

Le développement Ionic sous Android et iOS

Table des matières

I. Contexte	3
II. Installation d'Android Studio et configuration d'un projet Ionic sous Android	3
III. Quiz	13
IV. Environnement iOS et Configuration d'un projet Ionic sous iOS	14
V. Quiz	21
VI. Essentiel	22
VII. Exercice final	22
Solutions des exercices	24

I. Contexte

Durée : 2h

Environnement de travail : Visual Studio Code, Android Studio, Xcode

Prérequis : Bases Angular, installation de ionic

Contexte

Que serait une application mobile si on ne la fait pas fonctionner sur un smartphone ? Dans ce cours, nous allons entrer dans le monde du **Framework Capacitor**, qui, combiné à **Ionic**, permet de développer entièrement une application destinée à être utilisée sur un smartphone ou une tablette.

Le principal rôle d'**ionic** dans un contexte de développement d'applications, est d'offrir des composants d'interface utilisateurs cohérents, modernes, élégants et faciles d'utilisation.

Mais on se pose maintenant la question de comment l'application va pouvoir se transformer afin d'être accessible via un smartphone et ainsi pouvoir utiliser les fonctionnalités natives telles que l'appareil photo, l'enregistrement audio, etc. et c'est ici que la technologie **Capacitor** intervient dans le flux de développement.

Dans ce cours nous allons voir comment utiliser l'ensemble de ces technologies pour émuler une simple application **ionic** sous **Android** et **iOS**.

II. Installation d'Android Studio et configuration d'un projet Ionic sous Android

Objectifs

- Installer Android Studio
- Créer un appareil virtuel
- Créer un projet Ionic
- Configurer Android

Mise en situation

Android Studio est l'IDE idéal pour les développements Android, car son simulateur d'appareil virtuel permet de tester rapidement et simplement le résultat en conditions réelles. Nous allons donc commencer par l'installation de l'outil et ses prérequis. Ensuite nous verrons les différentes étapes de configuration nécessaires pour pouvoir démarrer un projet Ionic pour Android dans les meilleures conditions.

Android Studio

Pour développer une application pour des appareils basés sur le système **Android**, il est nécessaire de posséder les outils **Android** tel que le **SDK Android**.

Installation de Android Studio et du SDK Android

Android Studio est l'application **IDE** permettant la création d'applications Android natives. Il comprend le SDK qui devra être configuré.

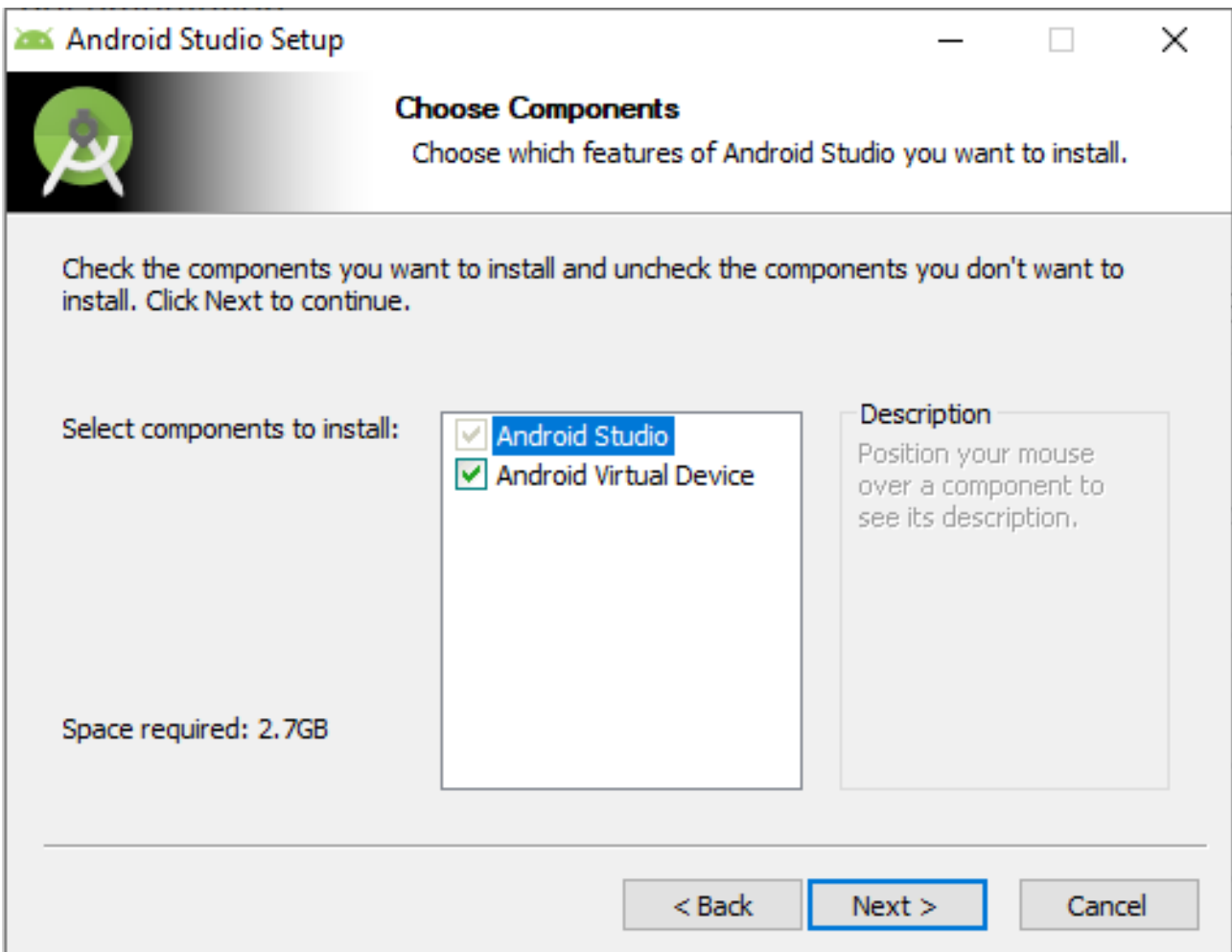
Android Studio est généralement utilisé pour la création d'émulateurs d'appareils **Android** (smartphone, tablette, etc.). Il faut savoir qu'il est possible de lancer directement une application **ionic** sur un appareil réel si on en possède un.

android studio

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

[DOWNLOAD ANDROID STUDIO](#)

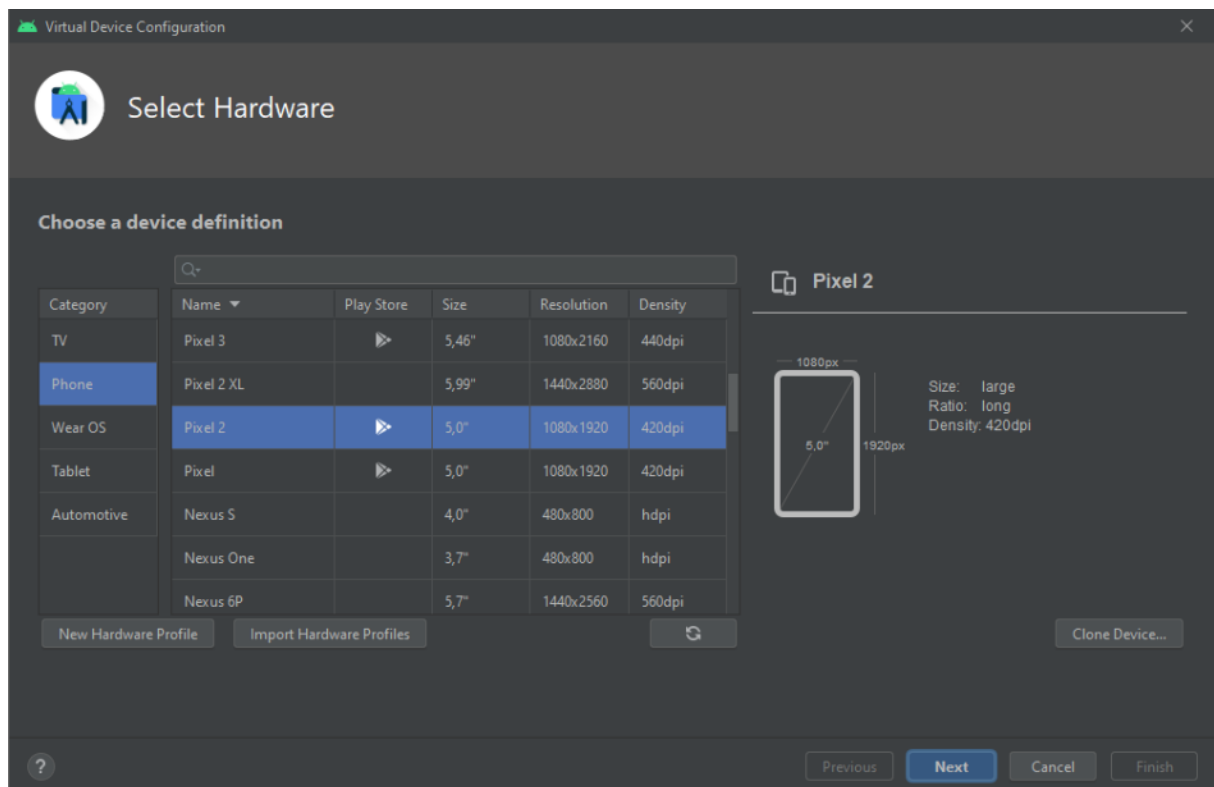
Le téléchargement de l'**IDE** est possible via le site web d'**Android** : <https://developer.android.com/studio/>¹ et l'on trouvera de plus amples informations sur le guide utilisateur visible à cette adresse²

