Apache Cordova

Sommaire

1. Apache Cordova . . . 1
1.1. Installation . . . 3
1.2. Cordova pour Android . . . 5
1.2.1. Installation Android 4.1 (Ubuntu 20.04) . . . 6
1.2.2. Java SDK . . . 6
1.2.3. Cordova CLI . . . 6
1.3. Cordova pour iOS . . . 7
2. Test : premier projet . . . 7
2.1. Android . . . 7
2.2. iOS . . . 15
3. Les plugins . . . 23
4. Voir aussi . . . 24

Thierry Vaira - <tvaira@free.fr> - version v1.0 - 05/01/2021

1. Apache Cordova

Apache Cordova (ou plus anciennement **PhoneGap**) est un *framework* open source développé par la Fondation Apache qui permet de créer des applications pour différentes plateformes (**Android**, Firefox OS, **iOS**, Windows 8 ...) en **HTML, CSS et JavaScript**.

Liste des fonctionnalités disponibles pour les plateformes **Android** et **iOS** :

Fonctionnalités	Android	iOS
Accéléromètre	✓ Oui	✓ Oui
Caméra	✓ Oui	✓ Oui
Boussole	✓ Oui	✓ Oui
Contacts	✓ Oui	✓ Oui
Écran d'accueil	✓ Oui	✓ Oui
Gestion des fichiers	✓ Oui	✓ Oui
Géolocalisation	✓ Oui	✓ Oui
Internationalisation	✓ Oui	✓ Oui
Media	✓ Oui	✓ Oui
Réseau	✓ Oui	✓ Oui
Notifications (Apple Push, pop-up)	✓ Oui	✓ Oui
Stockage	✓ Oui	✓ Oui

Les applications qui en résultent sont couramment appelées **hybrides**, ce qui signifie qu'elles ne sont ni vraiment natives, ni purement basées sur les langages HTML, CSS et JavaScript.

https://fr.wikipedia.org/wiki/Apache_Cordova

Lien : https://cordova.apache.org/

1.1. Installation

Cordova fournit une interface en ligne de commande (CLI). Celle-ci nécessite l'environnement `Node.js`.

Node.js est un environnement d'exécution JavaScript multiplateforme conçu pour exécuter du code JavaScript côté serveur. Il est généralement utilisé pour créer des applications *back-end*, mais il est également populaire en tant que solution *full-stack* et *front-end*. `npm` est le gestionnaire de *packages* par défaut pour Node.js.

L'outil `nvm` (*Node Version Manager*) permet d'installer plusieurs versions de Node.js sur la même machine. Pour les développeurs, c'est la méthode conseillée.

Installation de la dernière version LTS de `Node.js` et `npm` à l'aide de `nvm` :

- Sous Ubuntu 20.04 :

```
$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.35.3/install.sh | bash
 % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                Dload  Upload   Total   Spent    Left  Speed
100 13527  100 13527    0     0  57561      0 --:--:-- --:--:-- --:--:-- 57561
=> Downloading nvm from git to '/home/tvaira/.nvm'
=> Clonage dans '/home/tvaira/.nvm'...
remote: Enumerating objects: 290, done.
remote: Counting objects: 100% (290/290), done.
remote: Compressing objects: 100% (257/257), done.
remote: Total 290 (delta 34), reused 111 (delta 20), pack-reused 0
Réception d'objets: 100% (290/290), 163.37 Kio | 217.00 Kio/s, fait.
Résolution des deltas: 100% (34/34), fait.
=> Compressing and cleaning up git repository

=> Appending nvm source string to /home/tvaira/.bashrc
=> Appending bash_completion source string to /home/tvaira/.bashrc
=> Close and reopen your terminal to start using nvm or run the following to use it
now:

$ source $HOME/.bashrc

$ nvm --version
0.35.3

$ nvm install --lts
Installing latest LTS version.
Downloading and installing node v14.15.4...
Downloading https://nodejs.org/dist/v14.15.4/node-v14.15.4-linux-x64.tar.xz...
```

```
######################################################################################################
#################################### 100,0%
Computing checksum with sha256sum
Checksums matched!
Now using node v14.15.4 (npm v6.14.10)
Creating default alias: default -> lts/* (-> v14.15.4)

$ nvm ls
>     v14.15.4
         system
default -> lts/* (-> v14.15.4)
node -> stable (-> v14.15.4) (default)
stable -> 14.15 (-> v14.15.4) (default)
iojs -> N/A (default)
unstable -> N/A (default)
lts/* -> lts/fermium (-> v14.15.4)
lts/argon -> v4.9.1 (-> N/A)
lts/boron -> v6.17.1 (-> N/A)
lts/carbon -> v8.17.0 (-> N/A)
lts/dubnium -> v10.23.1 (-> N/A)
lts/erbium -> v12.20.1 (-> N/A)
lts/fermium -> v14.15.4

$ node --version
v14.15.4

$ npm --version
6.14.10
```

- sous macOS :

```
$ cd ~
$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.37.2/install.sh | bash
$ vim ~/.zshrc
export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm

$ source ~/.zshrc

$ nvm install --lts
```

Installation de Cordova :

```
$ npm install -g cordova

$ cordova --version
10.0.0
```

Ensuite, il faut préparer les dépendances nécessaires pour les plateformes supplémentaires comme **Android** et/ou **iOS**.

Liens :

- Guide pour la plate-forme Android
- Guide pour la plate-forme iOS

1.2. Cordova pour Android

Cordova pour Android nécessite le **SDK Android**. Pour cela, il est possible de l'installer directement ou par **Android Studio**.

Android Studio est un environnement de développement pour développer des applications Android. Il est basé sur IntelliJ.

Android est un système d'exploitation mobile basé sur le noyau Linux et développé actuellement par Google.

Lien : Les différentes versions d'Android

Les différents kits de développement sont :

- l'Android **NDK** (*Android Native Development Kit*) est une API du système d'exploitation Android permettant de développer directement dans le langage C/C++ du matériel cible, par opposition au Android SDK qui est une abstraction en *bytecode* Java, indépendante du matériel.
- l'Android **SDK** est un ensemble complet d'outils de développement (pour Linux, MAC OS ou Windows). Il inclut un débogueur, des bibliothèques logicielles, un émulateur basé sur QEMU, de la documentation, des exemples de code et des tutoriaux.

