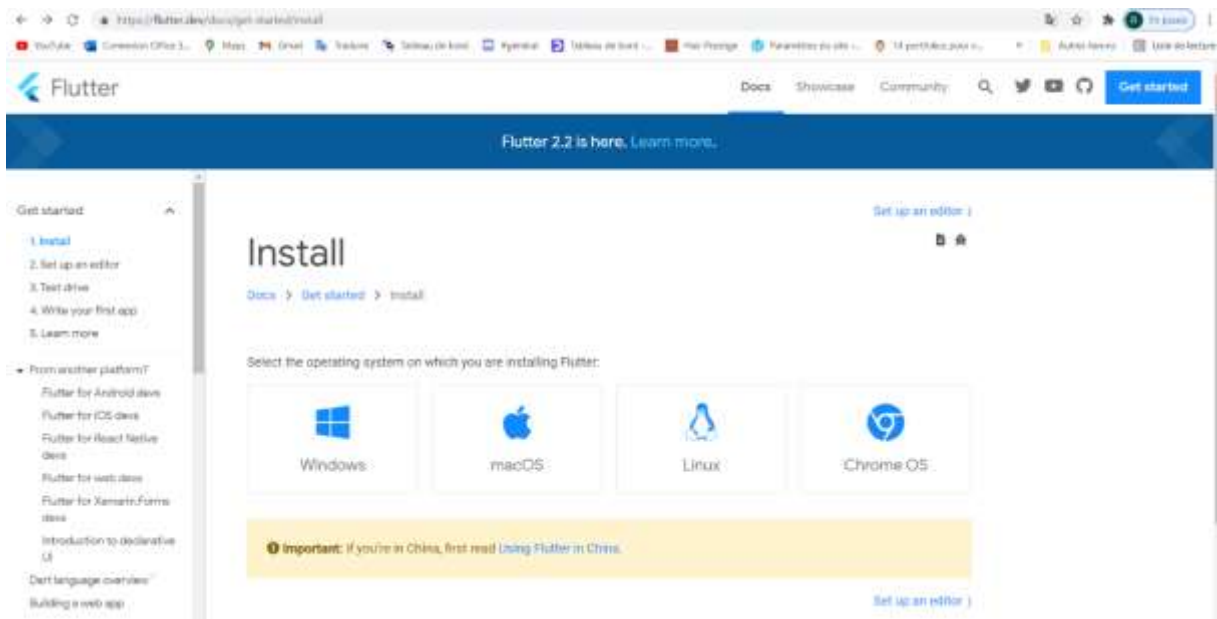
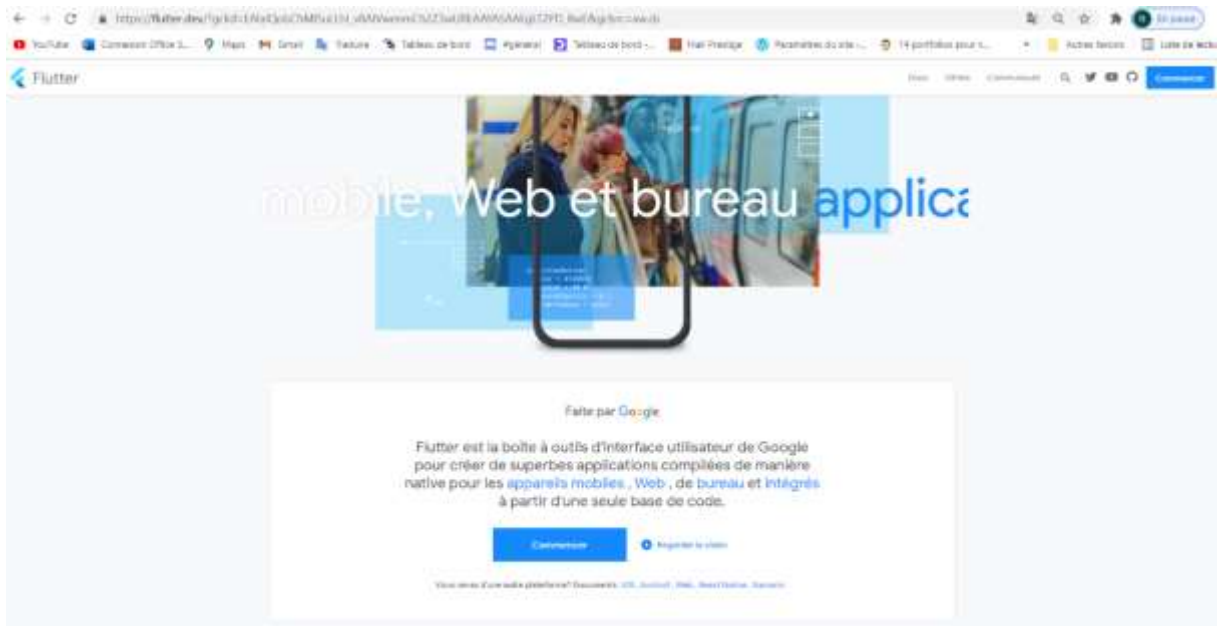


Installation Flutter :

1- Télécharger le fichier sur : <https://flutter.dev/>



Installer le SDK Flutter

Se rendre sur la page <https://flutter.dev/docs/get-started/install/windows> puis cliquer sur le bouton permettant de télécharger la dernière version stable de Flutter

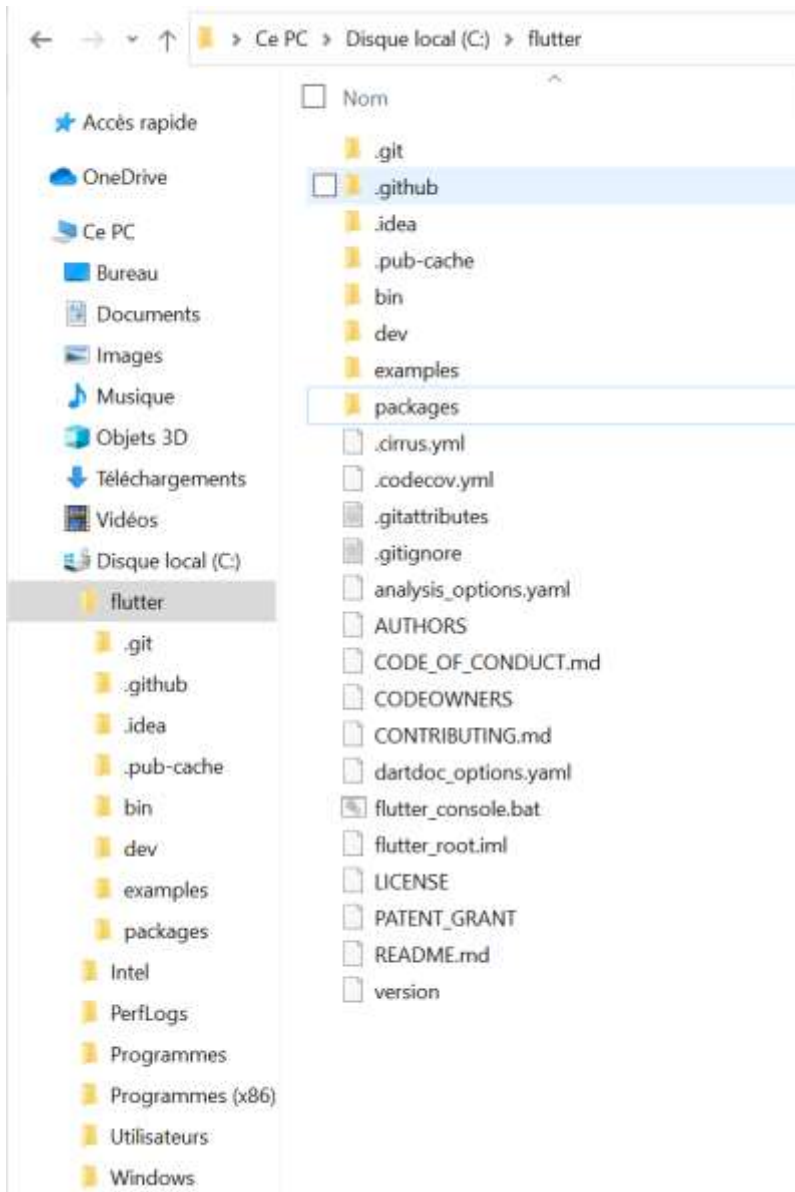
Get the Flutter SDK

1. Download the following installation bundle to get the latest stable release of the Flutter SDK:

[flutter_windows_2.2.1-stable.zip](#)

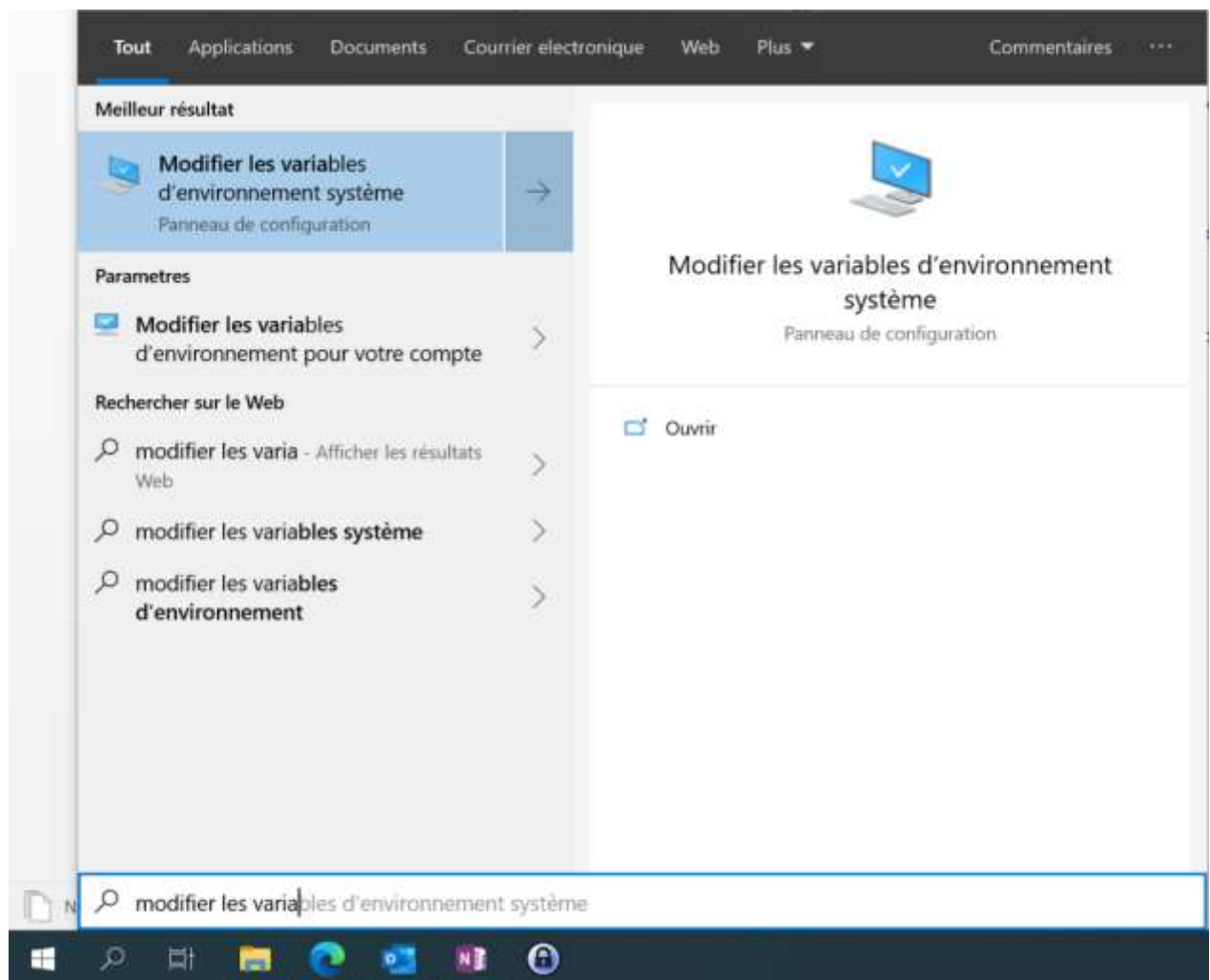
For other release channels, and older builds, see the [SDK releases](#) page.

Une fois l'archive zip téléchargée, extraire les fichiers et placez le contenu à l'emplacement d'installation souhaité pour le SDK Flutter, par exemple, **C:\flutter**.

