

# Développement Mobile

Préparation de l'environnement  
de développement mobile et résolution des  
problèmes d'installation courants



2022 - Alexandre Leroux (alex@sherpa.one)

# Développement Android



# ENVIRONNEMENT DE DÉV. MOBILE

Pour développer une application mobile **Android**, il faut :

- un PC avec **Windows** ou **MacOS** ou **Linux** ou **ChromeOS**,
- **Android Studio** (ou Android SDK),

<https://developer.android.com/studio>





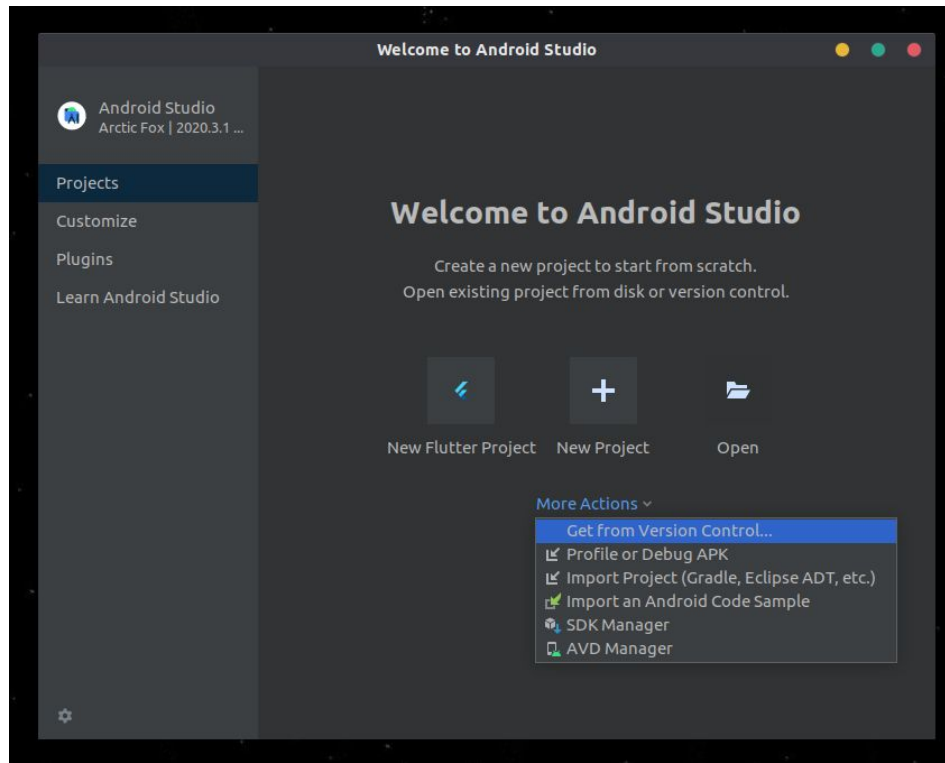
Android Studio provides the fastest tools for building apps on every type of Android device.

**Download Android Studio**

2020.3.1 for Linux 64-bit (935 MiB)

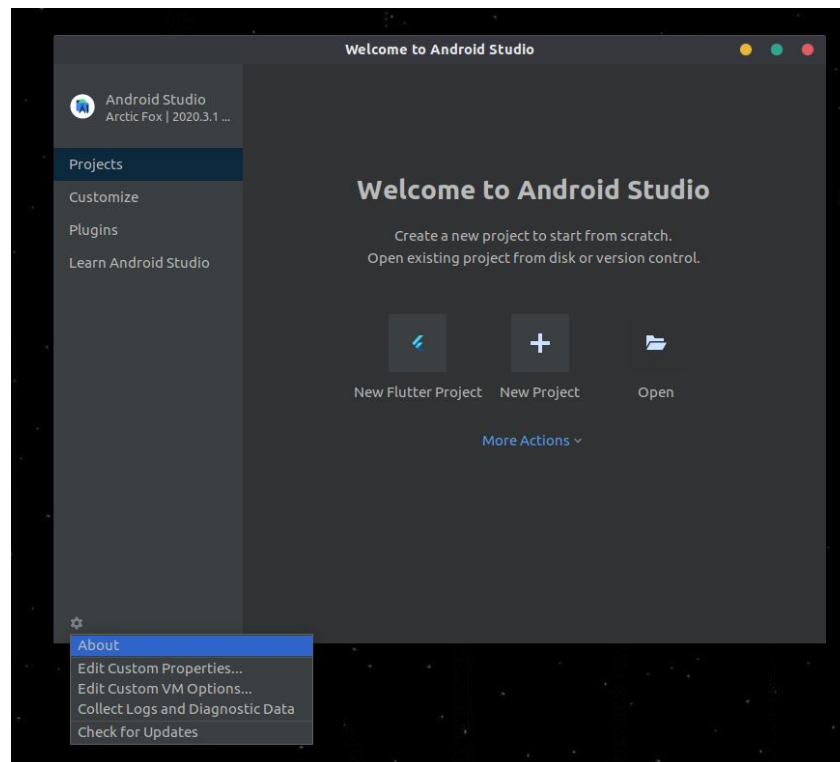
<https://developer.android.com/studio>





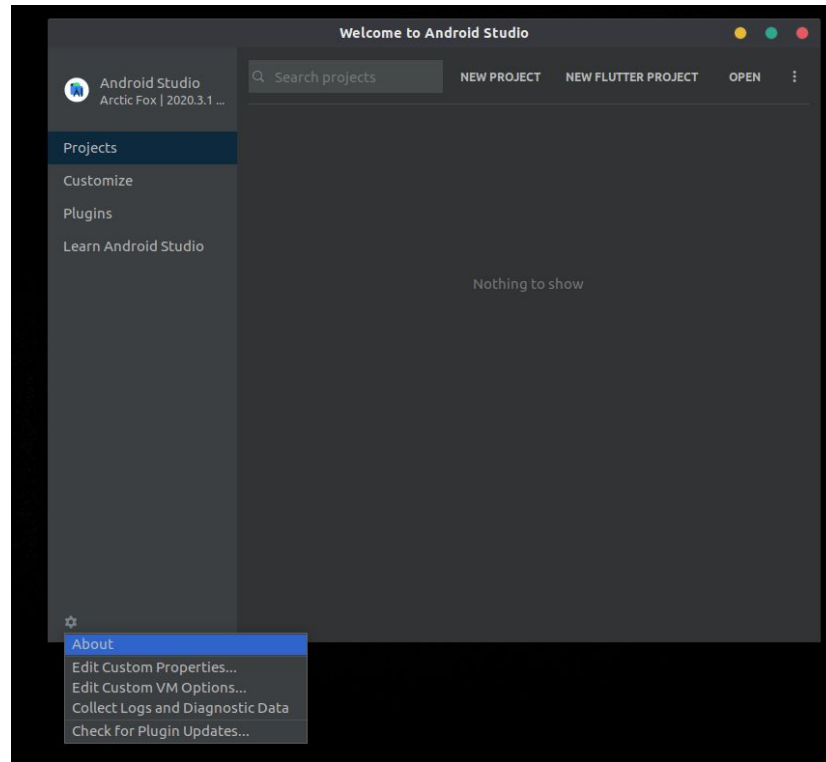
Le lien "More Actions" donne accès  
à AVD Manager et SDK Manager