L'**ADB** (*Android Debug Bridge*) est un outil inclus dans le package Android SDK. Il se compose d'un programme client et d'un programme serveur communicant entre eux et qui permet :

- la copie de fichier ;
- l'accès à la console Android ;
- la sauvegarde de la mémoire ROM ;
- l'installation de logiciel.

Ces outils sont disponibles aussi bien sous GNU/Linux que sous Windows et Mac OS. Les applications Android étant exécutées par une machine virtuelle, il n'y a pas d'avantages particuliers à développer sur un système plutôt qu'un autre (même si une machine Mac pourra développer pour **iOS** et Android ce qui ne sera pas le cas pour des machines GNU/Linux ou Windows pour iOS !) ...

1.2.1. Installation Android 4.1 (Ubuntu 20.04)

http://tvaira.free.fr/dev/android/installation-android.html (PDF)

Il sera probablement nécessaire d'installer :

- Java JDK (version 8)
- Android (API 22) platform SDK
- Android SDK Build-tools 19.1.0 ou supérieur

1.2.2. Java SDK

```
$ sudo apt install openjdk-8-jdk

$ sudo update-alternatives --config java
$ sudo update-alternatives --config javac

$ javac -version
javac 1.8.0_275

$ export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

1.2.3. Cordova CLI

Pour l'utiliser avec Cordova en ligne de commande (CLI), il faut inclure les répertoires des outils Android dans la variable d'environnement `PATH` et définir `ANDROID_HOME`.

L'idéal est de modifier le fichier `~/.bashrc` :

```
$ vim ~/.bashrc
export ANDROID_HOME=$HOME/Android/Sdk
export ANDROID_SDK_ROOT=$HOME/Android/Sdk
export PATH=${PATH}:$HOME/Android/Sdk/platform-tools:$HOME/Android/Sdk/tools
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

Pour les machines Mac, il faudra utiliser maintenant `~/.zshrc` pour le *shell* par défaut `zsh`. Et évidemment, changer les chemins !

Puis de le recharger :

```
$ source ~/.bashrc
```

La fabrication d'une application Android nécessite maintenant Gradle. Il faut le télécharger et l'installer (https://gradle.org/) ou utiliser celui déjà présent dans `$HOME/.gradle` si Android Studio a déjà été utilisé sur la machine :

```
$ ls -l $HOME/.gradle/wrapper/dists/
$ ls -l $HOME/.gradle/wrapper/dists/gradle-6.5-bin/

$ export PATH=$PATH:$HOME/.gradle/wrapper/dists/gradle-6.5-
bin/xxxxxxxxxxxx/gradle-6.5/bin
```

1.3. Cordova pour iOS

→ Installation de XCode : https://apps.apple.com/fr/app/xcode/id497799835?mt=12

→ Installation des outils de déploiement iOS :

```
$ npm install -g ios-sim
$ sudo xcode-select -switch /Applications/Xcode.app/Contents/Developer
$ npm install -g ios-deploy
```

2. Test : premier projet

```
$ cordova create hello com.example.hello HelloWorld
Creating a new cordova project.

$ cd hello/
```

2.1. Android

```
$ cordova platform add android
Using cordova-fetch for cordova-android@^9.0.0
Adding android project...
Creating Cordova project for the Android platform:
    Path: platforms/android
    Package: com.example.hello
    Name: HelloWorld
    Activity: MainActivity
    Android target: android-29
Subproject Path: CordovaLib
Subproject Path: app
Android project created with cordova-android@9.0.0
Discovered plugin "cordova-plugin-whitelist". Adding it to the project
Installing "cordova-plugin-whitelist" for android
Adding cordova-plugin-whitelist to package.json

$ cordova prepare
```

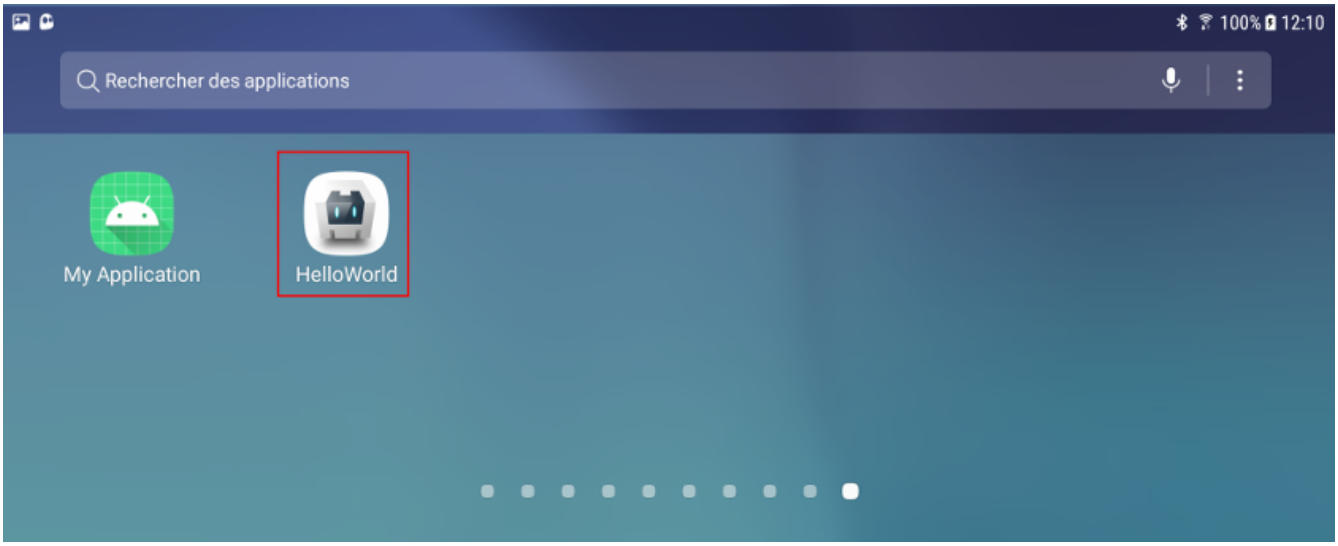
La commande `cordova build` permet d'effectuer successivement `cordova prepare` et `cordova compile`.