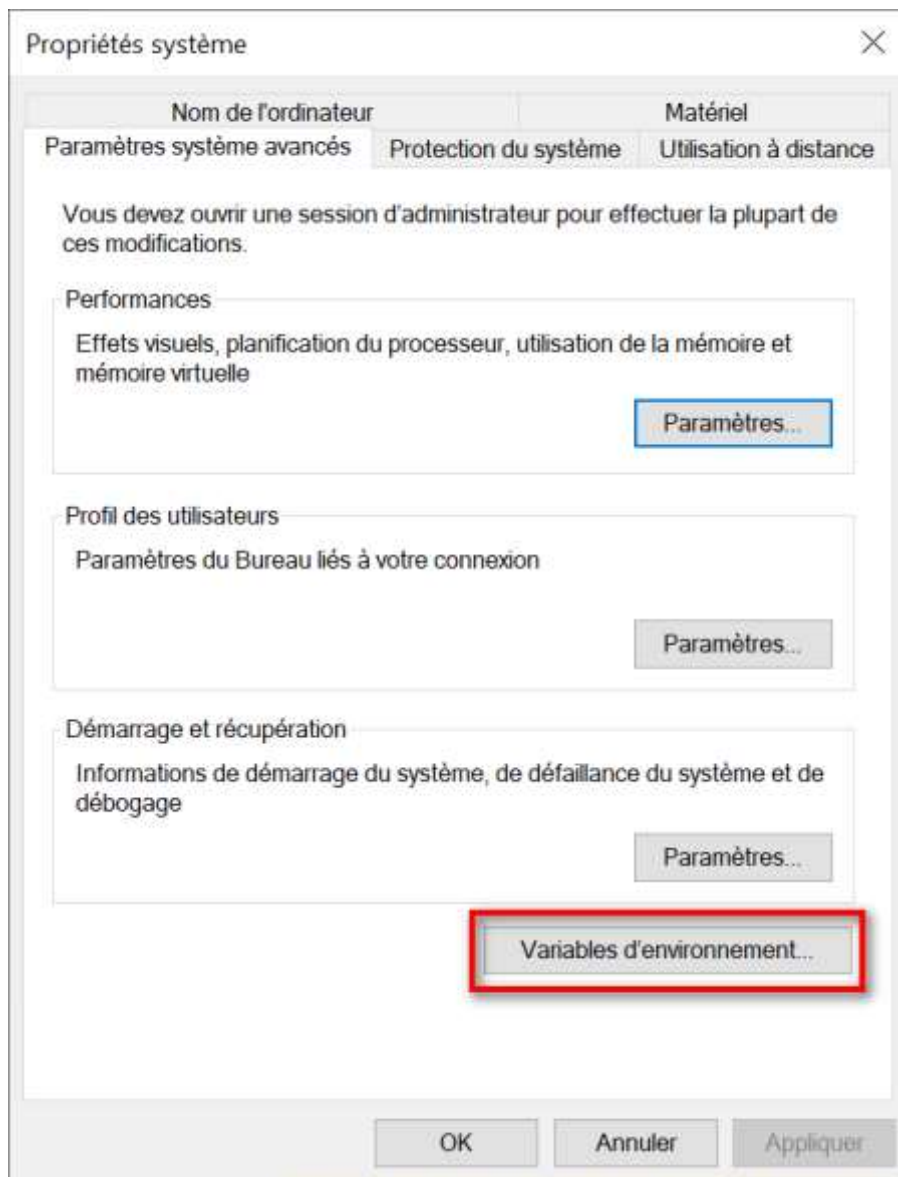


Il faut maintenant modifier nos variables d'environnement Windows pour lui indiquer l'emplacement de Flutter.

Dans la barre de recherche **Démarrer**, saisissez "**modifier les varia**" et sélectionnez **Modifier les variables d'environnement**.

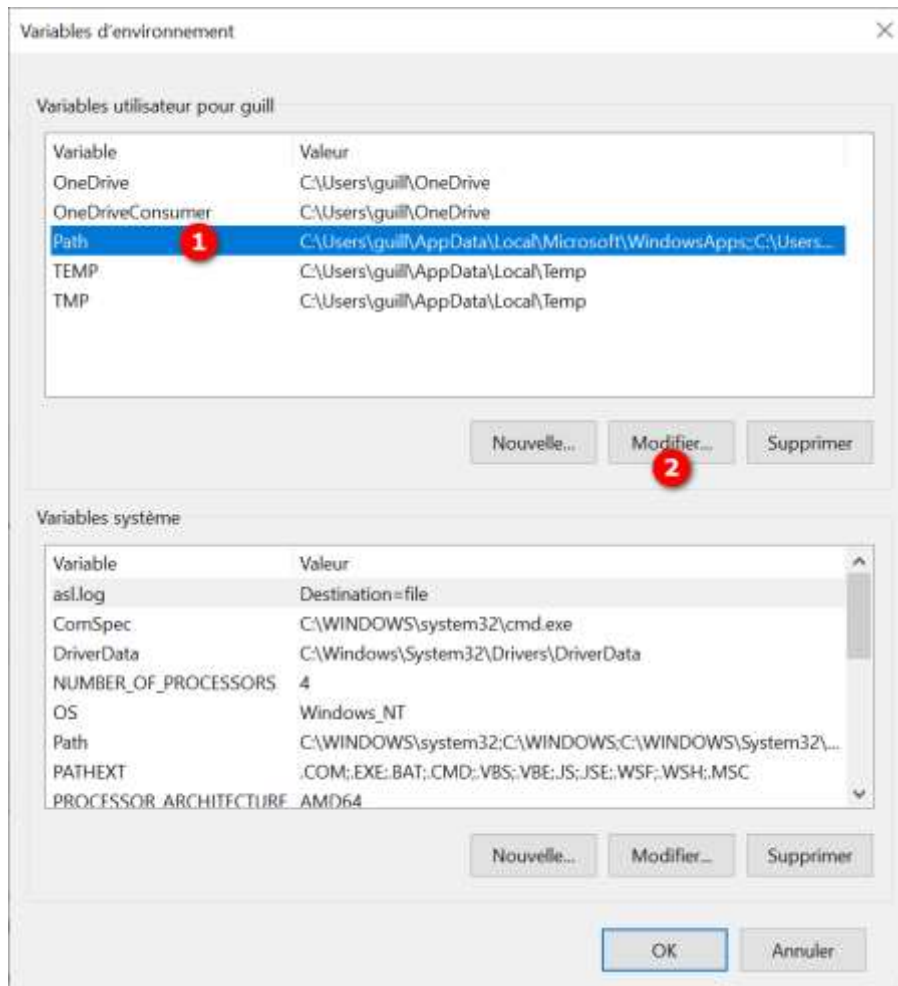


Ensuite dans la fenêtre **Propriétés Système**, cliquez sur le bouton **Variables d'environnement...**

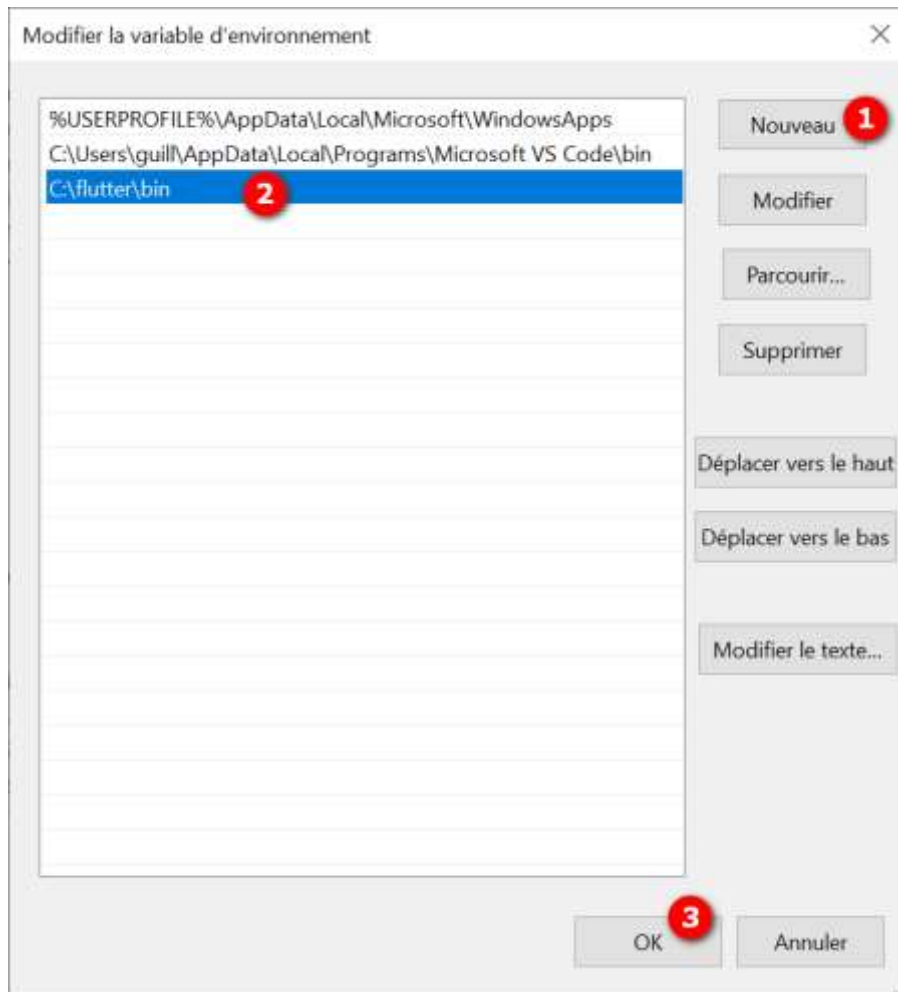


Dans la nouvelle fenêtre **Variables d'environnement**, Sous **Variables utilisateur pour xxx** (où xxx est votre nom d'utilisateur), vérifiez s'il existe une entrée appelée **Path**.

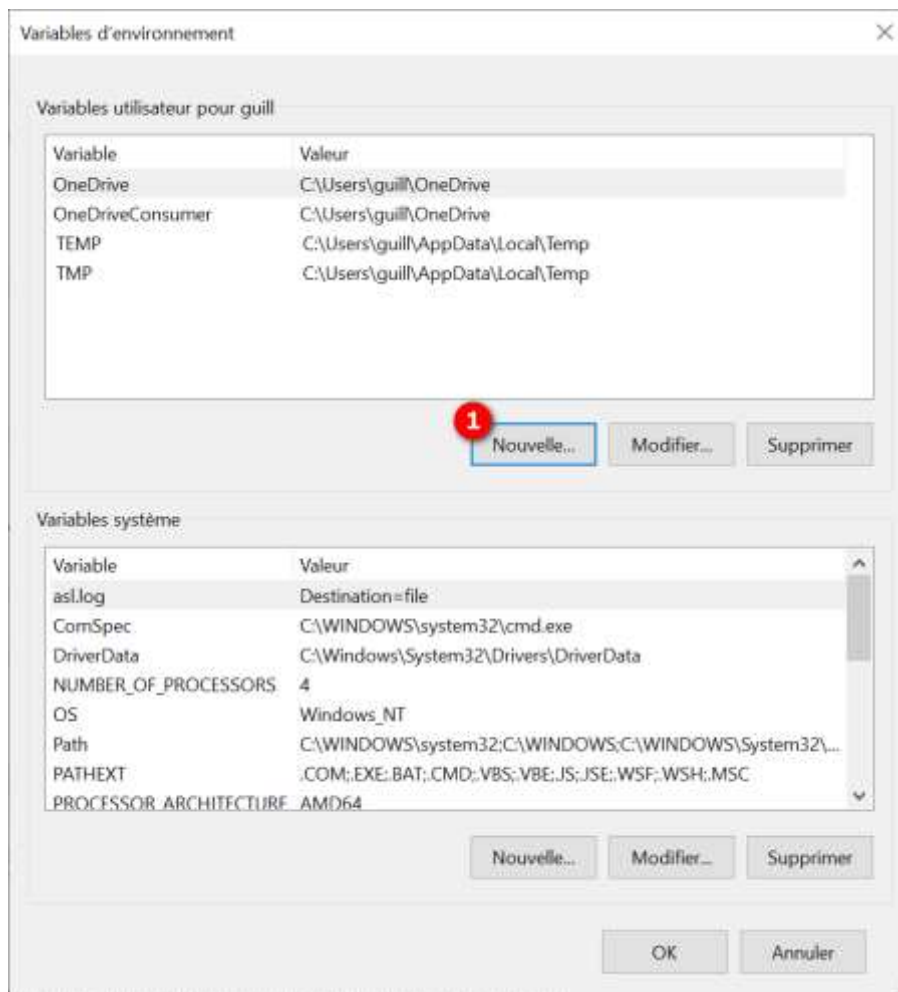
Si l'entrée existe, ajoutez le chemin complet à vers **flutter\bin** (dans notre exemple c:\flutter\bin) en utilisant « ; » comme séparateur des valeurs existantes. Pour cela, sélectionnez l'entrée **Path**, puis cliquez sur le bouton **Modifier...**



Dans la fenêtre **Modifier la variable d'environnement**, cliquez sur **Nouveau**, puis saisissez la nouvelle entrée, ici **C:\flutter\bin** et enfin, cliquez sur **OK**.



