éviter la pixellisation à l'affichage en permettant au système de choisir la résolution adéquate.
- **values** : ce répertoire contient les variables que l'on ne souhaite pas écrire en dur dans le code (comme les libellés et les messages).

Pour notre application, le répertoire `app/res/layout` contient le fichier `activity_main.xml`. Il s'agit d'un fichier de layout, c'est-à-dire un fichier qui décrit le rendu visuel de l'activité. Android Studio intègre un éditeur de layout qui vous permet d'éditer graphiquement votre layout. L'éditeur est ouvert par défaut lorsque vous double-cliquez sur le nom du fichier. Vous pouvez à tout moment consulter le source XML en cliquant sur l'onglet Text sous l'éditeur.