

# Guide installation du projet Locapp

## I. Mise en place du projet

## II. Installation de l'appli

### A. Prérequis

### B. Installation d'Android Studio

### C. Ouvrir le projet

### D. Virtualiser et lancer l'appli

## III. Installation de la base de données

## IV. Installation de l'API

### A. Adaptation des chemins

#### 1. Fichiers de config

#### 2. Imports dans l'API

#### 3. Connexion à l'API depuis l'appli

## V. En cas de bugs

## I. Mise en place du projet

Pour suivre ce guide d'installation, vérifiez de bien être muni du fichier zip contenant tous les fichiers nécessaires. Dans le fichier zip, vous deviez avoir trouvé cette documentation et d'autres dossiers : un dossier « **API** », un dossier « **Appli** » et un dossier « **Base de données** ». Munissez vous de ces dossiers et suivez ce guide pour mettre en place l'environnement nécessaire pour permettre à l'application de fonctionner.

## II. Installation de l'appli

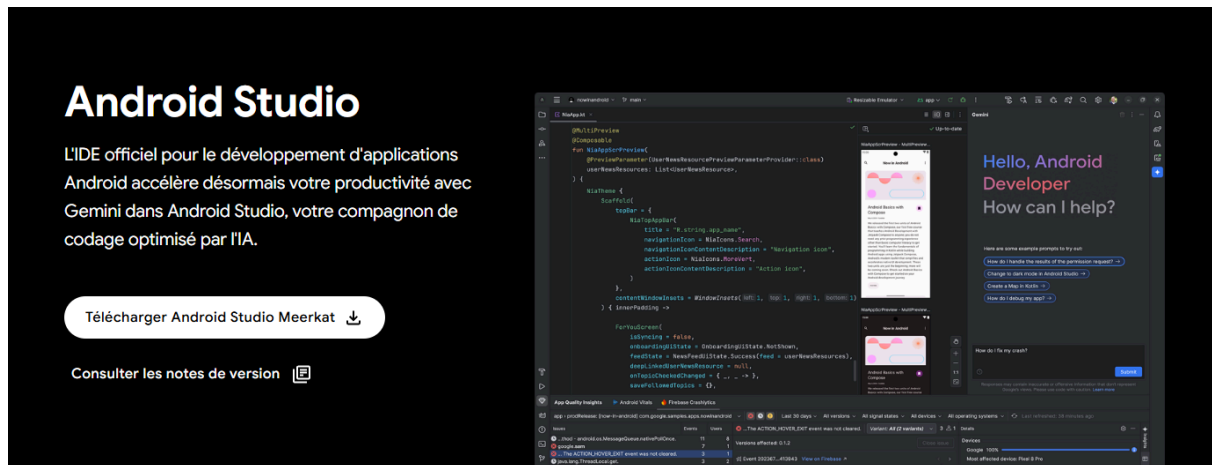
### A. Prérequis

Avant de commencer l'installation, assurez-vous d'abord d'être sur **Windows 10** ou **11**, les versions antérieures ne fonctionnent pas. **16GO de RAM** est recommandé car le logiciel est gourmand, il utilise la **virtualisation** pour afficher un appareil pour faire tourner l'application en temps réel. De ce fait, assurez-vous que votre processeur permet la virtualisation. Aussi, vérifiez si votre version de **Java** est à jour, sinon cela pourrait causer des bugs.

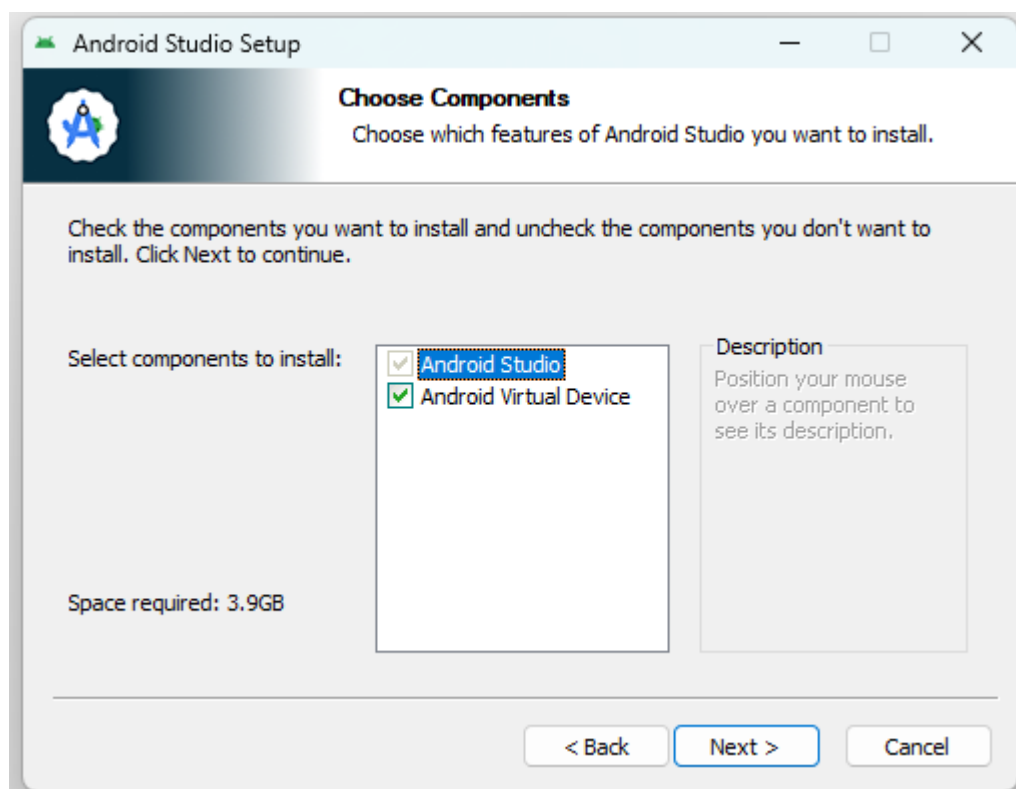
## B. Installation d'Android Studio

Pour télécharger le logiciel, rendez-vous sur ce lien :

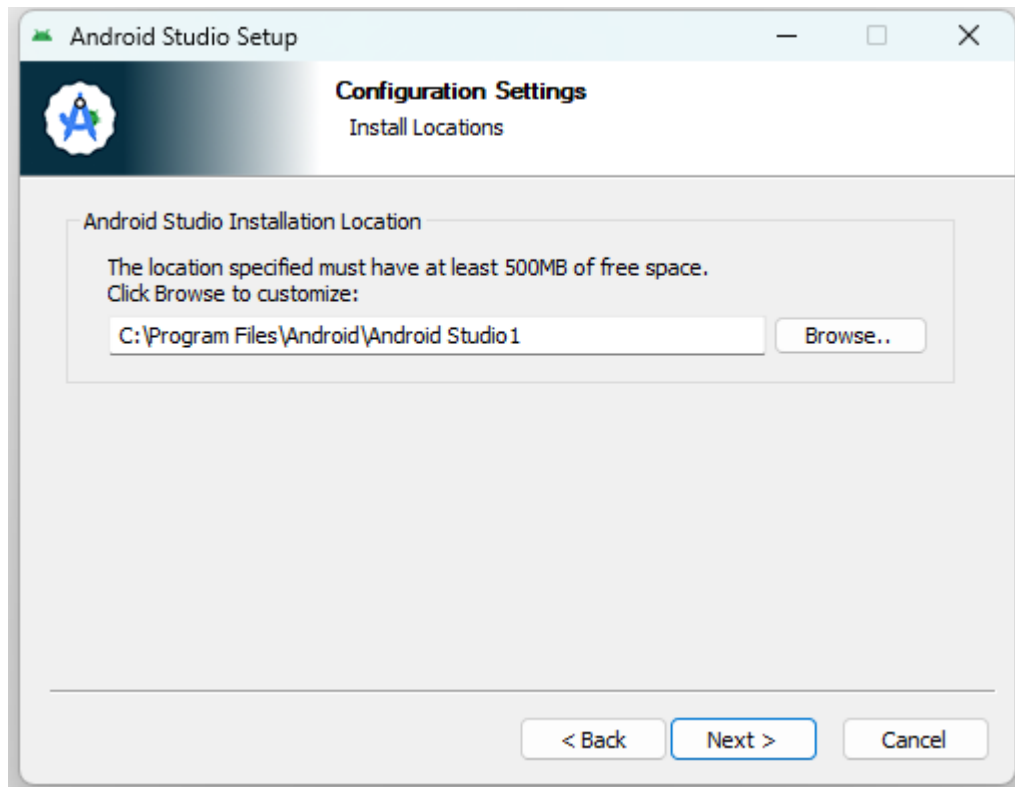
[developer.android.com/studio](https://developer.android.com/studio), puis cliquez sur le bouton « **Télécharger** ».