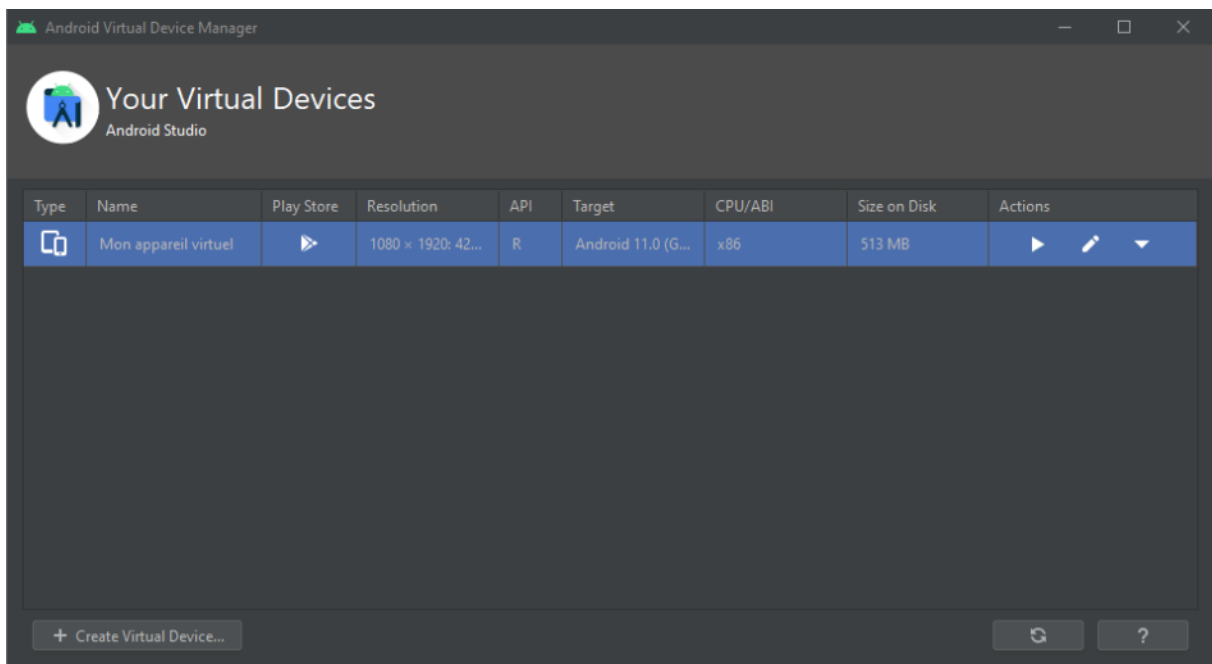
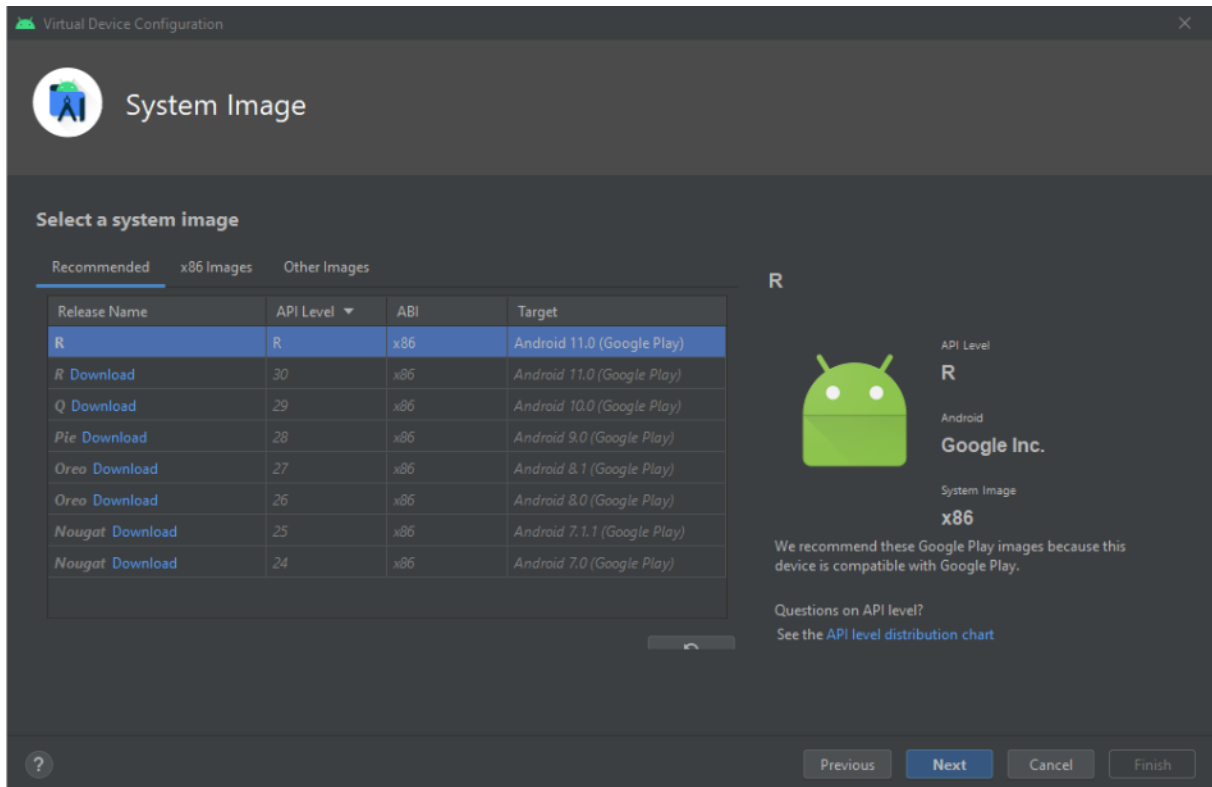


1 Android Studio
2 documentation installation

Création d'un appareil virtuel / simulateur Android

Pour créer un émulateur **Android**, il est nécessaire d'ouvrir le **ADV MANAGER** via **Android Studio**