Si l'entrée n'existe pas, créez une nouvelle variable utilisateur nommée **Path** avec le chemin complet de flutter\bin comme valeur (dans notre exemple c:\flutter\bin). Pour cela, cliquez sur le bouton **Nouveau...**



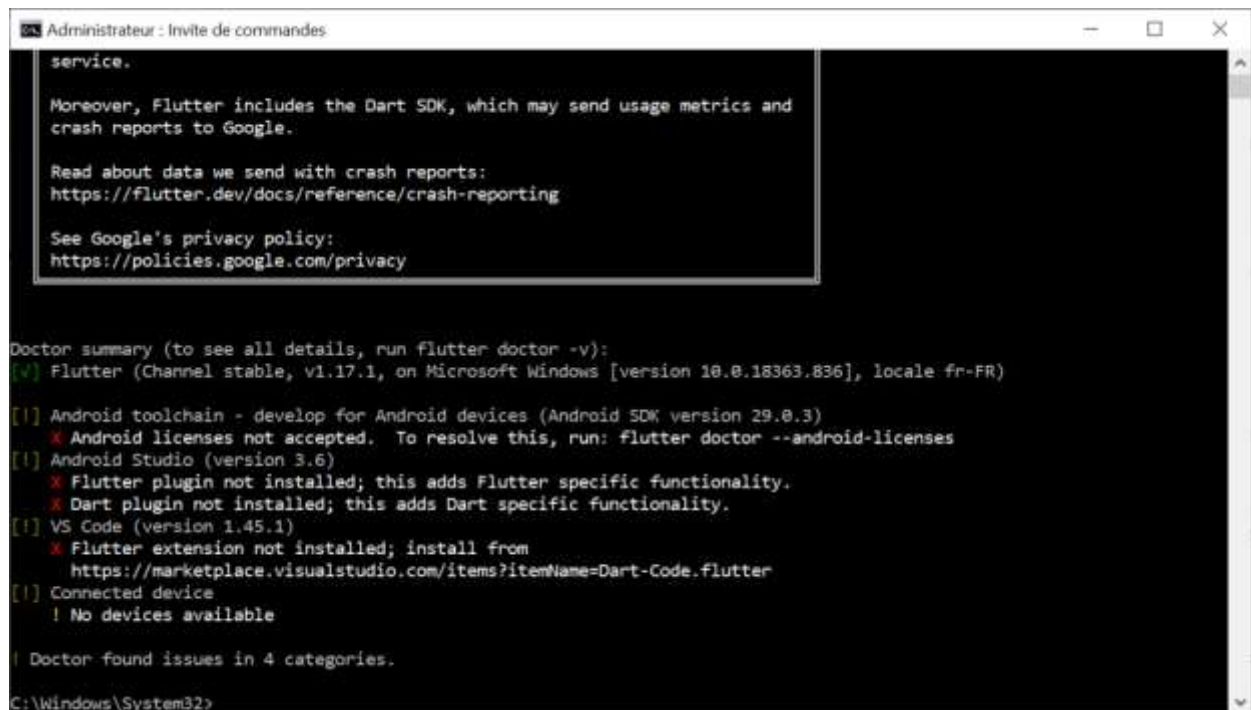
Dans la fenêtre **Nouvelle variable utilisateur**, saisissez **Path** comme **Nom de la variable** et le chemin vers **flutter\bin** dans **Valeur de la variable**, puis cliquez sur **OK**.



A partir de là, il est possible d'exécuter des commandes Flutter et notamment la commande **flutter doctor**.

Flutter doctor

Cette commande vérifie votre environnement et affiche un rapport sur l'état de votre installation Flutter. A ce stade, vous devriez avoir des erreurs que nous allons corriger dans l'étape suivante **Configuration de l'environnement.(accepter les licences Android et installer les extensions sur Visual Studio Code)**



```
service.

Moreover, Flutter includes the Dart SDK, which may send usage metrics and
crash reports to Google.

Read about data we send with crash reports:
https://flutter.dev/docs/reference/crash-reporting

See Google's privacy policy:
https://policies.google.com/privacy

Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, v1.17.1, on Microsoft Windows [version 10.0.18363.836], locale fr-FR)

[!] Android toolchain - develop for Android devices (Android SDK version 29.0.3)
    ✗ Android licenses not accepted. To resolve this, run: flutter doctor --android-licenses
[!] Android Studio (version 3.6)
    ✗ Flutter plugin not installed; this adds Flutter specific functionality.
    ✗ Dart plugin not installed; this adds Dart specific functionality.
[!] VS Code (version 1.45.1)
    ✗ Flutter extension not installed; install from
      https://marketplace.visualstudio.com/items?itemName=Dart-Code.flutter
[!] Connected device
    ! No devices available

! Doctor found issues in 4 categories.

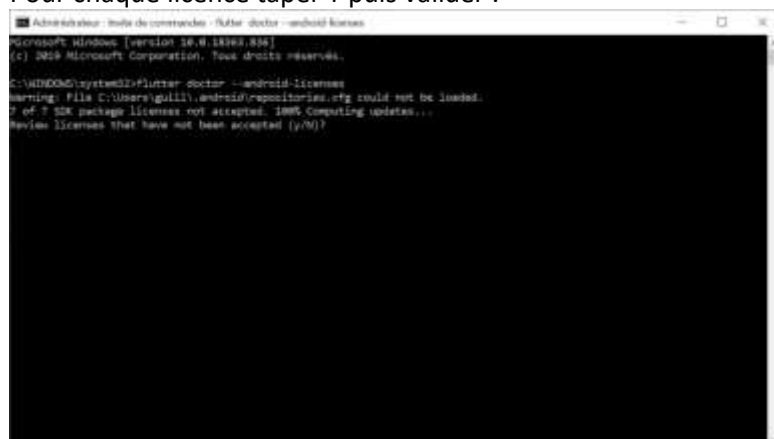
C:\Windows\System32>
```

Accepter les contrats de licences Android :

Ouvrir une invite de commande saisir la commande suivante :

Flutter doctor --android-licenses

Pour chaque licence taper Y puis valider :

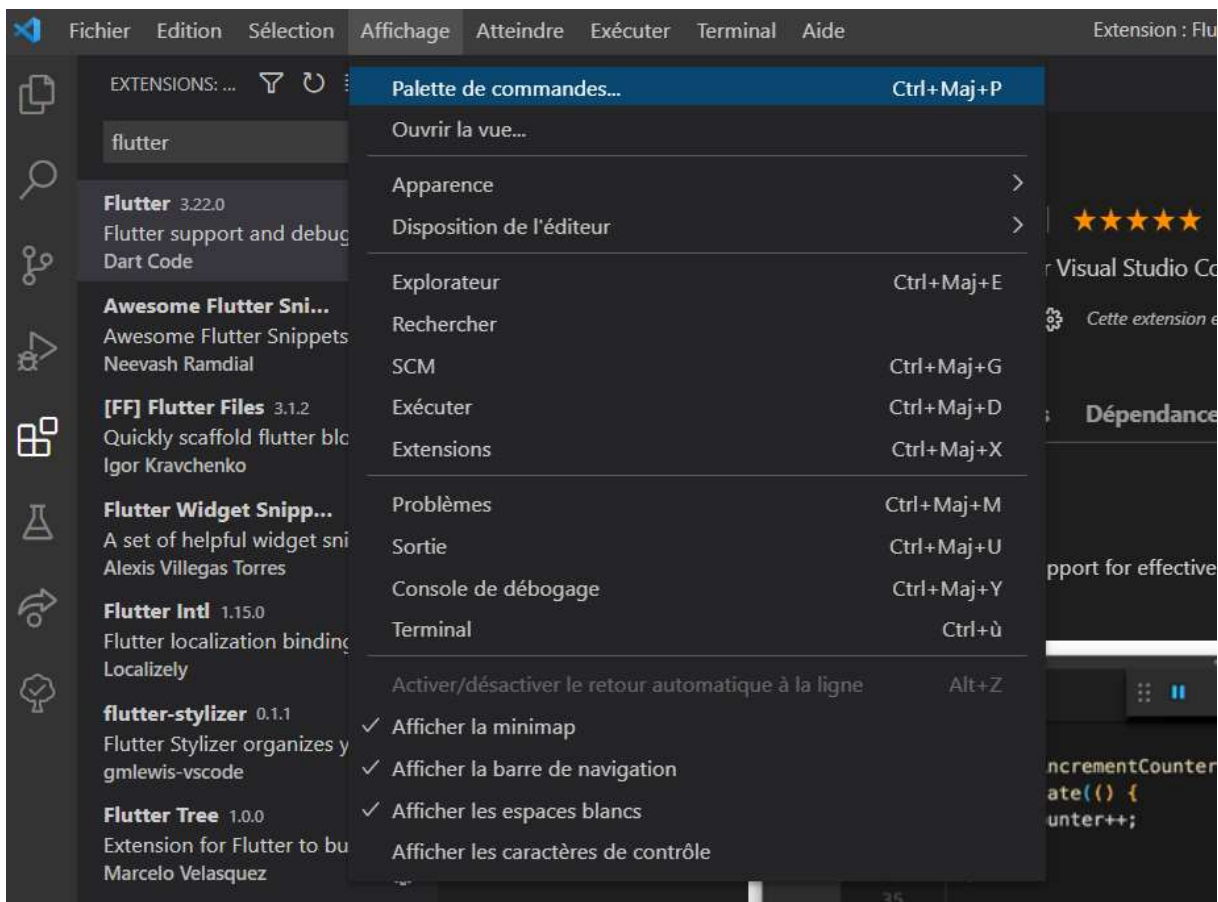
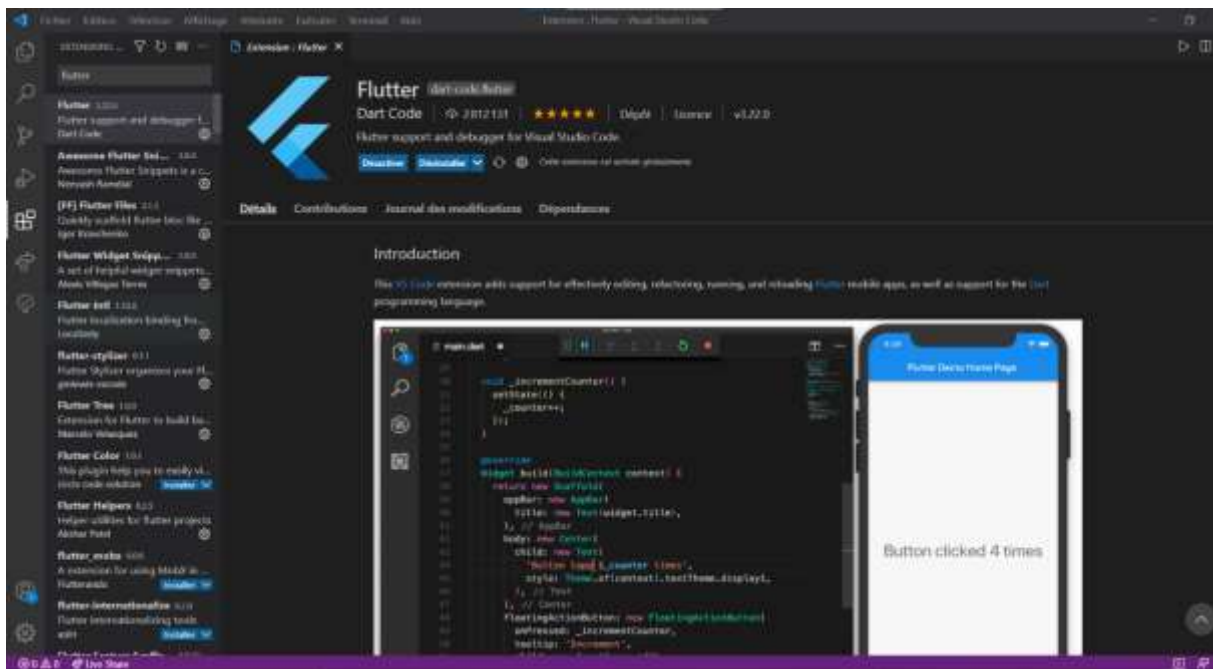


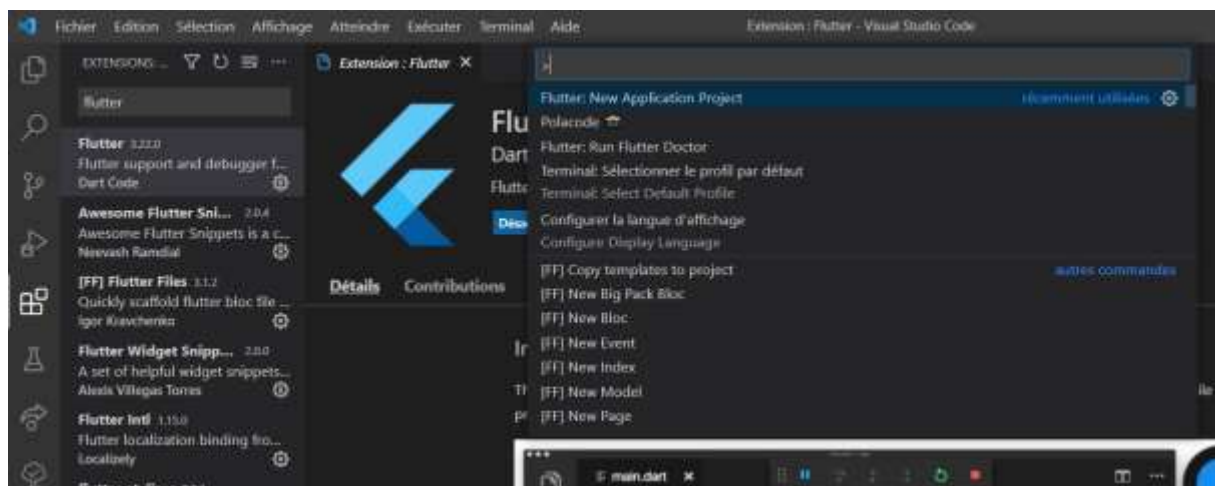
```
Administrateur : Invite de commandes - flutter doctor --android-licenses
Microsoft Windows [version 10.0.18363.836]
(c) 2019 Microsoft Corporation. Tous droits réservés.

C:\WINDOWS\system32>flutter doctor --android-licenses
Warning: File C:\Users\guill\android\repositories.cfg could not be loaded.
0 of 5 SDK package licenses not accepted. INFO: Computing updates...
Review licenses that have not been accepted (y/n)?
```