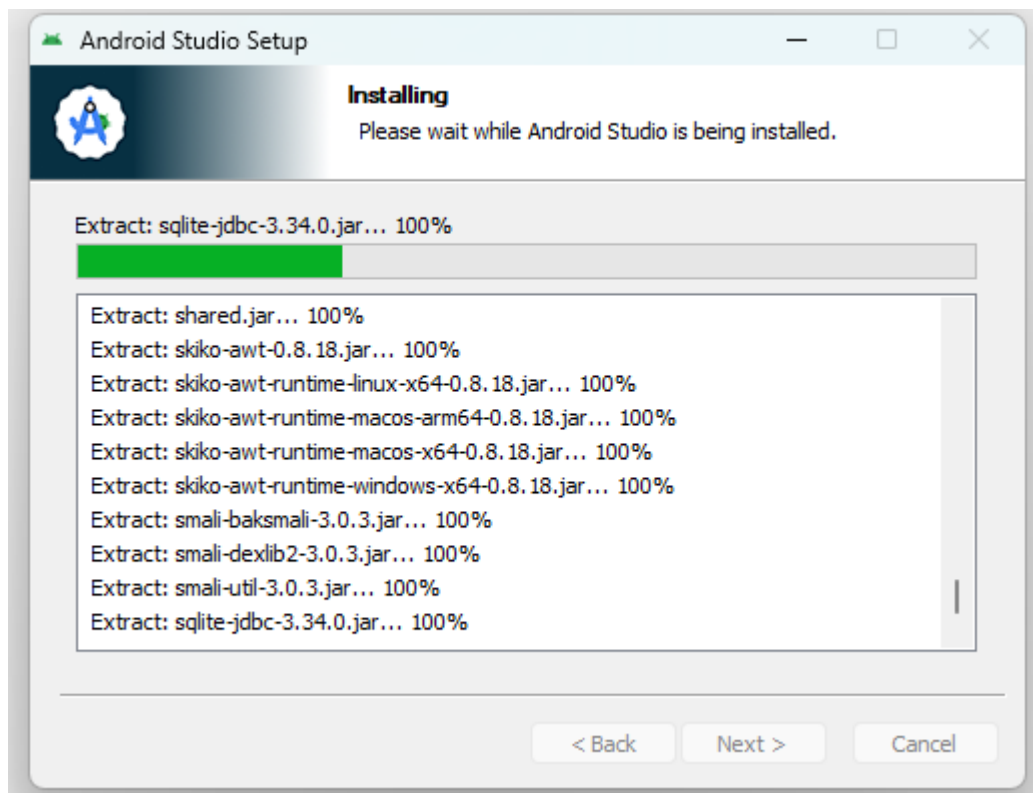
Vous obtiendrez un fichier **.exe**, exécutez-le.



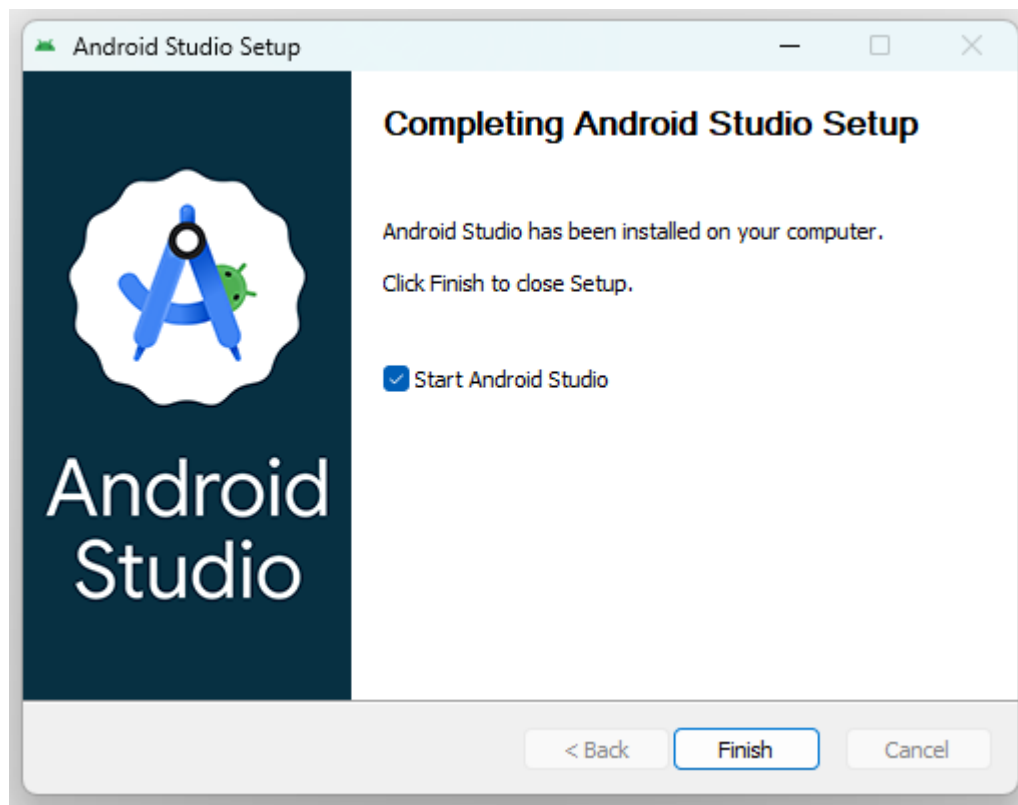
Cliquez sur « **Next** », en vérifiant bien qu'**Android Virtual Device** est coché.