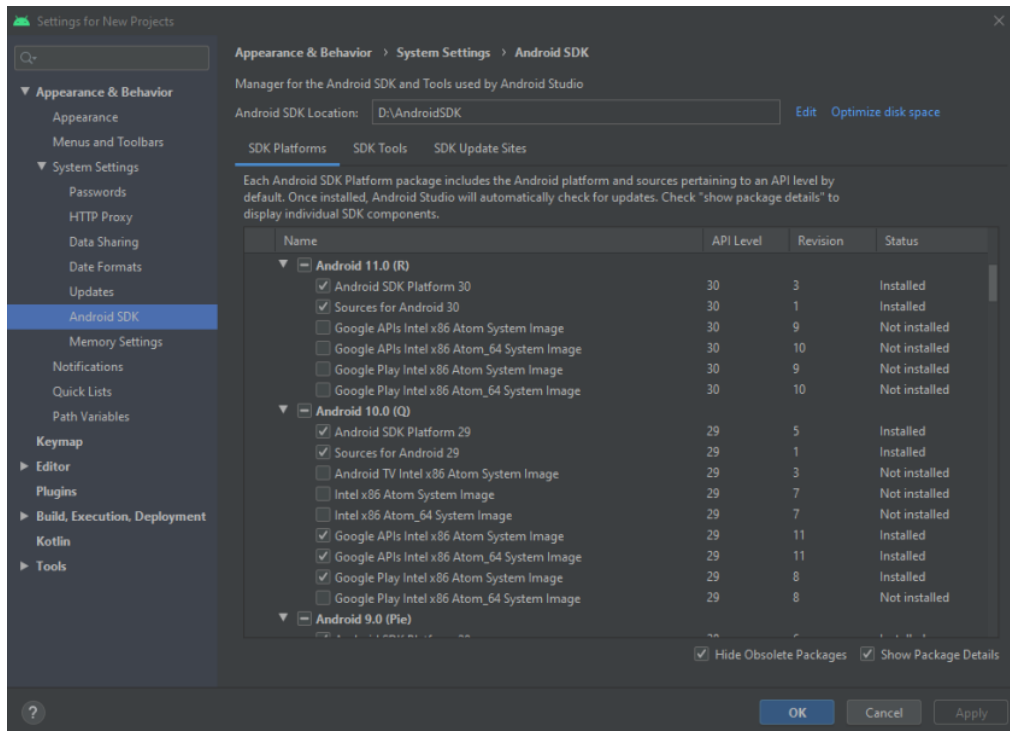


Installation des différents SDK Android

Par défaut, lors de l'installation d'**Android Studio**, seul le dernier SDK est téléchargé et installé.

Il est possible d'installer manuellement d'autres versions du **SDK Android** notamment pour gérer la rétrocompatibilité avec d'anciennes versions du système d'Android.

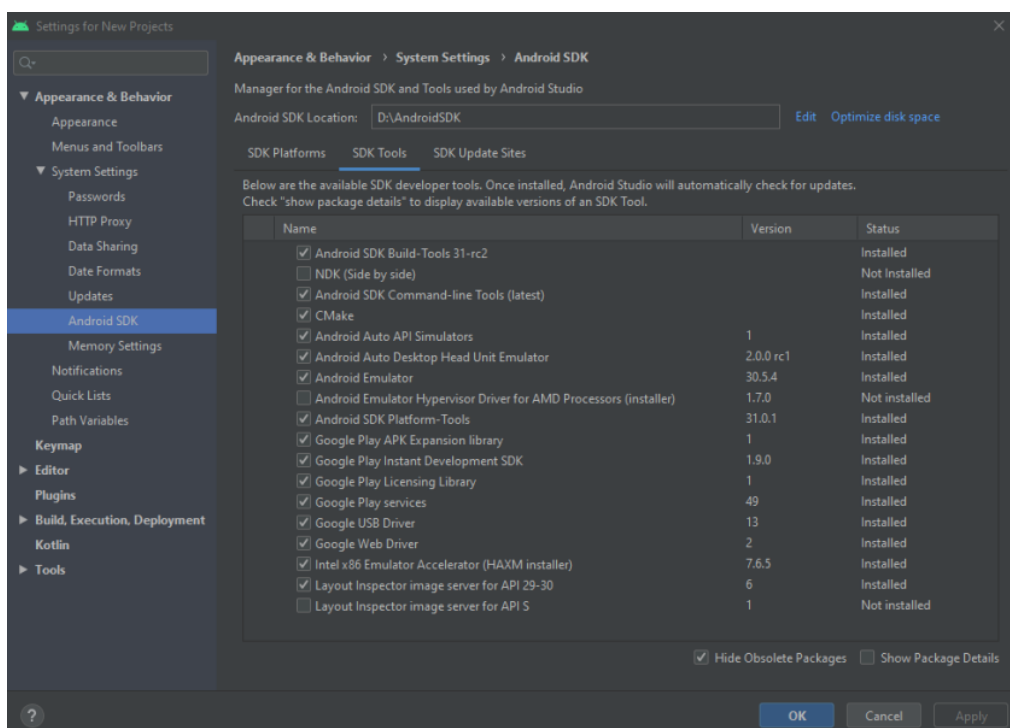
Pour cela il faut ouvrir le **SDK Manager** dans Android Studio.



Il est nécessaire également d'installer les **SDK Tools**, ces outils permettent de compiler l'application mais également de fournir des fonctions annexes.

Dans l'onglet SDK Tools, il faut installer au minimum la dernière version des outils suivants :

- Android SDK Build Tools
- Android SDK Command Line Tools
- Android Emulator
- Android SDK Platform Tools



Configuration des Variables d'Environnement

Certaines variables d'environnement sont nécessaires pour faire fonctionner correctement **Android Studio** et la compilation de projet **Ionic**.

Ajouter les variables suivantes :

- **ANDROID_SDK_ROOT** : {Chemin_dossier_SDK}
- **ANDROID_AVD_HOME** : {Chemin_dossier_SDK}\avd

Dans la variable **PATH**, ajouter les lignes suivantes :

- %**ANDROID_SDK_ROOT**%\platform-tools
- %**ANDROID_SDK_ROOT**%\tools
- %**ANDROID_SDK_ROOT**%\tools\bin

Configuration supplémentaire

Android Studio a besoin des outils de développement **Java** et **Gradle** pour fonctionner correctement.

Ces outils sont normalement installés via l'installation d'Android Studio mais il peut arriver de devoir les installer manuellement.

- **Java** : JDK8 pour compiler l'application Android : <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>¹
- **Gradle**: Outil de `build` installable depuis l'url suivante : <https://gradle.org/install/>²

Remarque

Pour revenir sur ce qui a été dit en début de cours , bien qu'étant l'outil idéal pour tester les applications android développées avec Ionic, il n'est pas recommandé d'utiliser **Android Studio** pour développer directement ces applications **ionic**, pour cela nous recommandons Visual Studio Code ou tout autre IDE conseillé pour les framework JS comme Angular, Vue.js ou ReactJs.

Android Studio est cependant fortement recommandé pour la création et l'exécution d'applications **Android** natives et pour gérer le **SDK Android** et les émulateurs permettant de tester les applications android générées via Ionic.

Création d'un projet Ionic de base

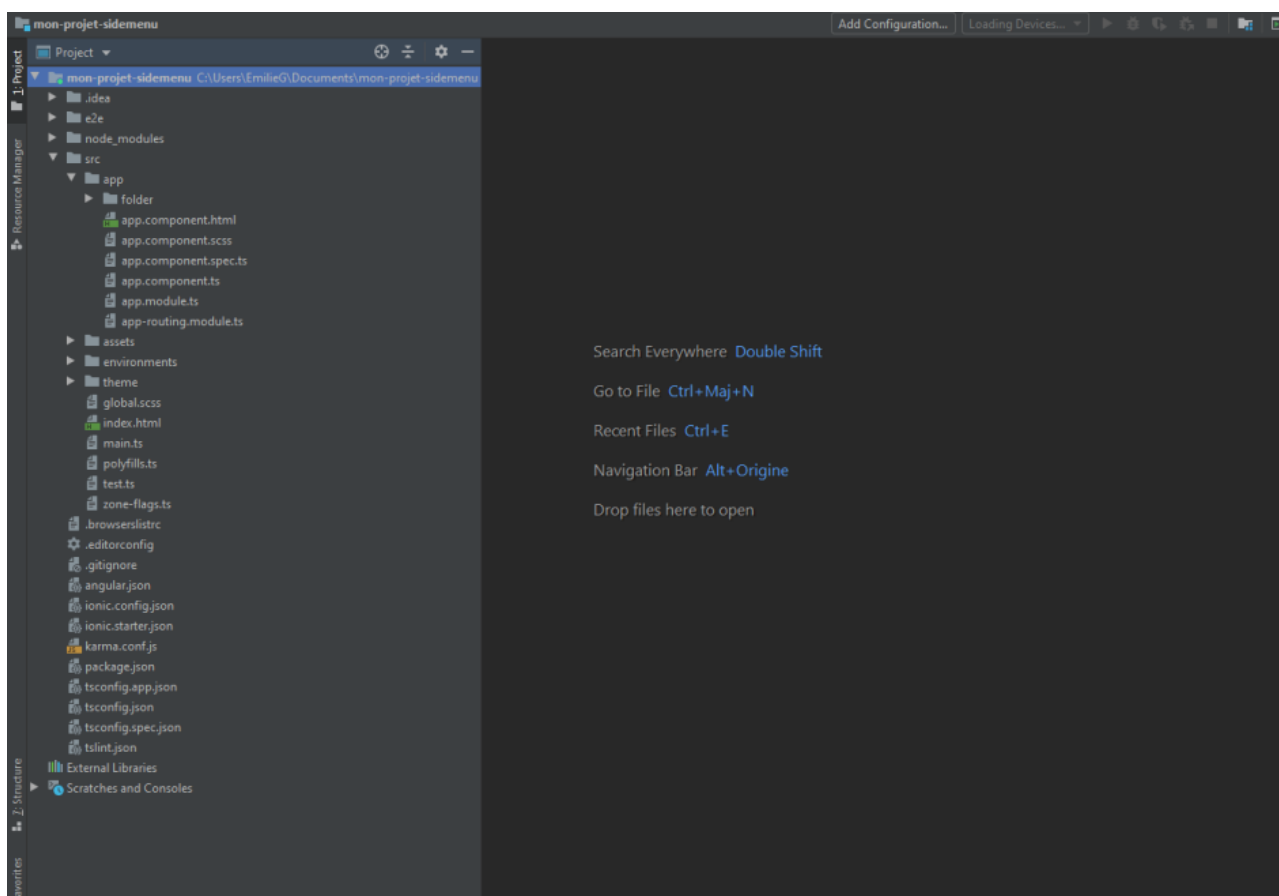
Avant de se pencher sur Android, nous allons au préalable créer une application simple sous ionic, celle-ci servira de référence par la suite pour son passage sous Android.

