Créer une application Android avec Cordova et VueJS

17 décembre 2018 · 7 min de lecture

Nous avons vu qu'il est possible de développer des applications de bureau avec les technologies du web (Javascript/HTML/CSS). De même, je vous propose de créer une application Android, de l'installation des briques logicielles, jusqu'à la publication sur le Play Store. Nous utiliserons VueJS pour coder l'interface et Cordova pour empaqueter l'application.

Dans cet article, le système utilisé est Linux mais les outils fonctionnent également sur Windows.

Installation des composants

NodeJS

Téléchargez NodeJS qui est livré avec le gestionnaires de librairies "npm". Décompressez le dossier par exemple dans `/home/user/logiciels/node` (où `user` est votre nom d'utilisateur).

Pour informer le système que la commande `node` se trouve dans ce dossier, ajoutez la ligne suivante dans votre fichier `/home/user/.profile` :

```
export PATH=$PATH:/home/user/logiciels/node/bin
```

Tapez la commande suivante pour prendre en compte cette modification :

```
source /home/user/.profile
```

Vous pouvez maintenant saisir les commandes `node` et `npm` depuis n'importe quel dossier. Essayez par exemple avec l'option `-v` pour afficher le numéro de version.

Cordova

Cordova est l'outil permettant d'empaqueter une application Javascript dans une application Android. La commande suivante va l'installer sur votre ordinateur :

```
npm install -g cordova
```

VueJS

Vue CLI 3 est un programme très utile qui crée des modèles de projets Vue en incluant, entre autres, des tests unitaires, des tests de fonctionnement et l'empaquetage avec Webpack. Pour l'installer tapez :

```
npm install -g @vue/cli
```

Java JDK

Les applications Android sont créées nativement en Java. Le kit de développement est donc nécessaire. Cordova l'utilisera. Téléchargez et installez le JDK ou tapez les commandes suivantes sous Linux :

```
sudo add-apt-repository ppa:webupd8team/java
sudo apt-get update
sudo apt-get install oracle-java8-installer
```

Android SDK

Pour développer une application Android, ce kit doit également être installé. Je recommande l'installation d'Android Studio, un logiciel gratuit pour le développement d'applications natives Android. Des émulateurs et le SDK seront installés automatiquement. Prévoyez quelques Go d'espace libre sur votre disque dur.

A moins d'avoir spécifié un dossier précis lors de l'installation, le SDK a été copié dans votre dossier utilisateur. Pour y accéder via le terminal depuis n'importe quel dossier, modifiez le fichier `/home/user/.profile` de la façon suivante :