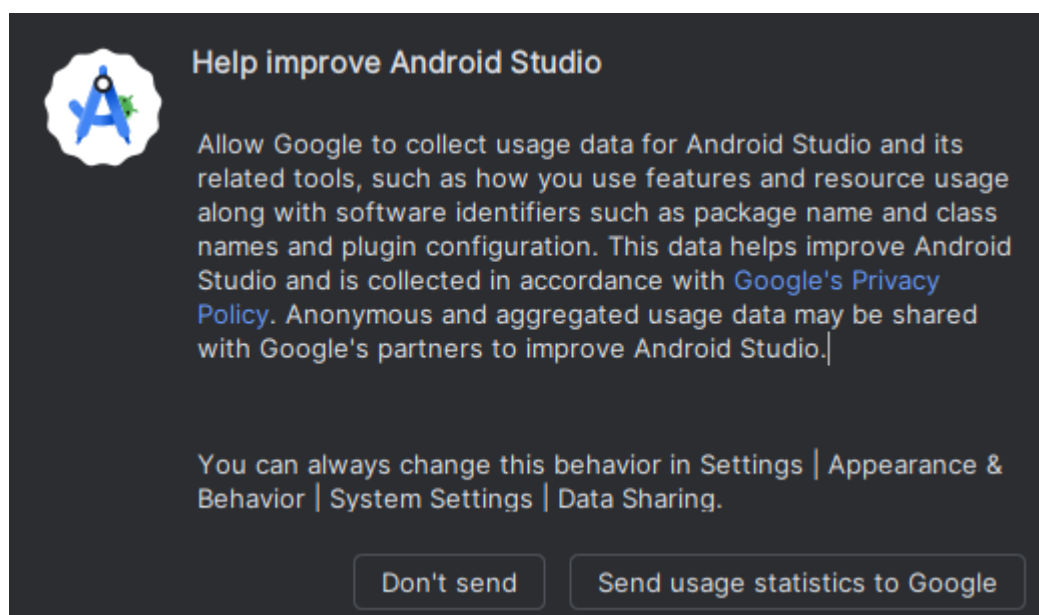
Choisissez le chemin d'installation que vous désirez, vous pouvez laisser celui de base. Cliquez sur « **Next** » et à l'étape suivante appuyez sur « **Install** ».



Patientez le temps que le logiciel termine son installation. Une fois terminée, appuyez sur le bouton « **Next** ».

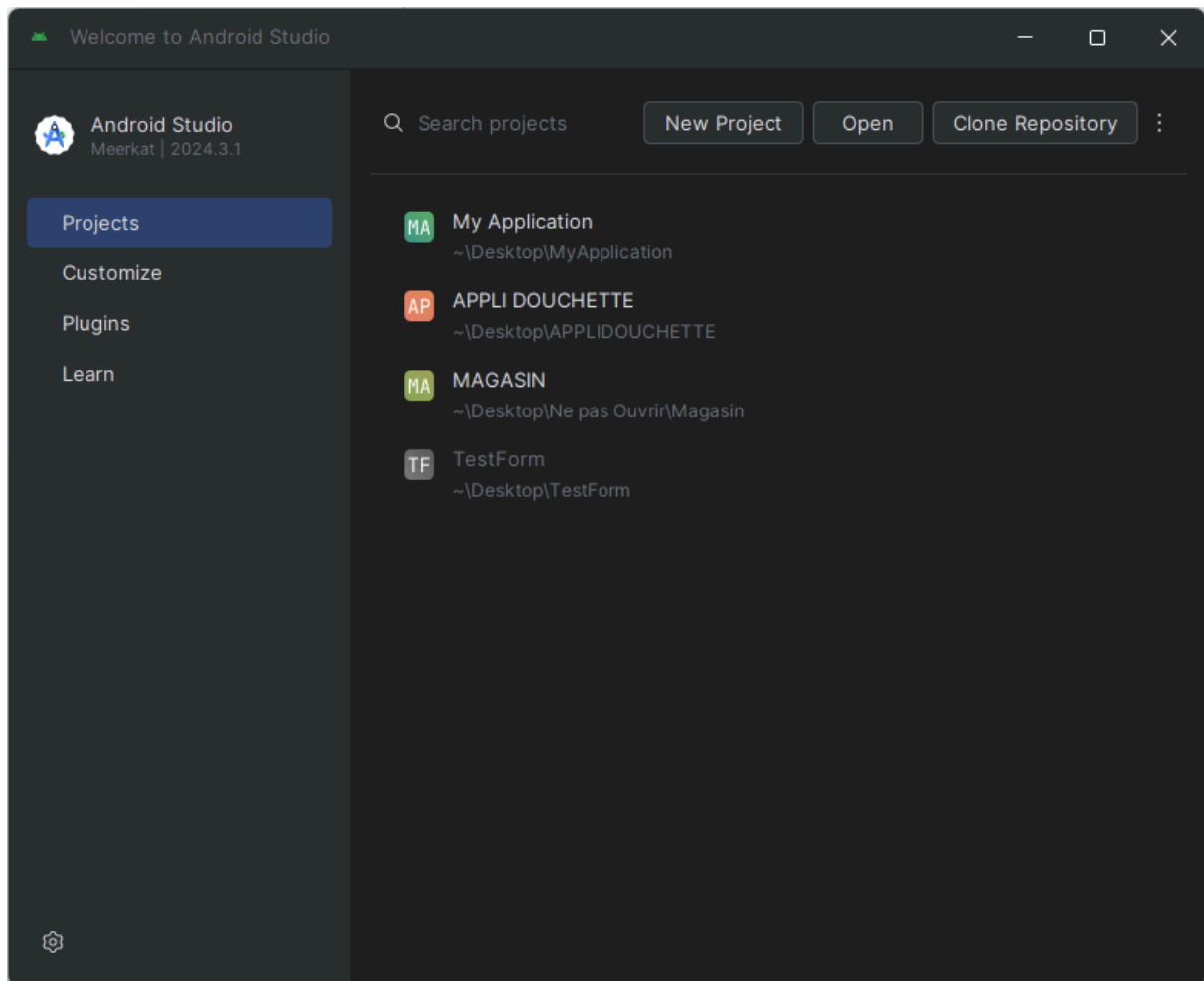


**Android Studio** est maintenant installé, vous pouvez appuyer sur le bouton « **Finish** ». Le logiciel se lancera automatiquement. Sinon lancez-le.