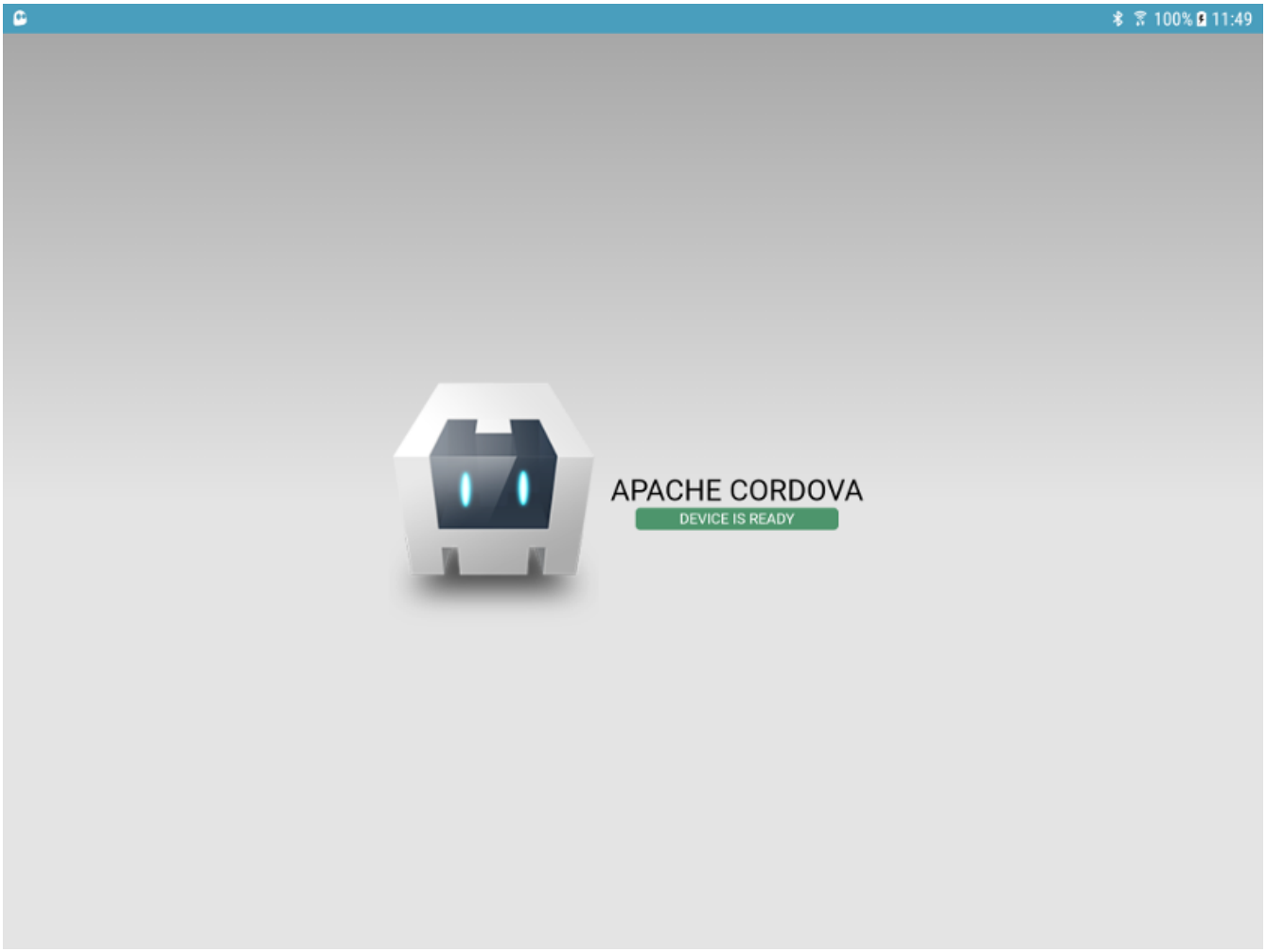
Les informations sur le projet :

```
$ cordova info

Cordova Packages:

    cli: 10.0.0
        common: 4.0.2
        create: 3.0.0
        lib: 10.0.0
            common: 4.0.2
            fetch: 3.0.0
            serve: 4.0.0

Project Installed Platforms:

    android: 9.0.0

Project Installed Plugins:

    cordova-plugin-whitelist: 1.3.4

Environment:

    OS: Ubuntu 20.04.1 LTS (linux 5.8.0-38-generic) x64
    Node: v14.15.4
    npm: 6.14.10

...
```

- Version *debug* :

```
$ cordova build android --debug
...
BUILD SUCCESSFUL in 1m 31s
40 actionable tasks: 40 executed
Built the following apk(s):
        /home/iris/hello/platforms/android/app/build/outputs/apk/debug/app-debug.apk

$ cordova run android --list
Available android devices:
d95f698401ba0ee8
Available android virtual devices:
Pixel_3a_API_30_x86

$ cordova run android --device
...
BUILD SUCCESSFUL in 2s
40 actionable tasks: 40 up-to-date
Built the following apk(s):
        /home/iris/hello/platforms/android/app/build/outputs/apk/debug/app-debug.apk
Checking Java JDK and Android SDK versions
ANDROID_SDK_ROOT=/home/iris/Android/Sdk (recommended setting)
ANDROID_HOME=/home/iris/Android/Sdk (DEPRECATED)
Using Android SDK: /home/iris/Android/Sdk
Using apk: /home/iris/hello/platforms/android/app/build/outputs/apk/debug/app-
debug.apk
Package name: com.example.hello
LAUNCH SUCCESS
```





