

Tutoriel installation Apache Cordova

Outil pour faire du développement mobile (iOS/Android/Windows) en Javascript.

Prérequis : Java (JDK 8), Node.JS, SDK Android v30 (pour développement Android), SDK Windows 10 (pour développement Windows 10).

Ouvrez un terminal (Windows powershell ou invite de commandes), en mode administrateur.

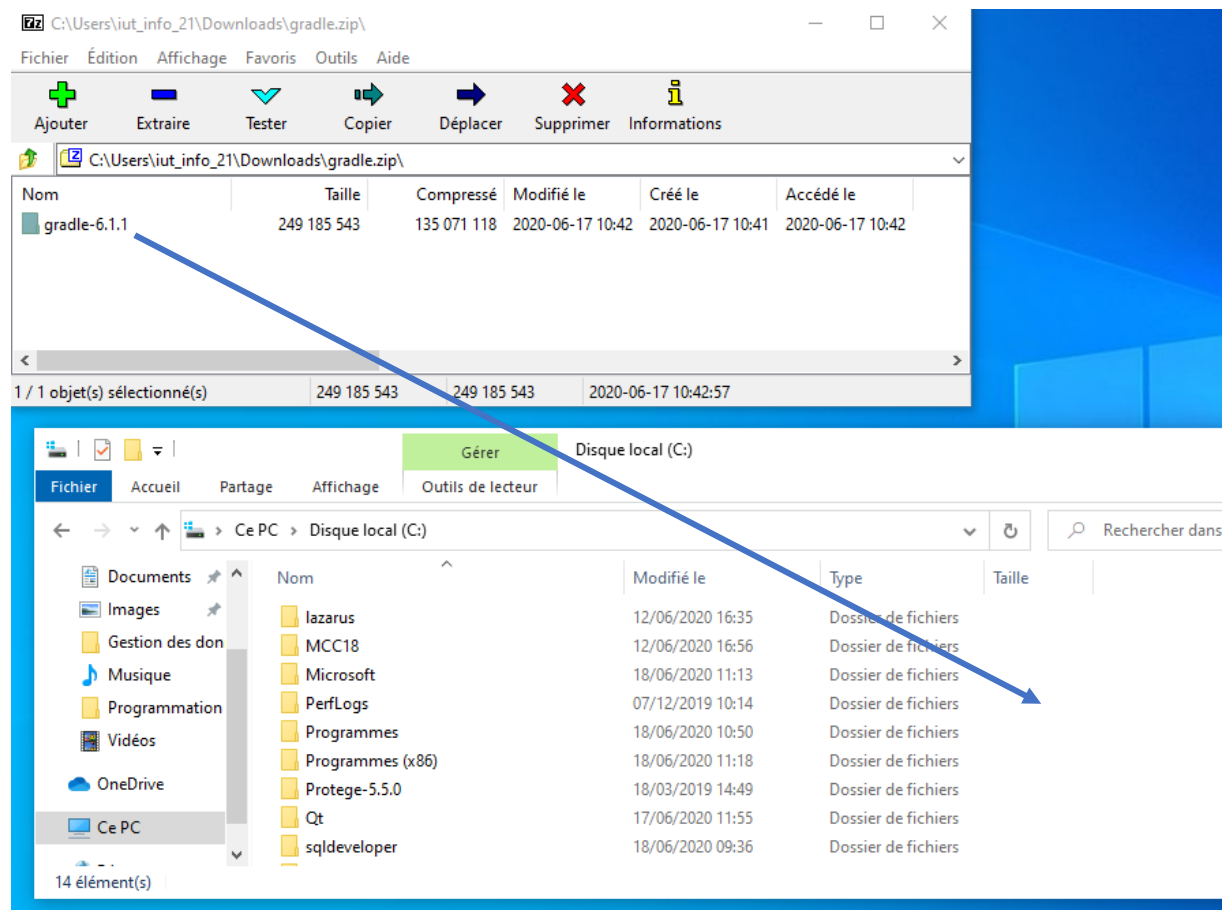
Tapez la commande suivante :

```
npm install -g cordova
```

Si la commande npm n'est pas trouvée, il faut installer Node.JS.

Si vous voulez faire du développement Android, installez Gradle. Pour cela, téléchargez ce fichier : [cd](#)

Une fois le fichier téléchargé, il faut l'ouvrir et extraire le dossier contenu dans la racine.