L'icône "rouage" donne accès  
aux mises à jour d'Android Studio



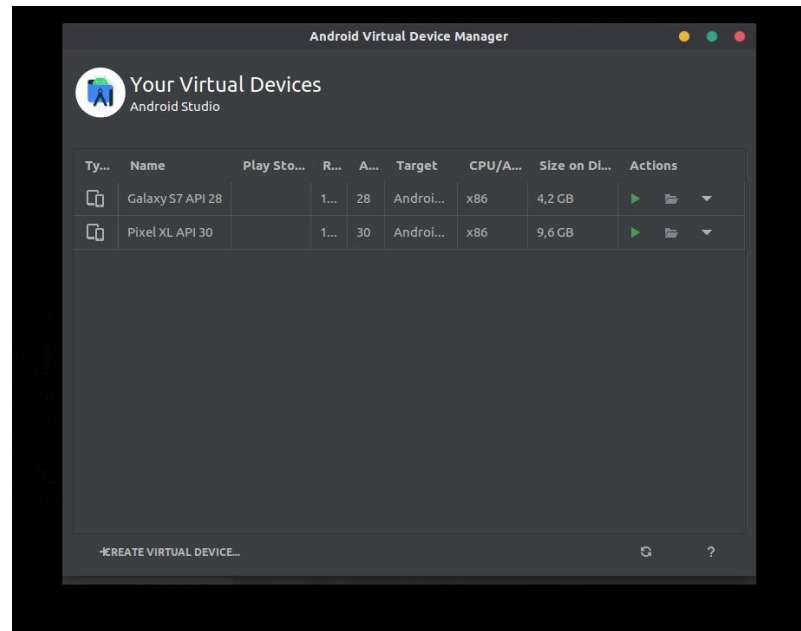


**Accès aux configurations d'Android Studio**

# ENVIRONNEMENT DE DÉV. MOBILE

- un émulateur représentatif du ou des smartphones / tablettes visé(es)
- le modèle d'émulateur "Pixel" le plus récent

<https://developer.android.com/studio/run/managing-avds>







## Select Hardware

### Choose a device definition



Category	Name	Play S...	Size	Resol...	Density
TV	Pixel XL		5,5"	1440x...	560dpi
Phone	Pixel 5		6,0"	1080x...	440dpi
Wear OS	Pixel 4a		5,8"	1080x...	440dpi
Tablet	Pixel 4 XL		6,3"	1440x...	560dpi
Automot...	Pixel 4	▶	5,7"	1080x...	440dpi
	Pixel 3a XL		6,0"	1080x...	400dpi
	Pixel 3a	▶	5,6"	1080x...	440dpi

### Pixel 5



Size: large  
Ratio: long  
Density: 440dpi

NEW HARDWARE PROFILE

IMPORT HARDWARE PROFILES

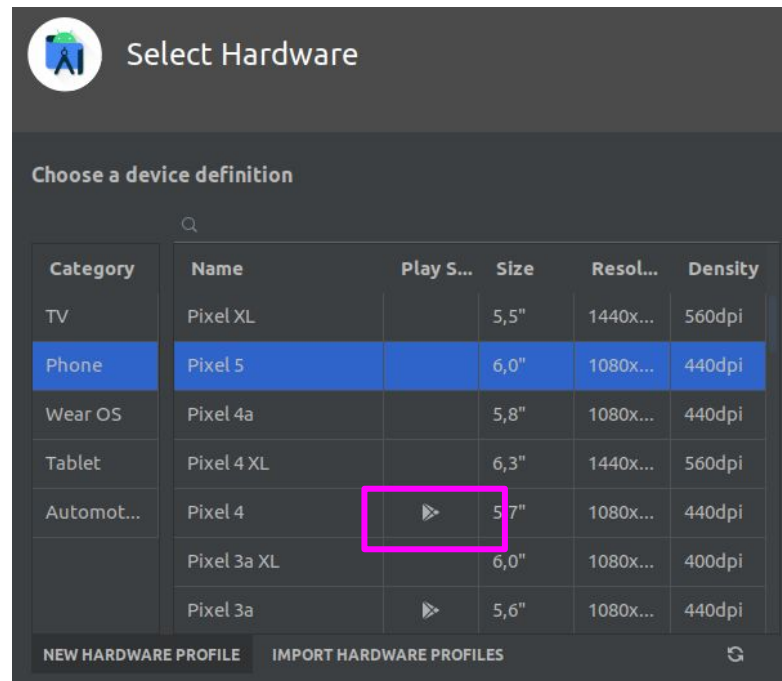


CLONE DEVICE...



# ENVIRONNEMENT DE DÉV. MOBILE

Il est possible de télécharger une application depuis Google Play Store sur l'émulateur.



# ENVIRONNEMENT DE DÉV. MOBILE

En cas de dysfonctionnement de l'émulateur :  
vérifier que le numéro de version du SDK Android nécessaire au  
fonctionnement de l'émulateur est bien téléchargé (cf. SDK  
Manager dans Android Studio).

Si ce n'est pas le cas, installer la version du SDK nécessaire, ou  
créez un nouvel émulateur compatible avec la version du SDK  
Android déjà disponible.