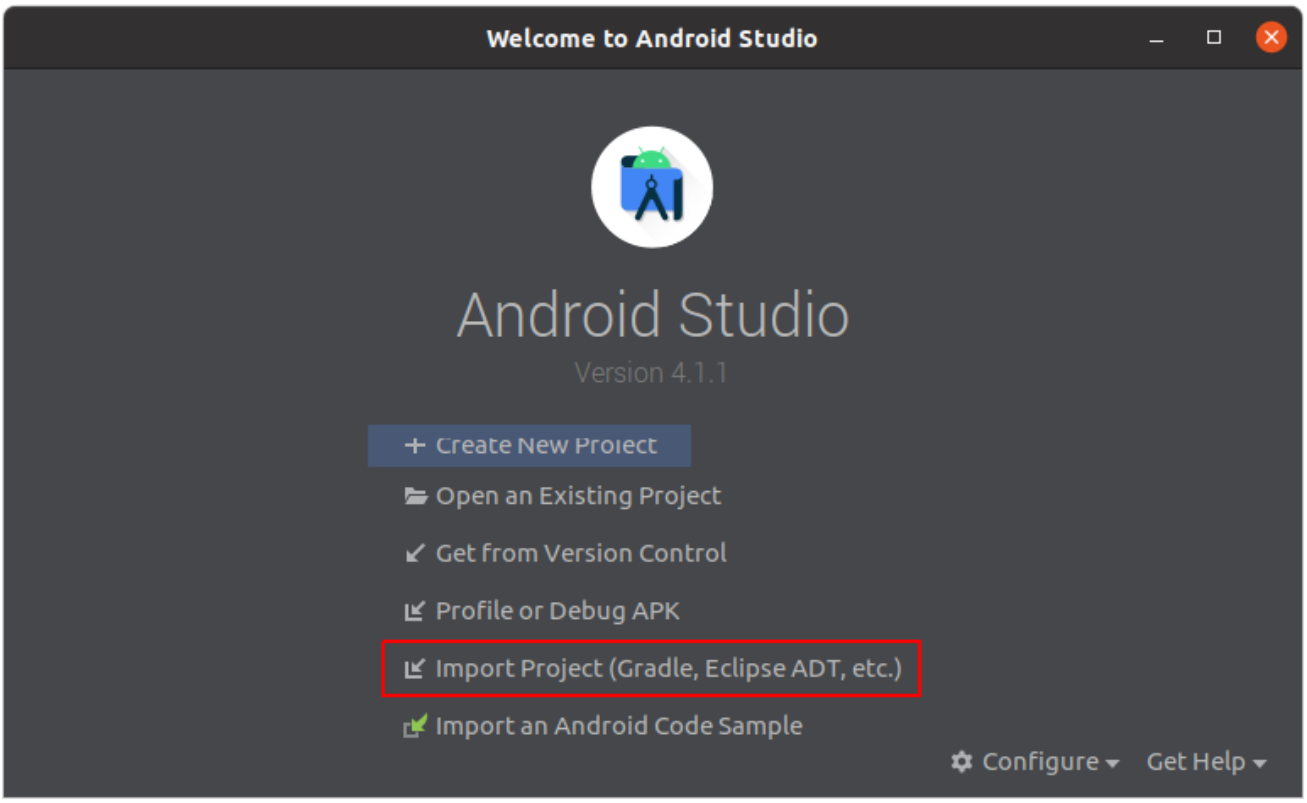
- Version *release* :

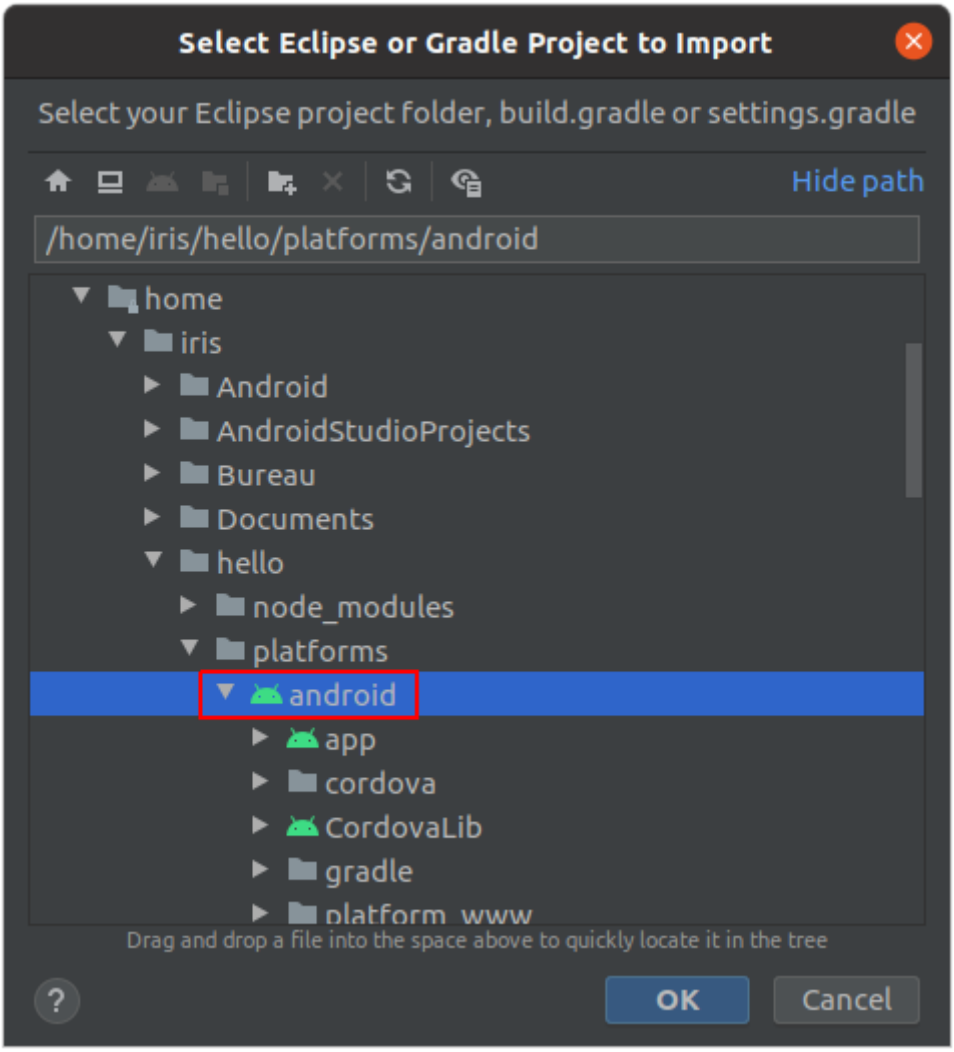
```
$ keytool -genkey -v -keystore HelloWorld.keystore -alias hello -validity 20000

$ keytool -list -keystore HelloWorld.keystore -storepass zg3pa04f -v
Type de fichier de clés : PKCS12
Fournisseur de fichier de clés : SUN

Votre fichier de clés d'accès contient 1 entrée

Nom d'alias : hello
Date de création : 17 janv. 2021
Type d'entrée : PrivateKeyEntry
Longueur de chaîne du certificat : 1
Certificat[1]:
Propriétaire : CN=vaira thierry, OU=lasalle, O=btssn, L=avignon, ST=vaucluse, C=fr
Emetteur : CN=vaira thierry, OU=lasalle, O=btssn, L=avignon, ST=vaucluse, C=fr
Numéro de série : 3a0b326
Valide du Sun Jan 17 13:29:24 CET 2021 au Mon Oct 21 14:29:24 CEST 2075
Empreintes du certificat :
     SHA 1: 98:5C:85:E0:82:D4:26:4D:98:20:F7:D4:59:D6:EB:26:52:81:EE:4B
     SHA 256:
AD:49:79:98:0D:34:59:5F:E5:73:3C:D1:08:DC:DD:6D:EF:CC:CB:4C:E1:6F:B2:63:87:67:F3:C8:22
:8D:B8:28
Nom de l'algorithme de signature : SHA256withDSA
Algorithme de clé publique du sujet : Clé DSA 2048 bits
Version : 3

Extensions :

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: F2 B0 FB EC 27 41 1F 92   5B AB 0B A5 CB 4B E2 3A  ....'A..[....K.:
0010: 19 0C D6 03                                        ....
]
]

*******************************************
*******************************************
```

