

 eBook Gratuit

# APPRENEZ

---

# Cordova

eBook gratuit non affilié créé à partir des  
**contributeurs de Stack Overflow.**

#cordova

# Table des matières

À propos.....	1
<b>Chapitre 1: Commencer avec Cordova.....</b>	<b>2</b>
Remarques.....	2
Exemples.....	2
Installation ou configuration.....	2
Créer une application.....	3
<b>Preliminaire.....</b>	<b>3</b>
<b>Créer l'application.....</b>	<b>3</b>
<b>Ajout de plates-formes et de plugins.....</b>	<b>3</b>
Plates-formes.....	4
Plugins.....	4
<b>Exécuter votre application.....</b>	<b>4</b>
Installez Cordova sur Windows.....	5
Des versions.....	9
<b>Chapitre 2: Commencer avec Cordova.....</b>	<b>10</b>
Introduction.....	10
Exemples.....	10
Créer une version Android (.apk).....	10
<b>Chapitre 3: Comment détecter l'état de la connexion réseau.....</b>	<b>13</b>
Exemples.....	13
Utilisation du plug-in cordova-plugin-network-information.....	13
<b>Chapitre 4: Comment installer / désinstaller un plugin Cordova personnalisé.....</b>	<b>15</b>
Exemples.....	15
Utilisez la commande plugman pour installer / désinstaller le plug-in cordova.....	15
<b>Chapitre 5: Comment personnaliser le dossier www spécifique à la plate-forme dans cordova... 16</b>	<b>16</b>
Exemples.....	16
créer des css / js spécifiques à une plateforme (android / ios).....	16
<b>Chapitre 6: Cordova Crop Image Plugin.....</b>	<b>18</b>
Exemples.....	18

Recadrer l'image après avoir cliqué en utilisant l'appareil photo ou en sélectionnant l'im.....	18
<b>Chapitre 7: Cordova ios construire.....</b>	<b>20</b>
Introduction.....	20
Remarques.....	20
Exemples.....	20
Cordova HelloWorld Project.....	20
<b>Chapitre 8: Créer votre première application avec Cordova.....</b>	<b>23</b>
Remarques.....	23
Exemples.....	23
Utilisation de l'outil de ligne de commande.....	23
<b>Chapitre 9: Déboguer l'application.....</b>	<b>25</b>
Remarques.....	25
Exemples.....	25
Déboguer sur un appareil Android via USB.....	25
Déboguer les applications Cordova utilisant GapDebug.....	25
Déboguer sur un appareil iOS avec USB.....	26
<b>Chapitre 10: Firebase Push Notification Cordova.....</b>	<b>28</b>
Exemples.....	28
Firebase Push Notification dans Cordova Android.....	28
Cordova Firebase Push Notification Plugin.....	28
<b>Chapitre 11: Google Analytics à Cordova.....</b>	<b>31</b>
Exemples.....	31
Google Analytics en Cordova sans aucun plugin.....	31
<b>Chapitre 12: Notification Push dans Android et iOS.....</b>	<b>33</b>
Exemples.....	33
Utilisation du nouveau plug-in-plug-in phonegap.....	33
<b>Chapitre 13: Plugins Cordova: comment installer, comment ils fonctionnent, exemples.....</b>	<b>36</b>
Exemples.....	36
Que sont les plugins Cordova?.....	36
Comment les plugins Cordova peuvent-ils être utiles?.....	36
Installation de Cordova Plugin.....	36

Plugins les plus populaires.....	36
<b>Chapitre 14: Rendre l'application libérée de Cordova CLI.....</b>	<b>38</b>
Exemples.....	38
Android.....	38
iOS.....	39
<b>Chapitre 15: Signez Android avec Cordova 5.....</b>	<b>40</b>
Exemples.....	40
Ajoutez la configuration de génération pour signer le fichier .apk.....	40
<b>Chapitre 16: Visual Studio Tools pour Apache Cordova.....</b>	<b>41</b>
Exemples.....	41
Obtenir les outils Apache Cordova dans Visual Studio.....	41
Mettre à jour les outils Apache Cordova dans Visual Studio.....	41
<b>Crédits.....</b>	<b>42</b>

# À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [cordova](#)

It is an unofficial and free Cordova ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official Cordova.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Chapitre 1: Commencer avec Cordova

## Remarques

Apache Cordova est utilisé pour créer des applications mobiles avec HTML, CSS et JS.

Apache Cordova cible plusieurs plates-formes avec une base de code.

Apache Cordova est gratuit et open source.