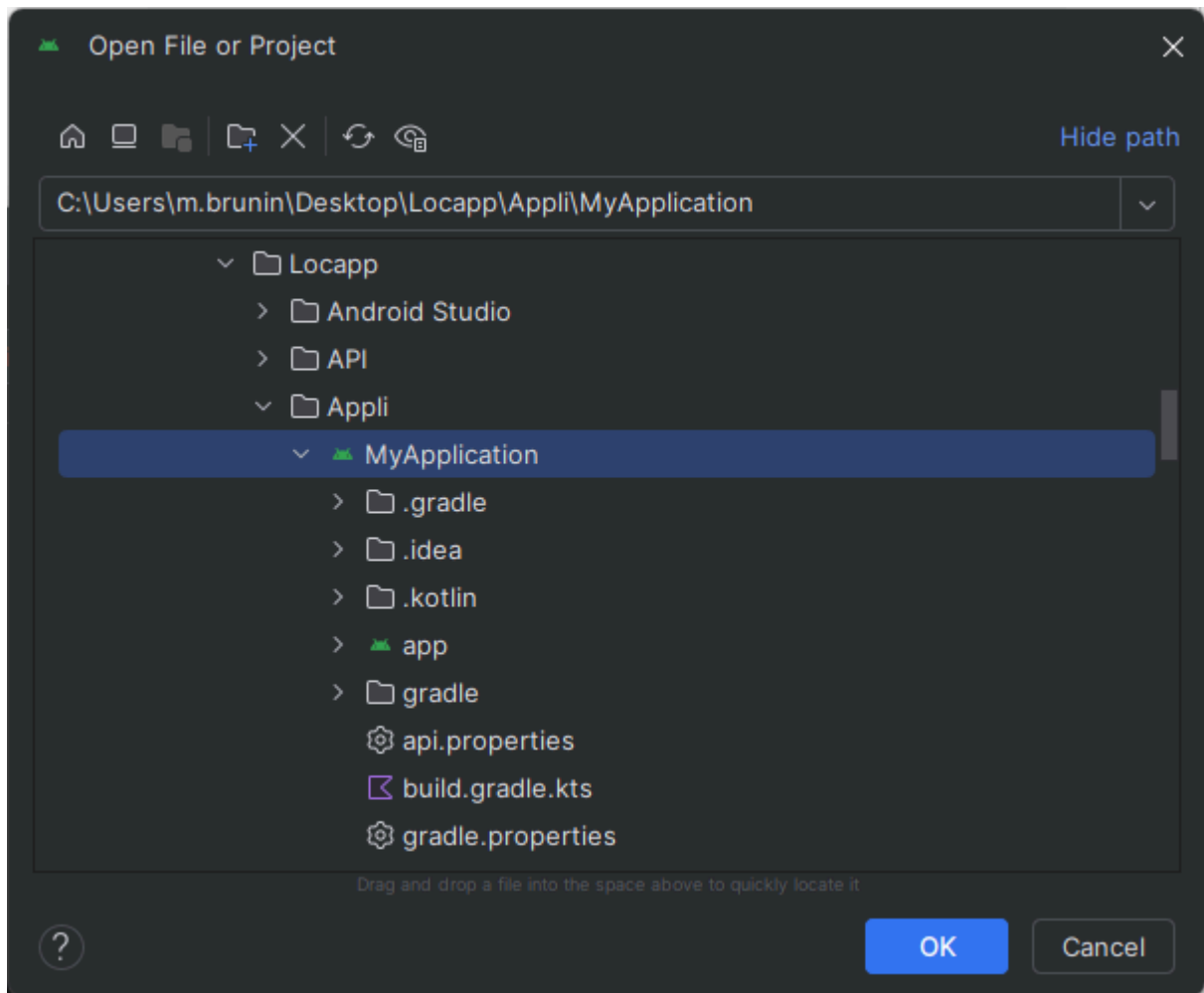


Choisissez de n'envoyer aucune statistique à Google avec le bouton de gauche.

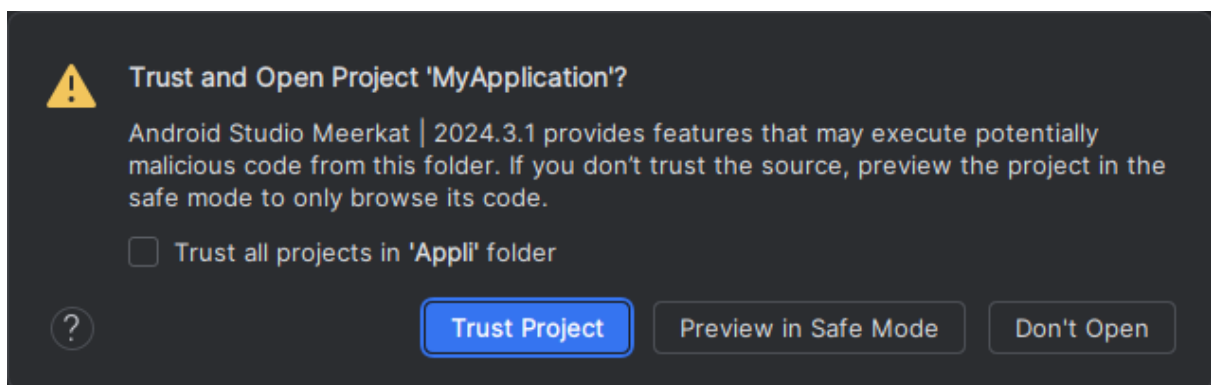
## C. Ouvrir le projet



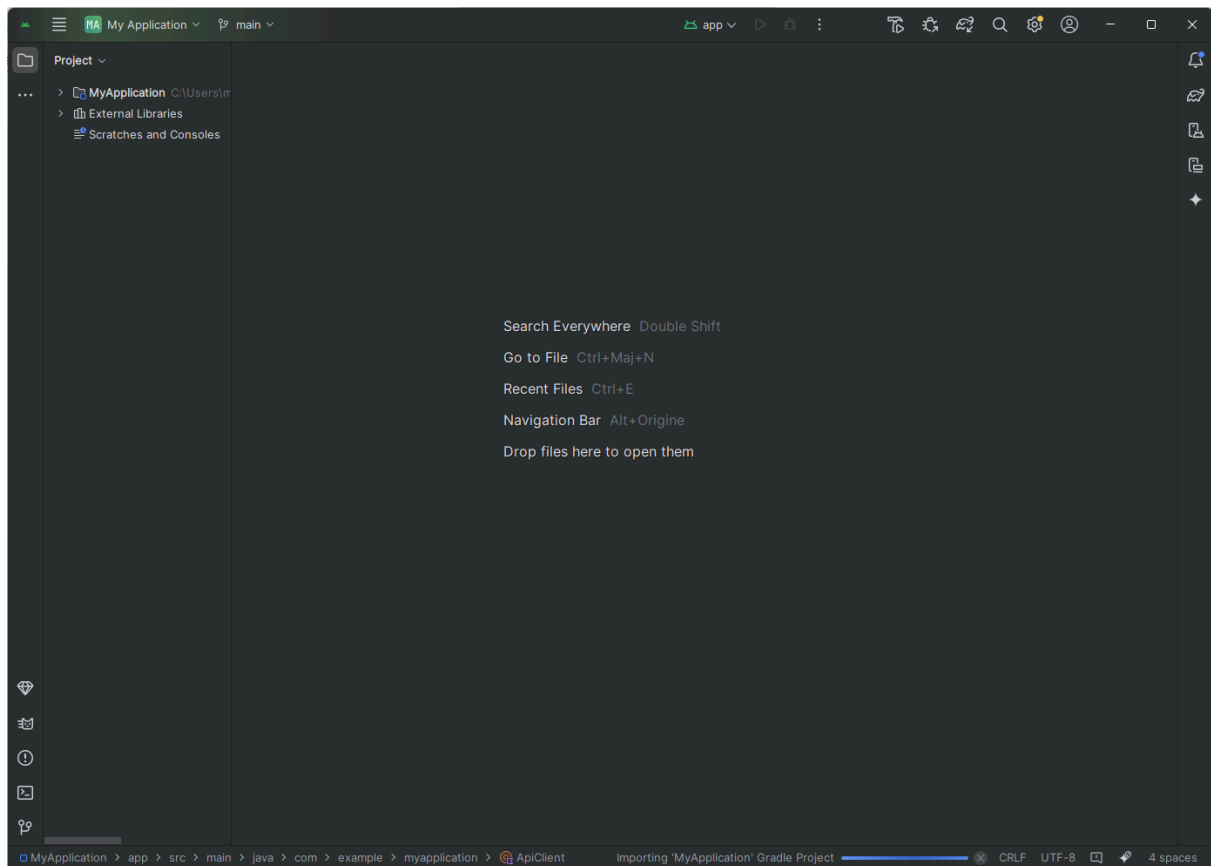
Vous devriez avoir cette page. Mais elle devrait être vide. Nous allons ajouter le projet. Appuyez sur le bouton « **Open** ».



Allez chercher le dossier « **My Application** » disponible dans le dossier **Appli** dans le fichier zip.

