

TP1 : Installation et première application

Objectifs :

- Installer et configurer Cordova
- Découvrir la ligne de commande de Cordova
- Créer la première application mobile

1. Présentation de l'environnement du travail

Les applications mobiles sont de plus en plus utilisées à travers les smartphones, les tablettes, les montres connectées, les voitures connectées, etc. Il existe trois solutions pour le développement des applications mobiles :

1. Une application native : développée avec des outils qui lui sont propres (Swift, Java, Kotlin, ...). La distribution et la mise à jour se font par l'intermédiaire d'un store dédié.
2. Une application web (webapp) : développée avec les technologies Web HTML5, CSS3 et JavaScript. L'application est distribuée via un serveur Web et est exécutée via le navigateur Web du mobile sur n'importe quelle plateforme mobile.
3. Une application hybride : utilisant le navigateur web et les technologies web encapsulées dans un composant natif WebView en compilant l'exécutable compatible avec le système d'exploitation.

Cordova est un framework opensource d'Apache, qui permet de développer des applications hybrides et qui présente des plugins pré-développés en JAVA pour android ou en Objective-C pour iOS. Ces derniers permettent, via une interface Javascript, de communiquer avec l'appli web et d'accéder aux composants natifs d'un appareil, tel que l'appareil photo, le GPS, le vibreur, etc.



Le site officiel est <https://cordova.apache.org/>

2. Installation

Les étapes d'installation détaillées ci-dessous sont pour le système windows. Pour la configuration sous linux, vous pouvez consulter la documentation officielle de cordova.

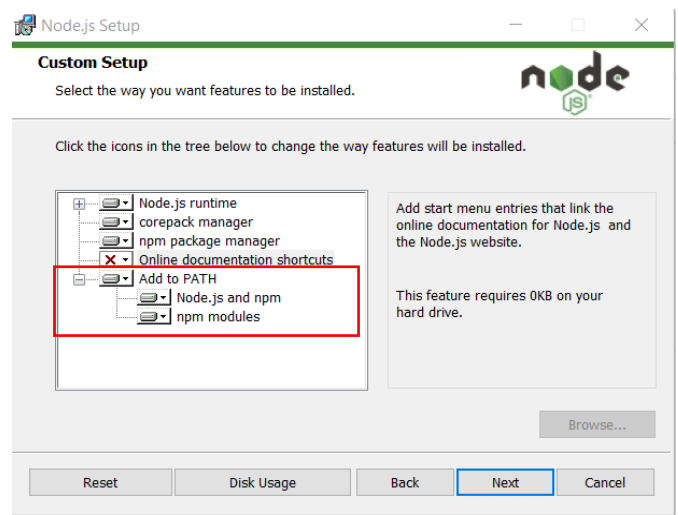
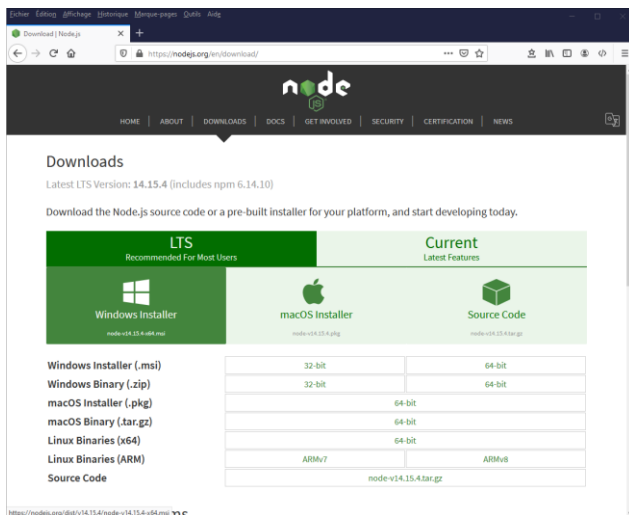
(<https://cordova.apache.org/docs/en/10.x/guide/platforms/android/index.html#requirements-and-support>)

1) Installer NodeJS et NPM

Télécharger « **NodeJS** » en choisissant le fichier adapté à votre système d'exploitation dans l'onglet « **LTS** » depuis le site <https://nodejs.org/en/download/> puis lancez l'installation.