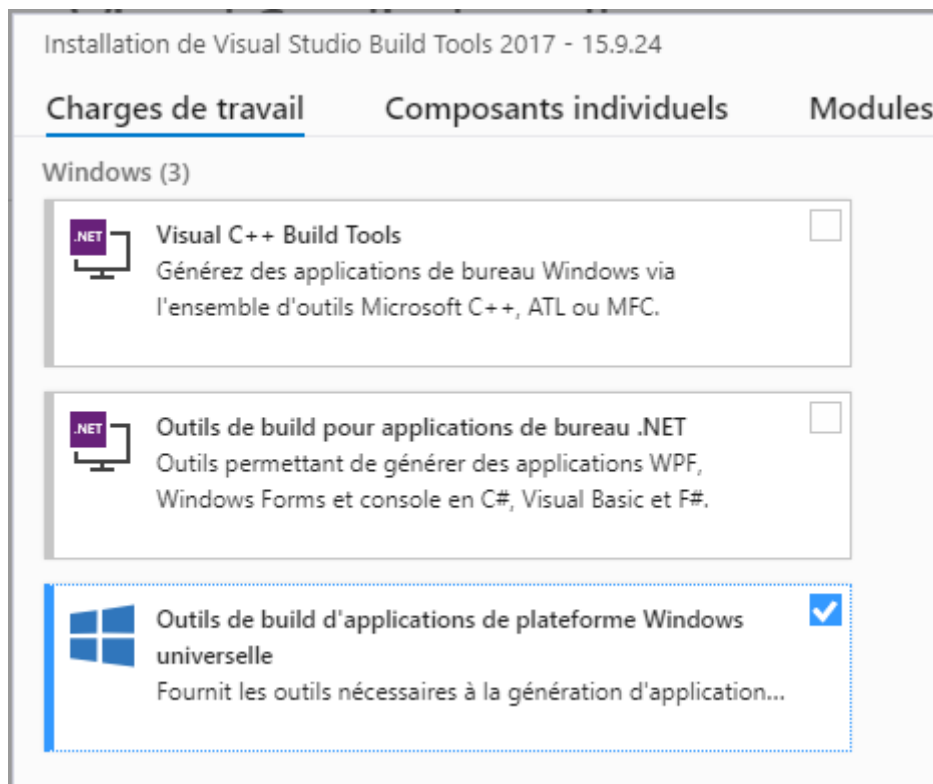


Si vous souhaitez développer pour Windows 10, il faut installer les outils de build de Visual Studio 2017 (Cordova n'est pas encore compatible avec les outils de la version 2019 ni de la version 2022).

Téléchargez ce fichier : <https://iutdijon.u-bourgogne.fr/intra/info/softs/files/vs2017.zip>

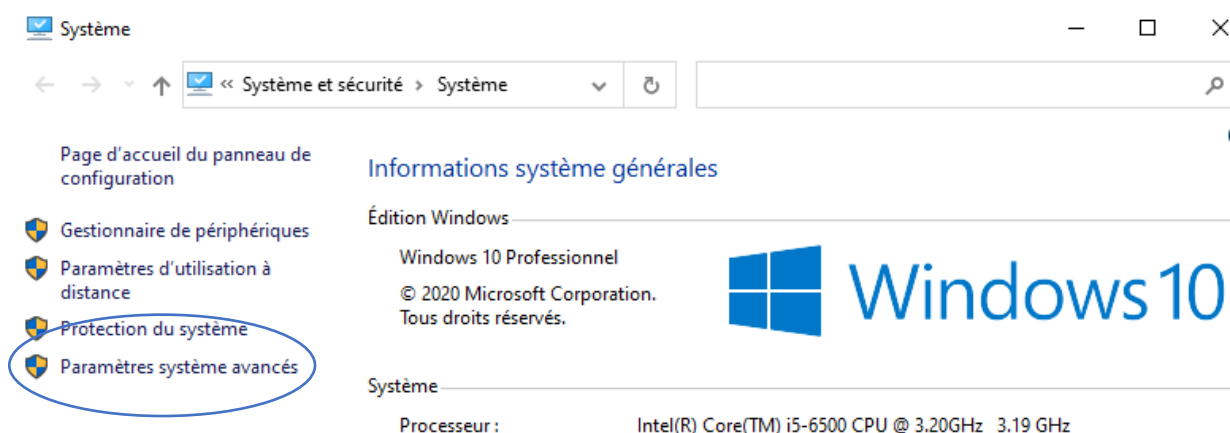
Ouvrez l'archive et exécutez l'installateur contenu.