Nous allons créer une application avec le template sidemenu

```
1 ionic start mon-projet sidemenu --capacitor
```

1 java jdk
2 gradle

L'arborescence du projet créé est la suivante :



Remarque Intégration de Capacitor sur un projet existant

Il est possible d'intégrer **Capacitor** dans un projet Ionic existant.

```
1 ionic integrations enable capacitor
```

Initialisation du projet natif

Dans un premier temps, il va falloir renseigner deux variables qui sont nécessaires pour la création du projet Android. Lancer la commande suivante avec **appName** le nom de l'application et **appId** identifiant du domaine (ex : com.exemple.myapp)

```
1 npx cap init [appName] [appId]
```

Compilation du projet Ionic

Avant toutes opérations sur le projet **Android**, il faut que le projet **ionic** soit compilé au format **WebApp**. Il faut lancer la commande

```
1 ionic build
```

Cette commande va créer un dossier **dist** ou **www** à la racine du projet. Selon le dossier créé, il faut modifier dans le fichier **capacitor.config.json** la propriété **webDir**.

Création du projet Android

Pour ajouter et configurer la plateforme **Android** sur un projet **Ionic**, il suffit de se rendre dans le dossier du projet qui aura été au préalable créé et taper les commandes :

```
1 npx cap add android
```

L'exécution de cette commande permet de créer un dossier **android** à la racine du projet. Ce dossier va comporter un projet natif Android (Java/Kotlin) qui pourra être modifié directement depuis Android Studio.

Capacitor permet de faire une distinction nette entre le projet **Ionic** et le projet **Natif**. Tous ce qui concerne l'application Web devra être géré via Ionic. Tout ce qui concerne la partie Native (Icônes, Permissions, Signatures, etc.) devront être gérés via Android Studio.

Il est **important** que ce dossier soit enregistré dans le gestionnaire de sources pour ne pas perdre les configurations.

Mise à jour du projet Android

Si on effectue une mise à jour du projet Ionic, on peut mettre à jour notre projet natif via les deux commandes suivantes :

```
1 npx cap sync android  
2 npx cap copy android
```

La première commande va mettre à jour les **dépendances** du projet natif et copier les **assets Web**. Cette commande doit être exécutée si on ajoute un nouveau plugin Capacitor/Cordova ou qu'une version est modifiée.

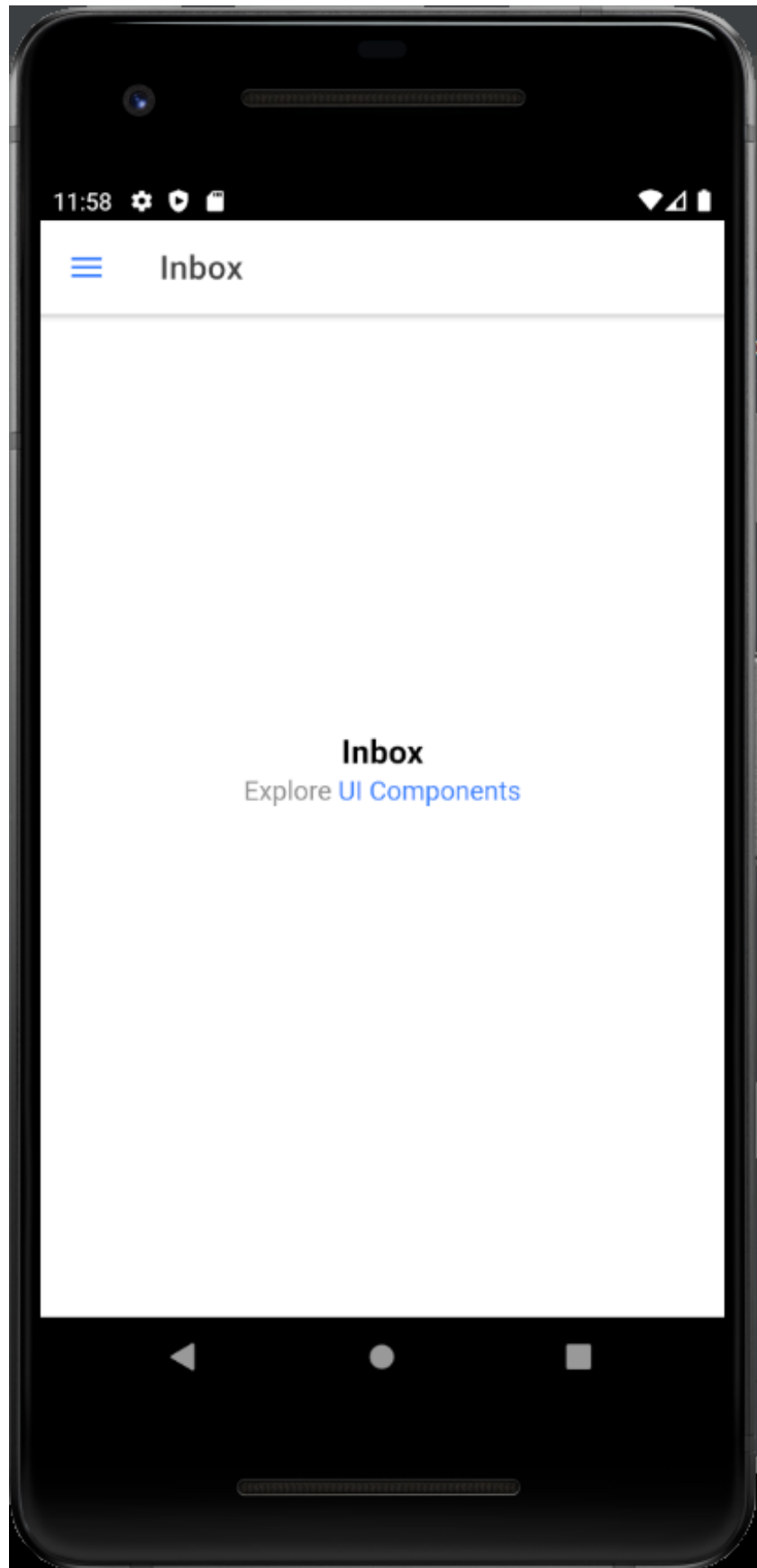
La seconde commande va uniquement copier les assets Web.