# ENVIRONNEMENT DE DÉV. MOBILE

Si le projet ne se lance pas sur l'émulateur avec la commande :

```
flutter run
```

tentez :

```
flutter run --enable-software-rendering
```

<https://developer.android.com/studio/run/emulator-acceleration>



# Tests sur devices



# ENVIRONNEMENT DE DÉV. MOBILE

Il est recommandé de tester un projet d'application mobile directement sur device et pas uniquement sur émulateur,

- Pour un projet **Android** :  
vous devez disposer d'un **smartphone** ou d'une **tablette**  
disposant d'une version d'Android récente  
+ **Android Studio** (à jour)  
sur **PC Windows / MacOS / Linux**,



# ENVIRONNEMENT DE DÉV. MOBILE

- Pour un projet **iOS** :  
vous devez disposer d'un **iPhone** ou d'un **iPad** disposant  
d'une **version iOS récente**  
+ **MacOS récent** + **Xcode récent**



# ENVIRONNEMENT DE DÉV. MOBILE

Il est nécessaire d'**activer le mode développeur** sur le smartphone ou la tablette et **autoriser le débogage USB**.

Sur **Android**, l'activation du mode développeur **peut différer selon le fabricant du device**

[https://www.frandroid.com/comment-faire/tutoriaux/184906\\_comment-acceder-au-mode-developpeur-sur-android](https://www.frandroid.com/comment-faire/tutoriaux/184906_comment-acceder-au-mode-developpeur-sur-android)



# ENVIRONNEMENT DE DÉV. MOBILE

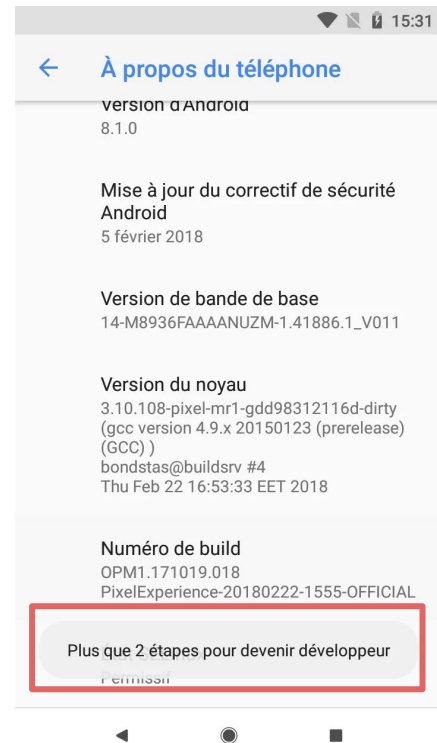
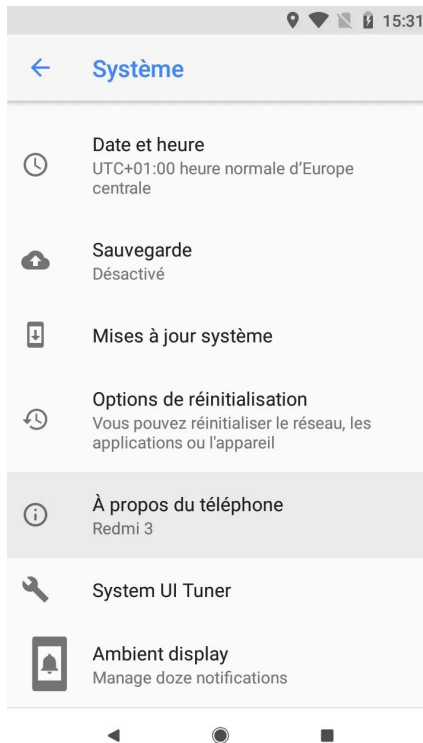
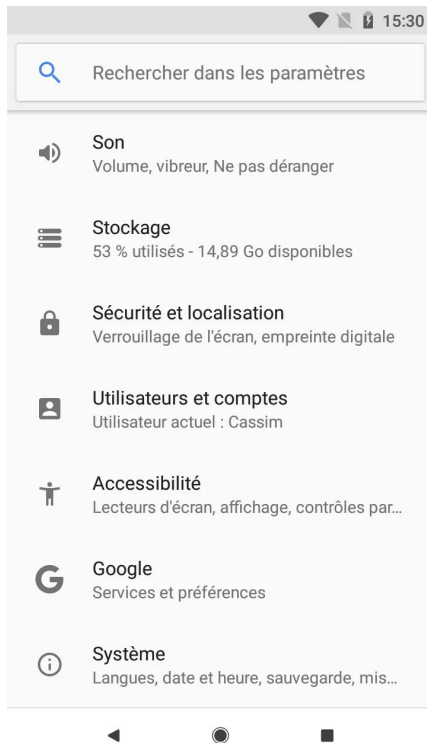
Sur **iOS**, l'activation du mode développeur est **identique pour tous les devices**.



# Activation du mode développeur sur un device Android



android



Taper plusieurs fois sur le numéro de build jusqu'à l'activation du mode développeur

Activer le mode développeur sur un device Android



## Activation du mode développeur sur un terminal Android

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

### Enable developer options and USB debugging

On Android 4.1 and lower, the **Developer options** screen is available by default. On Android 4.2 and higher, you must enable this screen. To enable developer options, tap the **Build Number** option 7 times. You can find this option in one of the following locations, depending on your Android version:

- Android 9 (API level 28) and higher: **Settings > About Phone > Build Number**
- Android 8.0.0 (API level 26) and Android 8.1.0 (API level 26): **Settings > System > About Phone > Build Number**
- Android 7.1 (API level 25) and lower: **Settings > About Phone > Build Number**

At the top of the **Developer options** screen, you can toggle the options on and off (figure 1). You probably want to keep this on. When off, most options are disabled except those that don't require communication between the device and your development computer.