Cochez uniquement les outils de *build* pour la plateforme universelle windows (ce qui est déjà assez gros) :

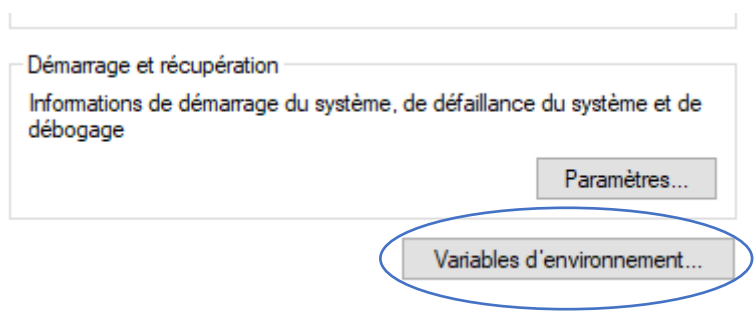


Cette installation terminée, il faut ensuite rentrer un certain nombre de variables d'environnement.

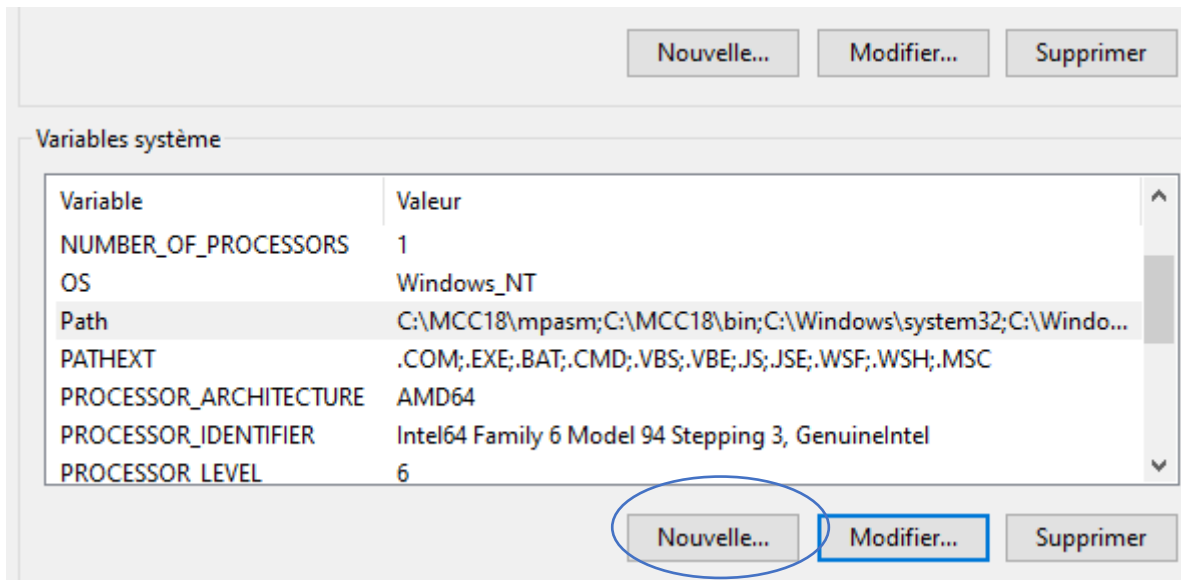
Tapez la combinaison de touches Windows+Pause (ou tapez « Système » dans la barre de recherche). Choisir ensuite « paramètres système avancés ».



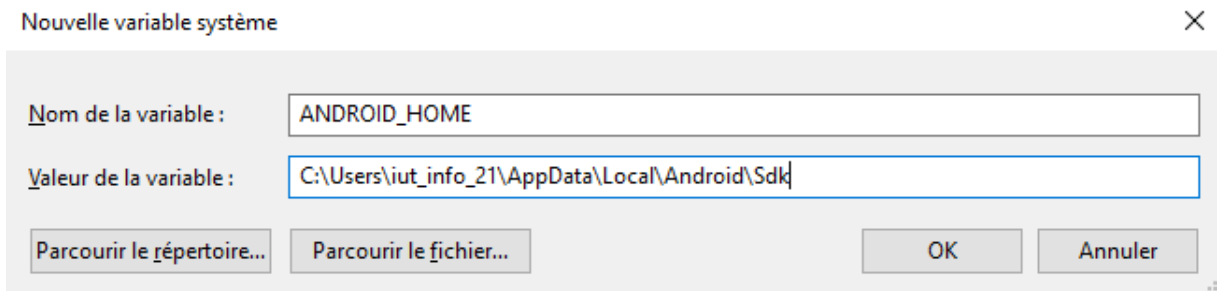
Cliquez ensuite sur « variables d'environnement »



Cliquez sur le bouton « nouvelle » en bas de la liste des variables système :