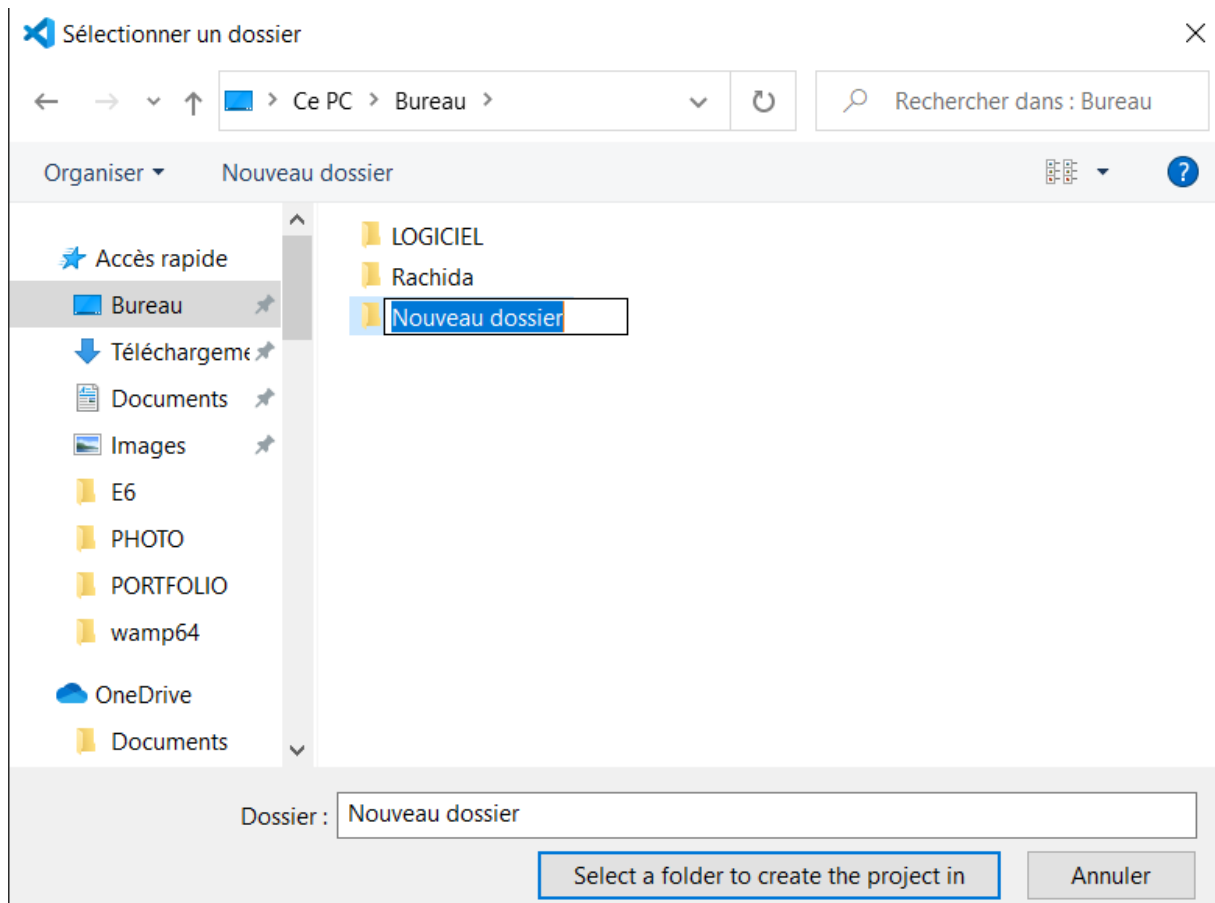

Visual Studio Code extension :

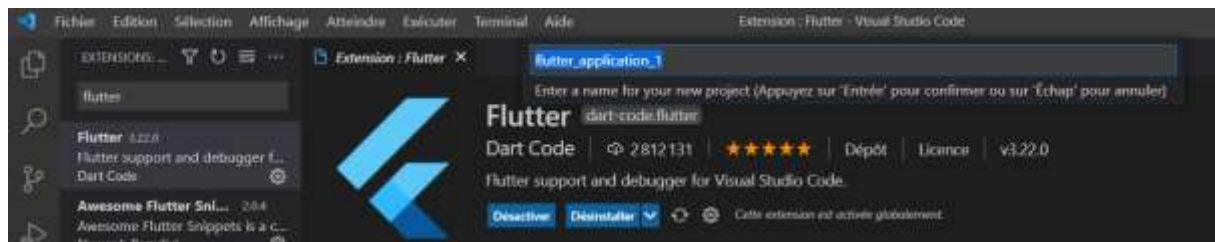
Extension Flutter :



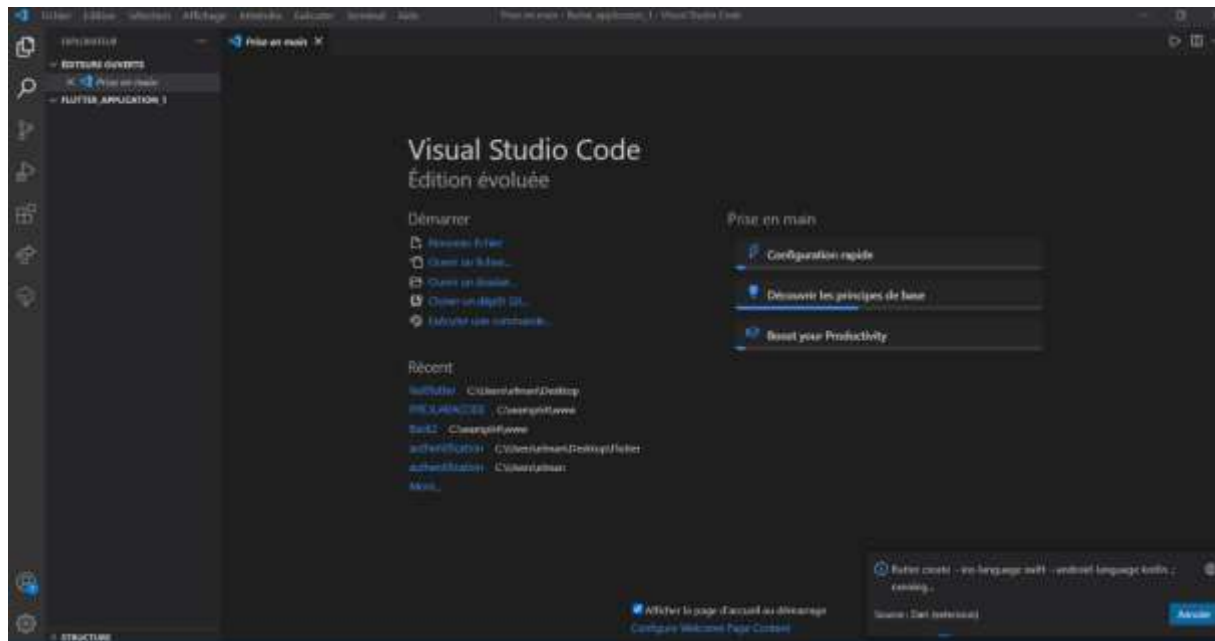


Création d'un nouveau dossier

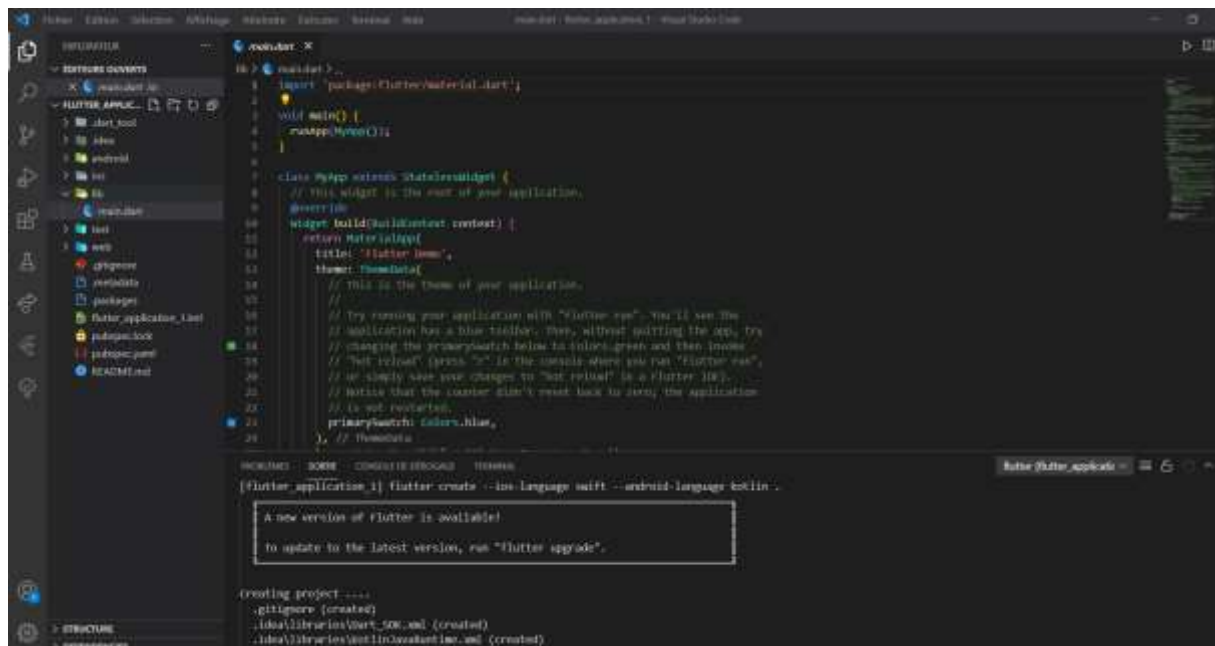




L'application sous Flutter se crée



L'application sous flutter est créée :



The screenshot shows an IDE interface with a project structure on the left and the main.dart file open in the editor. The project structure includes a flutter_application_1 directory with subdirectories like lib, test, and assets. The main.dart file contains the following code:

```
import 'package:flutter/material.dart';

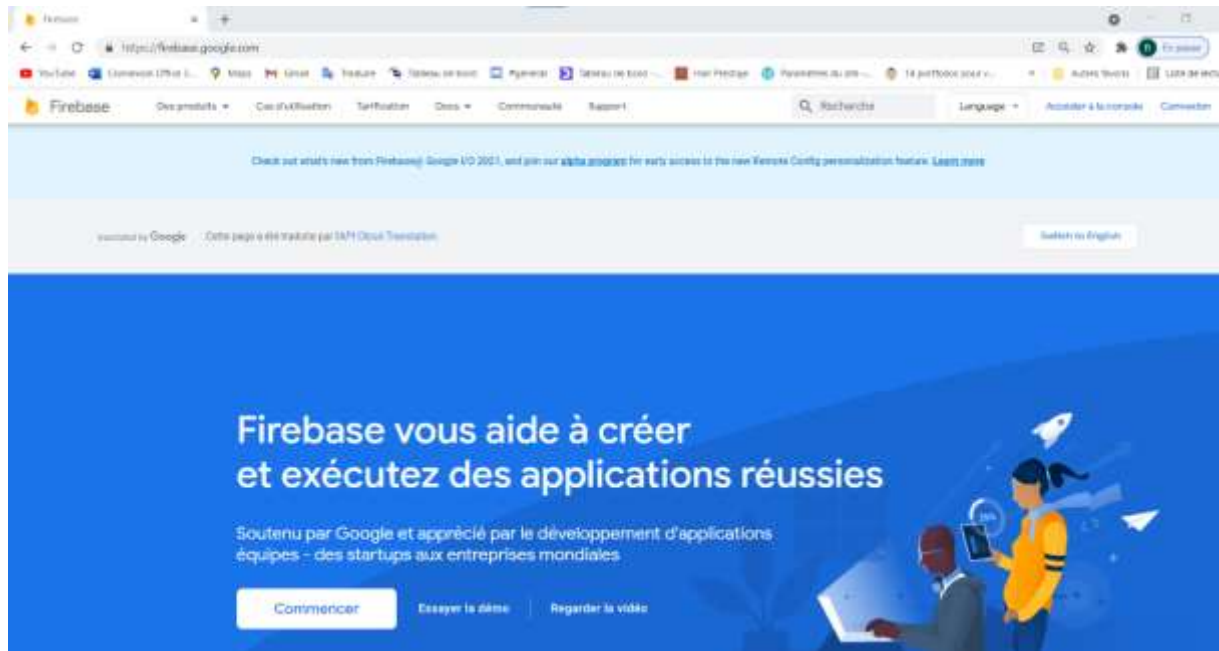
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        // This is the theme of your application.
        // Try running your application with "flutter run". You'll see the
        // application has a blue toolbar. Then, without quitting the app, try
        // changing the primarySwatch below to Colors.green and then invoke
        // "hot reload" (press "r" in the console where you run "flutter run",
        // or simply save your changes to "hot reload" in a Flutter IDE).
        // Notice that the counter didn't reset back to zero; the application
        // is not restarted.
        primarySwatch: Colors.blue,
      ), // This widget is the root of your application.
    );
```

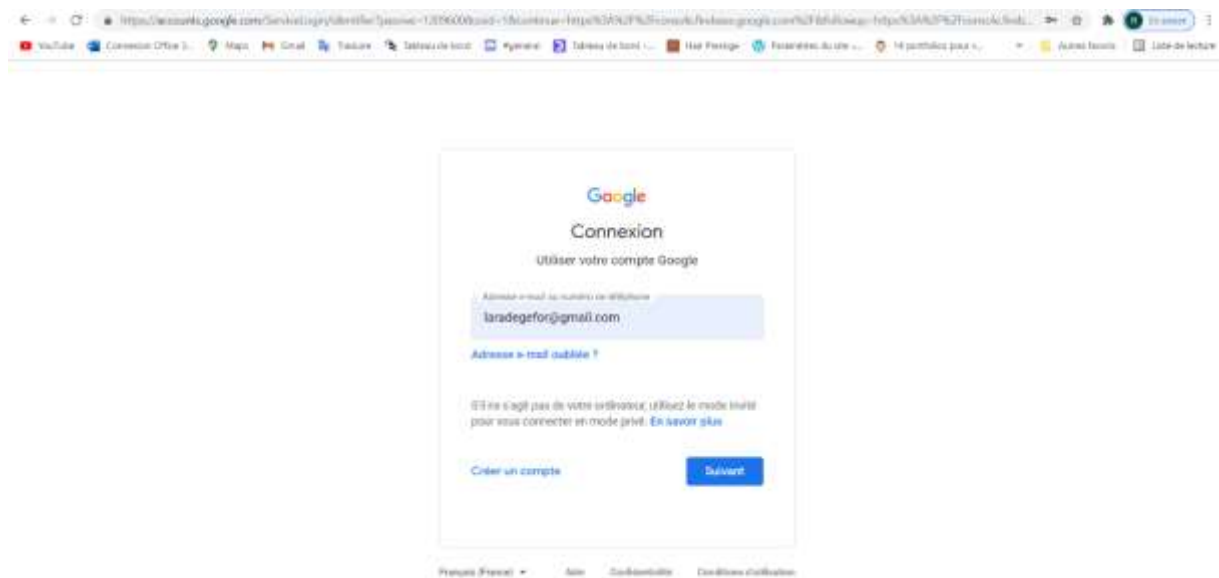
Below the code, the IDE shows the command line output for the flutter create command, indicating that the project was created successfully and providing instructions on how to run the application.