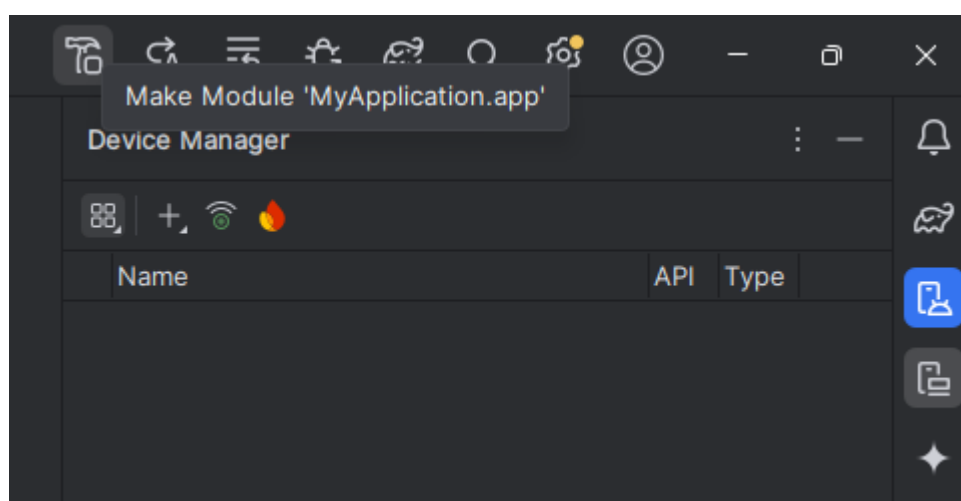


Cliquez sur « **Trust Project** »

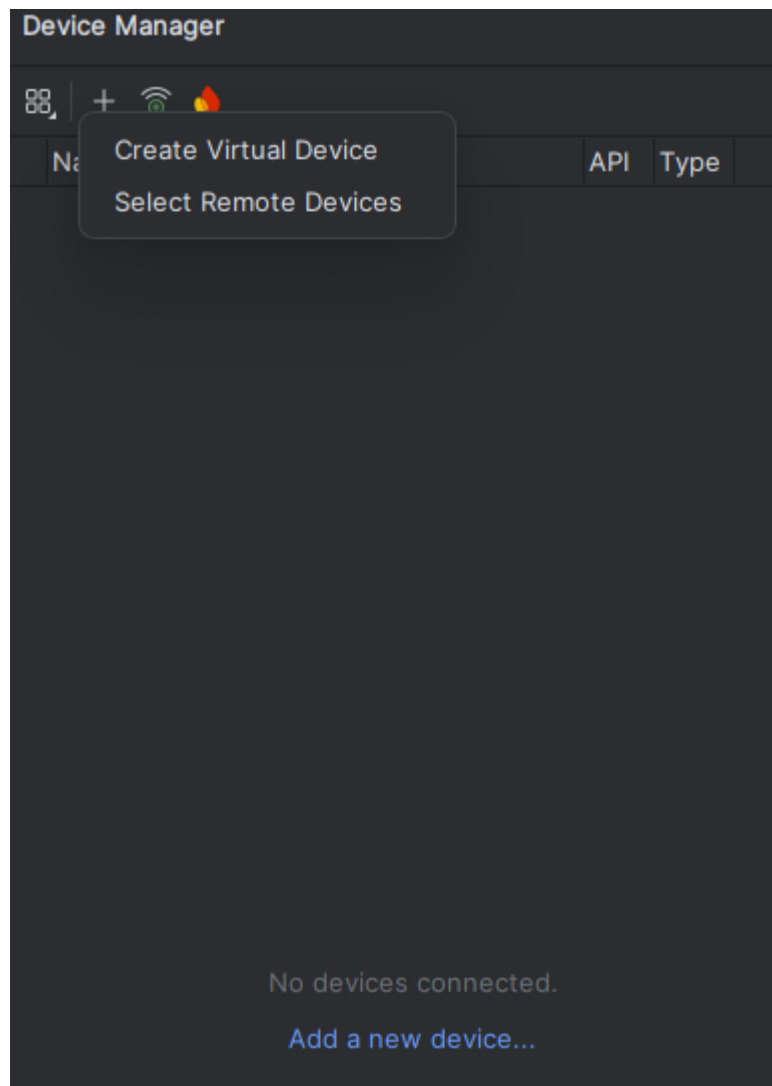


Vous devriez voir cette fenêtre, ici patientez le temps qu'**Android Studio** télécharge bien les fichiers nécessaires pour le projet. La barre bleue en bas à droite indique l'avancement de ces tâches.

## D. Virtualiser et lancer l'appli



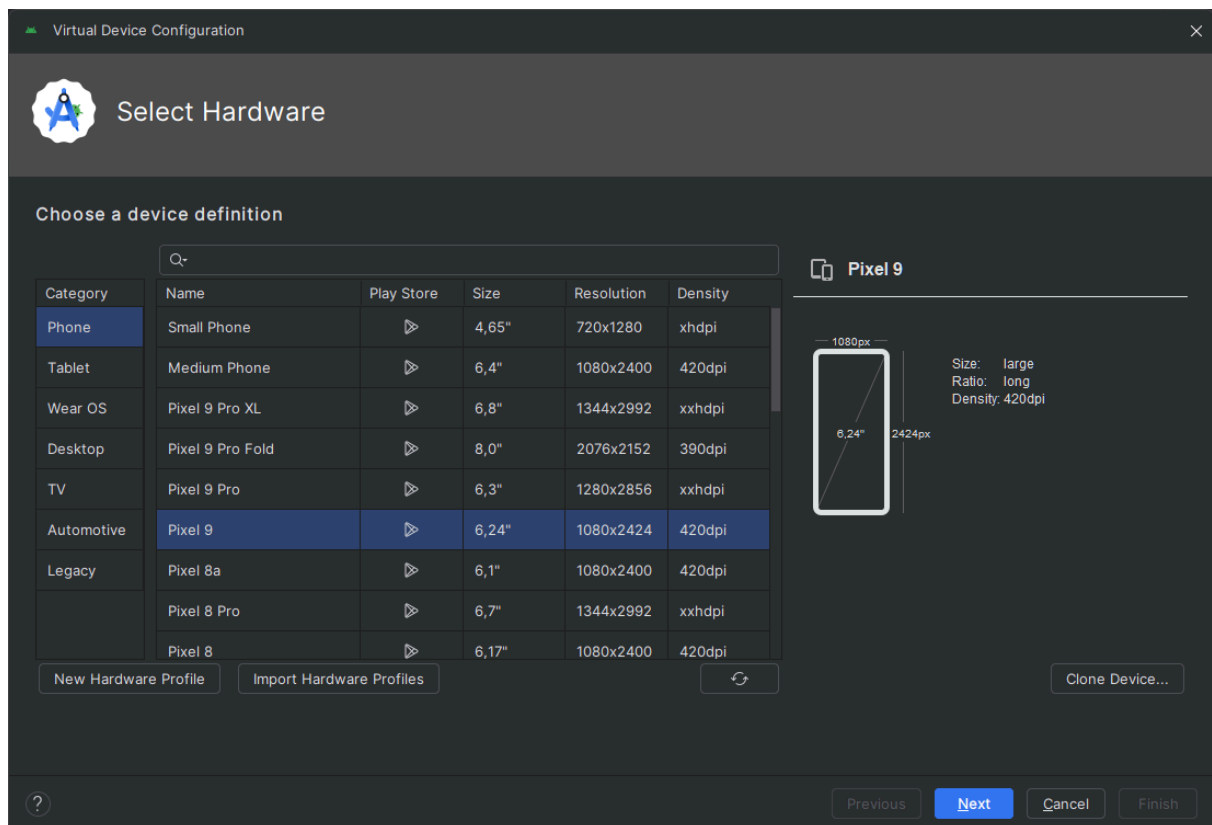
Sur la barre d'action à droite, cliquez sur le bouton en bleu sur le screen