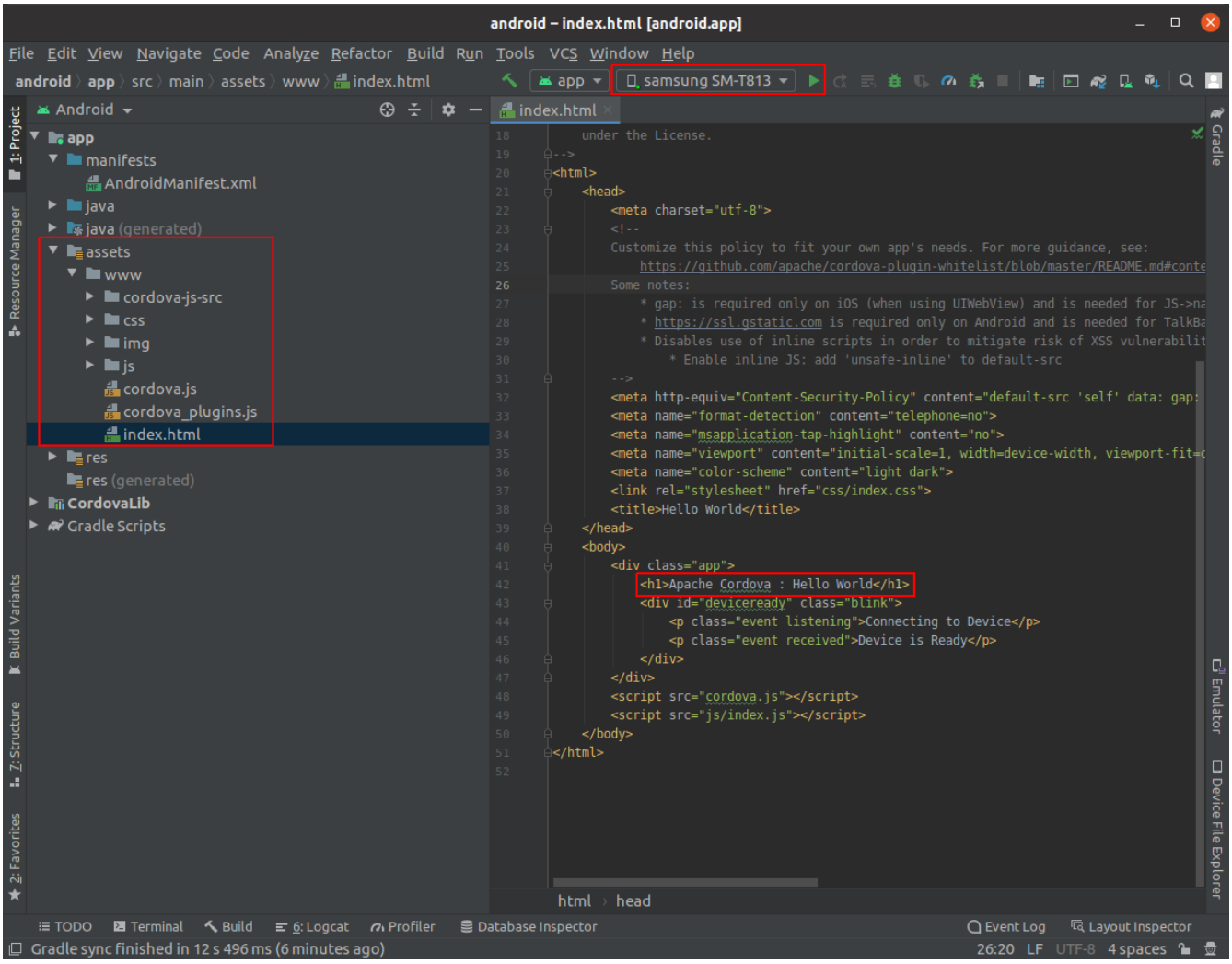
Il faut desinstaller le paquet précédent de la tablette.

```
$ cordova run android --release -- --keystore=./HelloWorld.keystore --storeType=jks
--storePassword=xxxxxxx --alias=hello --password=password
...
BUILD SUCCESSFUL in 1s
47 actionable tasks: 1 executed, 46 up-to-date
Built the following apk(s):
        /home/iris/hello/platforms/android/app/build/outputs/apk/release/app-
release.apk
Checking Java JDK and Android SDK versions
ANDROID_SDK_ROOT=/home/iris/Android/Sdk (recommended setting)
ANDROID_HOME=/home/iris/Android/Sdk (DEPRECATED)
Using Android SDK: /home/iris/Android/Sdk
No target specified, deploying to device 'd95f698401ba0ee8'.
Using apk: /home/iris/hello/platforms/android/app/build/outputs/apk/release/app-
release.apk
Package name: com.example.hello
LAUNCH SUCCESS
```

Pour finir, il est possible d'importer le projet dans **Android Studio** :




Select Eclipse or Gradle Project to Import
Select your Eclipse project folder, build.gradle or settings.gradle
Hide path
/home/iris/hello/platforms/android
home
iris
Android
AndroidStudioProjects
Bureau
Documents
hello
node_modules
platforms
android
app
cordova
CordovaLib
gradle
platform_www
Drag and drop a file into the space above to quickly locate it in the tree
OK
Cancel



2.2. iOS

```
$ cordova platform add ios
Using cordova-fetch for cordova-ios@^6.1.0
Adding ios project...
Creating Cordova project for the iOS platform:
    Path: platforms/ios
    Package: com.example.hello
    Name: HelloWorld
iOS project created with cordova-ios@6.1.1
Discovered plugin "cordova-plugin-whitelist". Adding it to the project
Installing "cordova-plugin-whitelist" for ios
Adding cordova-plugin-whitelist to package.json

$ cordova prepare
```

La commande `cordova build` permet d'effectuer successivement `cordova prepare` et `cordova compile`.