Installation et configuration de la Firebase :

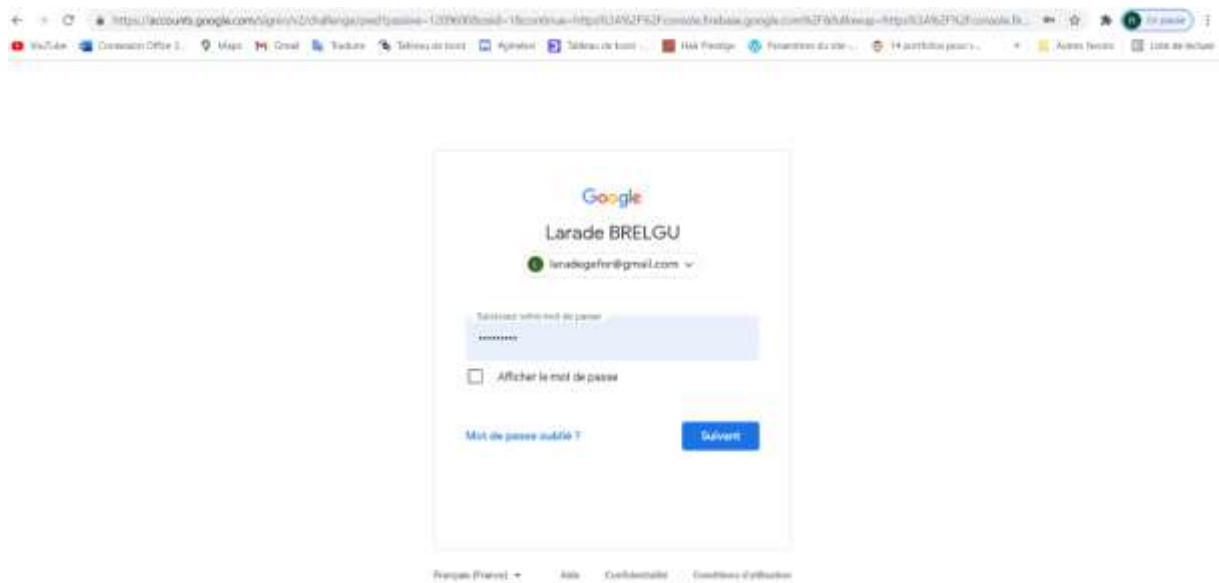
Site ➔ <https://firebase.google.com/>



Commencer ➔ s'enregistrer avec une adresse mail



Mettre le mot de passe :



The screenshot shows a web browser window with the URL <https://accounts.google.com/signin/2challenge/pwd?passive=1&continue=https://laradefirebase.firebaseio.com/&hl=fr&passive=1&continue=https://laradefirebase.firebaseio.com/&hl=fr>. The page displays the Google logo, the name "Larade BRELGU", and the email address "laradegiffr@gmail.com". Below this, there is a password input field with a strength indicator (four dots). A checkbox labeled "Afficher le mot de passe" is present. A link "Mot de passe oublié ?" is on the left, and a blue "Suivant" button is on the right. At the bottom, there are links for "Français (France)", "Aide", "Confidentialité", and "Conditions d'utilisation".

Créer un projet :



Cliquer sur "créer un projet" :

Créer un projet(Étape 1 sur 3)

Commençons par donner un nom à votre projet[®]

Saisissez le nom de votre projet

Le nom doit être unique et ne pas être trop long.

☐ J'accepte les [Conditions d'utilisation de Firebase](#)

Continuer

Nom du projet : Même nom que le projet Flutter

Créer un projet(Étape 1 sur 3)

Commençons par donner un nom à votre projet[®]

Nom du projet

flutterApplication1

flutterApplication1-94573

☒ J'accepte les [Conditions d'utilisation de Firebase](#)

Continuer

```
pubspec.yaml
name: flutter_application1
description: A new Flutter project.

# The following line prevents the package from being
# published using 'pub publish'. This is preferred
# because this package is intended for development use only.
publish_to: 'none' # Remove this line if you wish to publish to pub.dev

# The following defines the version and build number
# A version number is required, a build number is optional
# both the version and the build number must be
# build by specifying a build number and a build
# In Android, build-number is used as version-name
# In iOS, build-number is used as version-string
# For more information on the versioning system, see
# https://pub.dev/packages/flutter/versioning
version: 1.0.0+1

environment:
  sdk: '>=2.12.0 <3.0.0'

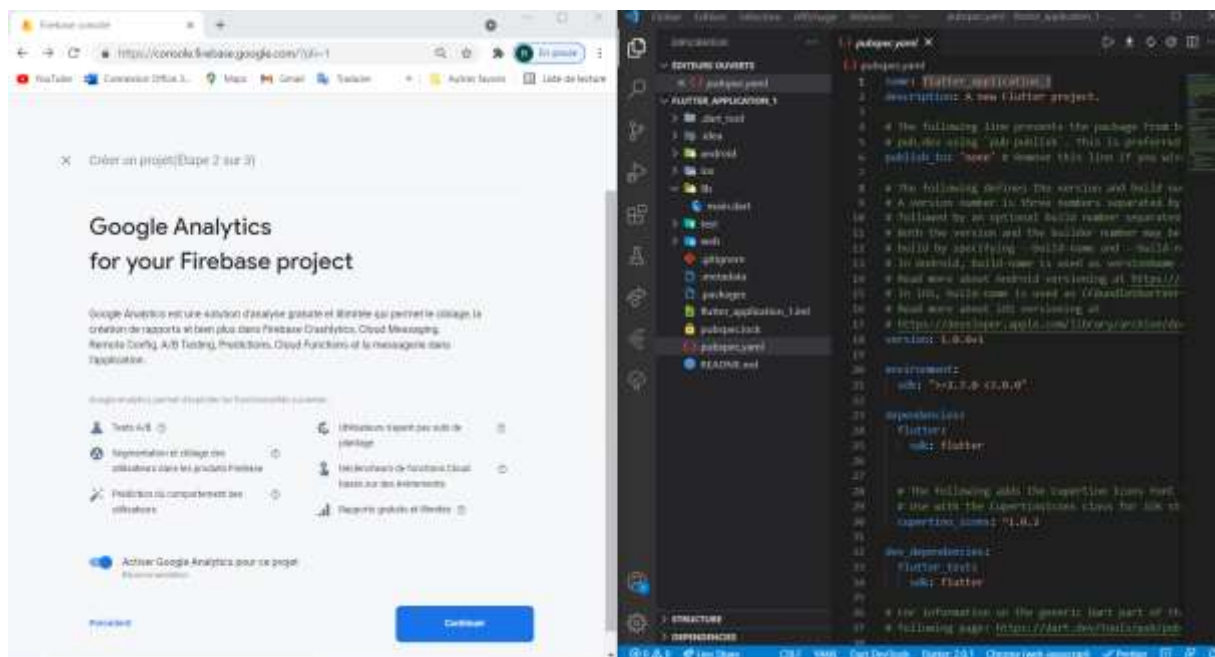
dependencies:
  flutter:
    sdk: flutter

# The following adds the Cupertino icon font to
# the assets directory. You can use the Cupertino class for iOS
cupertino_icons: ^1.0.2

dev_dependencies:
  flutter_test:
    sdk: flutter

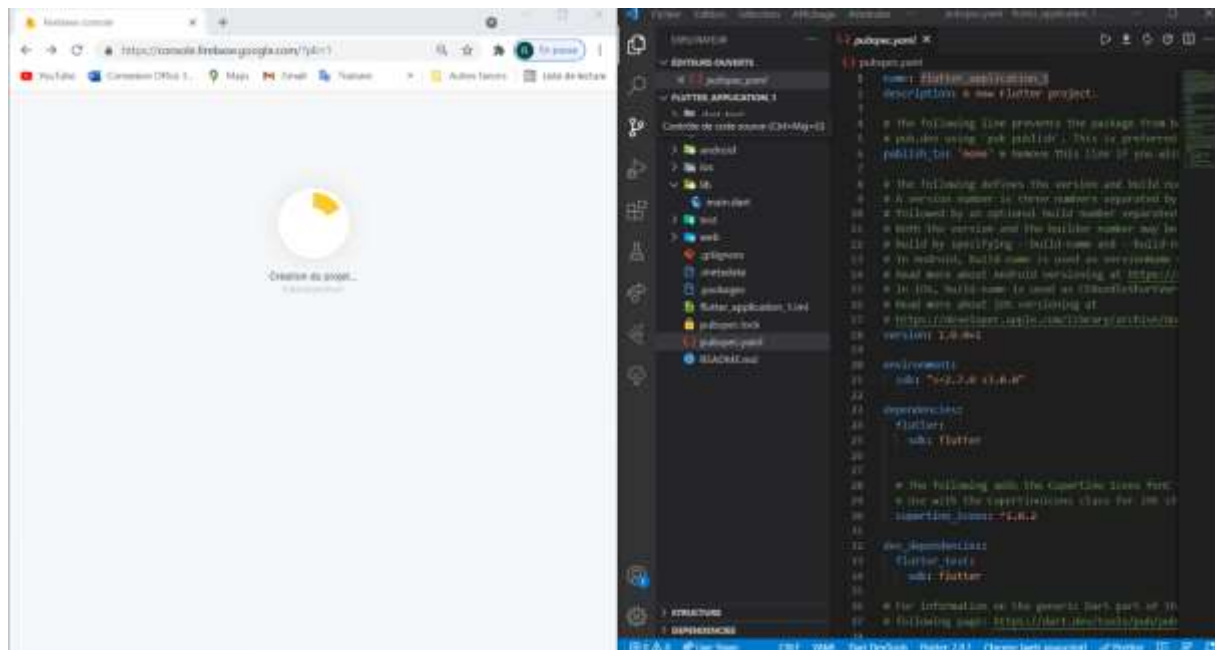
# For information on the generic Dart part of this
# file see the following page: https://dart.dev/doc/tooling
```