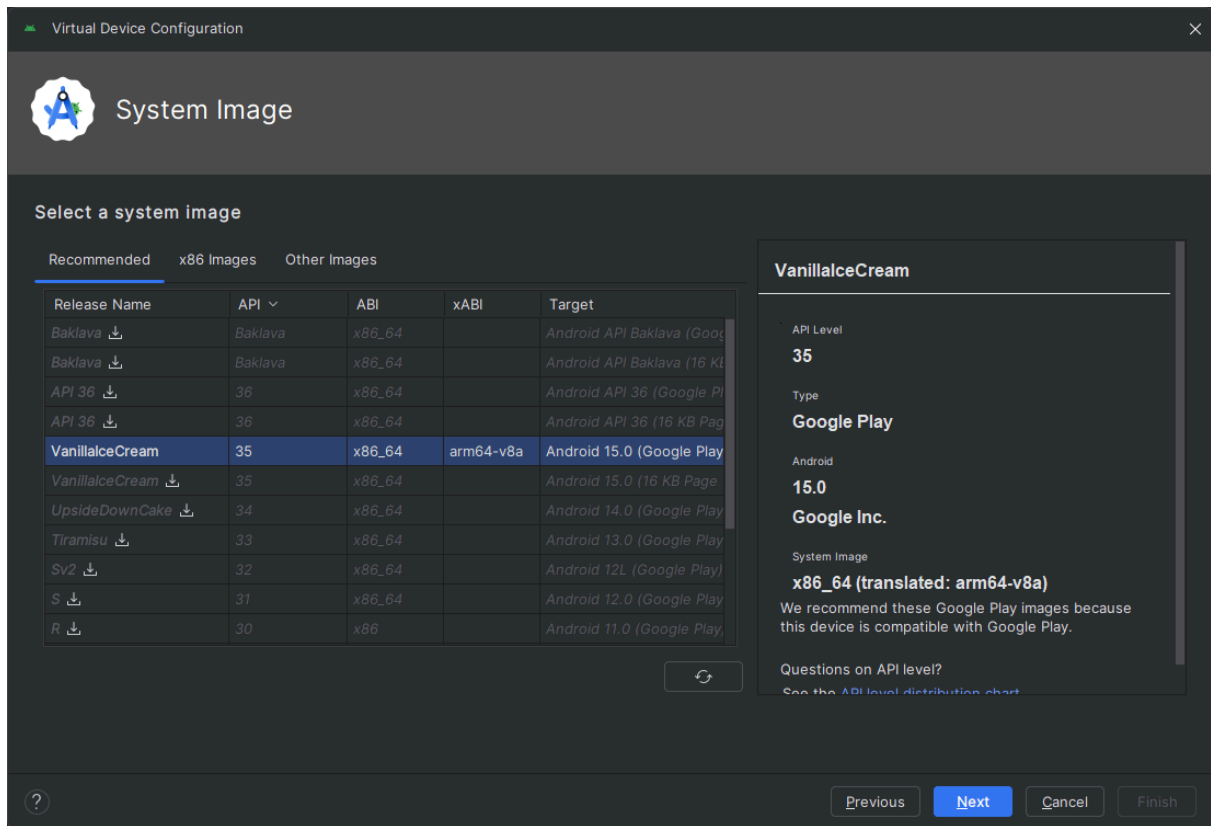
Cette fenêtre devrait s'ouvrir, cliquez sur « **Add a new device** » et ensuite sur « **Create Virtual Device** ».

Sélectionnez l'appareil désiré et appuyez sur « **Next** ». Ici j'ai sélectionné un Pixel 9.

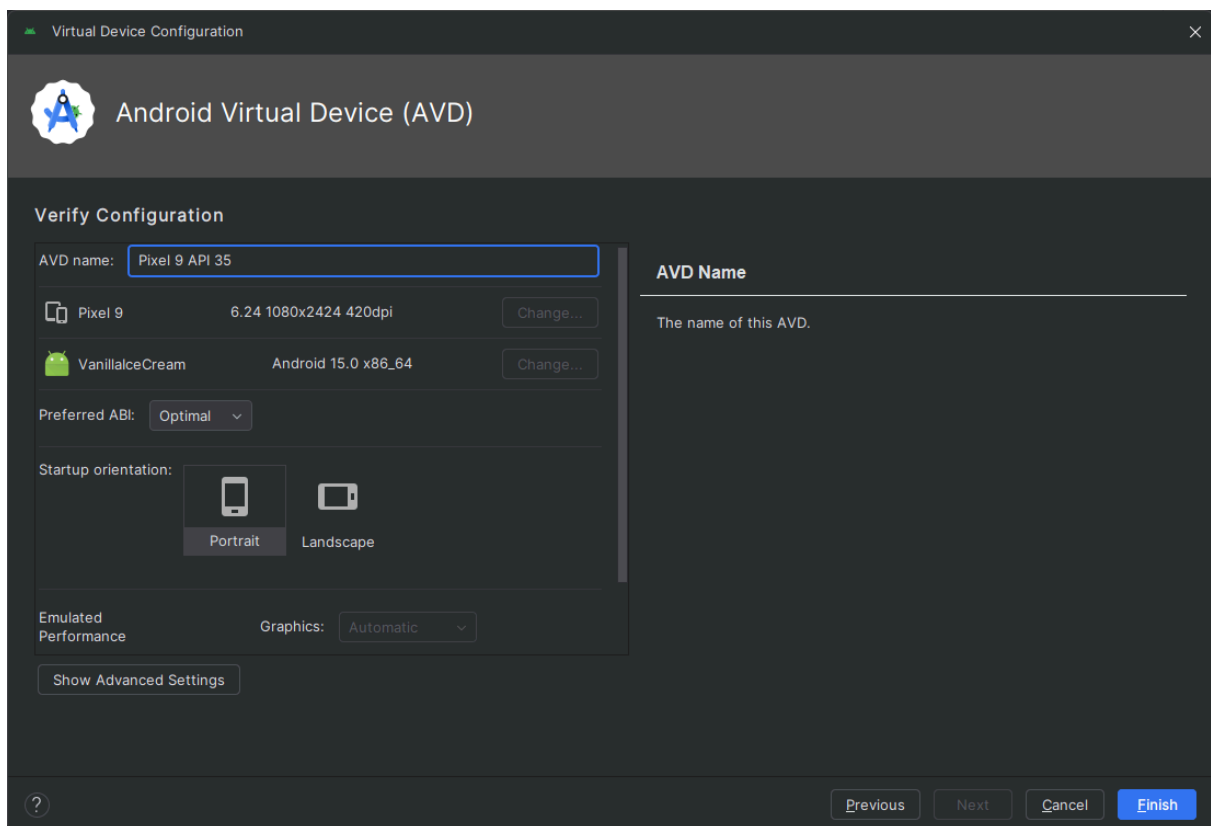




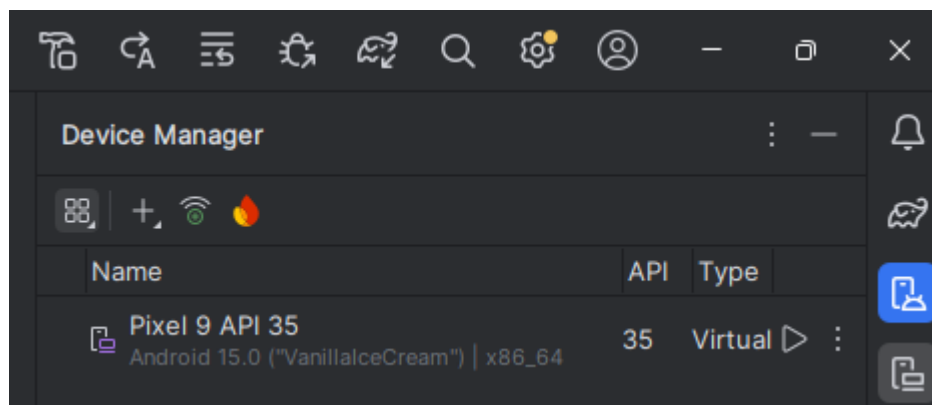
Ensuite vous devez sélectionner une image pour le système. Sélectionnez uniquement un système avec un niveau **API de minimum 26**, sinon l'application ne sera pas compatible avec l'appareil. Plus le niveau est haut, plus il est récent. Ici je vais partir avec un **VanillaIceCream API 35**.