Before you can use the debugger and other tools, you need to enable USB debugging, which allows Android Studio and other SDK tools to recognize your device when connected via USB. To enable USB debugging, toggle the **USB debugging** option in the Developer Options menu. You can find this option in one of the following locations, depending on your Android version:

- Android 9 (API level 28) and higher: **Settings > System > Advanced > Developer Options > USB debugging**
- Android 8.0.0 (API level 26) and Android 8.1.0 (API level 26): **Settings > System > Developer Options > USB debugging**
- Android 7.1 (API level 25) and lower: **Settings > Developer Options > USB debugging**

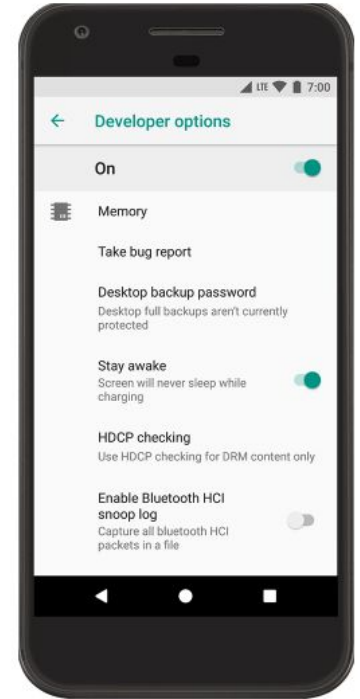


Figure 1. Developer options



# Activation du mode développeur sur un device iOS



# What You Need To Enroll



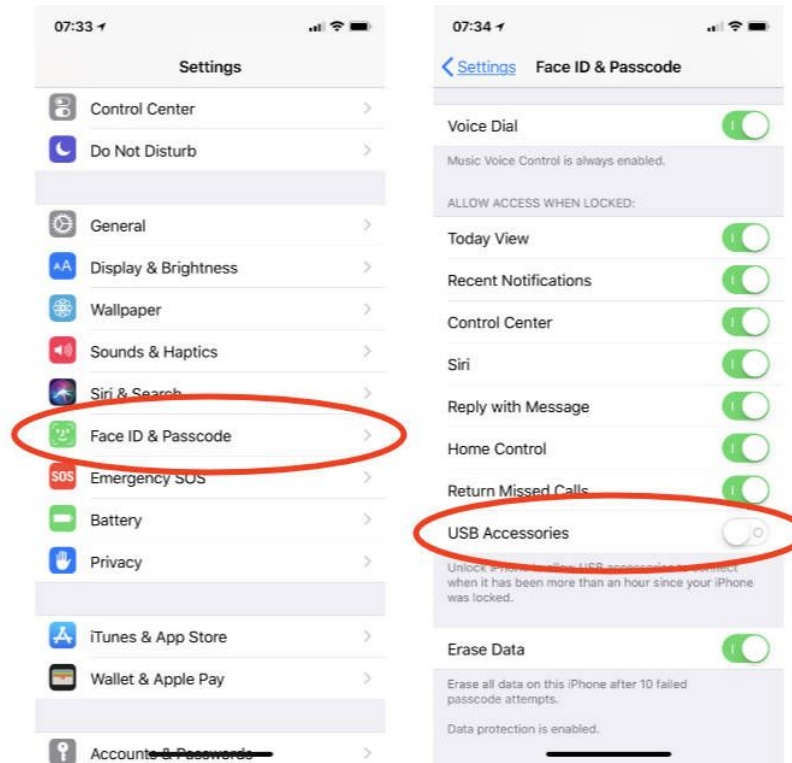
## Enrolling as an Individual

If you are an individual or sole proprietor/single person business, get started by signing in with your Apple ID with [two-factor authentication](#) turned on. You'll need to provide basic personal information, including your legal name and address.

**Pour tester une application iOS il faut d'abord créer un compte développeur Apple "individuel"**

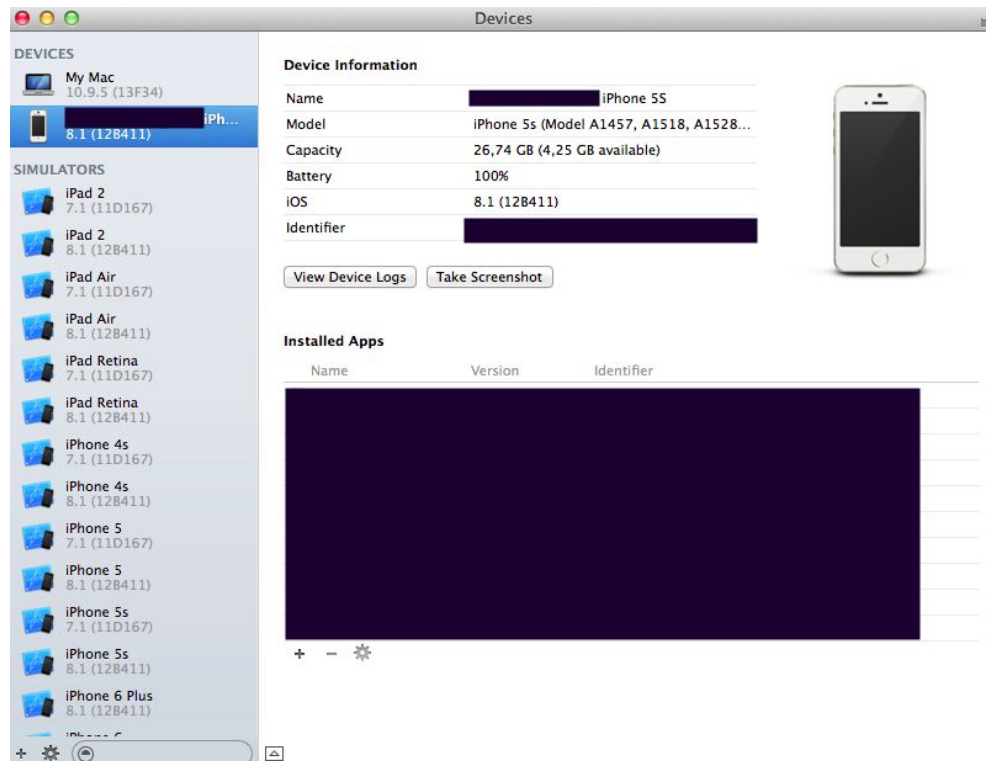
<https://developer.apple.com/programs/enroll/>