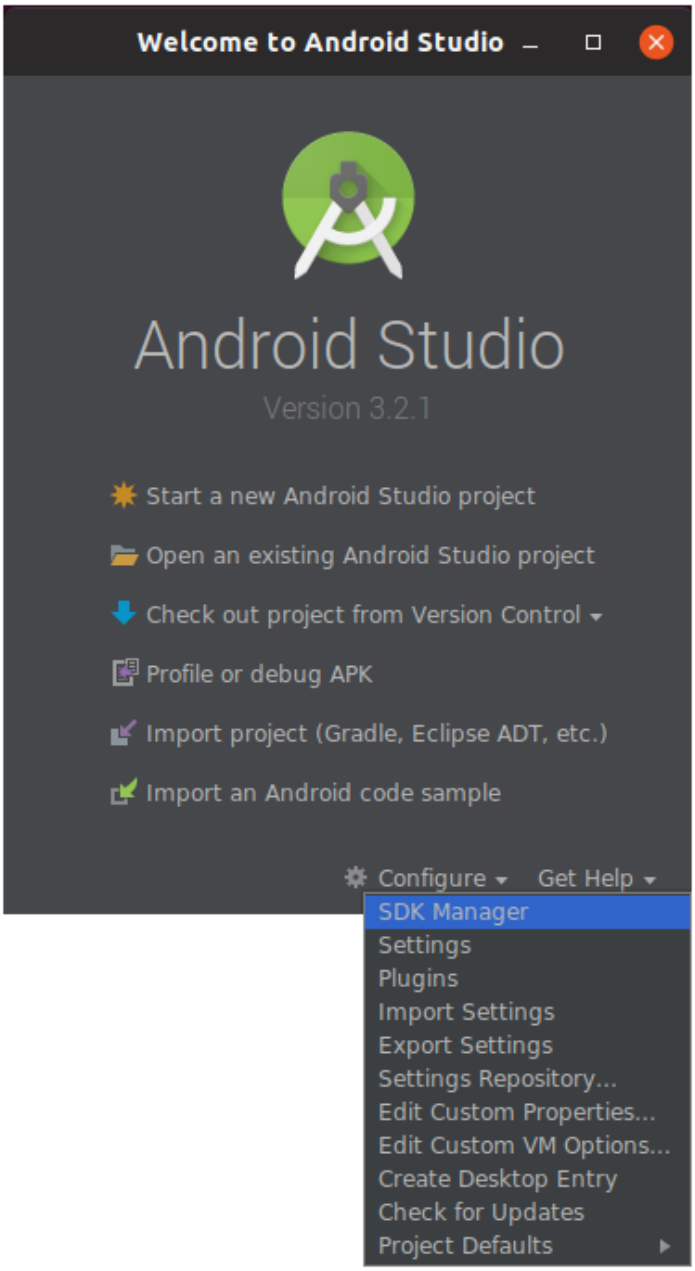
```
export ANDROID_HOME=/home/user/Android/Sdk
export PATH=$PATH:/home/user/logiciels/node/bin:${ANDROID_HOME}/emulator:${ANDROID_HOME
```

Pour prendre en compte les modifications, tapez :