Afin de faire des statistiques, possibilité d'activer Google Analytics

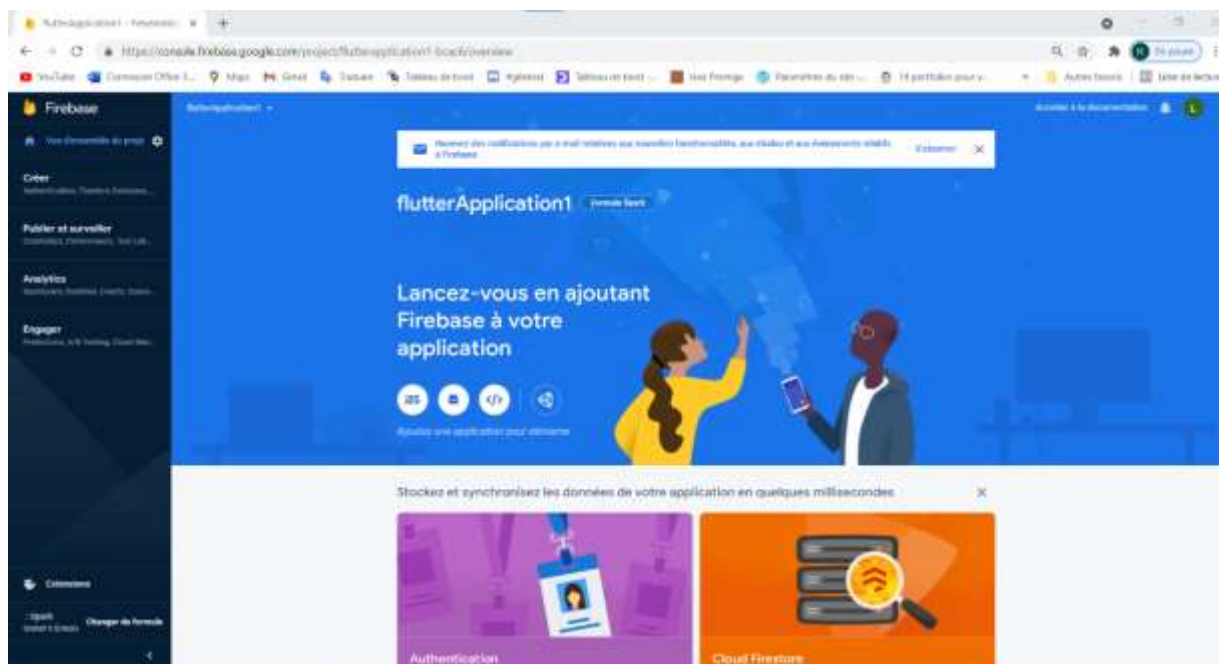


Si pas de Google Analytics ➔ cliquer sur continuer

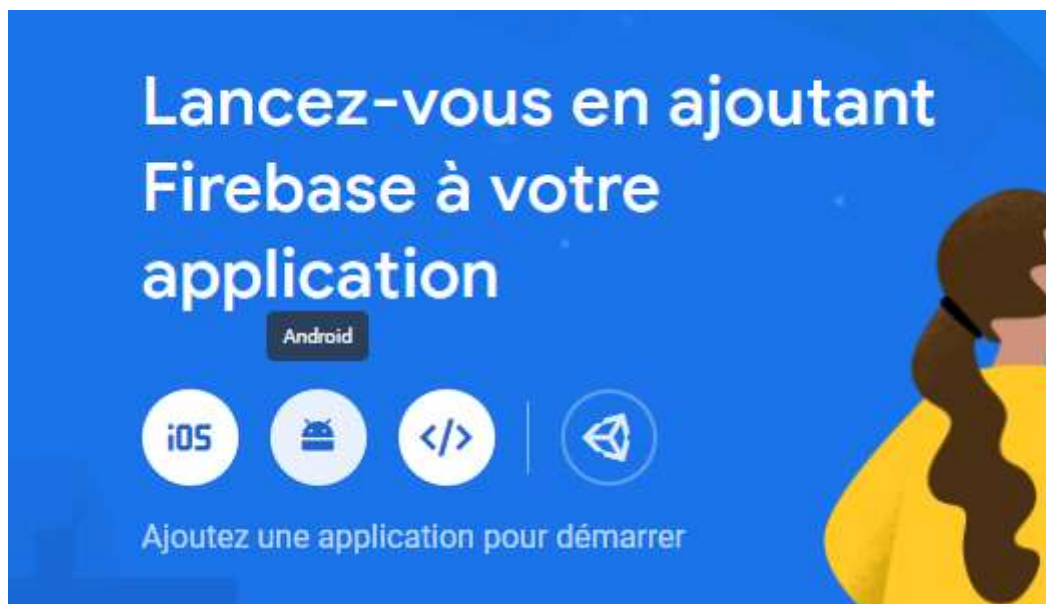
Création du projet :

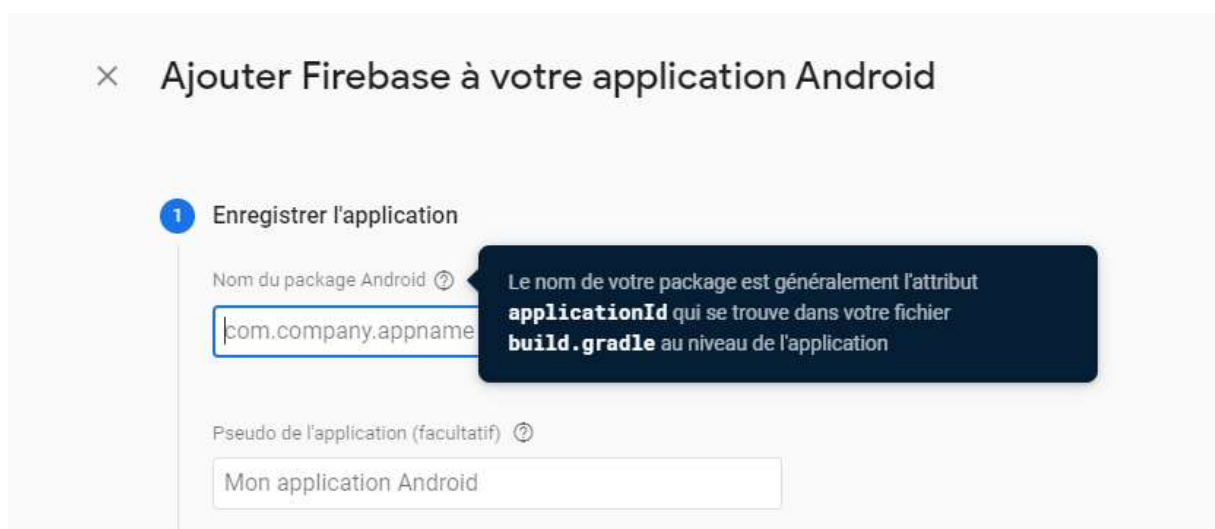
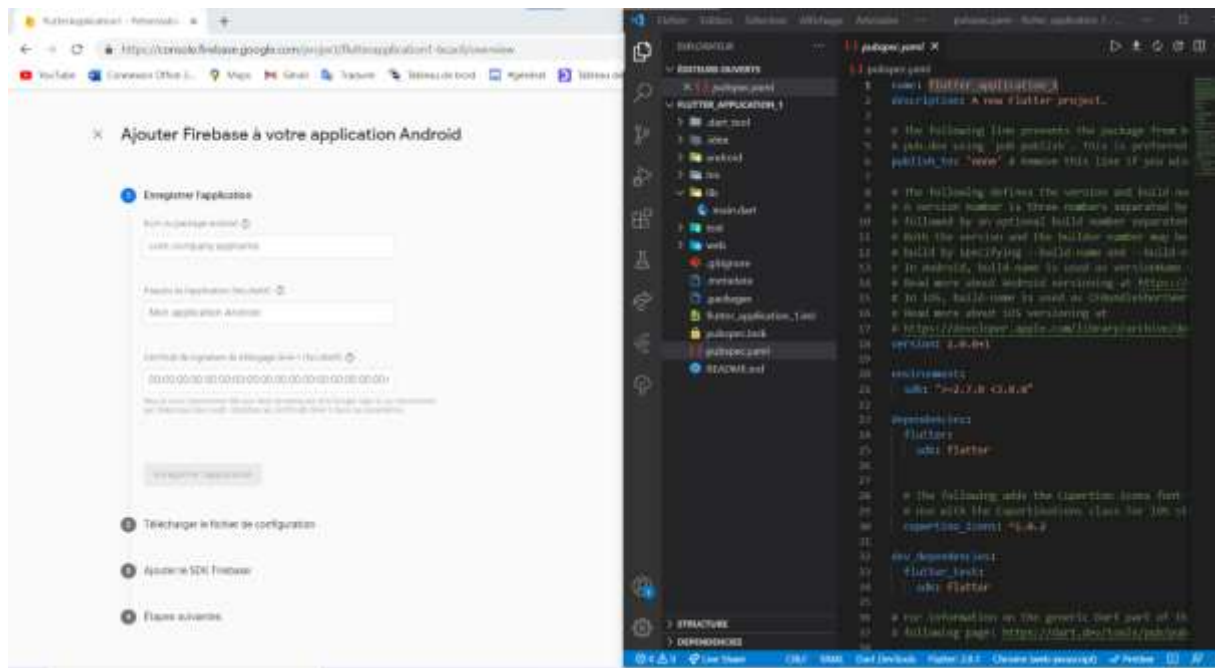


Le projet est créé



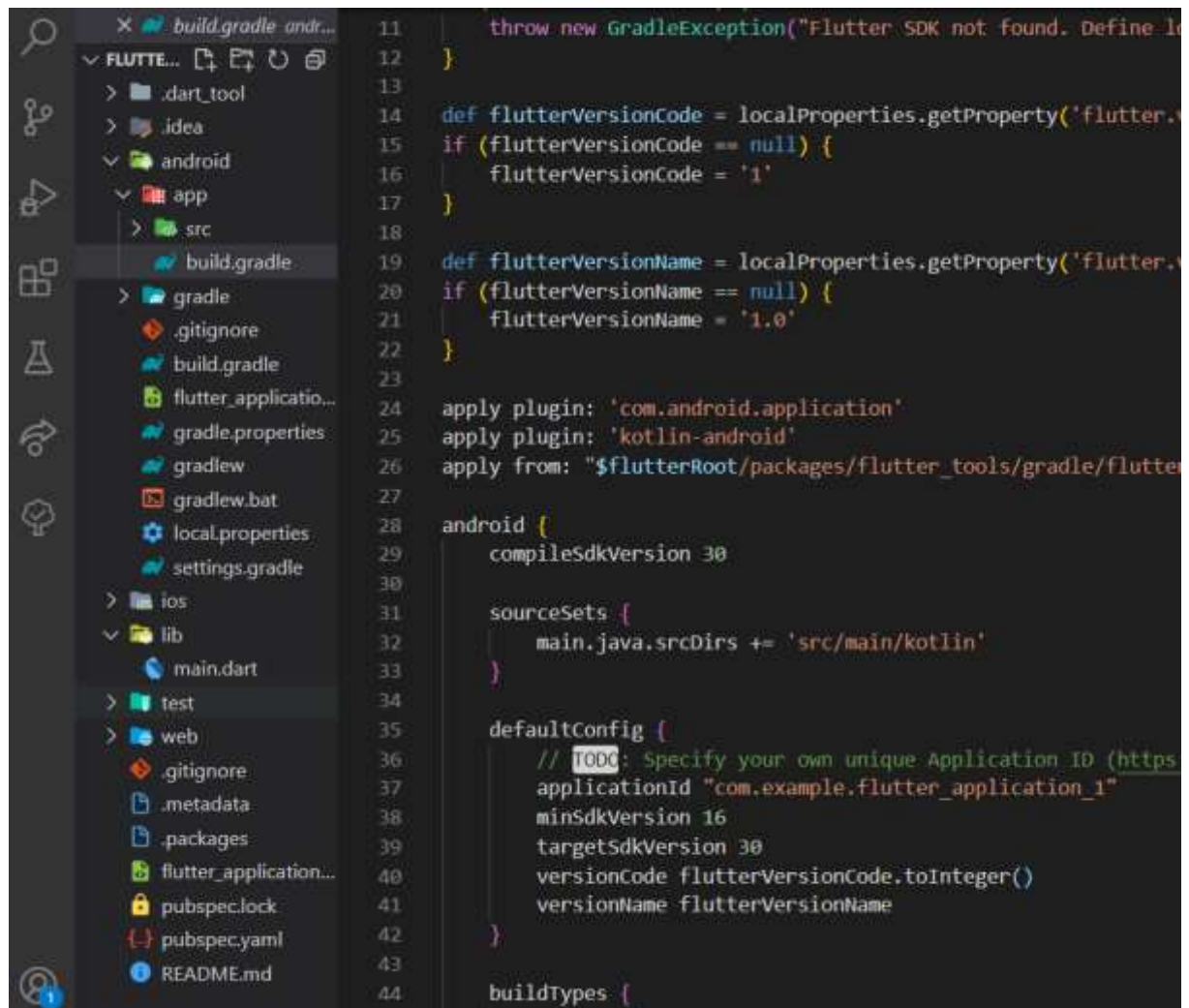
Cliquer sur Android





Aller sur Visual Studio Code, sur la liste des fichiers :

Android → App → Build.gradle : rechercher l'applicationId et l'indiquer sur la page de Firebase



```
11 |         throw new GradleException("Flutter SDK not found. Define lo
12 |     }
13 |
14 |     def flutterVersionCode = localProperties.getProperty('flutter.'
15 |     if (flutterVersionCode == null) {
16 |         flutterVersionCode = '1'
17 |     }
18 |
19 |     def flutterVersionName = localProperties.getProperty('flutter.'
20 |     if (flutterVersionName == null) {
21 |         flutterVersionName = '1.0'
22 |     }
23 |
24 |     apply plugin: 'com.android.application'
25 |     apply plugin: 'kotlin-android'
26 |     apply from: "$flutterRoot/packages/flutter_tools/gradle/flutter
27 |
28 |     android {
29 |         compileSdkVersion 30
30 |
31 |         sourceSets {
32 |             main.java.srcDirs += 'src/main/kotlin'
33 |         }
34 |
35 |         defaultConfig {
36 |             // TODO: Specify your own unique Application ID (https
37 |             applicationId "com.example.flutter_application_1"
38 |             minSdkVersion 16
39 |             targetSdkVersion 30
40 |             versionCode flutterVersionCode.toInteger()
41 |             versionName flutterVersionName
42 |         }
43 |
44 |         buildTypes {
```