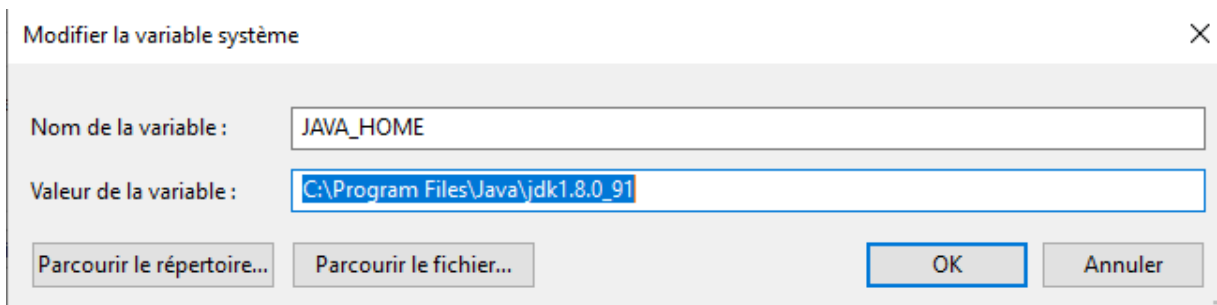


Créer une variable `ANDROID_HOME` pointant sur le dossier de votre SDK Android (adaptez à votre propre chemin. Si vous ne retrouvez pas le chemin du SDK, utilisez Android Studio pour le retrouver).

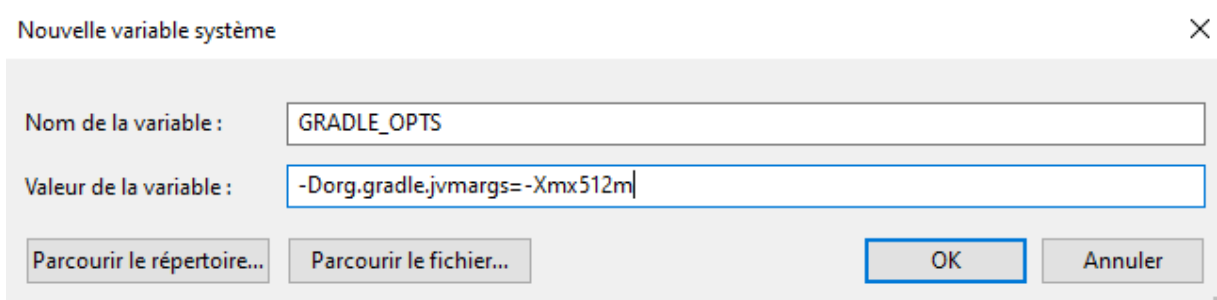


Créez de la même manière une variable `ANDROID_SDK_ROOT` possédant la même valeur.

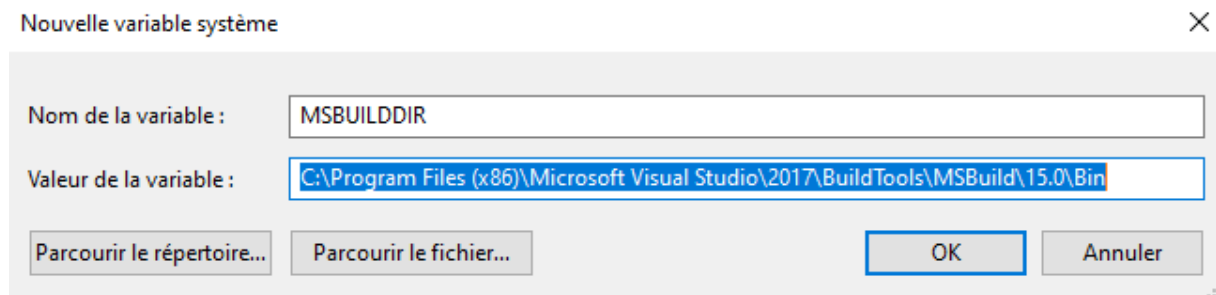
Créez ensuite une variable `JAVA_HOME` qui pointe sur le dossier de votre JDK.