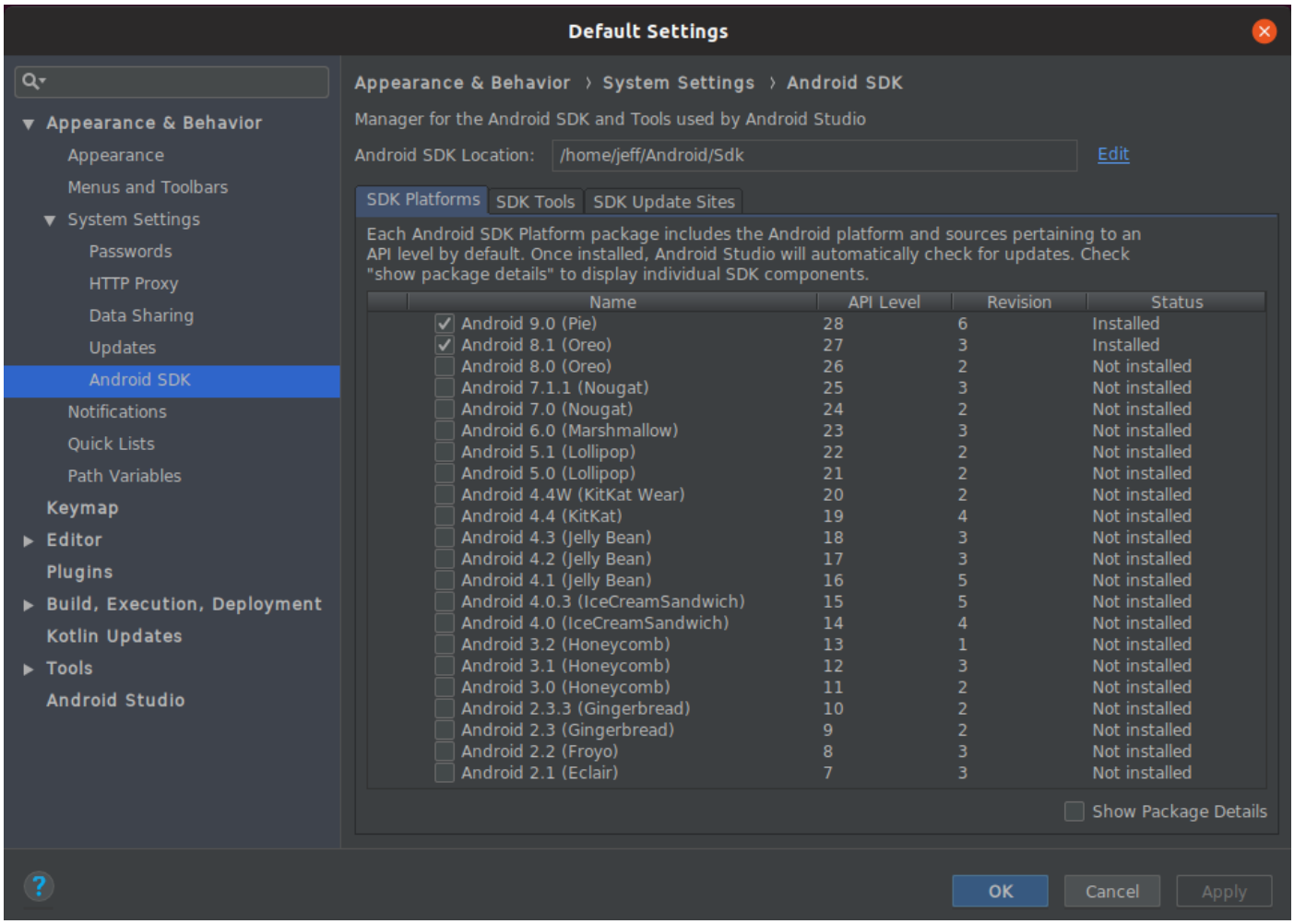
```
source ~/.profile
```

Vous disposez maintenant des commandes adb, sdkmanager, avdmanager…

Actuellement Cordova a besoin de l'API LEVEL 27 qui correspond à Android 8.1 (Oreo). Démarrez Android Studio, cliquez sur `Configure` en bas à droite de la fenêtre de démarrage puis sur `SDK Manager` :



Cochez `Android 8.1` puis cliquez sur `OK` pour le télécharger :



Gradle

Gradle permet de construire des projets d'après des tâches décrites dans un fichier. Rassurez-vous, ce fichier sera généré automatiquement par Cordova.

Il a été installé avec Android Studio dans le dossier `android-studio/gradle/gradle-4.6/bin`. C'est un script bash. Il faut le rendre exécutable :

```
chmod +x android-studio/gradle/gradle-4.6/bin/gradle
```

Il faut ensuite l'ajouter comme précédemment, au chemin d'accès. Modifiez le fichier `/home/user/.profile` :

```
export ANDROID_HOME=/home/user/Android/Sdk
export PATH=$PATH:/home/user/logiciels/node/bin:${ANDROID_HOME}/emulator:${ANDROID_HOME
```

Pour prendre en compte les modifications, tapez :

```
source ~/.profile
```

Nouveau projet

Cordova

Création du projet Android avec Cordova :

```
cordova create cordovatest com.exemple.test "Ma super appli"
```

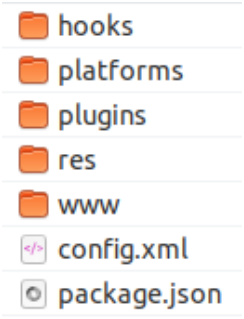
- `cordovatest` est le nom du dossier qui sera créé par Cordova.
- `com.exemple.test` est le nom du paquet de votre application. Par convention il s'agit de votre nom de domaine inversé et du nom de l'application.
- `Ma super appli`, comme vous l'avez deviné est le titre de l'application.

Ajoutons la plateforme Android :

```
cd cordovatest
cordova platform add android
```

NB : pour lister les plateformes disponibles tapez `cordova platform ls`.