Vérifier que le chemin sera rajouté au Path.



2) Installation de Cordova

La ligne de commande Cordova utilise NodeJS pour fonctionner. Pour cela, ouvrez un **terminal** **Cli** et vérifiez la bonne installation de NodeJS et NPM à l'aide des commandes respective

```
C:\Users\rtprof>npm -v
8.3.1

C:\Users\rtprof>node -v
v16.13.2
```

Installez « **Cordova** » à l'aide de la commande

```
npm install -g cordova
```

```
C:\Users\rtprof>cordova -v
11.0.0
```

3) Installation de JDK, Android sdk et de Gradle :

- Pour l'installation de java jdk8, téléchargez et installez la version adéquate du site.
<https://www.oracle.com/java/technologies/downloads/#java8-windows>
- Pour installer Gradle, il faut suivre la procédure de **l'installation manuelle** présentée sur le site.
<https://gradle.org/install/>
- Pour installer Android Sdk, téléchargez et lancer l'exécution de Android studio
<https://developer.android.com/studio/index.html>

4) Variables d'environnements

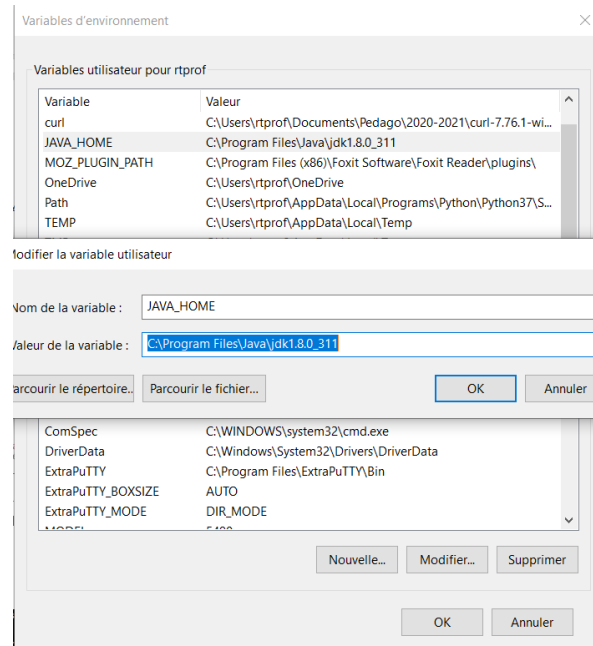
Pour les variables d'environnement, ouvrir ordinateur -> propriétés système -> paramètres avancés du système -> variables d'environnement.

Ajouter la variable d'environnement java :

Nouvelle variable d'environnement JAVA_HOME= « dossier d'installation de jdk »

Ci-dessous un exemple : C:\Program Files\Java\jdk1.8.0_311.

Ajouter ce chemin aussi à la variable d'environnement PATH.

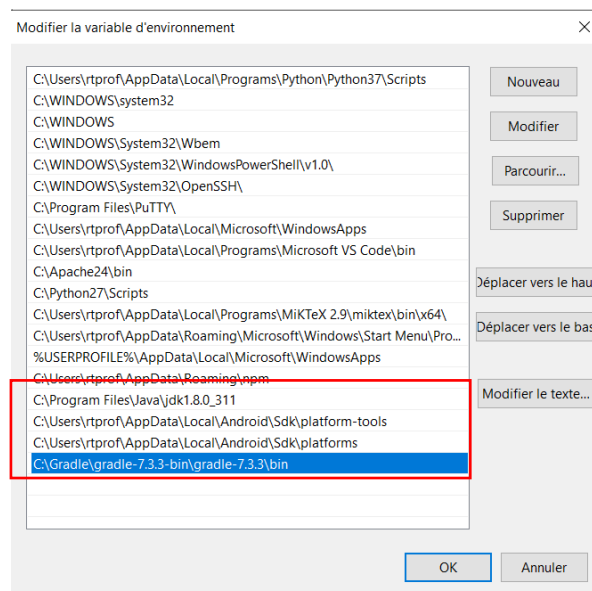


Après l'installation de Android Studio, Il faut rajouter les variables d'environnement pour platform-tools/ ; platforms de Android sdk et de gradle (en fonction du chemin d'installation)

Exemple :

C:\Users\[your user]\AppData\Local\Android\Sdk\platform-tools

C:\Users\[your user]\AppData\Local\Android\Sdk\tools



Pour vérifier le bon fonctionnement des outils, vous pouvez exécuter les commandes suivantes :

```
java -version
gradle -help
cordova --help
```

3. Première application

1) Création d'une application Cordova

Vous trouverez l'ensemble des explication relatives en suivant le lien sur le site officiel de Cordova <https://cordova.apache.org/docs/en/latest/guide/cli/index.html>.