Les informations sur le projet :

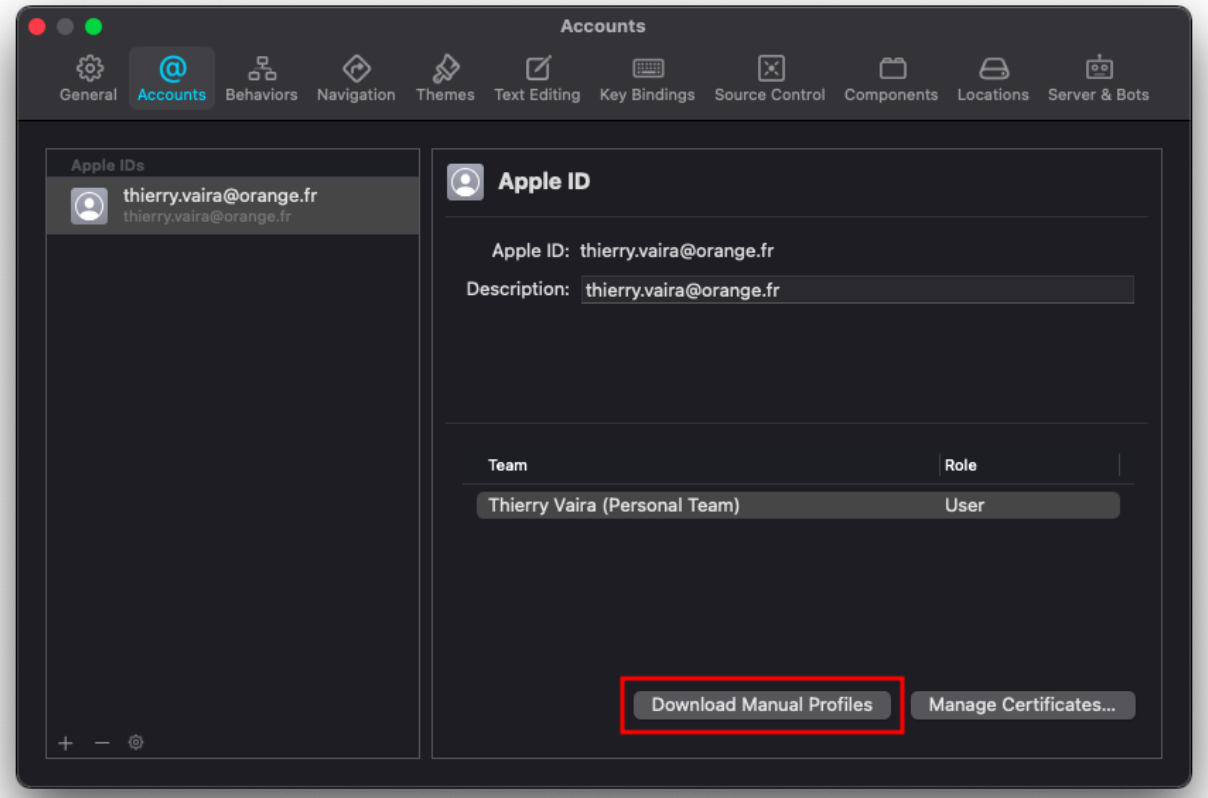
```
$ cordova info
Cordova Packages:

    cli: 10.0.0
        common: 4.0.2
        create: 3.0.0
        lib: 10.0.0
            common: 4.0.2
            fetch: 3.0.0
            serve: 4.0.0

Project Installed Platforms:

    ios: 6.1.1

Project Installed Plugins:

    cordova-plugin-whitelist: 1.3.4

Environment:

    OS: macOS 11.1 (20C69) (darwin 20.2.0) x64
    Node: v14.15.4
    npm: 6.14.10

ios Environment:

    xcodebuild:
Xcode 12.3
Build version 12C33
...
```

On va modifier l' `id` du projet :

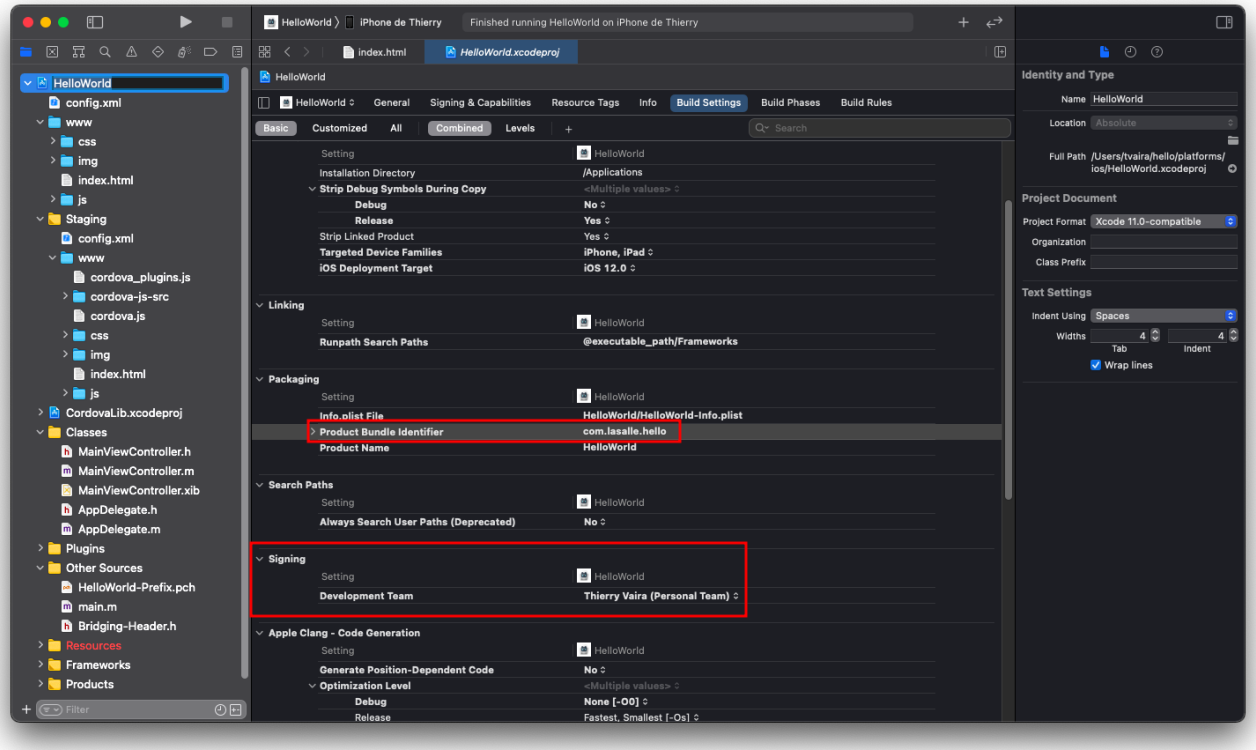
```
$ vim config.xml
<?xml version='1.0' encoding='utf-8'?>
<widget id="com.lasalle.hello" version="1.0.0" xmlns="http://www.w3.org/ns/widgets"
xmlns:cdv="http://cordova.apache.org/ns/1.0">
    <name>HelloWorld</name>
    <description>
        A sample Apache Cordova application that responds to the deviceready event.
    </description>
    <author email="dev@cordova.apache.org" href="http://cordova.io">
        Apache Cordova Team
    </author>
    <content src="index.html" />
    <access origin="*" />
    <allow-intent href="http://*/*" />
    <allow-intent href="https://*/*" />
    <allow-intent href="tel:*" />
    <allow-intent href="sms:*" />
    <allow-intent href="mailto:*" />
    <allow-intent href="geo:*" />
    <platform name="android">
        <allow-intent href="market:*" />
    </platform>
    <platform name="ios">
        <allow-intent href="itms:*" />
        <allow-intent href="itms-apps:*" />
    </platform>
</widget>
```