Dans le dossier `cordovatest` vous trouverez les fichiers suivants que vous pouvez parcourir :



- Le dossier `platforms` contient le code source des applications Android/iOS ou autre.
- Le dossier `plugins` contiendra les extensions que vous pouvez ajouter selon les besoins, pour utiliser par exemple la caméra du téléphone, la géolocalisation…
- Le dossier `res` contient les icônes et les écrans de démarrage de l'application (splash screen).
- Le dossier `www` contient le code source web de l'application. Son contenu peut être supprimé car nous générerons le code Vue dans ce dossier.
- Vous pouvez éditer le fichier `config.xml` pour modifier le nom de l'application, la description…

Pour voir si tout le nécessaire a bien été installé au paragraphe précédent, tapez la commande :

```
cordova requirements
```

qui doit retourner quelque chose comme ceci pour indiquer que tous les logiciels requis sont présents :

```
Android Studio project detected

Requirements check results for android:
Java JDK: installed 1.8.0
Android SDK: installed true
Android target: installed android-28,android-27
Gradle: installed /media/sdb/android-studio/gradle/gradle-4.6/bin/gradle
```

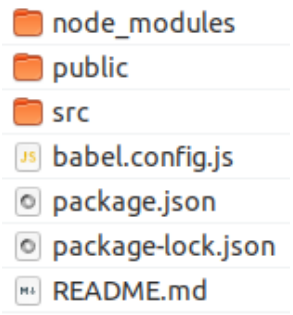
VueJS

Sortez du dossier `cordovatest` pour créer le dossier `vuejs` :

```
cd ..
vue create vuejs
```

Choisissez le réglage `default (babel, eslint)`. Pour information, dans les réglages avancés, vous avez les tests unitaires et de bon fonctionnement.

Vous pouvez déplacer ce dossier `vuejs` dans `cordovatest` pour plus de commodité si vous versionnez le code, mais ce n'est pas obligatoire. Voici son contenu :



- `node_modules` contient les dépendances javascript. Elles sont gérées par la commande `npm` et listées dans le fichier `package.json`. Vous n'avez rien à créer ou modifier dans ce dossier.
- `public` contient les fichiers statiques, comme vos images.
- `src` contient le code VueJS de votre application qu'il vous reste à écrire. Un exemple fonctionnel est présent.

Configuration

A la racine du dossier `vuejs`, à côté du fichier `package.json`, créez le fichier `vue.config.js` avec le contenu suivant :

```
module.exports = {
  outputDir: '/home/user/cordovatest/www',
  publicPath: './'
}
```

- `outputDir` est le dossier dans lequel sera compilé l'application Vue.
- `publicPath` permettra d'ajouter un chemin relatif aux fichiers de l'application. Sans cela, vous aurez une belle page blanche lors du démarrage de l'application dans votre smartphone.

Modifiez le fichier `package.json` comme suit :

```
  "scripts": {
    ...
    "build": "/bin/rm -rf /home/user/cordovatest/www/* && vue-cli-service build",
    ...
  },
```

Ainsi la commande `npm run build` compilera l'application dans le dossier `/home/user/cordovatest/www` après avoir supprimé son contenu.