Dans une console CLi, la commande « **cordova create** » permet de générer une application de base

```
Cordova create MyApp fr.myapp.demo MyApp
```

commande **path** **id** **name**

Le dossier MyApp est créé. C'est le dossier d'accueil de l'application. Il contient les dossiers et les fichiers qui composent l'application de base. MyApp est le nom que va prendre le dossier, fr.myapp.demo présente le nom du package de l'application (facultatif), MyApp est le nom (facultatif) que prendra votre application lorsqu'elle sera compilée ou déployée. Ces valeurs sont modifiables à tout instant dans le fichier config.xml qui va être créé dans le dossier.

- **path** : répertoire d'accueil de l'application
- **id** : utilise le style nom de domaine inverse utilisé pour les package généré pour les plateformes
- **name** : titre affiché de l'application
- **config** : optionnel – **chaîne JSON** répertoriant des couples **clé/valeur** à ajouter dans `<path>/cordova/config.xml`

Les plateformes mobiles (android ou ios) sont ajoutées dans le dossier **platforms**. Il faut se situer dans le dossier du projet pour lui ajouter les plateformes avec la commande :

```
cordova platform add android -save
```

L'application peut nécessiter l'ajout de plugins spécifiques aux fonctionnalités du device (accéléromètre, network, gps, camera, ...). Pour cela il faut se situer dans le dossier de l'application et exécuter la commande suivante :

```
cordova plugin add nom_du_plugin
```

Arborescence de l'application :

```

├── config.xml
├── hooks
│   └── README.md
├── platforms
├── plugins
├── www
│   ├── css
│   │   └── index.css
│   ├── img
│   │   └── logo.png
│   ├── index.html
│   └── js
│       └── index.js
└── 7 directories, 6 files
    
```

L'application est créée dans le dossier défini par la commande
Elle est structurée comme une **webapp** à **page unique**

La page d'accueil du projet correspond au fichier **www/index.html**

- Le dossier **www/**
 - Contient les éléments traditionnels d'une webapp (page html, feuille de style, script, image)
 - Sous arborescence classique (**css/**, **js/**, **img/**)
- Le dossier **hooks/** : scripts à exécuter avant une commande
<http://cordova.apache.org/docs/en/6.x/guide/appdev/hooks/index.html>
- Le dossier **platforms/** : code source et les fichiers produits pour la plateforme spécifiée. Ce dossier est celui dans lequel sera compilé l'application, contenant plus tard des dossiers au nom des plateformes concernées comme "ios" ou "android"
- Le dossier **plugins/** : fichiers des plugins utilisés dans l'application
- Le dossier **node_modules/** : ce dossier contient les modules NodeJS
- Le fichier **config.xml** permet notamment de modifier le nom de l'application, sa description, le nom du package et de l'auteur (utiles lors de l'ajout de l'app sur un Store).
- Le fichier **package.json** contient, lui, la liste des plugins et dépendances du projet.