Il faut ensuite ouvrir le projet `platforms/ios/HelloWorld.xcodeproj` dans **XCode** pour faire les réglages du compte du **développeur Apple**.

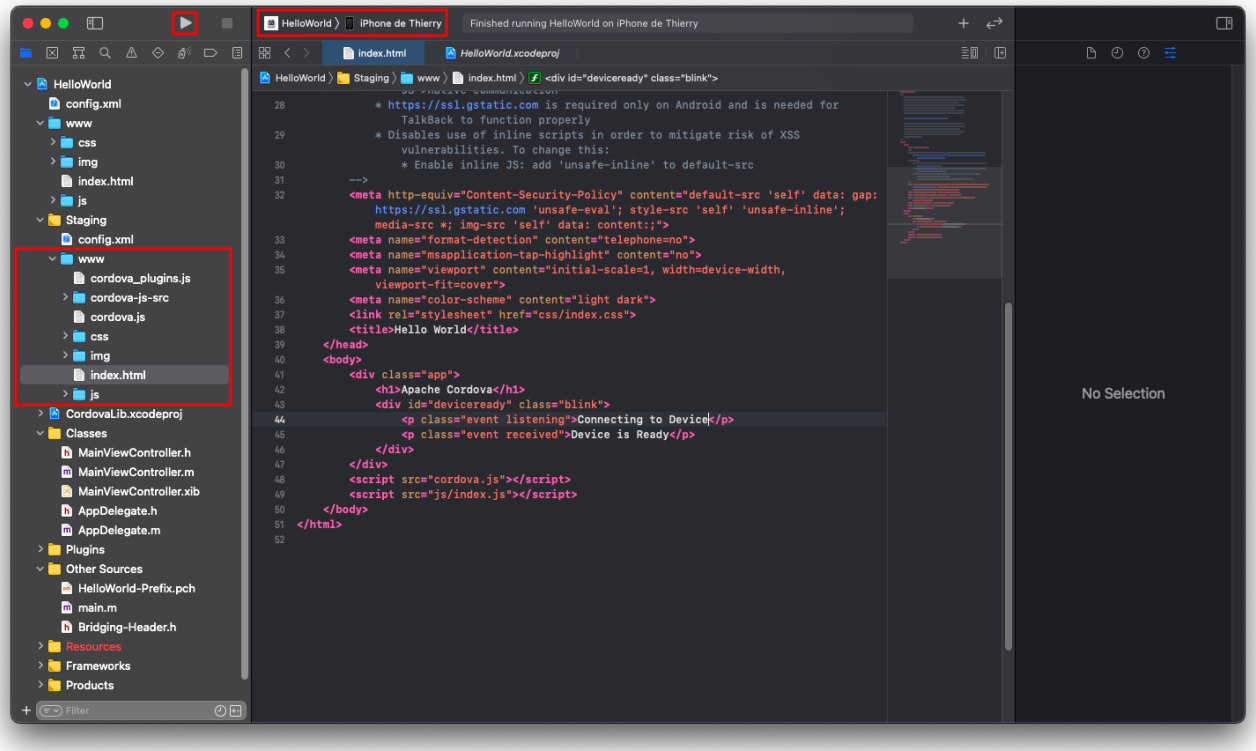
Aller dans `XCode` → `Preferences` → `Accounts` :



Puis faire les réglages du projet :



Il est possible de fabriquer et lancer l'application sur l'iPhone depuis XCode :

