

Documentation d'installation du projet

SAE_501

Cette documentation a pour objectif de guider l'installation et la configuration du projet [SAE_501](#). Nous avons opté pour le développement de l'application en utilisant **React Native**.

Prérequis pour Développer une Application React Native via Android Studio

Avant de commencer le développement d'une application **React Native** sous **Android Studio**, il est essentiel d'installer et configurer plusieurs outils. Voici les principaux prérequis que nous allons utiliser :

- [Node.js](#) : Version Stable
- [React Native CLI](#) : Dernière version stable (généralement installée via npm)
- [Android Studio](#) : Dernière version stable (mise à jour régulière, vérifier sur le site officiel)
- [Java Development Kit \(JDK\)](#) : Version 11 ou 17
- [Android SDK](#) : Inclus avec Android Studio, mise à jour via SDK Manager
- [Emulateur Android \(AVD\)](#) : Inclus avec Android Studio, configurable via AVD Manager

Installation d'un JDK (v.17) via Chocolatey

Pour installer le JDK, il existe plusieurs méthodes. Nous allons ici présenter la méthode d'installation via **Chocolatey**.

Installation de Chocolatey (windows)

Prérequis

- Windows 10 ou Windows 11
- Droits administrateur sur le système
- PowerShell v2 ou version plus récente

Étapes d'installation

1. **Ouvrir PowerShell en tant qu'administrateur :**
 - Cliquez sur [Démarrer](#) > Recherchez [PowerShell](#) > Clic droit > [Exécuter en tant qu'administrateur](#).
2. **Exécuter le script d'installation :** dans PowerShell, entrez la commande suivante pour installer Chocolatey :

```
Set-ExecutionPolicy Bypass -Scope Process -Force;  
[System.Net.WebClient]::new().DownloadString('https://community.chocolatey.org/install.ps1') | Invoke-Expression
```

3. **Vérifier l'installation :**
 - Tapez `choco --version` pour vérifier que Chocolatey est correctement installé.

Installation de OpenJDK 17

L'installation de **OpenJDK 17** à l'aide de **Chocolatey** est un moyen simple et efficace de configurer votre environnement de développement Java. Voici les étapes à suivre pour installer OpenJDK 17 et configurer le chemin d'accès.

Étape 1 : Installer OpenJDK 17

1. **Ouvrir une nouvelle invite de commande** (vous pouvez utiliser PowerShell ou l'invite de commande classique).
2. **Exécuter la commande d'installation :**
 - Pour installer OpenJDK 17, exécutez la commande suivante :

```
choco install openjdk17
```

3. **Attendre la fin de l'installation :**
 - Chocolatey téléchargera et installera OpenJDK 17. Une fois l'installation terminée, vous devriez voir un message de confirmation.

Étape 2 : Configurer la variable d'environnement [JAVA_HOME](#)

Pour que votre système puisse localiser le JDK, il est nécessaire de configurer la variable d'environnement [JAVA_HOME](#). Il est également important de noter que **Temurin 17** a été installé avec **OpenJDK 17**, il est donc nécessaire de trouver le dossier [Eclipse Adoptium](#).

1. Localiser le chemin d'installation du jdk:

- Chocolatey installe généralement le jdk correspondant à **Temurin 17** dans le dossier suivant :

```
C:\Program Files\Eclipse Adoptium\jdk-17.0.12.7-hotspot
```

2. Créer ou modifier **JAVA_HOME** :

- Chercher et ouvrir dans la barre de recherche windows **Variables d'environnement**.
- Dans la section **Variables système**, cliquez sur **Nouvelle** pour créer une nouvelle variable ou sélectionnez **JAVA_HOME** si elle existe et cliquez sur **Modifier**.
- Pour la **Variable nom**, entrez **JAVA_HOME**.
- Pour la **Variable valeur**, entrez le chemin menant à votre jdk :

```
C:\Program Files\Eclipse Adoptium\jdk-17.0.12.7-hotspot
```

3. Mettre à jour la variable **Path** :

- Dans la même fenêtre **Variables d'environnement**, trouvez la variable **Path** et cliquez sur **Modifier**.
- Cliquez sur **Nouvelle** et ajoutez le chemin suivant :

```
%JAVA_HOME%\bin
```

Étape 4 : Vérifier l'installation

1. Ouvrir une nouvelle invite de commande.
2. Vérifier la variable **JAVA_HOME** :

```
echo %JAVA_HOME%
```

Cela devrait afficher le chemin que vous avez défini pour **JAVA_HOME**.

3. Vérifier la version de Java :

```
java -version
```

Vous devriez voir une sortie confirmant que **OpenJDK 17** est installé.