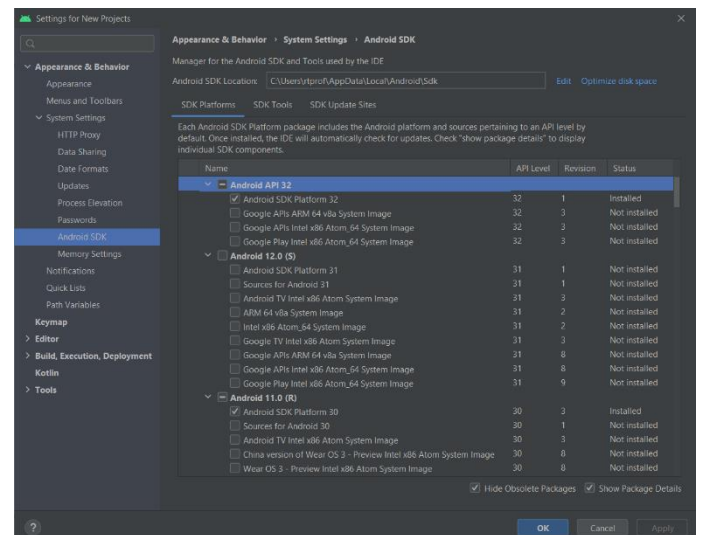
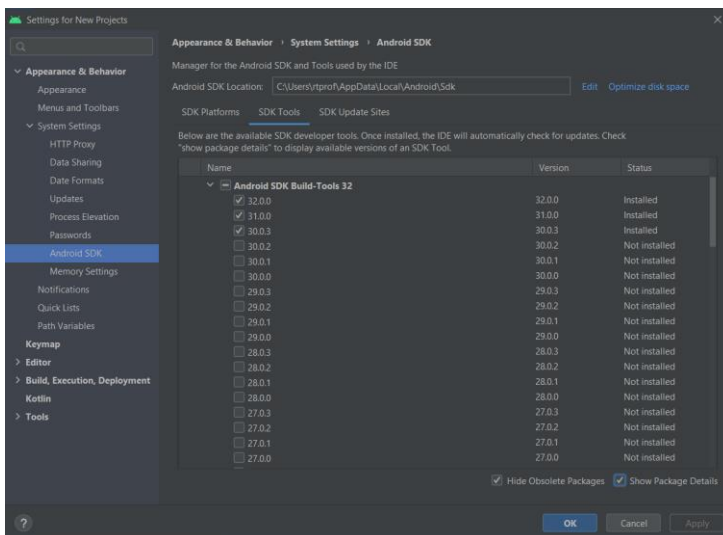
La commande cordova requirements permet de vérifier l'installation des sdk nécessaire à la plateforme avant compilation. Ci-dessous un exemple du résultat obtenu.

```
C:\Users\rtprof\Documents\Pedago\2021-2022\COURS\dev_mobile\Cours2022\Lab\Cordova\myapp>cordova requirements android

Requirements check results for android:
Java JDK: installed 1.8.0
Android SDK: installed true
Android target: not installed
Please install the Android SDK Platform "platforms;android-30"
Gradle: installed C:\Gradle\gradle-7.3.3-bin\gradle-7.3.3\bin\gradle.BAT
Some of requirements check failed

C:\Users\rtprof\Documents\Pedago\2021-2022\COURS\dev_mobile\Cours2022\Lab\Cordova\myapp>
```

Pour résoudre le problème, il faut ajouter les sdk Android nécessaires. Pour cela, il faut lancer android studio, aller dans tools → sdk manager → sdk tools et sélectionner android 30.0.3



Quand on ré-exécute la commande, on obtient le résultat ci-dessous.

```
C:\Users\rtprof\Documents\Pedago\2021-2022\COURS\dev_mobile\Cours2022\Lab\Cordova\myapp>cordova requirements android

Requirements check results for android:
Java JDK: installed 1.8.0
Android SDK: installed true
Android target: installed android-32,android-30
Gradle: installed C:\Gradle\gradle-7.3.3-bin\gradle-7.3.3\bin\gradle.BAT
```