Choisissez un nom et d'autres paramètres si vous le désirez, et appuyez sur « **Finish** ».

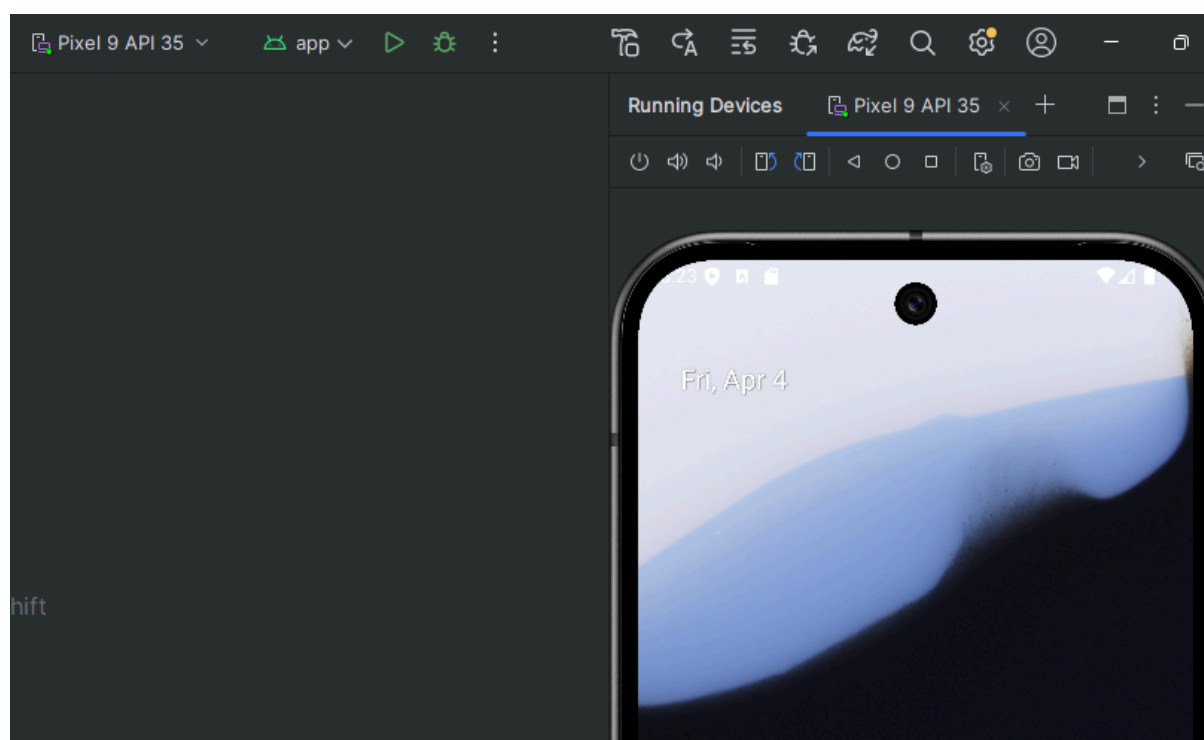


Une fois créée, l'appareil s'affichera dans le menu, vous avez juste à cliquer sur la petite flèche grise à côté de l'appareil pour le lancer. Vous pouvez ouvrir le menu des appareils en exécution en cliquant sur le bouton entouré de gris en dessous de celui en bleu sur le screen.

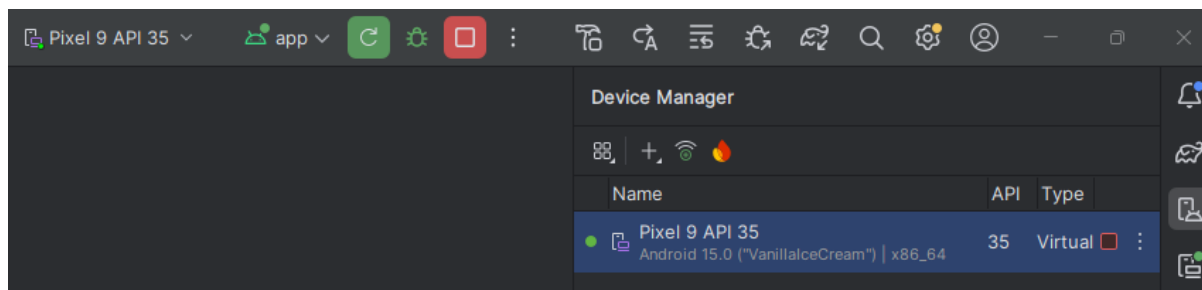


Au lancement de l'appareil, il fera les installations nécessaires pour fonctionner, attendez que ce dernier finisse de tout installer.

Une fois l'appareil lancé, vous pouvez cliquer sur la petite flèche verte en haut du logiciel pour lancer le build de l'application. Elle sera build et installée sur l'appareil en fonctionnement. Vous pouvez suivre l'avancement de ces étapes avec la barre bleu affichée en bas du logiciel. Patientez et l'application s'ouvrira à la fin du build.



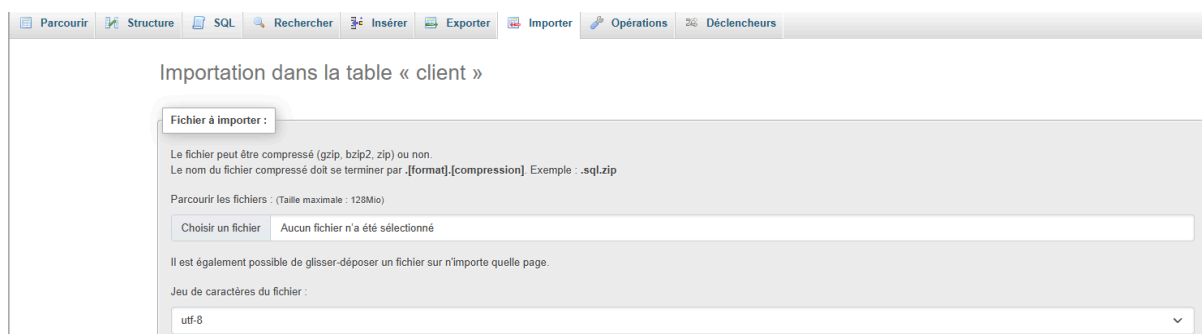
Pour stopper l'application, vous pouvez cliquer sur le bouton carré rouge en haut du logiciel. Pour stopper l'appareil vous pouvez cliquer sur le carré rouge à côté de l'appareil dans le menu « **Device Manager** ».