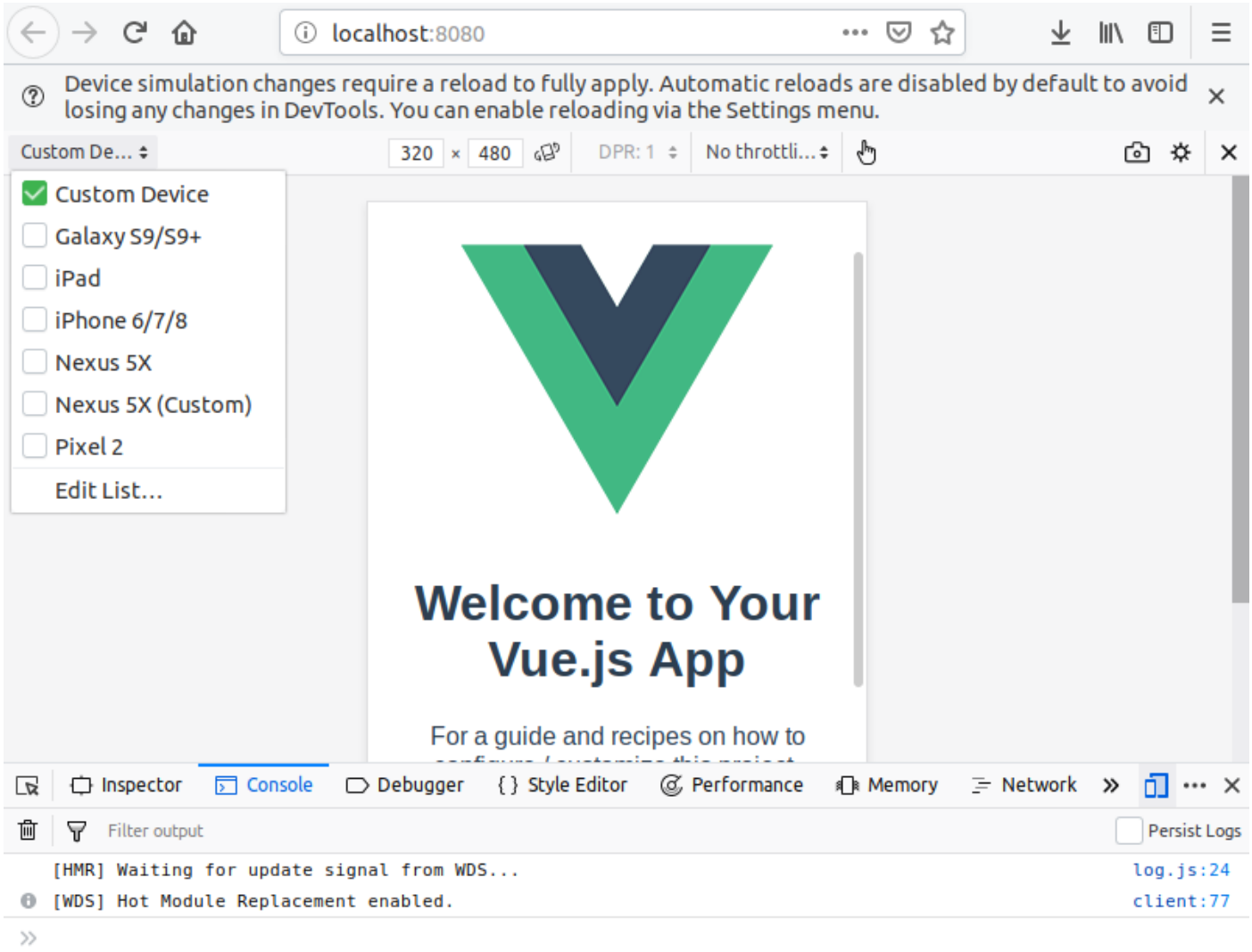
Développement

Dans le navigateur

Dans le dossier `vuejs`, démarrez le serveur de développement :

```
cd vuejs
npm run serve
```

Ouvrez le navigateur à l'adresse : `http://localhost:8080`



Abusez des outils pour les développeurs fournis par les navigateurs (`CRTL+MAJ+I`). La console vous informe d'éventuelles erreurs, et il est possible de visualiser l'application selon plusieurs tailles d'écrans de smartphones prédéfinies. Vous pouvez également vérifier le comportement sans réseau ou avec un réseau extrêmement lent.

Dès que vous modifiez le code dans le dossier `vuejs/src` , le navigateur est rafraîchi automatiquement.

Si vous débutez en VueJS, je vous invite à lire la documentation officielle et en français de VueJs qui est très bien écrite.