***Cordova encapsule votre application HTML / JavaScript dans un conteneur natif pouvant accéder aux fonctions de périphérique de plusieurs plates-formes. Ces fonctions sont exposées via une API JavaScript unifiée, vous permettant d'écrire facilement un ensemble de codes pour cibler presque tous les téléphones ou tablettes sur le marché aujourd'hui et de les publier dans leurs magasins d'applications.***

Qui pourrait utiliser Apache Cordova

1. un développeur mobile et souhaite étendre une application sur plusieurs plates-formes, sans avoir à la ré-implémenter avec la langue et les outils de chaque plate-forme.
2. un développeur Web et souhaitez déployer une application Web qui est emballée pour la distribution dans divers portails de la boutique d'applications.
3. un développeur mobile intéressé par le mélange de composants d'application natifs avec une fenêtre WebView (fenêtre de navigateur spéciale) capable d'accéder aux API au niveau du périphérique ou si vous souhaitez développer une interface de plug-in entre les composants natifs et WebView.

**Introduction à Cordova:** <https://cordova.apache.org/docs/en/latest/>

## Exemples

### Installation ou configuration

Pour installer l'outil de ligne de commande **cordova** , procédez comme suit:

1. Téléchargez et installez [Node.js](#). Lors de l'installation, vous devriez pouvoir invoquer **node** et **npm** sur votre ligne de commande.
  - Pour voir si Node est installé, ouvrez votre interface de ligne de commande (CLI). Pour Windows, il s'agit de l'invite de commande Windows. Pour MAC, c'est le terminal.  
Type:

```
$ node -v
```

Cela devrait imprimer un numéro de version, vous verrez donc quelque chose comme

ceci v0.10.35. Si Node n'est pas installé, trouvez votre système d'exploitation et suivez les instructions ici: <https://nodejs.org/en/download/package-manager/>

2. (Facultatif) Téléchargez et installez un [client git](#) , si vous n'en avez pas déjà un. Après l'installation, vous devriez pouvoir appeler **git** sur votre ligne de commande. La CLI l'utilise pour télécharger des actifs lorsqu'ils sont référencés à l'aide d'une URL vers un référentiel git.
3. Installez le module **cordova** à l'aide de l'utilitaire **npm** de Node.js. Le module **cordova** sera automatiquement téléchargé par l'utilitaire **npm** .

sur OS X et Linux:

```
$ sudo npm install -g cordova
```

sur Windows:

```
C:\>npm install -g cordova
```

L'indicateur **-g** ci-dessus indique à **npm** d'installer **cordova** globalement. Sinon, il sera installé dans le sous-répertoire **node\_modules** du répertoire de travail en cours.

Après l'installation, vous devriez être capable d'exécuter **cordova** sur la ligne de commande sans arguments et cela devrait imprimer du texte d'aide.

## Créer une application

# Préliminaire

Installez les outils cli de Cordova, si vous ne l'avez pas déjà fait.

```
$ npm install -g cordova
```

Accédez au dossier de travail de votre choix.

```
$ cd /path/to/coding/folder
```

# Créer l'application

Créer une nouvelle application

```
$ cordova create <appProjectName> <appNameSpace> <appName>
```

Pour cet exemple, nous allons créer une application "HelloWorld":

```
$ cordova create helloWorld com.example.helloworld HelloWorld
```

# Ajout de plates-formes et de plugins

# Plates-formes

Tout d'abord, accédez au dossier de l'application.

```
$ cd <appName>
```

Ajoutez les plates-formes pour lesquelles vous souhaitez construire. La liste des plates-formes prises en charge peut être trouvée [ici](#).

```
$ cordova platform add <platformList>
```

Nous allons ajouter la plate-forme Android, iOS et le navigateur. Utilisez la séparation d'espace pour ajouter plusieurs plates-formes à la fois. La plate-forme de `browser` sera utile pour les tests dans le navigateur.

L'utilisation de l'argument `--save` permet d'enregistrer la liste de plates-formes dans le fichier `config.xml` de Cordova.

```
$ cordova platform add android ios browser --save
```

Une liste complète des options concernant la commande de `platform - platform` se trouve dans la [documentation de cordova](#).

## Plugins

Les plug-ins Cordova peuvent vous donner accès au matériel de l'appareil, aux fonctions spécifiques du système d'exploitation et à bien d'autres fonctionnalités.