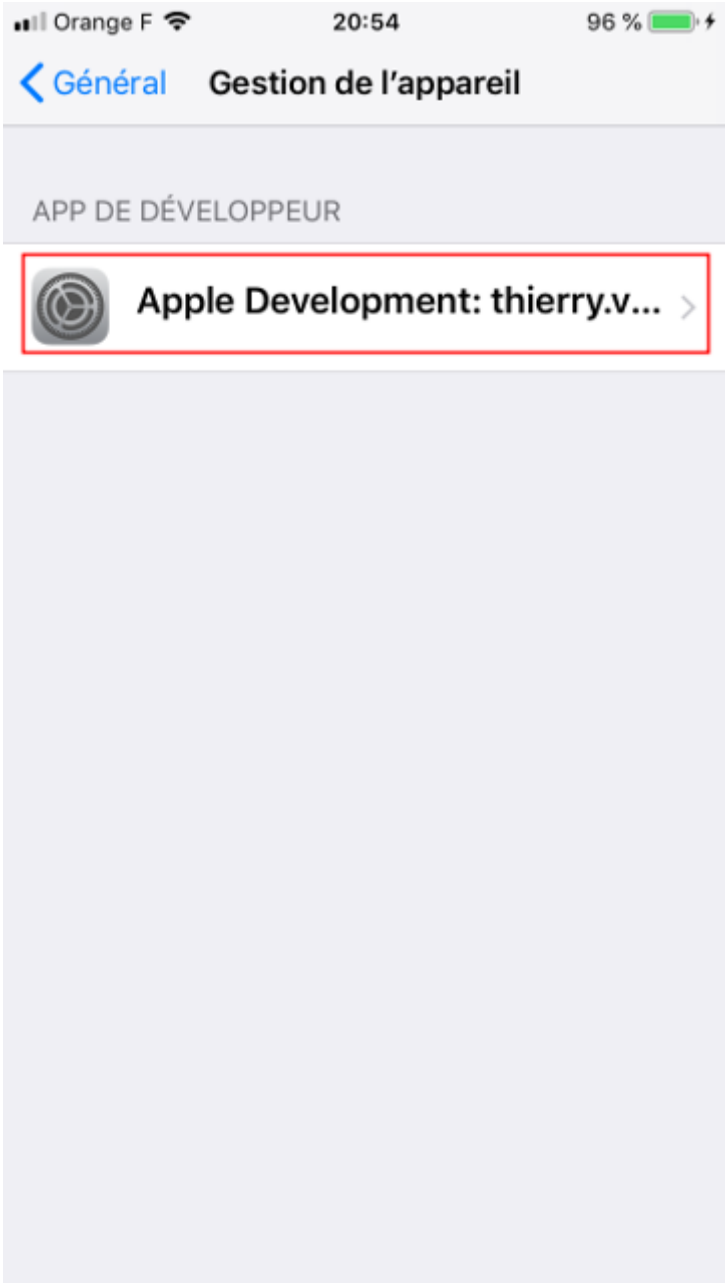


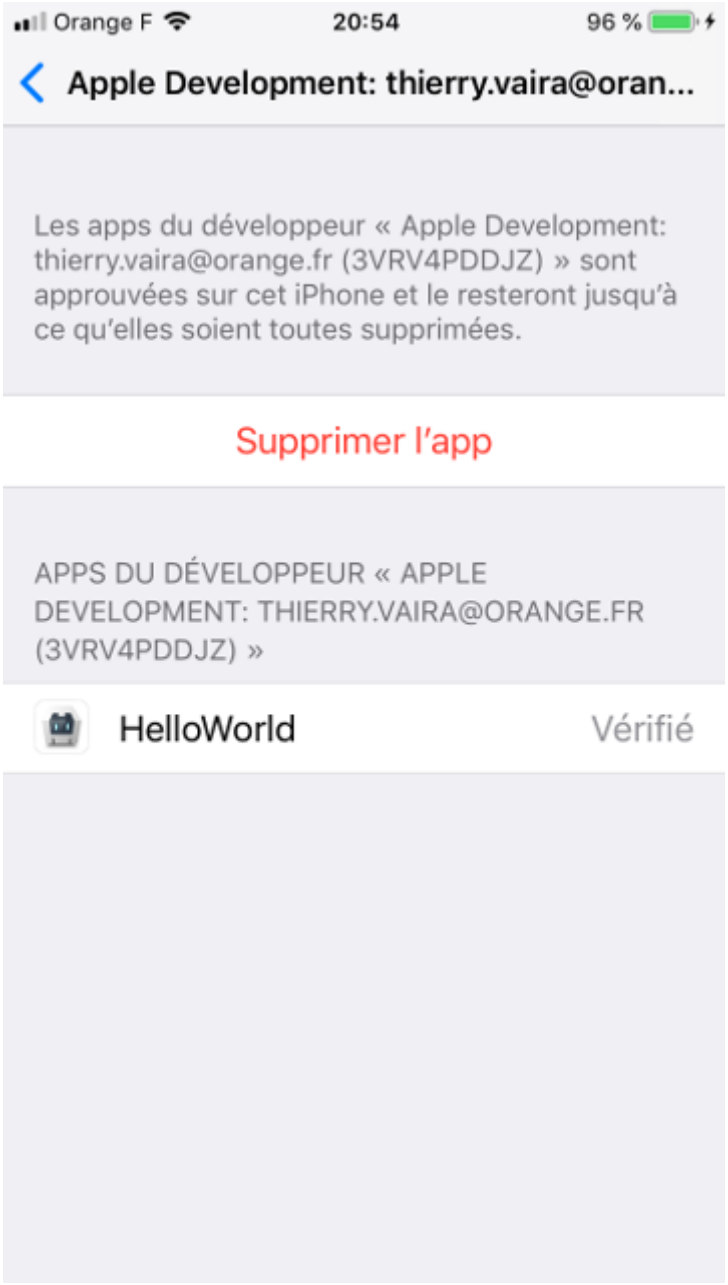
Ou avec Cordova depuis la ligne de commande :

```
$ cordova run ios --device
```

Il faut autoriser l’application à s’exécuter sur l’iPhone :


Orange F
20:54
96 %
Réglages
Général
Date et heure
Clavier
Langue et région
Dictionnaires
Synchronisation Wi-Fi iTunes
VPN
Non connecté
Gestion de l'appareil
Apple Develop...
Certifications
Réinitialiser
Éteindre


Orange F
20:54
96 %
Général
Gestion de l'appareil
APP DE DÉVELOPPEUR
Apple Development: thierry.v...



3. Les plugins

Un *plugin* Cordova est une API qui fournit une interface JavaScript aux composants natifs de la plateforme. Ils permettront à l'application d'utiliser les fonctionnalités natives de l'appareil.

Liste des fonctionnalités par plateforme : https://cordova.apache.org/docs/fr/latest/guide/support/index.html

Les *plugins* : https://cordova.apache.org/plugins/

```
$ cordova plugin list
cordova-plugin-whitelist 1.3.4 "Whitelist"
```

La commande `cordova plugin` permet aussi d'ajouter (`add`) et supprimer (`remove`) des *plugins*.

4. Voir aussi

- Personnalisation des icônes et écran de démarrage (*splash*) : https://cordova.apache.org/docs/fr/latest/config_ref/images.html

---

Thierry Vaira - <tvaira@free.fr> - version v1.0 - 05/01/2021