Grâce à ce framework, vous pouvez coder une application complète dans une seule page. Pour bénéficier d'un visuel proche du design d'une application Android, il existe le framework CSS Vuetify. On peut citer également iView, Cube-UI, Quasar, Element…

Astuces pour le navigateur Chrome :

- Dans la barre d'adresse, tapez `chrome://inspect/#devices` . Si vous avez branché votre smartphone via USB, en cliquant sur `inspect` , vous accédez à la console de développement reliée au téléphone.

- Dans la barre d'adresse, tapez `chrome://settings/languages` pour changer la langue du navigateur et tester l'application si elle est multilingue (voir la librairie Vue I18n).

Tests avec un smartphone ou un émulateur

Activez le mode développeur dans votre téléphone et connectez-le en USB. Dans le dossier `cordovatest` tapez :

```
cordova run android
```

Vous pouvez créer les émulateurs depuis Android Studio. Si un émulateur existe la commande précédente le démarrera automatiquement.

Compilation

Ou comment créer le binaire distribuable d'une application Android (extension `apk`).

Tout d'abord, il faut "compiler" ou "empaqueter" l'application VueJS dans le dossier `cordovatest/www`.

```
cd vuejs
npm run build
```

Création du fichier `apk` :

```
cd cordovatest
cordova build android --release
```

Le chemin du fichier `apk` créé sera indiqué. Vous pouvez l'envoyer par mail à un ami ou le copier sur un smartphone pour l'installer. Il faudra toutefois configurer le système pour autoriser l'installation d'applications tierces.

Distribution sur le Play Store

Le fichier `apk` doit être signé numériquement. Ouvrez le dossier `cordovatest/platforms/android` dans Android Studio. Après quelques minutes d'indexation des fichiers, allez dans le menu `Build > Generate Signed Bundle/APK`. Choisissez `APK`, puis `Create new` pour générer un fichier clé (keystore). Il sera demandé de créer un mot de passe. Pour le `Build type`, sélectionnez `release` et cochez `V1 (Jar Signature)`.

Un fichier `apk` signé sera créé. Vous pourrez le transférer dans votre compte développeur Google Play.

Attention ! Conservez bien le fichier `keystore` créé. Je recommande de le versionner. Car les prochaines mises à jour à diffuser sur le Play Store devront être signées avec ce même fichier.

Si vous possédez déjà ce fichier `keystore`, Cordova peut générer l'apk signé :

```
cd cordovatest
cordova build android --release -- --keystore=fichierkeystore --storePassword=password
```

Conclusion

VueJS permet de créer des applications "single-page" complètes et au design soigné qu'il est possible d'embarquer dans des applications pour PC ou smartphone, ce qui est bien pratique. L'avantage étant de ne pas réécrire le code pour chaque plateforme (Java pour Android, Swift pour iOS…).

Voici par exemple l'application Pocket Cube Solver que j'ai développé avec Cordova, VueJS et Vuetify :

#adb #apk #android #cordova #vuejs #nodejs

A LIRE EGALEMENT

- Frameworks Javascript : Par où commencer ?
- Développement d'applications multiplateforme avec Electron
- Le test du singe
- Enregistrement audio et vidéo d'un écran Android
- Un moteur de recherche interne pour votre site Hugo

0 COMMENTS 💬

Leave a comment...

User name... Screen name... Send

Mots-clés

2d adb android api apk bash bitcoin blog bot chiffrement css donnees echecs electron framework git gohugo html hugo internet java javascript jekyll jeu linux logiciel meles moteur mots mysql nodejs photos php python react sauvegarde script sqlite svelte twitter ubuntu usb vuejs web windows

Derniers articles

Créez votre barre d'état i3/sway en bash et python

21 juin 2020

Svelte JS : Configuration de VSCode et internationalisation

6 mai 2020

Trouvez de tête le jour de la semaine pour n'importe quelle date

1 janvier 2020

Automatisation de tâches dans un OS virtualisé

5 décembre 2019

Sauvegardes chiffrées dans le cloud

30 novembre 2019

Lire la suite →

Archives

2020 (3)
2019 (6)
2018 (10)
2017 (5)
2016 (4)
2015 (12)
2014 (13)