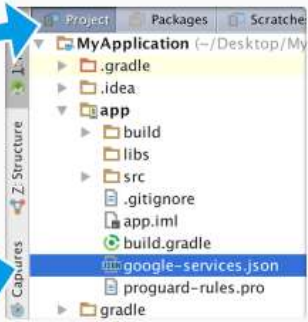
Cliquer sur "Enregistrer l'application"

flutterApplication1 - Présentation x +

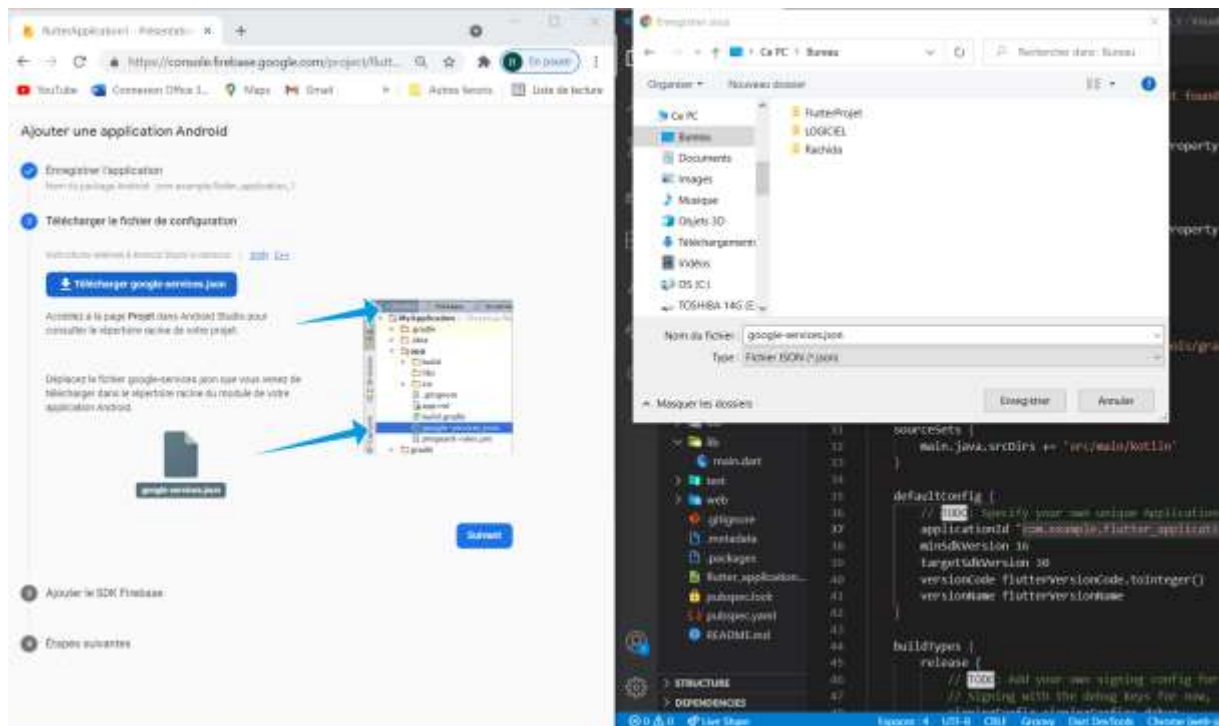
← → ↻ 🔒 https://console.firebase.google.com/project/flutt... 🔍 ☆ ⚙️ R En pause ⋮

YouTube Connexion Office 3... Maps Gmail » Autres favoris Liste de lecture

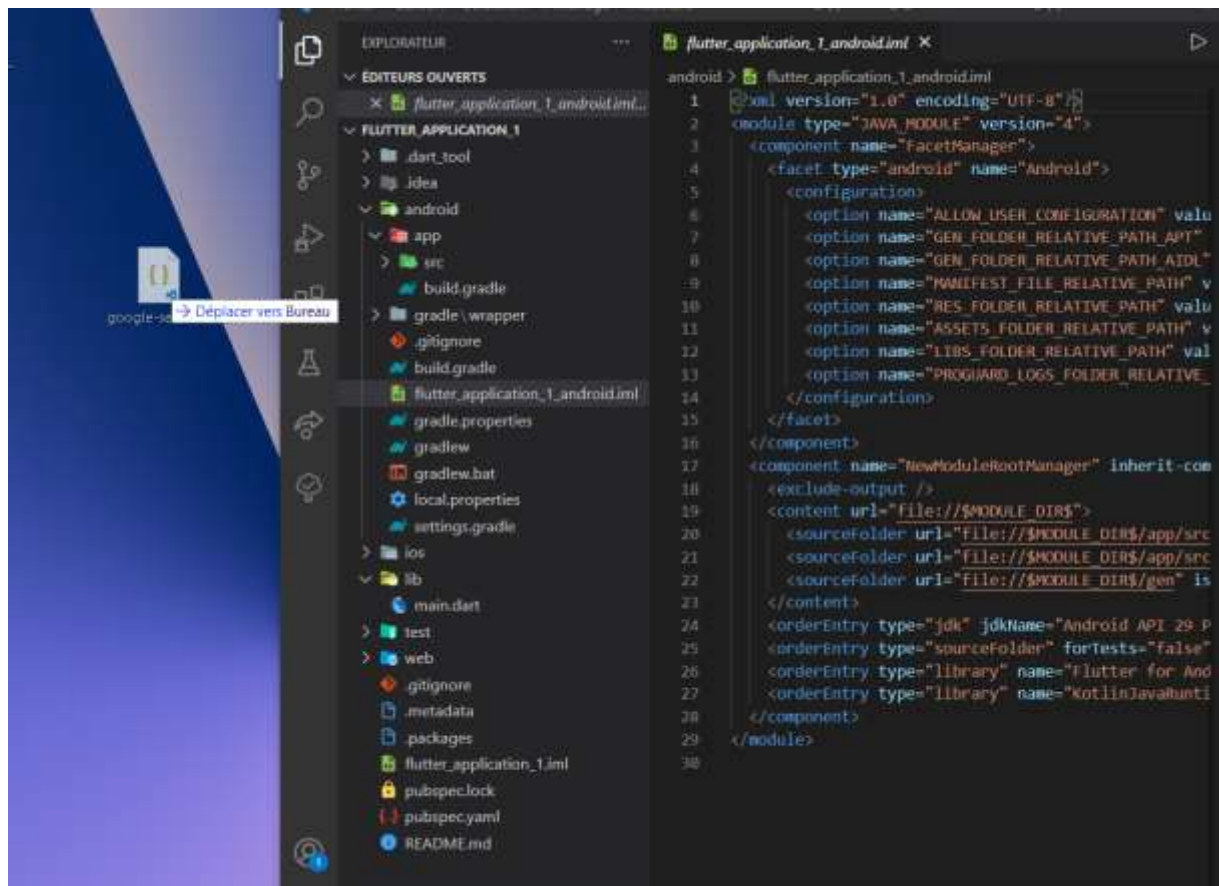
Ajouter une application Android

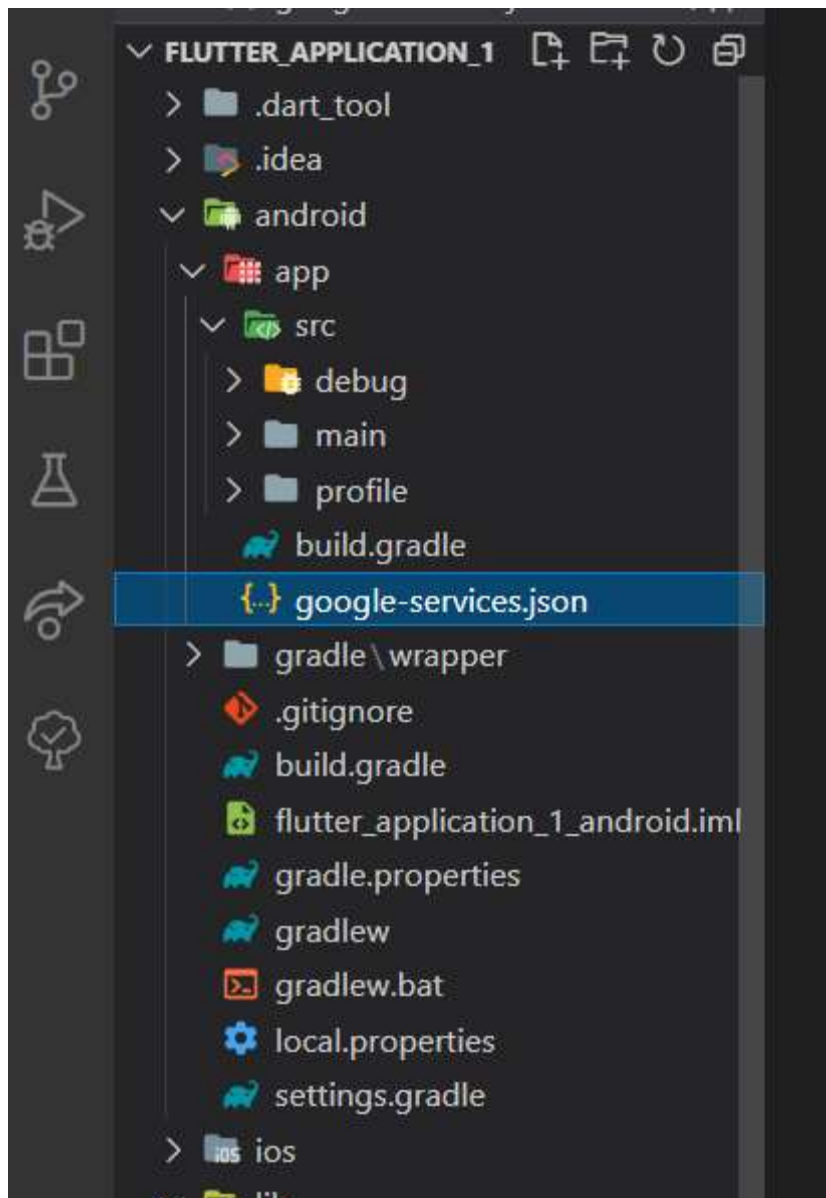
- ✓ Enregistrer l'application
Nom du package Android : com.example.flutter_application_1
- 2 Télécharger le fichier de configuration
Instructions relatives à Android Studio ci-dessous | [Unity](#) [C++](#)
[Télécharger google-services.json](#)
Accédez à la page **Projet** dans Android Studio pour consulter le répertoire racine de votre projet.
Déplacez le fichier google-services.json que vous venez de télécharger dans le répertoire racine du module de votre application Android.

[Suivant](#)
- 3 Ajouter le SDK Firebase
- 4 Étapes suivantes

Télécharger → enregistrer le fichier afin de le mettre ensuite dans le projet flutter dans Visual studio code

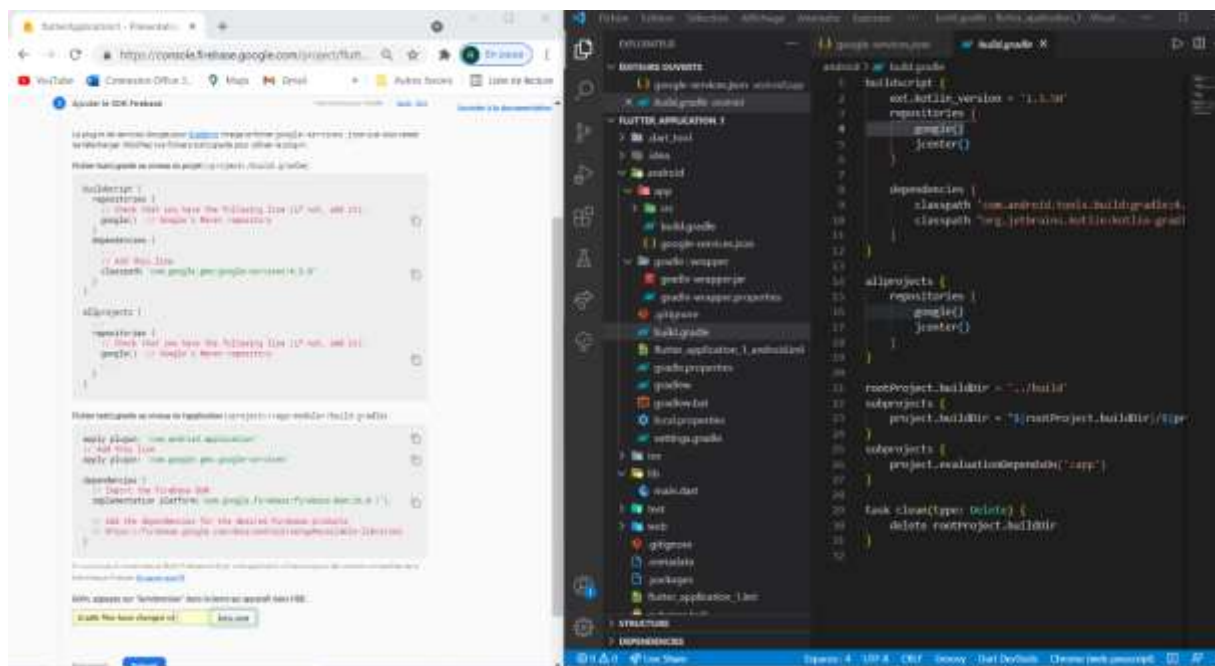


Mettre le fichier enregistré dans fichier App



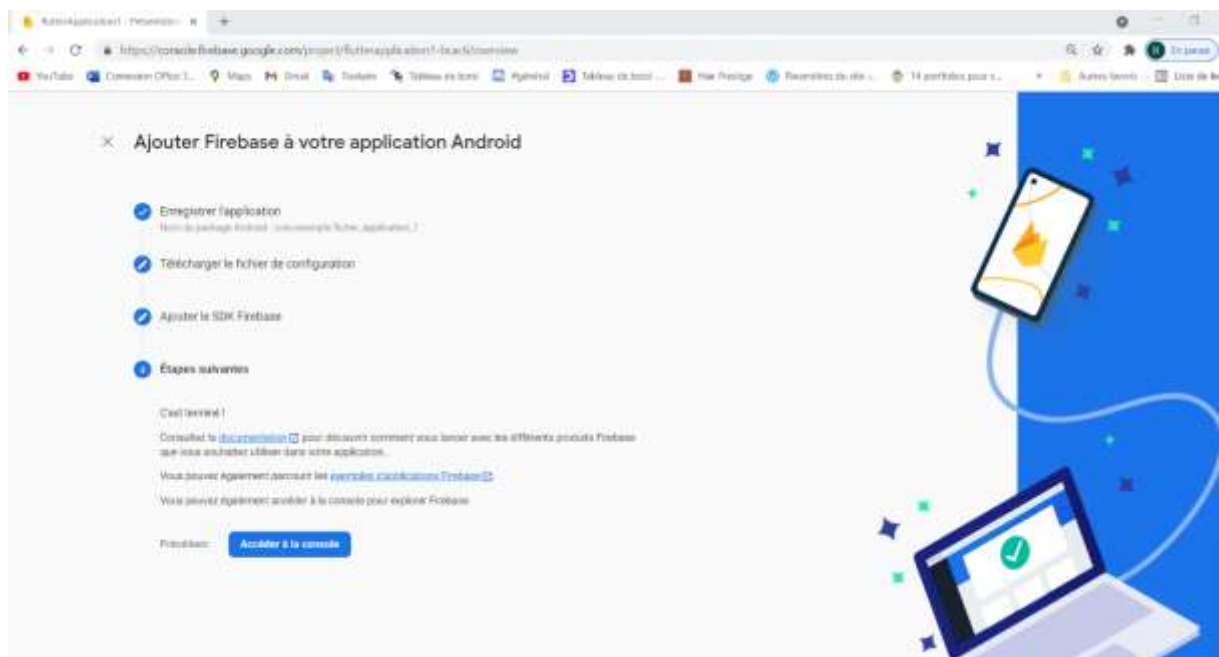


Ajouter le SDK Firebase dans le projet sur Visual Studio Code :