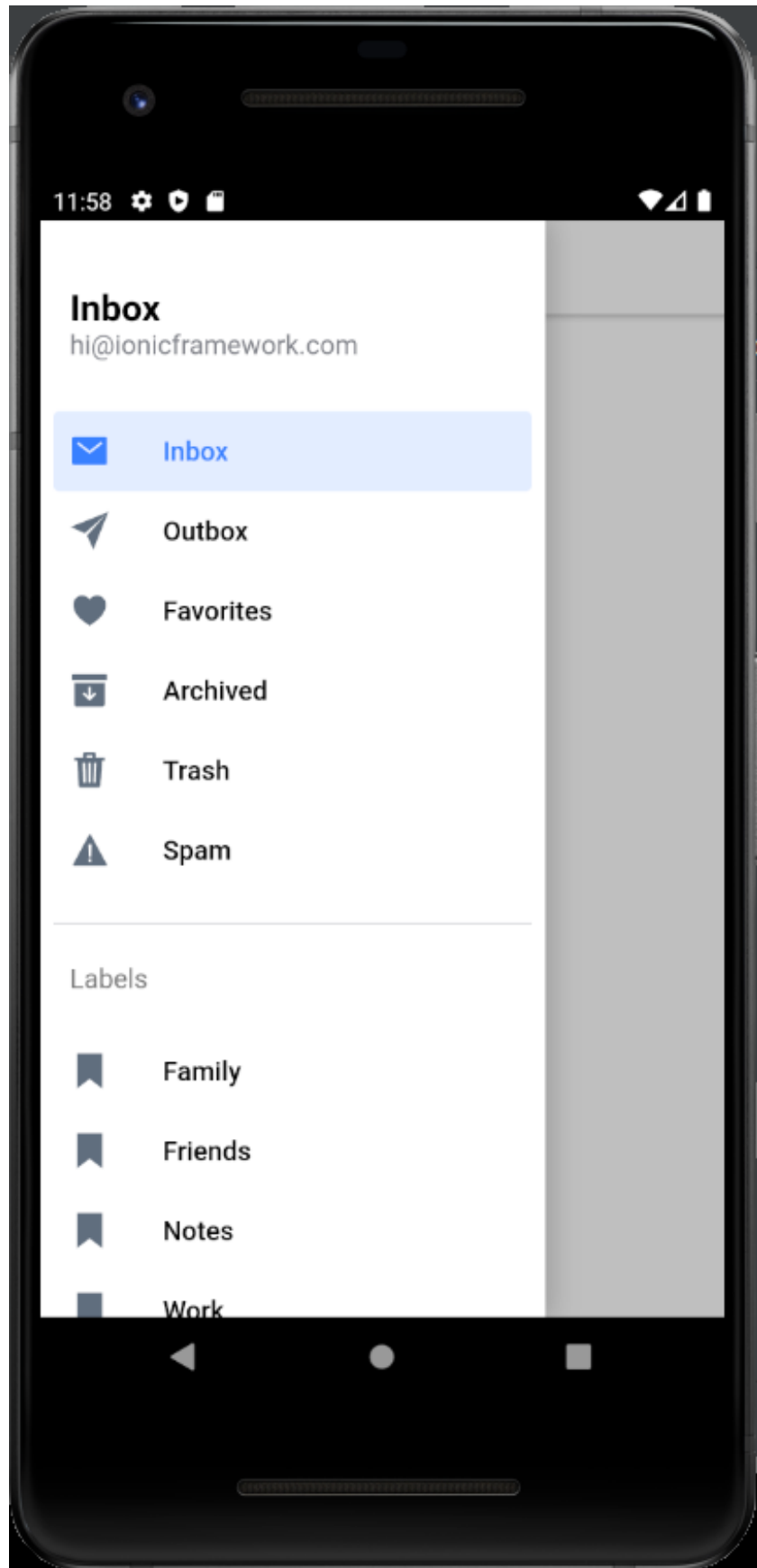
Lancement du projet Android

Pour ouvrir le projet Android, il faut lancer la commande

```
1 npx cap open android
```

Cette commande va ouvrir **Android Studio** directement sur le bon dossier. Ensuite il sera possible de lancer l'application sur un **émulateur** ou un **device réel** connecté à l'ordinateur.





Syntaxe **À retenir**

Pour développer sous Android il est indispensable d'avoir le **SDK Android**.

Android Studio Code est recommandé car il embarque le **SDK Android** et met à disposition des appareils virtuels pour tester les applications Android générées.

Il faut bien penser à gérer les **variables d'environnement** pour éviter tout problème lié.

Se rappeler que le projet **Ionic** doit être compilé au format **WebApp**.

Si l'on ajoute un nouveau plugin Capacitor/Cordova ou qu'une version est modifiée, la commande **npx cap sync android** doit être exécutée.

III. Quiz

Exercice 1

[solution n°1 p.25]

Exercice

Quel est l'outil qui permet la gestion des appareils émulés ?

- Android SDK
- Android AVD
- Android SKD
- Android ADV

Exercice

Quels sont les outils qu'il faut installer à minima sur Android Studio ?

- Android SDK Build Tools
- Ionic Emulator Android
- Android SDK Command Line Tools
- Android Emulator
- Ionic Build Tools
- Android SDK Platform Tools

Exercice

Quelle est la commande à utiliser pour intégrer Capacitor dans un projet Ionic existant ?

- `ionic integration enable capacitor`
- `ionic integrations activate capacitor`
- `ionic integrations enable capacitor`
- `ionic activate integrations capacitor`

Exercice

Une fois la commande d'initialisation du projet exécutée, quel est le nom du dossier qui va contenir les fichiers générés et le code de l'application ?

- dist
- src
- www
- root
- webapp

Exercice

Quelle est la commande qui permet d'ouvrir un projet Android ?

- android-studio open <projectname>
- npx cap open android
- npx cap launch android
- android open cap npx

IV. Environnement iOS et Configuration d'un projet Ionic sous iOS

Objectifs

- Installer XCode
- Créer un projet Ionic pour iOS
- Configurer XCode

Mise en situation

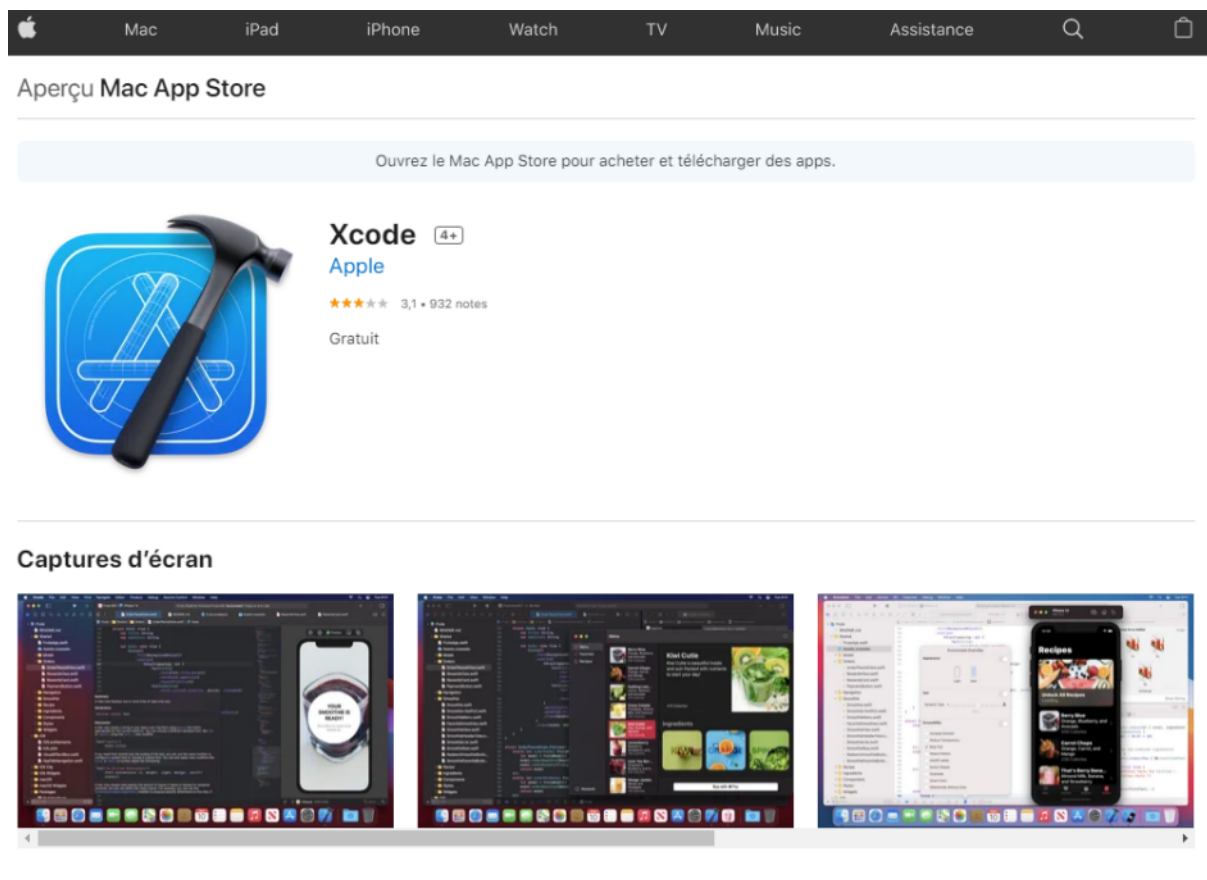
Pour développer une application iOS, il faut également mettre en place un environnement de développement approprié. Nous allons commencer par présenter les outils de base nécessaires pour travailler efficacement. Puis nous verrons ensuite comment configurer un projet Ionic pour iOS, cela ressemble beaucoup à la configuration d'un projet android à quelques détails près.