Installation et paramétrage de Android Studio

Android Studio est l'IDE officiel pour le développement d'applications Android. Il inclut tout ce dont vous avez besoin pour commencer à créer des applications Android, notamment le SDK Android, un émulateur, et des outils de débogage.

Installation de Android Studio

Étape 1 : Télécharger Android Studio

1. Allez sur le site officiel : [Télécharger Android Studio](#).
2. Cliquez sur **Download Android Studio** et acceptez les conditions d'utilisation.

Étape 2 : Installer Android Studio (windows)

1. Exécutez le fichier **.exe** téléchargé.
2. Suivez les étapes de l'assistant d'installation
3. Une fois l'installation terminée, lancez Android Studio.

Paramétrage de Android Studio

Une fois Android Studio lancé, quelques vérifications sont de mises, nous allons donc procéder à celles-ci pour pouvoir nous assurer du bon fonctionnement de notre application.

Étape 1 : SDK Platforms

1. Sur la page d'accueil de l'application cliquer sur **More Actions** > puis ouvrir **SDK Manager**, dans ce menu sélectionner ensuite **SDK Platforms**.
2. Dans **Languages & Frameworks** > **Android SDK**, cochez les deux versions les plus récentes en termes de niveau d'API (par exemple : 35 et 34). Cette étape assure la compatibilité de React Native avec l'Android SDK.

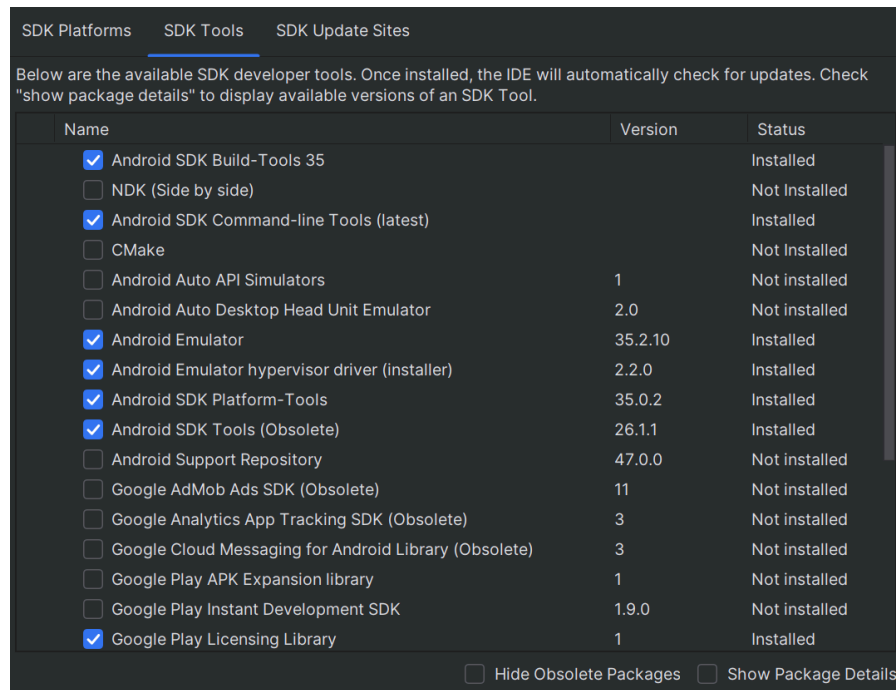
Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, the IDE will automatically check for updates. Check "show package details" to display individual SDK components.

	Name	API Level	Re...	Status
<input checked="" type="checkbox"/>	Android API 35	35	1	Installed
<input type="checkbox"/>	Android VanillaIceCream Preview	VanillaIceCream	4	Not install...
<input type="checkbox"/>	Android UpsideDownCakePrivacySandbox Prev	UpsideDownCakePrivacySand...	3	Not install...
⌵	<input checked="" type="checkbox"/> Android 14.0 ("UpsideDownCake")	34	3	Not install...
	<input type="checkbox"/> Android 14.0 ("UpsideDownCake")	34-ext8	1	Not install...
	<input type="checkbox"/> Android 14.0 ("UpsideDownCake")	34-ext10	1	Not install...
	<input type="checkbox"/> Android 14.0 ("UpsideDownCake")	34-ext11	1	Not install...
	<input type="checkbox"/> Android 14.0 ("UpsideDownCake")	34-ext12	1	Not install...
	<input type="checkbox"/> Android TiramisuPrivacySandbox Preview	TiramisuPrivacySandbox	9	Not install...
	<input type="checkbox"/> Android 13.0 ("Tiramisu")	33	3	Not install...
	<input type="checkbox"/> Android 13.0 ("Tiramisu")	33-ext4	1	Not install...
	<input type="checkbox"/> Android 13.0 ("Tiramisu")	33-ext5	1	Not install...
	<input type="checkbox"/> Android 12L ("Sv2")	32	1	Not install...
	<input type="checkbox"/> Android 12.0 ("S")	31	1	Not install...