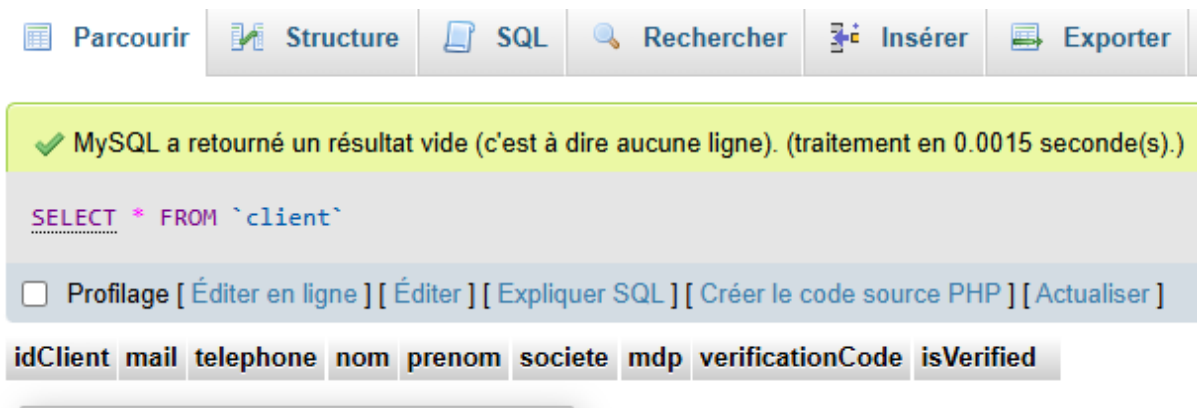
Maintenant que le logiciel est installé, que le projet est importé dans le logiciel, il faut installer l'API et la base de données.

### III. Installation de la base de données

Vous trouverez le script de la table « **client** » dans le dossier « **Base de données** », contenu dans le fichier zip fourni. Vous pouvez importer ce script dans une base de données **PHPMyAdmin**. Créez une base de données si vous n'en avez aucune, puis sélectionnez « **Importer** » dans le menu du haut et allez chercher le script « **client.SQL** ». Puis en bas de la page appuyez sur « **Exécutez** ».



Une fois créée, la base de données devrait ressembler à ça :



Une fois que la base de données est installée, vous devez maintenant installer l'API.

## IV. Installation de l'API

L'API devra être installée sur le domaine que vous souhaitez utiliser. Vous trouverez tous les fichiers constituant l'API dans le dossier « **API** » du fichier zip fourni. Selon la méthode que vous utilisez, l'import des fichiers peut différer, mais le principe restera le même. Voici comment vous devez placer vos fichiers dans votre domaine.

D'abord, vous devez placer les fichiers « **apiKey.php** », « **config.php** », « **smtp.php** » et « **token.php** » en dehors du fichier racine. Dans mon cas, ce fichier était « **www** ». Vous placerez aussi en dehors de la racine, le dossier « **libs** », qui contient les bibliothèques nécessaires à l'API.

Dans le dossier racine, venez juste déposer le dossier « **api** » qui se trouve avec les autres fichiers et le fichier **.htaccess**.

Une fois les fichiers mis dans votre domaine, il ne reste plus qu'à adapter tous les chemins et imports pour que tout fonctionne. Si vous voulez mieux visualiser l'arborescence de l'API, vous pouvez vous aider de ce schéma : [Arbo.](#)

### A. Adaptation des chemins

#### 1. Fichiers de config

Commençons par les fichiers de l'API.

Pour « **smtp.php** », le fichier contient les infos de votre **SMTP**, ouvrez-le et adaptez les identifiants pour votre SMTP.

Pour « **token.php** », vous avez uniquement besoin de changer le domaine pour le remplacer par le vôtre.

Pour « **config.php** », ce fichier renseigne les informations de votre base de données. Vous devez l'ouvrir et adapter ce fichier avec les informations de votre base de données. Si vous êtes sur **PHPMyAdmin**, vous pouvez voir l'host et le nom de votre base de données directement dessus. Pour le username et le password, il s'agit tout simplement d'un compte ayant les accès nécessaires pour faire des requêtes sur la base de données.

## 2. Imports dans l'API

Une fois les fichiers de config modifiés, il faut adapter les imports dans les fichiers PHP. Dirigez vous dans le dossier racine et dans le dossier « **api** ». Vous devez vérifier les points ci-dessous et apportez les modifications pour **TOUT** les fichiers PHP dans ce dossier et sous dossier « **src** », sans quoi le projet ne fonctionnera pas.

Dans le cas où l'import contient `'/home/locaapc/... '`, vous devez remplacer le début du chemin par celui que vous utilisez dans le domaine. Un moyen simple est d'afficher le chemin du fichier dans ses propriétés. Par exemple :

« **require\_once** `'/home/locaapc/libs/php-jwt/src/JWT.php'` ; » doit être changé en accord avec votre domaine. De même pour les imports du genre :  
« **\$config** = **include**(`'/home/locaapc/token.php'`) ; »

Si votre chemin d'import ne contient que `'src/fichier.php'`, alors aucun besoin de le changer. Par exemple : « **require\_once** `'src/Auth.php'` ; » ne demande aucun changement.

Une fois ces modifications faites dans **TOUT** les fichiers PHP du dossier « **api** », votre API devrait être fonctionnelle et reliée à votre base de données.

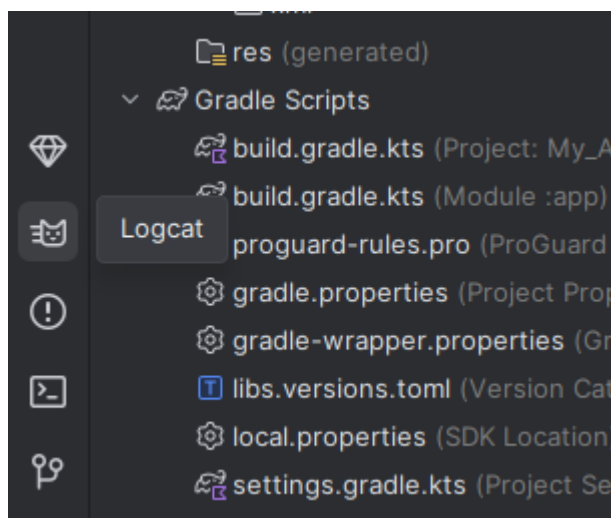
## 3. Connexion à l'API depuis l'appli

Dirigez vous dans le dossier du projet **Android Studio**, « **My Application** ». Dans la racine du projet, vous devriez apercevoir un fichier « **api** », ouvrez-le avec **Visual Studio Code** par exemple. Vous devriez voir deux choses, la **clé API** et **l'URL de l'API**. Ici, modifiez l'URL pour y mettre celle de qui héberge votre API. Si vous n'avez pas changé la clé sur l'API, ne changez surtout pas celle de l'appli, auquel cas les requêtes ne seront pas traitées. Par ailleurs, pour assurer une bonne sécurité, vérifiez bien que votre domaine est en **HTTPS**.

Une fois l'URL changée, votre application devrait être connectée à votre API, qui elle est reliée à votre base de données. Ainsi, le projet est prêt à être lancé et modifié si besoin.

## V. En cas de bugs

Si vous rencontrez des bugs, vous pouvez consulter dans le logiciel **Android Studio** la console **LogCat** :



Les erreurs s'afficheront dedans et vous pourrez les lire et les traiter. De manière générale, les erreurs indiqueront si le problème vient de l'appli, de l'API ou de la base de données.

La documentation Android Studio est plutôt bien fournie, mais sinon la communauté **Stack Overflow** est assez active sur Android Studio et la majorité des erreurs rencontrées peuvent faire l'objet d'un sujet sur le forum, je vous invite à y faire un tour en cas de bug persistant.