Activer le débogage USB sur iPhone ou iPad

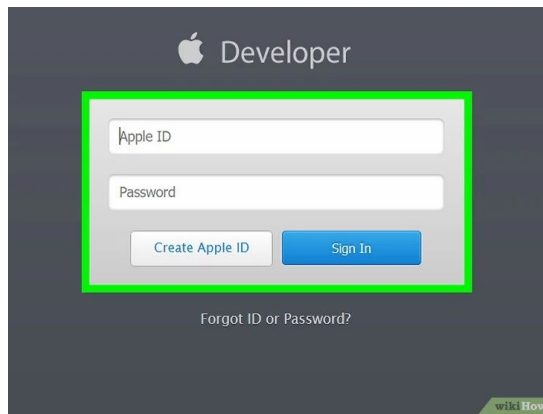




Ajouter et autoriser les tests sur un device iOS dans Xcode

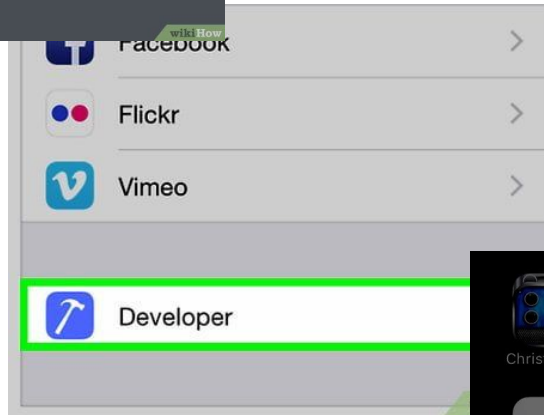






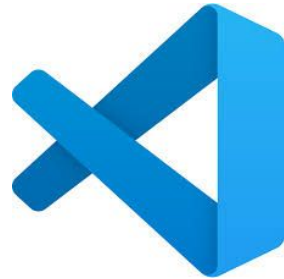
## Activation du mode développeur sur un terminal iOS

<https://fr.wikihow.com/activer-le-mode-d%C3%A9veloppeur-sur-un-iPhone>



# Visual Studio Code

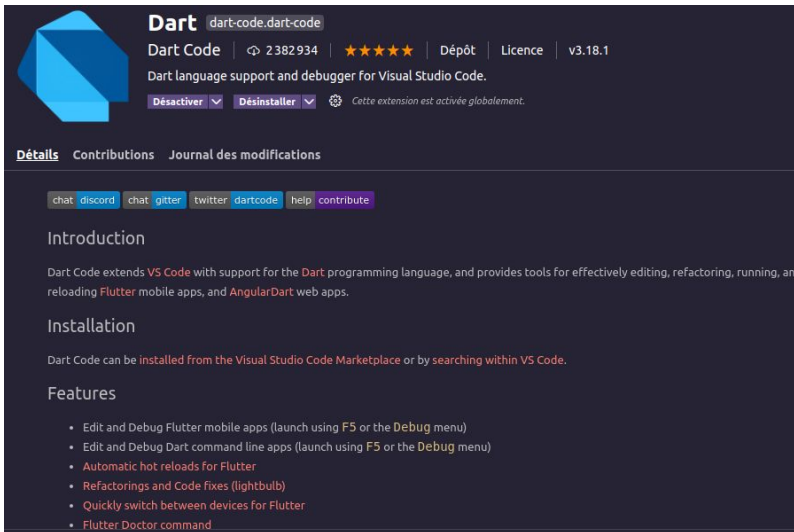
(facultatif)



# ENVIRONNEMENT DE DÉV. MOBILE

**Android Studio** et **Xcode** sont les **IDE par défaut** pour le développement mobile.

Cependant, il est possible d'utiliser **Visual Studio Code** avec les plugins **Flutter** et **Dart**.



**Dart** `dart-code.dart-code`

Dart Code | 2382934 | ★★★★★ | Dépôt | Licence | v3.18.1

Dart language support and debugger for Visual Studio Code.

Désactiver | Désinstaller | Cette extension est activée globalement.

Détails Contributions Journal des modifications

chat discord chat gitter twitter dartcode help contribute

### Introduction

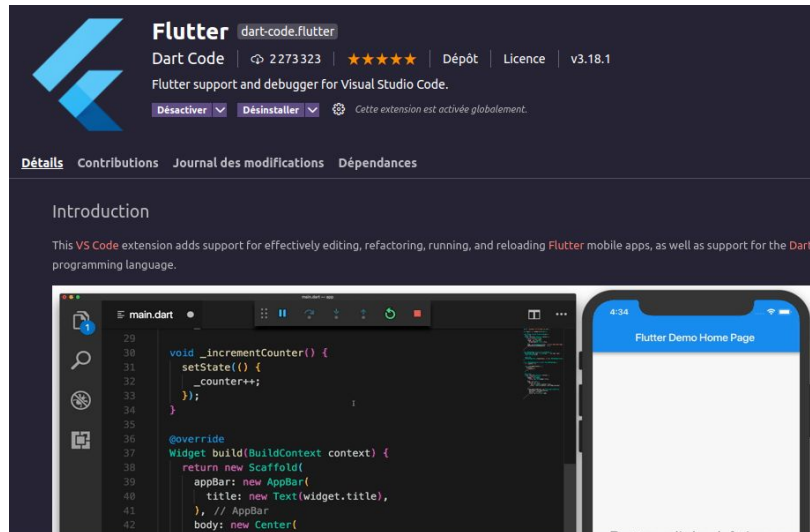
Dart Code extends **VS Code** with support for the **Dart** programming language, and provides tools for effectively editing, refactoring, running, and reloading **Flutter** mobile apps, and **AngularDart** web apps.

### Installation

Dart Code can be installed from the Visual Studio Code Marketplace or by searching within VS Code.

### Features

- Edit and Debug Flutter mobile apps (launch using F5 or the Debug menu)
- Edit and Debug Dart command line apps (launch using F5 or the Debug menu)
- Automatic hot reloads for Flutter
- Refactorings and Code fixes (lightbulb)
- Quickly switch between devices for Flutter
- Flutter Doctor command



**Flutter** `dart-code.flutter`

Dart Code | 2273323 | ★★★★★ | Dépôt | Licence | v3.18.1

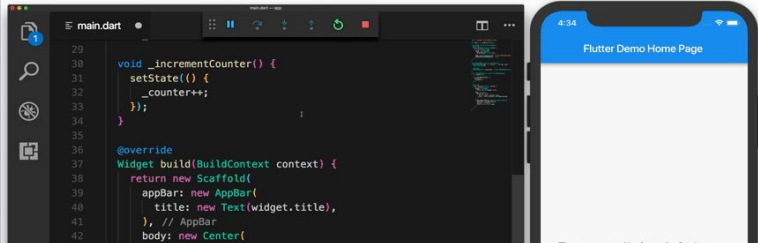
Flutter support and debugger for Visual Studio Code.