☒ Hide Obsolete Packages ☐ Show Package Details

Étape 2 : SDK Tools

1. Aller maintenant dans le menu **SDK Tools**
2. Décocher **Hide Obsolete Packages** en bas à gauche de la fenêtre
3. et vérifier que les outils soient cochés comme ci-dessous :



4. Une fois cela fait appuyer sur **Apply** et attendre la fin du téléchargement

Étape 3 : Variables d'environnements

Les outils Android maintenant téléchargés, il est nécessaire de les ajouter en tant que variable d'environnement.

1. Chercher et ouvrir dans la barre de recherche windows **Variables d'environnement**.
2. Dans la section **Variables système**, cherchez et cliquez sur la variable **Path**
3. Dans la nouvelle fenêtre qui vient de s'ouvrir ajoutez les variables de votre SDK Android (ne pas oublier de changer **User_Name** par le nom de votre ordinateur)

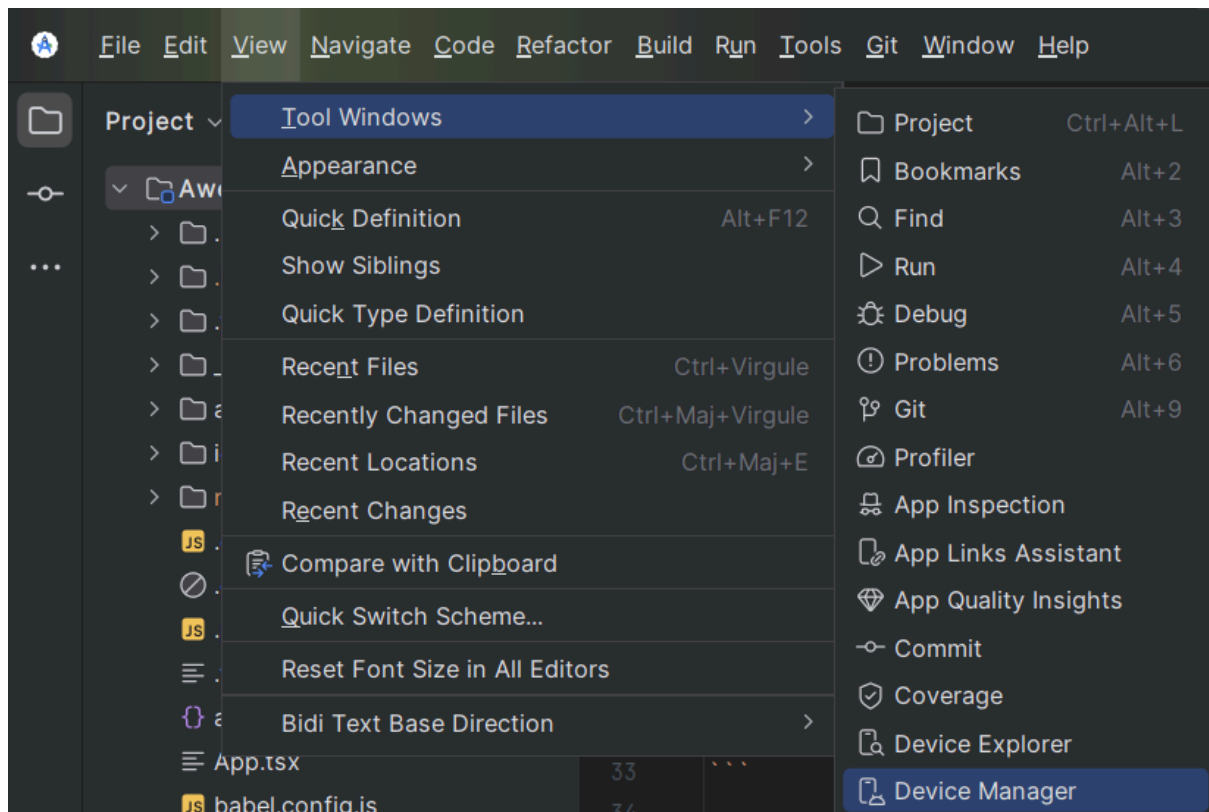
```
C:\Users\User_Name\AppData\Local\Android\Sdk  
C:\Users\User_Name\AppData\Local\Android\Sdk\platform-tools  
C:\Users\User_Name\AppData\Local\Android\Sdk\emulator  
C:\Users\User_Name\AppData\Local\Android\Sdk\tools  
C:\Users\User_Name\AppData\Local\Android\Sdk\tools\bin
```