2) Compiler l'application Cordova

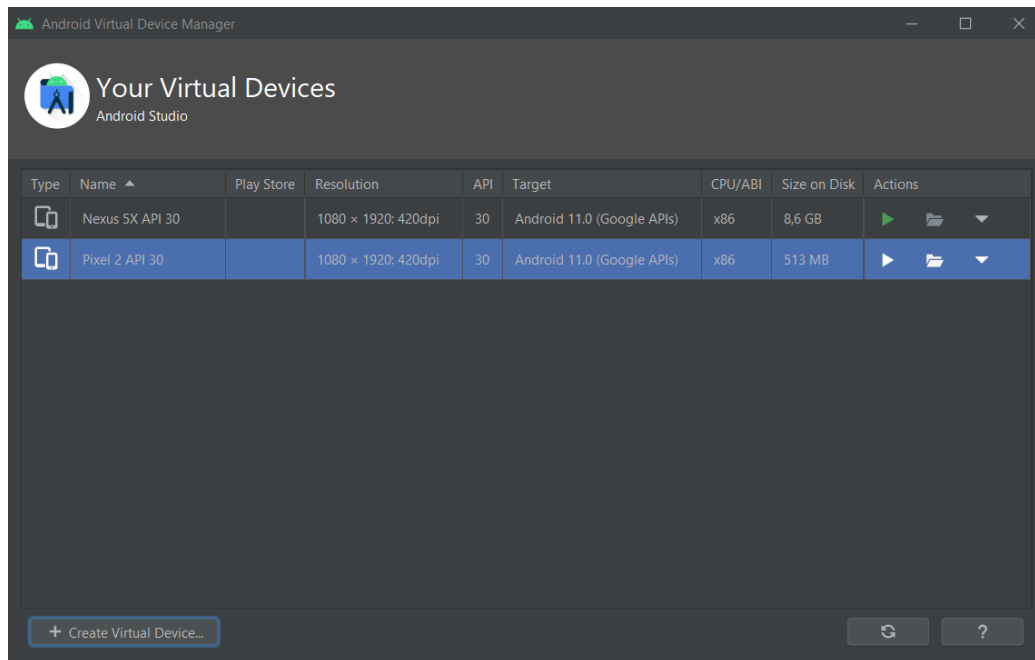
Compiler l'application pour la plateforme souhaitée avec l'une des commandes ci-dessous.

cordova build (pour toutes les plateformes ajoutées)
cordova build android

Pour tester l'application avec un émulateur, il faut ajouter un équipement virtuel.

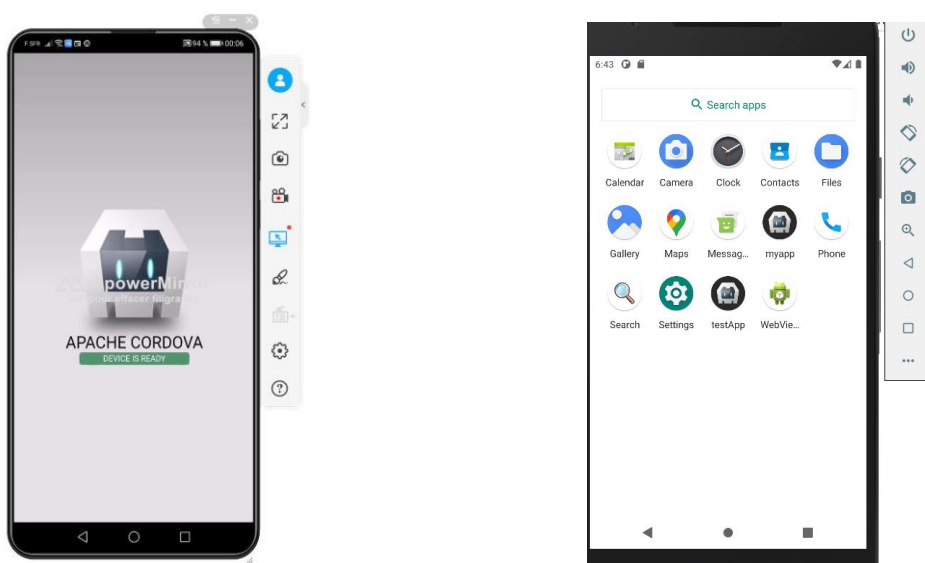
Lancer android studio, aller dans tools → AVD manager → create virtual device

Ensuite choisir le modèle de l'équipement smartphone ou tablette et l'image system.



Pour émuler le fonctionnement, on exécute la commande suivante:

cordova emulate android



Si la connexion avec l'émulateur se fait mal.

1. Lancer l'émulateur à la main

android avd

2. Charger l'application dans l'émulateur (depuis le dossier de votre projet) :

adb install chemin_debug.apk

exemple: adb install C:\Users\rtprof\Documents\Pedago\2021-2022\COURS\dev_mobile\Cours2022\Lab\Cordova\myapp\platforms\android\app\build\outputs\apk\debug\app-debug.apk

Dans le fichier index.html, la commande `<script src="cordova.js"></script>` demande au client Cordova d'injecter la version correspondant à votre plateforme (Android/iOS/...) au moment du build.