Désactiver | Désinstaller | Cette extension est activée globalement.

Détails Contributions Journal des modifications Dépendances

### Introduction

This **VS Code** extension adds support for effectively editing, refactoring, running, and reloading **Flutter** mobile apps, as well as support for the **Dart** programming language.



The screenshot shows the Flutter extension interface. On the left, a code editor displays Dart code for a Flutter app, including a `main.dart` file with a `void _incrementCounter()` function and a `build(BuildContext context)` method. On the right, a mobile app preview is shown, displaying a "Flutter Demo Home Page" with a blue header and a white body.

**Configuration de Visual Studio Code pour Flutter**

<https://flutter.dev/docs/development/tools/vs-code>

# Installation de Flutter



# Install



[Docs](#) > [Get started](#) > Install

Select the operating system on which you are installing Flutter:



Windows



macOS



Linux



Chrome OS

**Suivre intégralement la méthode d'installation  
dédiée à votre système d'exploitation**

**<https://flutter.dev/docs/get-started/install>**

# ENVIRONNEMENT DE DÉV. MOBILE

- **Windows :**

<https://flutter.dev/docs/get-started/install/windows>

- **MacOS :**

<https://flutter.dev/docs/get-started/install/macos>

- **Linux :**

<https://flutter.dev/docs/get-started/install/linux>

- **ChromeOS :**

<https://flutter.dev/docs/get-started/install/chromeos>

# Setting up Flutter for Windows

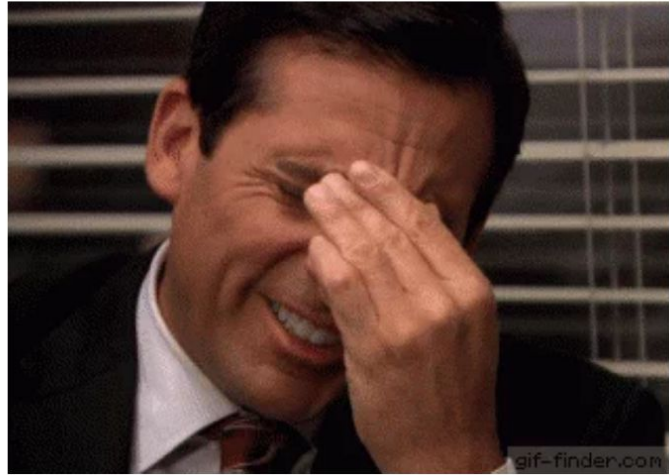
If your Flutter configuration is taking forever to set up, this article is for you. This article will guide you to set up flutter path and set up environment for both VS Code and Android Studio.



Pratik Singhal

Follow

Sep 16, 2019 · 6 min read



**Installation complète sur Windows**

<https://medium.com/fnplus/setting-up-flutter-for-windows-ca2c5e643fdf>



# Setting up Flutter on macOS Catalina



Alexandros Baramilis Oct 22, 2019 · 7 min read



**Installation complète sur Mac**

<https://medium.com/@alexandrosbaramilis/setting-up-flutter-on-macos-catalina-d023df8845ae>

# ENVIRONNEMENT DE DÉV. MOBILE

Renseignez correctement les **variables d'environnement** de votre système d'exploitation :

- `ANDROID_HOME`
- `ANDROID_SDK_ROOT`

Et renseignez l'accès à Flutter dans votre "***path***" afin d'accéder au CLI de Flutter depuis le terminal.



# ENVIRONNEMENT DE DÉV. MOBILE



Les problèmes d'installation sont très souvent liés à une mauvaise configuration des **variables d'environnement** :

- non renseignées
- ou mal renseignées.

# ENVIRONNEMENT DE DÉV. MOBILE

Modifier les variables d'environnement **Linux ou Mac** :

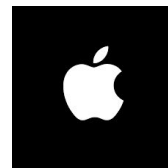
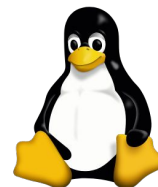
```
sudo nano ~/.bashrc
```

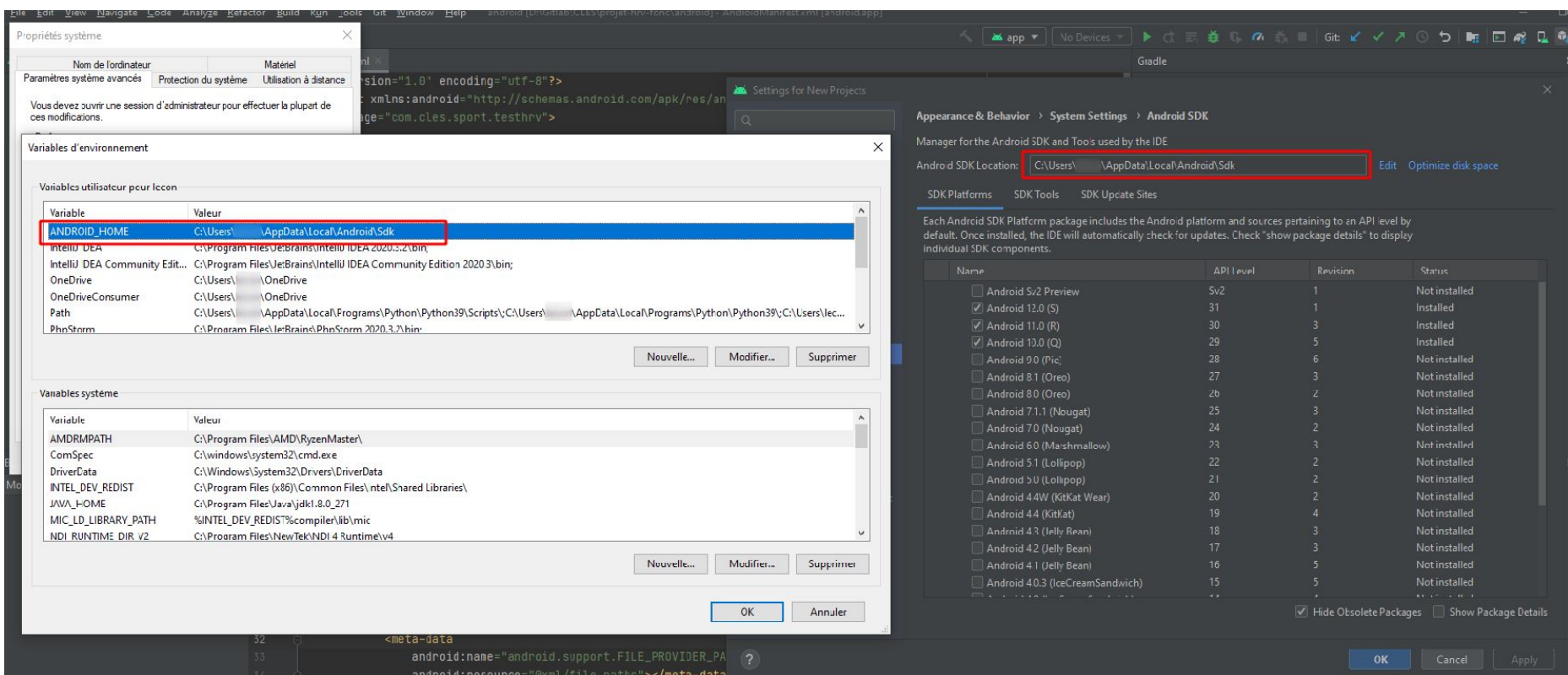
```
export
```

```
PATH="$PATH:<PATH-TO-FLUTTER>/flutter/bin"
```