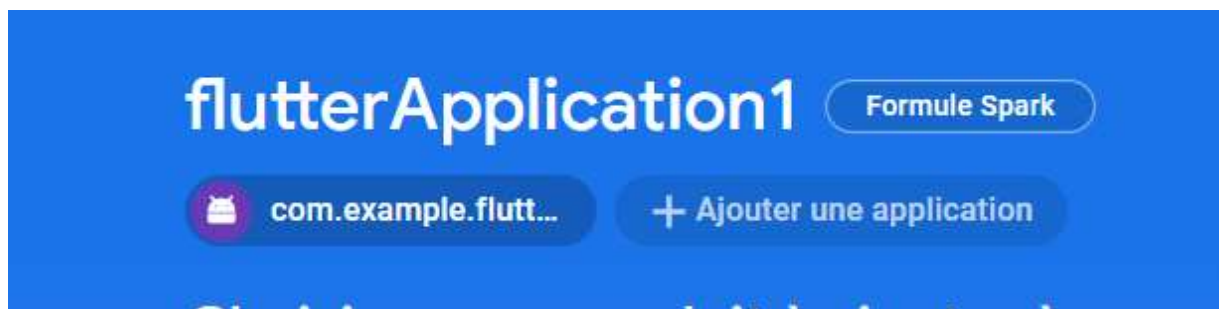


Ajouter les informations mentionnées par le site de Firebase

Cliquer sur "Suivant"



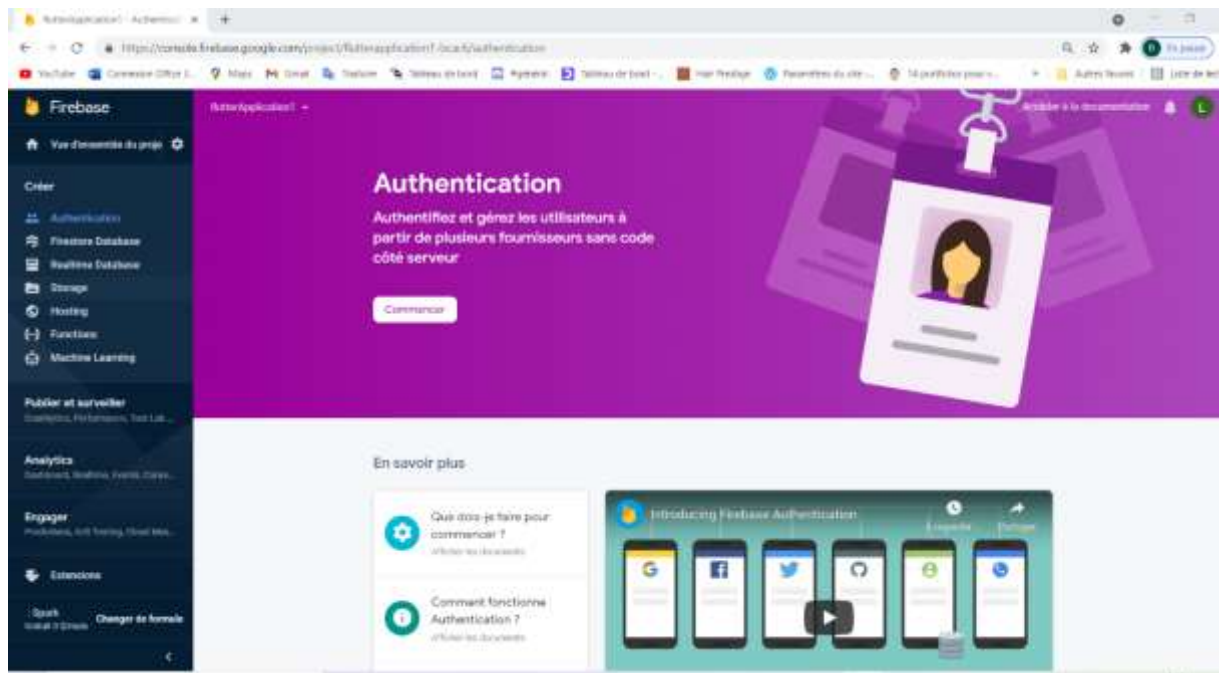
L'application est maintenant en lien avec le projet Flutter de Visual Studio Code



Sur la gauche de l'écran : onglet "Créer " ➔ Authentification

L'authentification :

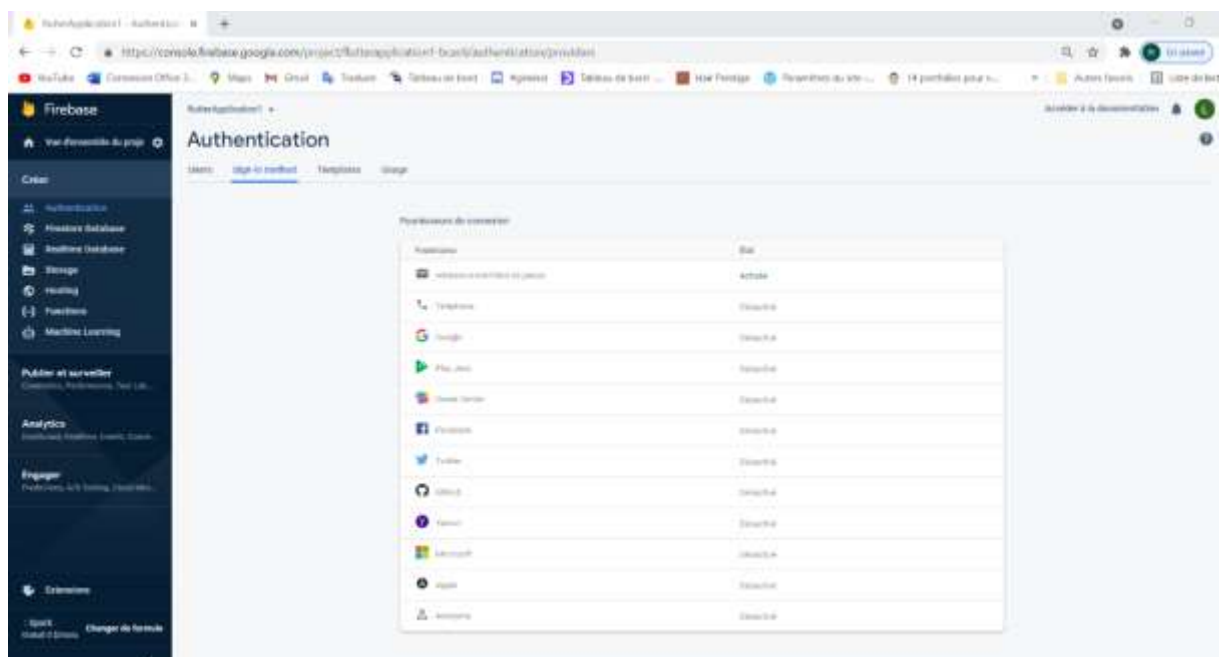
Cela permet de déterminer comment l'authentification sera effectuée sur l'application



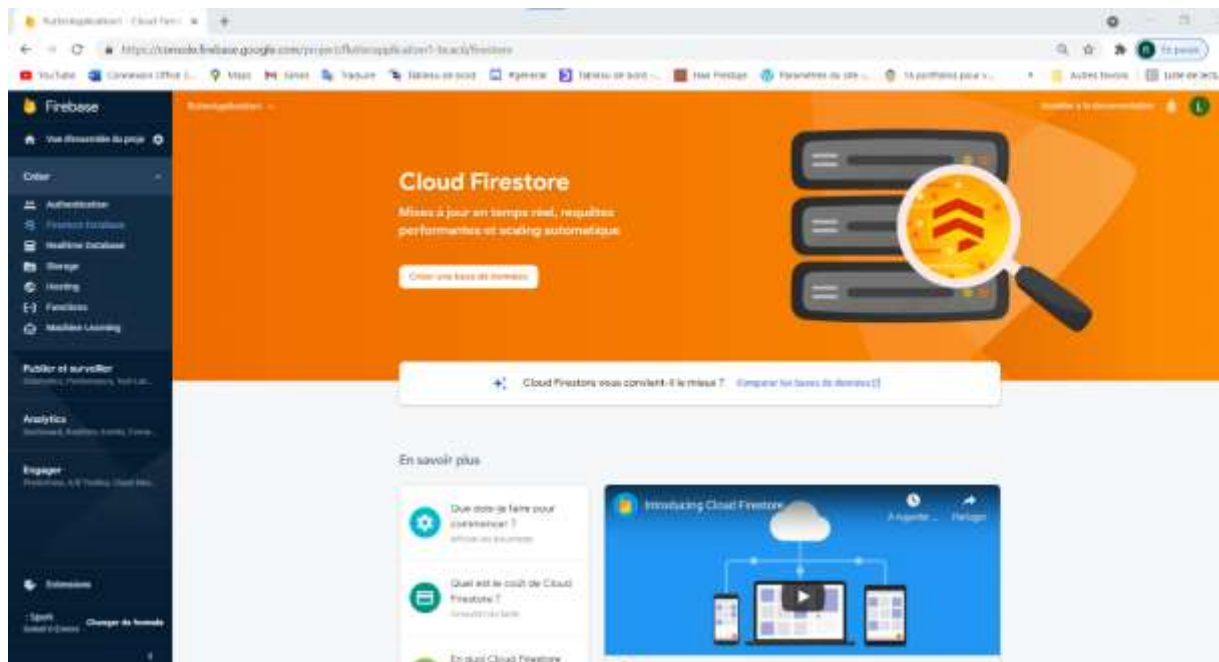
Cliquer sur commencer

Et dans l'onglet "Sign-in-method" :

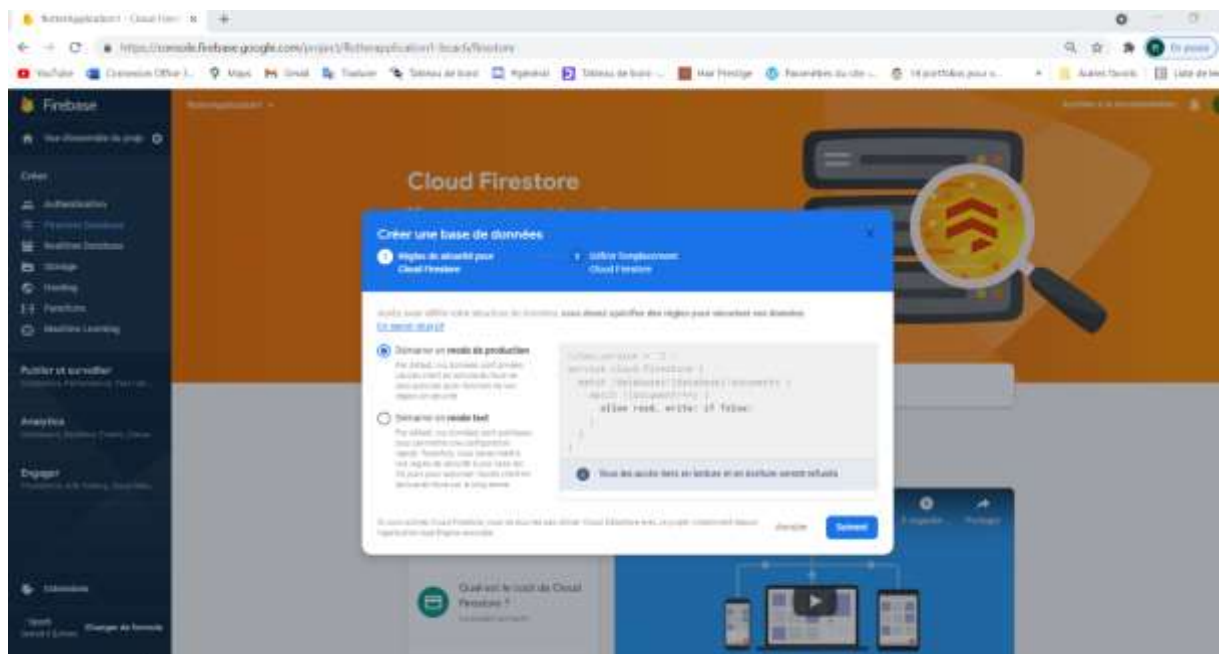
Activer : Adresse e-mail/Mot de passe



Sur la gauche de l'écran : onglet Créer ➔ Firestore Database



Configuration de Firestore database



La base de données est prête : cette dernière sera alimentée par le code réalisé sur le projet via Visual Studio Code