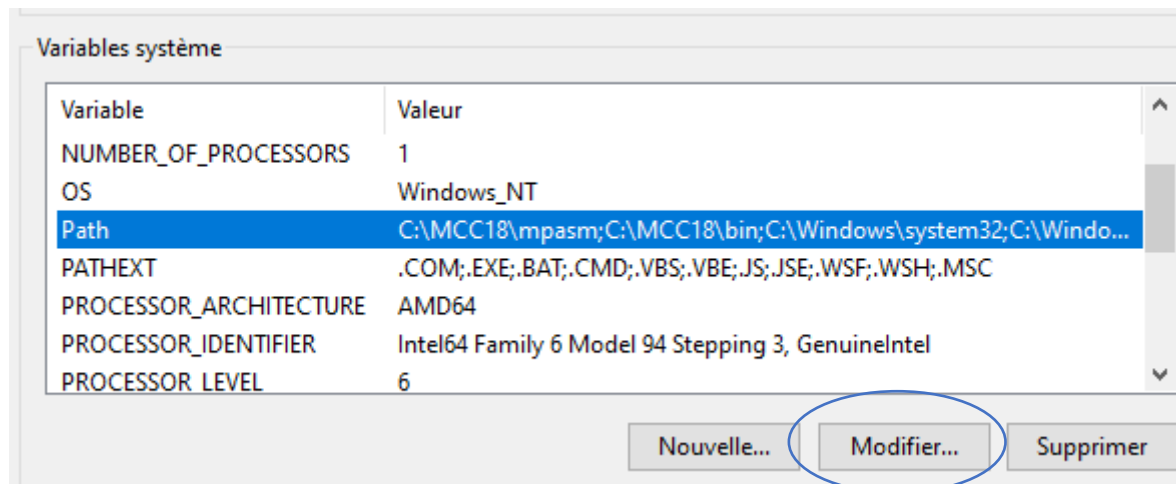
Créez une variable `GRADLE_OPTS` qui prend la valeur `-Dorg.gradle.jvmargs=-Xmx512m` :



Créez une nouvelle variable MSBUILDDIR et donnez-lui le chemin des outils de build installés précédemment (normalement c'est le chemin ci-dessous) :

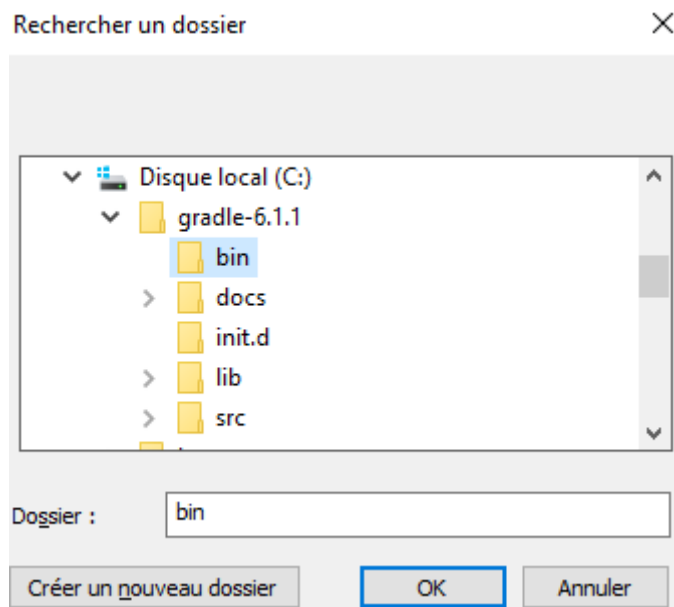


Dans la liste des variables système, sélectionnez « Path » et cliquez sur « modifier ».

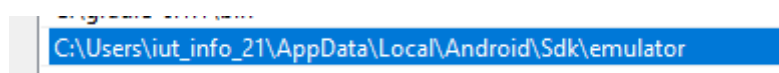


Dans la fenêtre suivante, cliquez sur « nouveau » puis « parcourir ».

Cherchez le dossier bin du dossier gradle précédemment installé.



Cliquez encore sur « nouvelle » puis parcourir, et allez chercher le chemin de l'émulateur Android :



Ouf ! Cordova est normalement installé, il reste à le tester...

Ouvrez un terminal et placez-vous dans le dossier où vous voulez créer votre application (par exemple, le bureau). Tapez la commande suivante pour créer une application de test :

```
cordova create test fr.iut.test Test
```

Une fois le projet créé, il faut se placer dans le répertoire créé (il porte le nom du projet)

```
cd test
```

Pour tester une application Windows, il faut saisir la ligne suivante :

```
cordova platforms add windows
```

Pour tester une application Android, il faut saisir la ligne suivante :

```
cordova platforms add android
```

Vous pouvez bien sûr ajouter les 2 plateformes.

Utilisez la commande suivante pour compiler (windows et/ou android) :

```
cordova build windows
```

Pour lancer l'exécution sous windows, il faut utiliser la commande

```
cordova run windows
```

Pour lancer l'exécution sous un périphérique Android connecté, il faut utiliser la commande

```
cordova run android
```

Pour lancer l'exécution sur un émulateur Android, il faut que celui-ci soit lancé et saisir la commande

```
cordova emulate android
```