Remplacer la partie **<PATH-TO-FLUTTER>** par le chemin réel de l'installation de Flutter sur votre PC.

Pour connaître ce chemin : `which flutter`



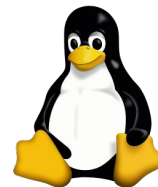


## Configuration des variables d'environnement sur Windows

# ENVIRONNEMENT DE DÉV. MOBILE

Après une modification du fichier *.bashrc*, tapez la commande suivante pour que les réglages prennent effet :

```
source ~/.bashrc
```



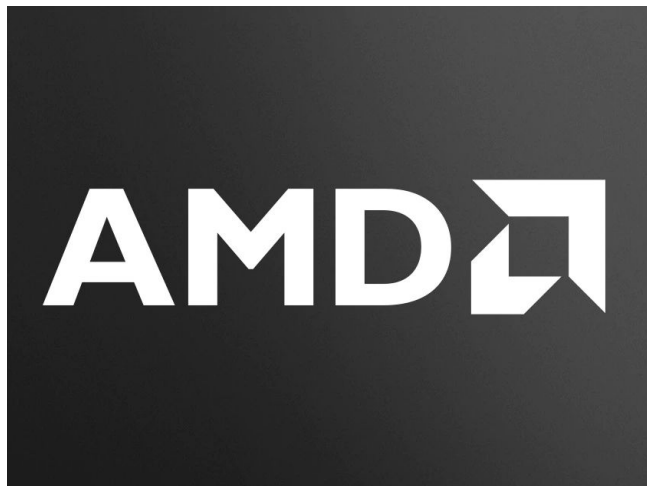
Sur les versions récentes de MacOS  
Bash a été remplacé par ZSH en tant  
qu'outil Shell par défaut.

L'utilisation de Bash est  
recommandée.

[https://www.howtogeek.com/444596/  
/how-to-change-the-default-shell-to-b  
ash-in-macos-catalina/](https://www.howtogeek.com/444596/how-to-change-the-default-shell-to-bash-in-macos-catalina/)



# ENVIRONNEMENT DE DÉV. MOBILE



Sur les ordinateurs équipés d'un **processeur AMD**, il est nécessaire d'**activer l'accélération matérielle dans le BIOS** pour utiliser les émulateurs Android Studio.

<https://developer.android.com/studio/un/emulator-acceleration>



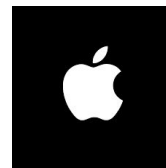


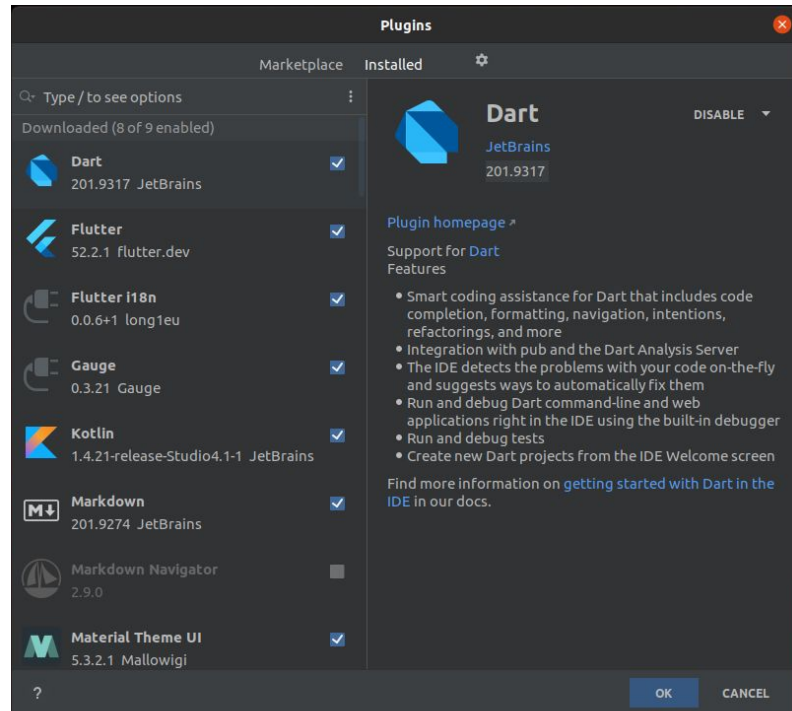
# ENVIRONNEMENT DE DÉV. MOBILE

De plus, il est nécessaire d'accepter les licences Xcode et/ou Android Studio :

```
sudo xcodebuild -license
```

```
flutter doctor --android-licenses
```





# Installation des plugins Dart et Flutter depuis Android Studio



Search: flutter Category: All Sort by: name

Name	Downloads	Rating	Updated
AutoJson	3 476	☆☆☆☆☆	3 weeks ago
<b>Flutter</b>	1 365 571	★★★★★	4 days ago
Flutter i18n	76 443	★★★★★	one month ago
Flutter JsonBeanFactory	13 430	☆☆☆☆☆	4 days ago
Flutter JsonHelper			
Json2Dart			

**Flutter**  
LANGUAGES  
1365571 downloads  
Updated 29/03/2019 v34.0.1

Support for developing **Flutter** applications. **Flutter** gives developers an easy and productive way to build and deploy cross-platform, high-performance mobile apps for both Android and iOS. Installing this plugin will also install the Dart plugin.