XCode

Pour développer une application pour **iOS**, il est nécessaire d'installer un environnement de développement spécifique à la plate-forme. Dans un premier temps, il faut absolument posséder une machine qui fonctionne via **MacOS**. L'installation de tous les outils nécessaires est effectuée via l'installation de l'IDE **XCode**.

Installation de XCode et des outils

Pour installer **XCode**, il faut passer depuis l'**AppStore** (d'où la nécessité d'avoir un compte **Apple**).



Xcode includes everything developers need to create great applications for Mac, iPhone, iPad, Apple TV, and Apple Watch. Xcode provides developers a unified workflow for user interface design, coding, testing, and debugging. The Xcode IDE combined with the Swift programming language make developing apps easy and fun.

[Plus](#)

Une fois l'installation terminée, il faut installer les outils via la ligne de commande. Lancer la commande suivante :

```
1 xcode-select --install
```

Remarque

Si la machine sous **MacOS** est différente de celle utilisée pour **Android**, il est nécessaire d'installer sur cette machine **NodeJS** et **NPM** ainsi que les packages **@ionic/cli**.

Connexion Compte Apple

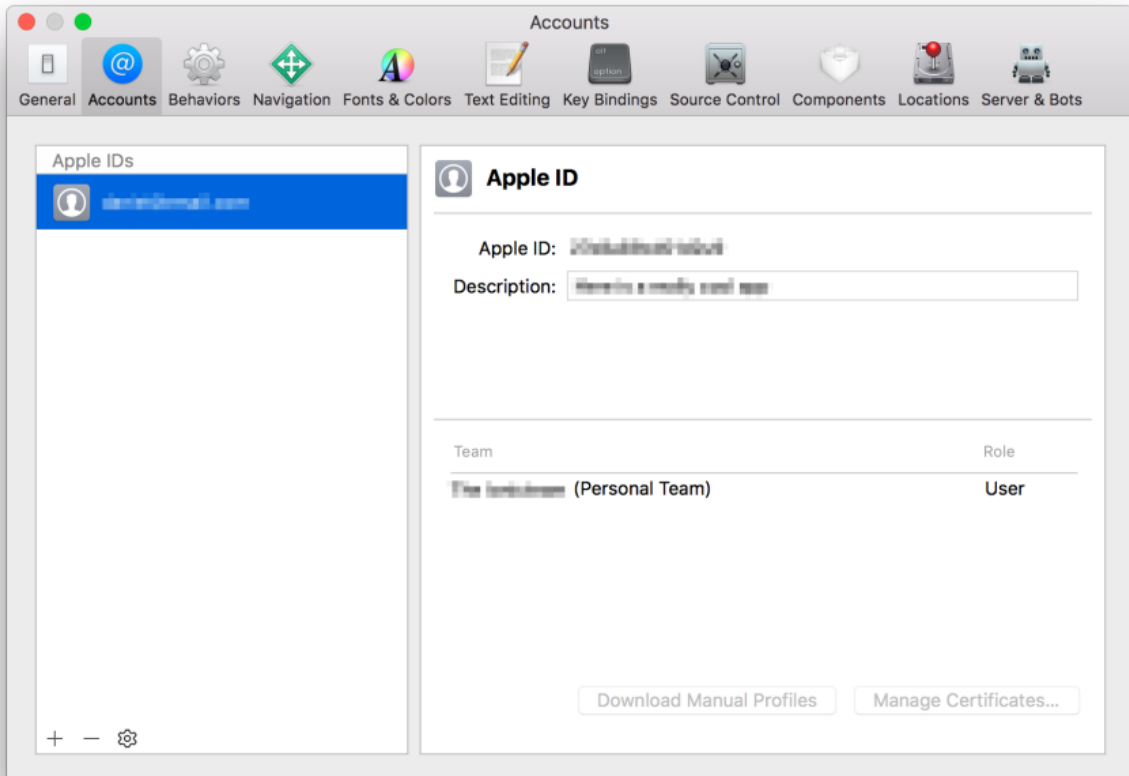
Pour permettre de tester des applications **iOS** sur des émulateurs ou des appareils physiques, il est obligatoire de se connecter à un compte **Apple** pour pouvoir récupérer les **Provisionnings Profiles** (Profils de signature).

En fonction du type de compte ajouté, les fonctions ne seront pas toutes disponibles.

- Utilisation de son compte Apple ID, création d'un profil "**Personnal Team**". Ce profil est le minimum requis.
- Utilisation d'un compte Developer, création d'un profil "**Developer Team**". Ce profil est obligatoire pour pouvoir connecter son device physique et pouvoir tester son application.
- Utilisation d'un compte Enterprise, création d'un profil "**Enterprise**". Ce profil permettra de déployer directement l'application sous les devices enregistrés dans le compte.

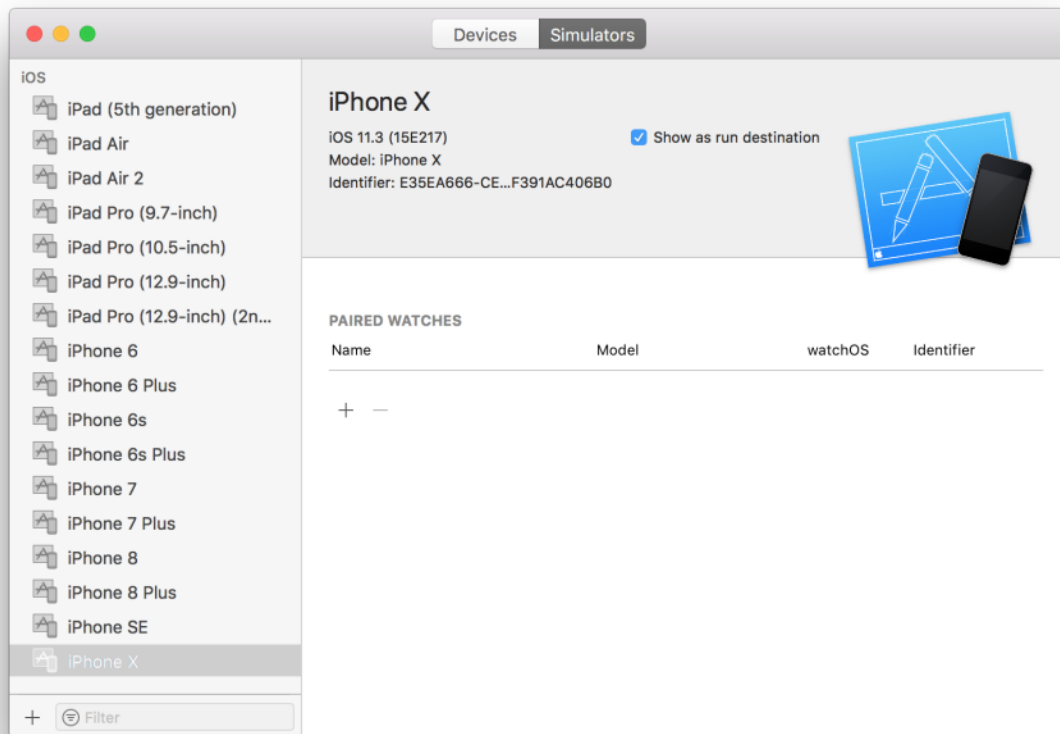
Pour activer son compte developper, il faut se connecter au site : <https://developer.apple.com/> et acheter une licence (99 \$ / an).

Ouvrir **XCode**, dans les préférences (**XCode** » **Préférences** » **Accounts**), ajouter les informations de connexion de son compte **Apple**. La Team va être automatiquement mise à jour.



Gérer les émulateurs

Par défaut, dans **XCode**, il existe déjà un certain nombre de simulateurs déjà disponible que ce soit pour les **iPhone** ou le **iPad**. Pour vérifier la liste, il faut aller dans le menu **Window** » **Devices and Simulators**. Il est possible d'ajouter des nouveaux modèles.



