

Marquer comme terminé

Construire l'application

Lorsque l'application est stable, il est nécessaire de la *construire* (ou emballer) pour la déployer sur un périphérique mobile. Il s'agit de générer un fichier au format **apk**.

Utilisation d'**expo**

On peut utiliser **expo** pour construire l'application\ :

```
$ expo build:android -t apk
```

Après avoir répondu à quelques questions (nom du package, clé de chiffage), la construction est lancée sur les serveurs d'**expo**. Le temps d'attente est variable selon leur occupation.

Le gros intérêt de cette pratique est qu'il n'est pas nécessaire de modifier le code de l'application s'il a été initié à l'aide d'**expo**.

[Des détails ici](#)

Cependant, utiliser **expo** procure de nombreux désagréments, voir par exemple [ce post en français](#).

Utilisation d'**androidstudio**

Utiliser **androidstudio** pose quelques difficultés\ :

- ce produit est très lourd et requiert de nombreux modules tout aussi lourds
- l'émulateur tourne sur une machine virtuelle incompatible avec **Virtualbox**

Cependant, **androidstudio** est utile pour installer l'émulateur. Une fois cette étape réalisée, on peut quitter **androidstudio**, il n'est plus nécessaire.

En particulier, il vaut mieux éviter de mettre à jour **gradle** comme le propose **androidstudio**.

[Des détails ici](#)

[Détails de l'installation d'androidstudio\ :](#)

Procédure de développement classique, sans **expo**

1. initialiser l'application à l'aide de

```
react-native init <nomProjet>
```

2. *éjecter expo*\ : il s'agit de se libérer du framework. À ce stade, on recopiera simplement dans le nouveau dossier du projet créé par **react-native** les fichiers et dossiers de l'application gérée par **expo**. Il faut éventuellement mettre à jour le fichier **package.json** et bien sûr **App.js**.

3. ajouter les modules pour la navigation

```
npm i --save react-native-safe-area-context
npm i --save react-native-screens
```

4. dans le dossier de l'application, lancer l'empaqueteur **metro**\ :

```
react-native start
```

5. lancer l'émulateur android (il faut parfois insister en relançant la commande, car l'émulateur prend du temps pour s'initialiser)\ :

```
react-native run-android
```

Cela construit un **.apk** de développement dans le dossier **android/app/build/outputs/apk/debug**.

Génération de l'**apk debug**

1. emballer le Javascript de l'application (*bundle*)

?

```
mkdir android/app/src/main/assets
react-native bundle --platform android --dev false --entry-file index.js --bundle-output android/app/src/main/assets/
index.android.bundle --assets-dest android/app/src/main/res
```

2. On peut également explicitement construire l'**apk**\:

```
cd android
./gradlew assembleDebug
```

Pour cette dernière étape, il n'est pas nécessaire d'avoir lancé **androidstudio** (mais il faut avoir fermé **Virtualbox**).

On peut cependant\:

- lancer **androidstudio**
- ouvrir comme projet le dossier **android**
- sélectionner le bon périphérique et exécuter le projet\ : il s'exécute dans l'émulateur à l'intérieur d'**androidstudio**.

Génération d'un **.apk release**

<https://medium.com/geekculture/react-native-generate-apk-debug-and-release-apk-4e9981a2ea51>

1. générer une clé de signature\:

```
keytool -genkey -v -keystore your_key_name.keystore -alias your_key_alias -keyalg RSA -keysize 2048 -validity 10000
```

Un fichier **your_key_name.keystore** est généré. Le positionner dans le dossier **android/app**\:

```
mv your_key_name.keystore android/app/
```

2. éditer le fichier de configuration de **gradle** comme suit\:

```
signingConfigs {
    debug {
        ...
    }
    release {
        storeFile file('your_key_name.keystore')
        storePassword 'password'
        keyAlias 'your_key_alias'
        keyPassword 'passwd (le même)'
    }
}
buildTypes {
    debug {
        signingConfig signingConfigs.debug
        if (nativeArchitectures) {
            ndk {
                abiFilters nativeArchitectures.split(',')
            }
        }
    }
    release {
        // Caution! In production, you need to generate your own keystore file.
        // see https://reactnative.dev/docs/signed-apk-android.
        signingConfig signingConfigs.release // <- changer ici debug en release
        ...
    }
}
```

3. autoriser l'application à faire des échanges avec l'API. Dans **android/app/src/main/AndroidManifest.xml**, ajouter un attribut à la balise **application**\:

```
<application
    ...
    android:usesCleartextTraffic="true"
```

4. paqueter l'application

```
mkdir android/app/src/main/assets
react-native bundle --platform android --dev false --entry-file index.js --bundle-output android/app/src/main/assets/
index.android.bundle --assets-dest android/app/src/main/res
```

Voir également [ce post](#).

5. construire l'**apk**

?

```
cd android
./gradlew assembleRelease
```

6. s'il y a des problèmes avec des ressource dupliquées\ : les détruire\ !

```
rm -f app/src/main/res/drawable-*/*
```

L'**apk** construit est dans **app/build/outputs/apk/release/app-release.apk**.

Conclusion

Pour la même application\ :

- APK expo : 64Mo, 132 Mo de stockage interne
- APK debug : 42 Mo (dont 1 Mo de bundle Javascript), 65,41 Mo de stockage interne
- APK release : 32 Mo, 63,28 Mo de stockage interne

expo est un bon choix s'il l'on souhaite travailler rapidement, mais l'**apk** produit est très lourd.

La démarche classique (pas d'**expo**) est à privilégier en contexte professionnel. Il faudra alors dès le début travailler en condition réelles et tester la génération d'**apk** à chaque étape du développement.

Liens / tips

```
https://medium.com/reactbrasil/being-free-from-expo-in-react-native-apps-310034a3729
https://apiko.com/blog/expo-vs-vanilla-react-native/
https://fulcrum.rocks/blog/react-native-init-vs-expo/

npm i react-native-screens --save
npm install @react-navigation/native --save
npm install react-native-reanimated react-native-gesture-handler react-native-screens react-native-safe-area-context @react-native-community/masked-view --save
```

Modifié le: lundi 9 septembre 2024, 11:36

◀ **hierarchie.png**

Choisir un élément

Aller à...

CM 7 ▶

[mentions légales](#) . [vie privée](#) . [charte utilisation](#) . [unicaen](#) . [cemu](#) . [moodle](#)

[f](#) [t](#) [v](#) [@](#) [in](#)