Modifier le contenu du fichier index.html en ajoutant le code ci-après :

```
<body>
  <div class="app">
    <h1>Apache Cordova</h1>
    <h2>First Mobile App!!</h2>
    <div id="deviceready" class="blink">
      <p class="event listening">Connecting to Device</p>
      <p class="event received">Device is Ready</p>

      <p>Click the button to display an alert box.</p>

      <button id= "bt1" onclick="">Try it</button>
    </div>
  </div>
```

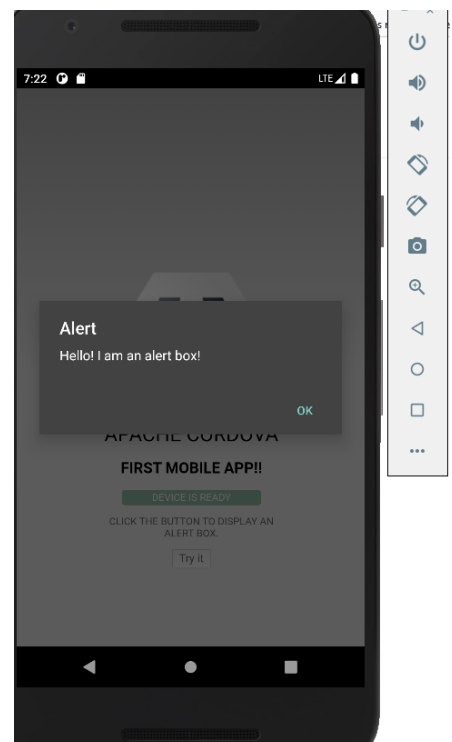
Ajouter le code suivant au fichier index.js :

```
var bt1 = document.getElementById("bt1");
bt1.addEventListener('click', function (event) {
  alert("Hello! I am an alert box!");
}, false);
```

Recompiler et exécuter l'application. Vous obtenez le résultat suivant :

3) Conclusion :

Tout l'intérêt est évidemment, dans ce dossier www/, de pouvoir inclure des projets web complexes, à partir de frameworks tels que VueJS et autres. Cordova servira alors à "encapsuler" notre application, et à faire la "passerelle" entre notre code web et le téléphone.



4. Annexe

Commandes Cordova

La liste des commandes est bien détaillée dans le site officiel de cordova.

<https://cordova.apache.org/docs/en/10.x/reference/cordova-cli/index.html#cordova-build-command>

```
C:\Users\rtprof>cordova --help
Synopsis

  cordova command [options]

Global Commands
  create ..... Create a project
  help ..... Get help for a command
  telemetry ..... Turn telemetry collection on or off
  config ..... Set, get, delete, edit, and list global cordova options

Project Commands
  info ..... Generate project information
  requirements ..... Checks and print out all the requirements
                    for platforms specified

  platform ..... Manage project platforms
  plugin ..... Manage project plugins

  prepare ..... Copy files into platform(s) for building
  compile ..... Build platform(s)
  clean ..... Cleanup project from build artifacts

  run ..... Run project
             (including prepare && compile)
  serve ..... Run project with a local webserver
             (including prepare)

Learn more about command options using 'cordova help <command>'

Aliases
  build -> cordova prepare && cordova compile
  emulate -> cordova run --emulator

Options
  -v, --version ..... prints out this utility's version
  -d, --verbose ..... debug mode produces verbose log output for all activity,
  --no-update-notifier ..... disables check for CLI updates
  --nohooks ..... suppress executing hooks
                    (taking RegExp hook patterns as parameters)

Examples
  cordova create myApp org.apache.cordova.myApp myApp
  cordova plugin add cordova-plugin-camera
  cordova platform add android
  cordova plugin add cordova-plugin-camera --nosave
  cordova platform add android --nosave
  cordova requirements android
  cordova build android --verbose
  cordova run android
  cordova build android --release -- --keystore="..\android.keystore" --storePassword=android --alias=
mykey
  cordova config ls
```