Remarque

Au niveau de la création et la configuration du projet pour être compatible avec iOS est identique aux étapes effectuées pour Android. Il n'est pas nécessaire de les refaire, il est possible d'utiliser le même projet pour les deux systèmes.

Compilation du projet Ionic

Avant toutes opérations sur le projet **iOS**, il faut que le projet **Ionic** soit compilé au format **WebApp**. Il faut lancer la commande

```
1 ionic build
```

Cette commande va créer un dossier **dist** ou **www** à la racine du projet. Selon le dossier créé, il faut modifier dans le fichier **capacitor.config.json** la propriété **webDir**.

Création du projet iOS

Pour ajouter et configurer la plateforme **iOS** sur un projet **Ionic**, il suffit de se rendre dans le dossier du projet qui aura été au préalable créé et taper les commandes :

```
1 npx cap add ios
```

L'exécution de cette commande permet de créer un dossier **ios** à la racine du projet. Ce dossier va comporter un projet natif **iOS** (Swift / Objective-C) qui pourra être modifié directement depuis **XCode**.

Capacitor permet de faire une distinction nette entre le projet **Ionic** et le projet **Natif**. Tout ce qui concerne l'application Web devra être géré via Ionic. Tout ce qui concerne la partie Native (Icônes, Permissions, Signatures, etc.) devront être gérés via XCode.

Il est **important** que ce dossier soit enregistré dans le gestionnaire de sources pour ne pas perdre les configurations.

Mise à jour du projet iOS

Si on effectue une mise à jour du projet Ionic, on peut mettre à jour notre projet natif via les deux commandes suivantes :

```
1 npx cap sync ios
2 npx cap copy ios
```

La première commande va mettre à jour les **dépendances** du projet natif et copier les **assets Web**. Cette commande doit être exécutée si on ajoute un nouveau plugin Capacitor/Cordova ou qu'une version est modifiée.

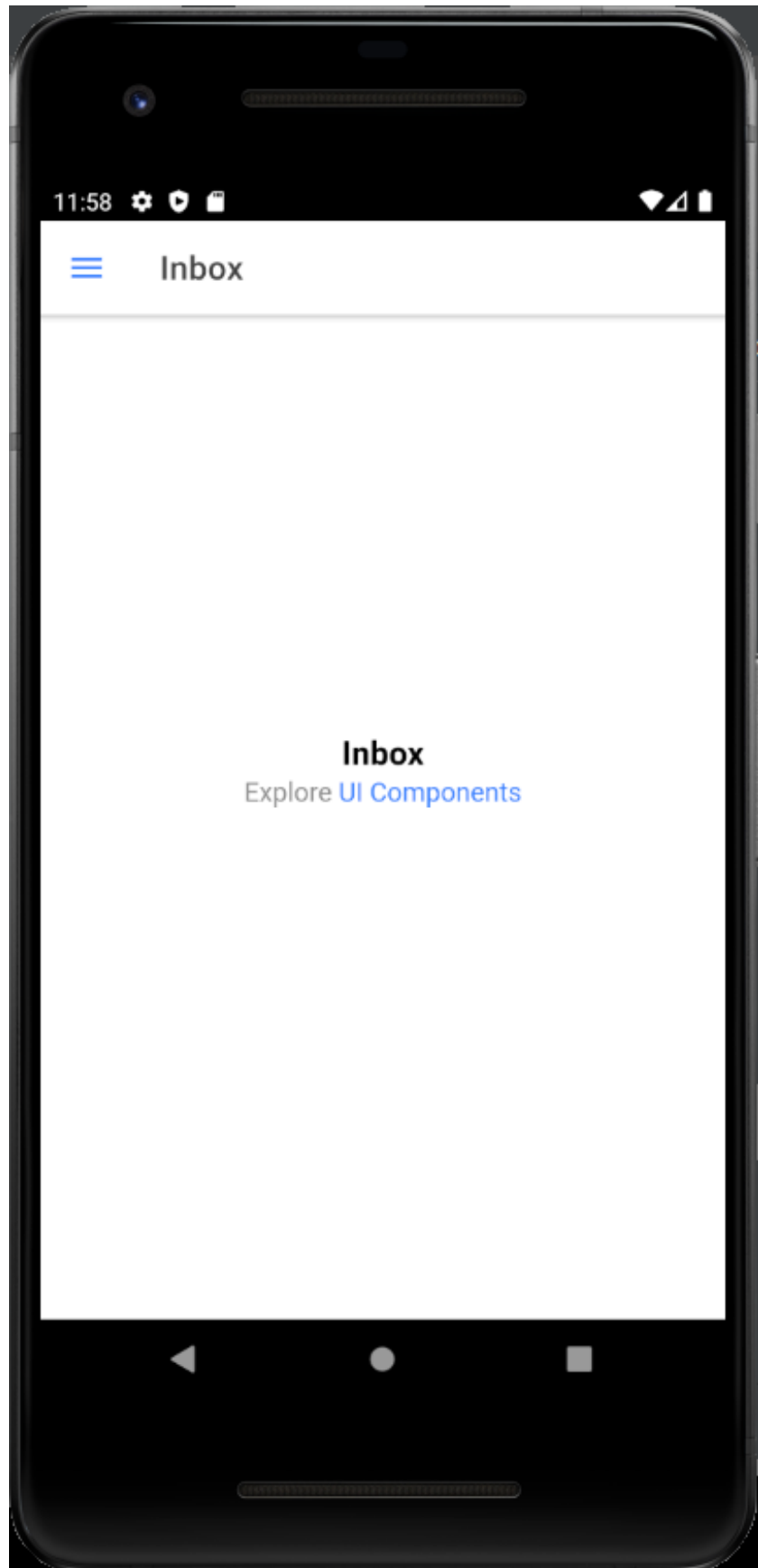
La seconde commande va uniquement copier les assets Web.

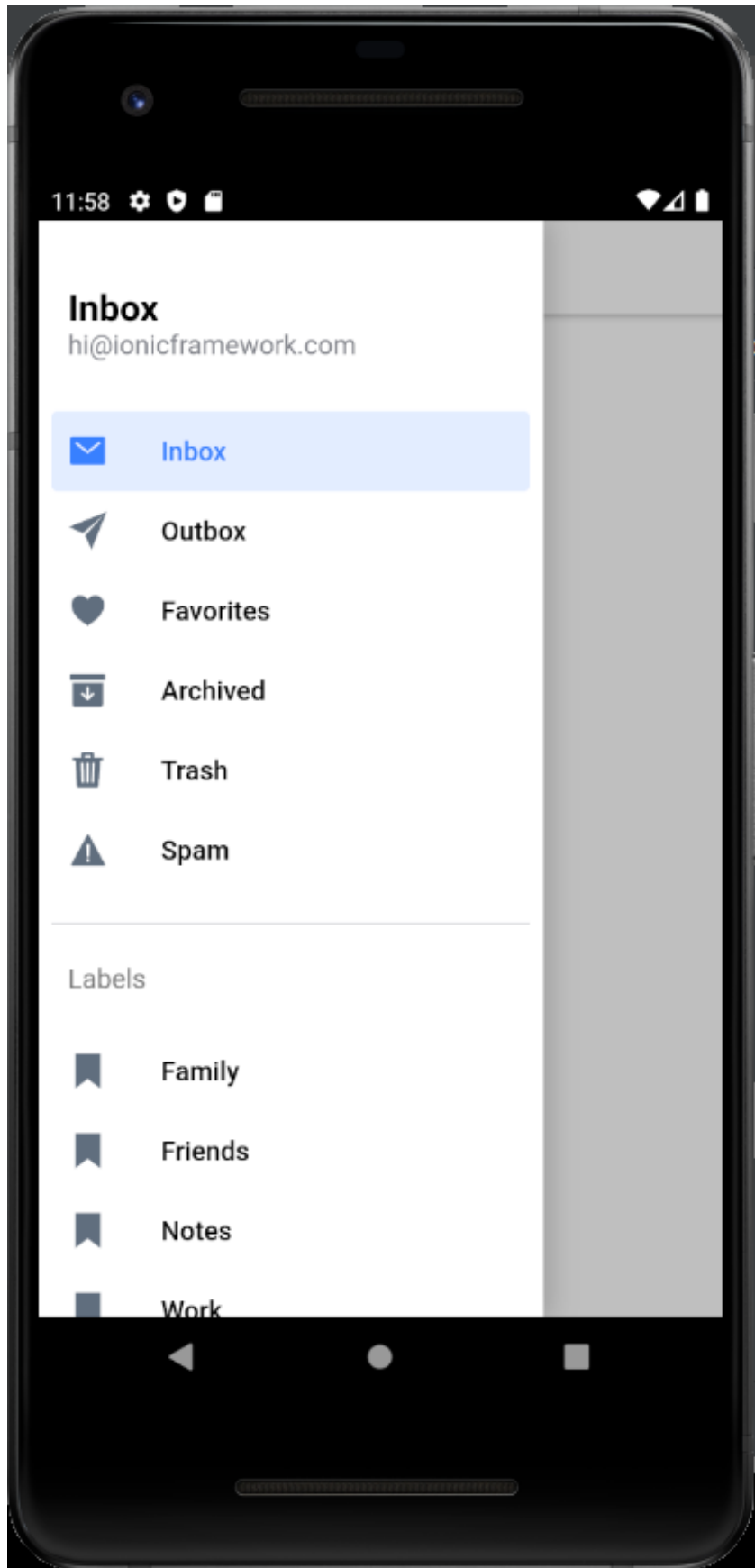
Lancement du projet iOS

Pour ouvrir le projet **iOS**, il faut lancer la commande

```
1 npx cap open ios
```