Étape 4 : Création de device

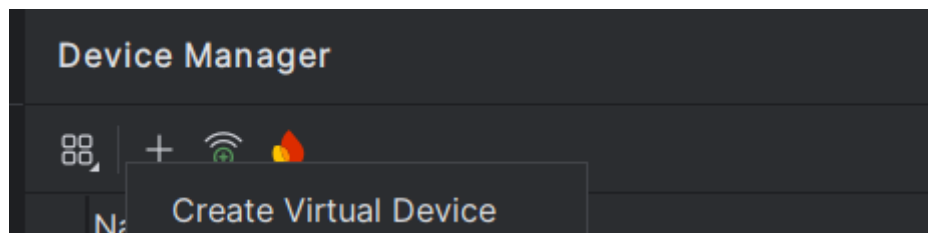
Créer un projet avec les settings de base ou en ouvrir un.

Afin d'émuler une application mobile sur un téléphone il nous faut créer un device qui est la représentation virtuel d'un téléphone.

Pour ce faire il suffit d'aller dans **View > Tool Windows > Device Manager**

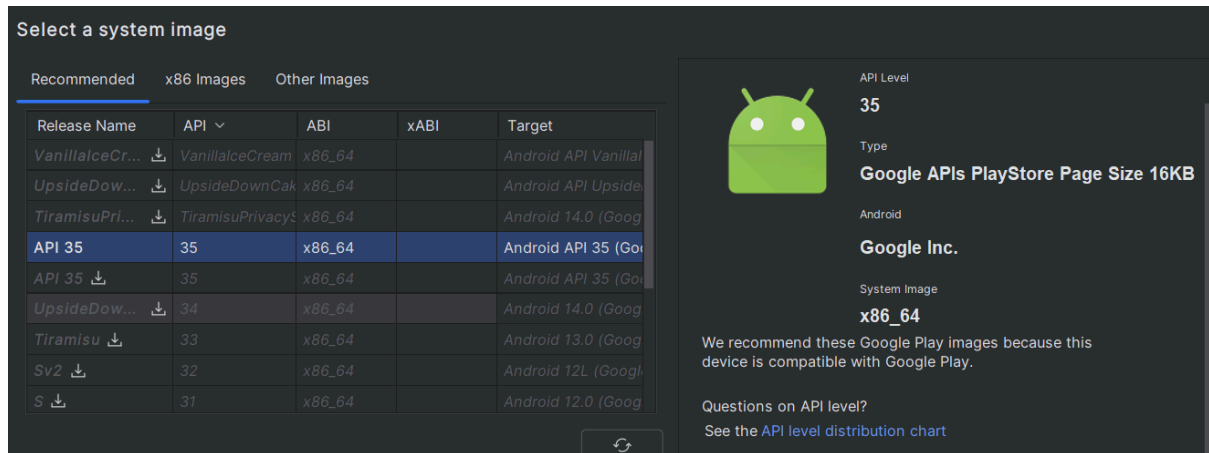


Puis en allant sur la croix (add a new device) nous choisissons d'ajouter créer un téléphone virtuel.



Puis dans le menu suivant nous sélectionnons dans la catégorie “Phone” les téléphones de taille médium “Médium Phone” puis next.

Et pour finir nous installons l’API 35, la sélectionnons et appuyons sur next et finish.



Installation de react native

Etape 1 : Installation de react

Lancement du projet

Etape 1 : Installation de Node JS :

Pour ce faire, installé l'exécutable sur le lien : [Node.js — Download Node.js® \(nodejs.org\)](https://nodejs.org/en/download/) et lancé le en suivant ses instructions.

Etape 2 : Création du projet (si le projet est inexistant)

Afin de créer un projet il faut utiliser la commande ci-dessous en remplaçant `MyReactNativeApp` par le nom du projet :

```
npx create-expo-app@latest
```

Etape 3 : Lancement du projet

Afin d'utiliser l'émulateur il suffit de lancer la commande :

```
npm run android
```

Ensuite, un émulateur s'ouvrira et le projet sera compilé à l'intérieur. L'émulateur lancera automatiquement notre application React via Expo Go. **Si cela ne se produit pas**, vous devrez suivre ces étapes :

1. Ouvrez l'application "Expo Go" sur l'émulateur.



2. Dans l'application, entrez l'adresse exp:// que vous trouverez dans le terminal.

```
Starting project at C:\Users\LFste\OneDrive\Bureau\test\testApp
Starting Metro Bundler
> Opening emulator Medium_Phone_API_35_2
> Opening exp://10.11.20.237:8081 on Medium_Phone_API_35_2
Downloading the Expo Go app [=====] 100% 0.0s
```

3. Appuyez sur "Connect " pour lancer manuellement l'application.

Cela vous permettra d'accéder à notre application React.