**Change Notes**

- Don't show a background square in the inspector summary tree. (#3326)
- Make **Flutter**ModuleUtils consistently robust to disposed projects. (#3323)
- Fix NPE issue sometimes hit evaluating expressions. (#3324)
- Fix widget names. (#3322)
- Make Perf and Inspector views only display when a **Flutter** app is being debugged. (#3320)
- Support the inspector for **flutter\_web** libraries. (#3310)
- Detect when integrations tests are running (#3308)
- Add in support for reloading and restarting all running apps (#3268)
- Log tree path selection fixes (#3302)
- Throttle logger updates (#3280)
- New method in **Flutter** class: **FlutterMethodChannel**

**Downloading Plugins**

Dart  
Downloading plugin 'Flutter'

Cancel  
Background

HTTP Proxy Settings... Manage repositories... Close

Installation des plugins  
Flutter et Dart dans  
Android Studio

configure / plugins /



# ENVIRONNEMENT DE DÉV. MOBILE

## Vérification de l'installation de Flutter

En cas de problème, tapez :

```
flutter doctor
```

puis suivez les consignes indiquées dans le terminal.

Pour plus de détails, tapez :

```
flutter doctor -v
```

# ENVIRONNEMENT DE DÉV. MOBILE

```
[!] Android Studio
• Android Studio at /snap/android-studio/current/android-studio
✗ Flutter plugin not installed; this adds Flutter specific functionality.
✗ Dart plugin not installed; this adds Dart specific functionality.
• android-studio-dir = /snap/android-studio/current/android-studio
• Java version OpenJDK Runtime Environment (build 1.8.0_242-release-1644-b3-6222593)
```

La commande **flutter doctor** peut indiquer (par erreur) que les plugins Flutter et Dart ne sont pas installés.

Si les plugins sont effectivement installés, ne pas tenir compte du message indiqué par le terminal.

# ENVIRONNEMENT DE DÉV. MOBILE

## Création d'un projet de base

créer un projet Flutter :

```
flutter create helloworld
```

se placer à la racine du dossier helloworld qui vient d'être créé :

```
cd helloworld
```

ouvrez un émulateur ou branchez un device, puis lancez l'app :

```
flutter run
```

# ENVIRONNEMENT DE DÉV. MOBILE

## Test d'une application Flutter sur émulateur

Obtenir la liste des émulateurs disponibles :

```
flutter emulators
```

Copier l'id de l'émulateur :

ex : `Galaxy_S7_API_28 • Galaxy S7 API 28 • User • android`

Lancer l'émulateur (en remplaçant <id-emulator>)

```
flutter emulators --launch <id-emulator>
```

soit :

```
flutter emulators --launch Galaxy_S7_API_28 • Galaxy S7 API 28 • User • android
```

# ENVIRONNEMENT DE DÉV. MOBILE

## Test d'une application Flutter sur émulateur

Lorsque l'émulateur est lancé, placez vous à la racine d'un dossier de projet Flutter puis tapez :

```
flutter run
```



# ENVIRONNEMENT DE DÉV. MOBILE

L'utilisation de l'émulateur peut parfois être perturbée :

- par l'usage d'un logiciel Antivirus,
- le manque d'espace libre sur l'émulateur (cf. "Wipe Data" sur Android Studio),
- le manque d'espace disque sur le PC...

<https://developer.android.com/studio/run/emulator-troubleshooting>

# ENVIRONNEMENT DE DÉV. MOBILE

## Test d'une application Flutter sur device

Obtenir la liste des devices disponibles :

```
flutter devices
```

Copier l'id du device :

ex : `mon_iphone`

Lancer l'émulateur (en remplaçant <id-emulator>)

```
flutter run --launch <id-device>
```

soit :

```
flutter emulators --launch Galaxy S7 API 28 • Galaxy S7 API 28 • User • android
```

# ENVIRONNEMENT DE DÉV. MOBILE

Configuration pour le test d'une application Android sur device :

<https://developer.android.com/studio/run/device>



# Développement Xcode



# Developing with Flutter on Apple Silicon

Tim Sneath edited this page on 31 Dec 2020 · 14 revisions

Flutter includes preliminary support for developing on macOS devices with [Apple Silicon \(M1\) hardware](#). This wiki page documents limitations and temporary workarounds as we complete work for this new architecture.

► Pages 120

## Using macOS on Apple Silicon to develop Flutter apps (host)

You can use Apple Silicon-based Mac devices as a developer workstation (host) for building Flutter apps. While some support is in a preliminary stage or requires use of the [Rosetta 2 translation environment](#), it's possible to develop and build on [Apple Silicon at this time](#).

Baseline support for Apple Silicon is available in the current stable channel release (Flutter 1.22.4+), which contains required hotfixes to support this hardware. However, depending on your tolerance for risk, [we recommend use of the](#) [beta](#) or [dev](#) [channel](#) to take early advantage of improvements and fixes as they become available.

[Issue 60118](#) tracks the full set of work to support this feature.



Flutter Wiki

**Flutter sur Mac M1 / Apple Silicon**

<https://github.com/flutter/flutter/wiki/Developing-with-Flutter-on-Apple-Silicon>



# Communication avec une API

# ENVIRONNEMENT DE DÉV. MOBILE

Lors de la communication avec une API une erreur de type **"*CERTIFICATE\_VERIFY\_FAILED*"** peut se produire.

Pour résoudre ce problème, il est nécessaire d'autoriser le certificat du client (cf. code ci-après).

[https://flutteragency.com/solve-flutter-certificate\\_verify\\_failed-error-while-performing-a-post-request/](https://flutteragency.com/solve-flutter-certificate_verify_failed-error-while-performing-a-post-request/)

# ENVIRONNEMENT DE DÉV. MOBILE

```
Dio dio = new Dio();  
(dio.httpClientAdapter as  
DefaultHttpClientAdapter).onHttpClientCreate =  
    (HttpClient client) {  
        client.badCertificateCallback =  
            (X509Certificate cert, String host, int port)  
=> true;  
        return client;  
    };
```