Cette commande va ouvrir **XCode** directement sur le bon dossier. Ensuite il sera possible de lancer l'application sur un **émulateur** ou un **device réel** connecté à l'ordinateur.





Syntaxe **À retenir**

XCode se récupère sur l'**AppStore** et nécessite un compte **Apple**.

L'installation de xCode permet également de configurer tout les outils nécessaire au développement d'application Ionic pour iOS.

Il est possible d'ajouter des modèles de simulateurs supplémentaires si ceux par défaut ne conviennent pas aux besoins.

La configuration d'un projet iOS est relativement similaire à celle d'un projet Android.

V. Quiz

Exercice 1

[solution n°2 p.26]

Exercice

Quels sont les systèmes d'exploitation qui permettent le développement d'applications d'iOS ?

- Ubuntu
- Windows
- Manjaro
- MacOS
- ArchLinux

Exercice

Quels sont les types de profil nécessaires pour développer, tester et déployer une application sur l'AppStore ?

- Personal Team
- Family Team
- Enterprise
- Developer Team
- Professionnal Team

Exercice

Quelles sont les étapes à réaliser pour rendre un projet Ionic android compatible avec iOS ?

- ionic migrate-ios <projectname>
- ionic convert <projectname>
- npx cap migrate ios
- Aucune

Exercice

Quel est le coût de la licence pour déposer une application sur l'AppStore ?

- 50\$
- 99\$
- 10\$
- 159.99\$

VI. Essentiel

VII. Exercice final

Exercice 1

[solution n°3 p.27]

Exercice

Quelle solution est utilisée dans ce cours pour obtenir des applications natives ?

- Cordova
- PhoneGap
- Capacitor
- NativeScript
- React-Native

Exercice

Quel est le nom de l'IDE pour développer des applications pour Android ?

- Android Studio
- Visual Studio Code
- XCode
- Android Code

Exercice

Quel est le nom de l'IDE pour développer des applications pour iOS ?

- Android Studio
- Visual Studio Code
- XCode
- Android Code

Exercice

Quel est le nom de la commande pour créer un projet natif Android ?

- npx cap init android
- npx cap add android
- npx cap sync android
- npx cap copy android

Exercice

Quel est le nom de la commande pour mettre à jour les dépendances pour un projet natif iOS ?

- npx cap init ios
- npx cap add ios
- npx cap sync ios
- npx cap copy ios

Exercice

Quel est le nom de la commande pour mettre à jour l'application Web pour Android ?

- npx cap init android
- npx cap add android
- npx cap sync android
- npx cap copy android

Exercice

Quel est le nom du fichier pour gérer la configuration d'un projet natif Android ?

- AndroidManifest.xml
- AndroidProperty.xml
- android.plist
- android.properties

Exercice

Quel est le nom du fichier pour gérer la configuration d'un projet natif iOS ?

- iosManifest.xml
- iosProperty.xml
- Info.plist
- ios.properties

Exercice

Quel est l'outil qui permet de récupérer les dépendances à des librairies pour les applications Android ?

- NPM
- Maven
- Nuget
- Gradle

Exercice

Quel type de compte est nécessaire pour déployer une application sur un device réel sous iOS ?

- Compte **Apple ID**
- Compte **Apple Developer**
- Compte **Apple Enterprise**

Exercice

En quel format est compilé un projet Ionic ?

- Apk
- WebApp
- Npx
- Natif

Exercice

En quel langage est codé une application Android ?

- Kotlin
- Python
- Scala
- C++
- Java

Solutions des exercices

Exercice p. 13 Solution n°1**Exercice**

Quel est l'outil qui permet la gestion des appareils émulés ?

- Android SDK
- Android AVD
- Android SKD
- Android ADV

Exercice

Quels sont les outils qu'il faut installer à minima sur Android Studio ?

- Android SDK Build Tools
- Ionic Emulator Android
- Android SDK Command Line Tools
- Android Emulator
- Ionic Build Tools
- Android SDK Platform Tools

Exercice

Quelle est la commande à utiliser pour intégrer Capacitor dans un projet Ionic existant ?

- `ionic integration enable capacitor`
- `ionic integrations activate capacitor`
- `ionic integrations enable capacitor`
- `ionic activate integrations capacitor`

Exercice

Une fois la commande d'initialisation du projet exécutée, quel est le nom du dossier qui va contenir les fichiers générés et le code de l'application ?

- `dist`
- `src`
- `www`
- `root`
- `webapp`

Exercice

Quelle est la commande qui permet d'ouvrir un projet Android ?

- android-studio open <projectname>
- npx cap open android
- npx cap launch android
- android open cap npx

Exercice p. 21 Solution n°2**Exercice**

Quels sont les systèmes d'exploitation qui permettent le développement d'applications d'iOS ?

- Ubuntu
- Windows
- Manjaro
- MacOS
- ArchLinux

Exercice

Quels sont les types de profil nécessaires pour développer, tester et déployer une application sur l'AppStore ?

- Personal Team
- Family Team
- Enterprise
- Developer Team
- Professionnal Team

Exercice

Quelles sont les étapes à réaliser pour rendre un projet Ionic android compatible avec iOS ?

- ionic migrate-ios <projectname>
- ionic convert <projectname>
- npx cap migrate ios
- Aucune

Exercice

Quel est le coût de la licence pour déposer une application sur l'AppStore ?

- 50\$
- 99\$
- 10\$
- 159.99\$

Exercice p. 22 Solution n°3**Exercice**

Quelle solution est utilisée dans ce cours pour obtenir des applications natives ?

- Cordova
- PhoneGap
- Capacitor
- NativeScript
- React-Native

Exercice

Quel est le nom de l'IDE pour développer des applications pour Android ?

- Android Studio
- Visual Studio Code
- XCode
- Android Code

Exercice

Quel est le nom de l'IDE pour développer des applications pour iOS ?

- Android Studio
- Visual Studio Code
- XCode
- Android Code

Exercice

Quel est le nom de la commande pour créer un projet natif Android ?

- npx cap init android
- npx cap add android
- npx cap sync android
- npx cap copy android

Exercice

Quel est le nom de la commande pour mettre à jour les dépendances pour un projet natif iOS ?

- npx cap init ios
- npx cap add ios
- npx cap sync ios
- npx cap copy ios

Exercice

Quel est le nom de la commande pour mettre à jour l'application Web pour Android ?

- npx cap init android
- npx cap add android
- npx cap sync android
- npx cap copy android

Exercice

Quel est le nom du fichier pour gérer la configuration d'un projet natif Android ?

- AndroidManifest.xml
- AndroidProperty.xml
- android.plist
- android.properties

Exercice

Quel est le nom du fichier pour gérer la configuration d'un projet natif iOS ?

- iosManifest.xml
- iosProperty.xml
- Info.plist
- ios.properties

Exercice

Quel est l'outil qui permet de récupérer les dépendances à des bibliothèques pour les applications Android ?

- NPM
- Maven
- Nuget
- Gradle

Exercice

Quel type de compte est nécessaire pour déployer une application sur un device réel sous iOS ?

- Compte **Apple ID**
- Compte **Apple Developer**
- Compte **Apple Enterprise**

Exercice

En quel format est compilé un projet Ionic ?

- Apk
- WebApp
- Npx
- Natif

Exercice

En quel langage est codé une application Android ?

- Kotlin
- Python
- Scala
- C++
- Java