La structure de la commande `plugin` est la même que celle des plates-formes

```
$ cordova plugin add <plugins.value>
```

Nous allons ajouter le plug-in de fichier cordova (pour un accès facile au stockage de l'appareil) et le plug-in de caméra, qui donne accès à la caméra de l'appareil pour créer des photos et des vidéos.

```
$ cordova plugin add cordova-plugin-file cordova-plugin-camera --save
```

**Rappelez - vous :** l'utilisation de l'argument `--save` écrit vos paramètres dans le fichier `config.xml`. Très utile pour recréer facilement le projet sur une autre machine.

Cordova a une excellente configuration de page de recherche de plugin pour votre commodité. Vous pouvez le trouver [ici](#).

---

## Exécuter votre application

L'exécution de l'application est assez simple. Utilisez simplement la commande suivante.

```
$ cordova run <platform name>
```

Pour notre exemple, nous allons exécuter notre application de test dans le navigateur.

```
$ cordova run browser
```

Cela ouvre votre navigateur par défaut avec votre application prête pour le test.



## Installez Cordova sur Windows

### Tout d'abord, installez Java SE Development Kit

Cela peut être aussi simple que de télécharger, de double-cliquer sur le fichier téléchargé et de suivre les instructions d'installation. Pour installer Java SE Development Kit, téléchargez-le à partir du site Web officiel. [Kit de développement Java SE. Téléchargements](#)

Une fois l'installation du JDK terminée, vous devez ajouter une nouvelle variable système `JAVA_HOME` avec le chemin d'accès à votre JDK.

À côté de la variable système `PATH`, ajoutez un chemin vers bin directory de JDK

Maintenant, vous pouvez tester l'installation. Ouvrir l'invite de commande et utiliser la commande

```
javac -version
```

Si vous voyez un numéro de version, vous avez tout fait correctement!

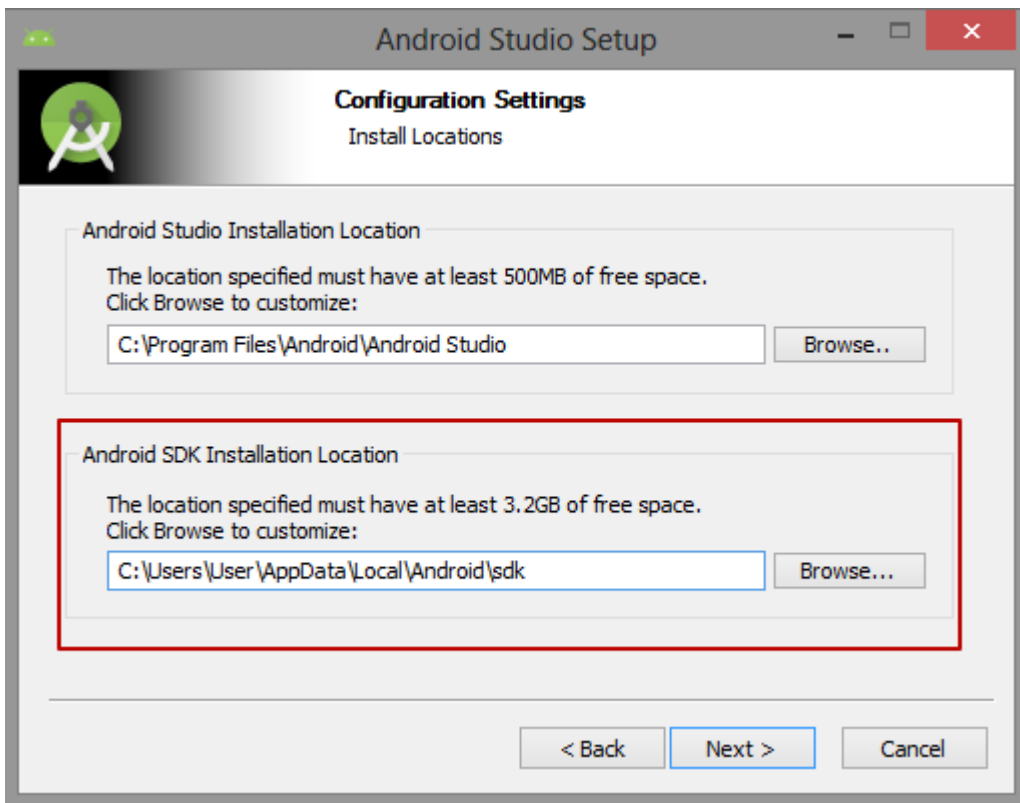
### Maintenant, installez les outils Android SDK avec Android Studio

J'ai recommandé d'installer Android Studio car pour le moment c'est le meilleur moyen d'installer rapidement et facilement toutes les choses nécessaires pour le développement Android. La liste des choses comprend:

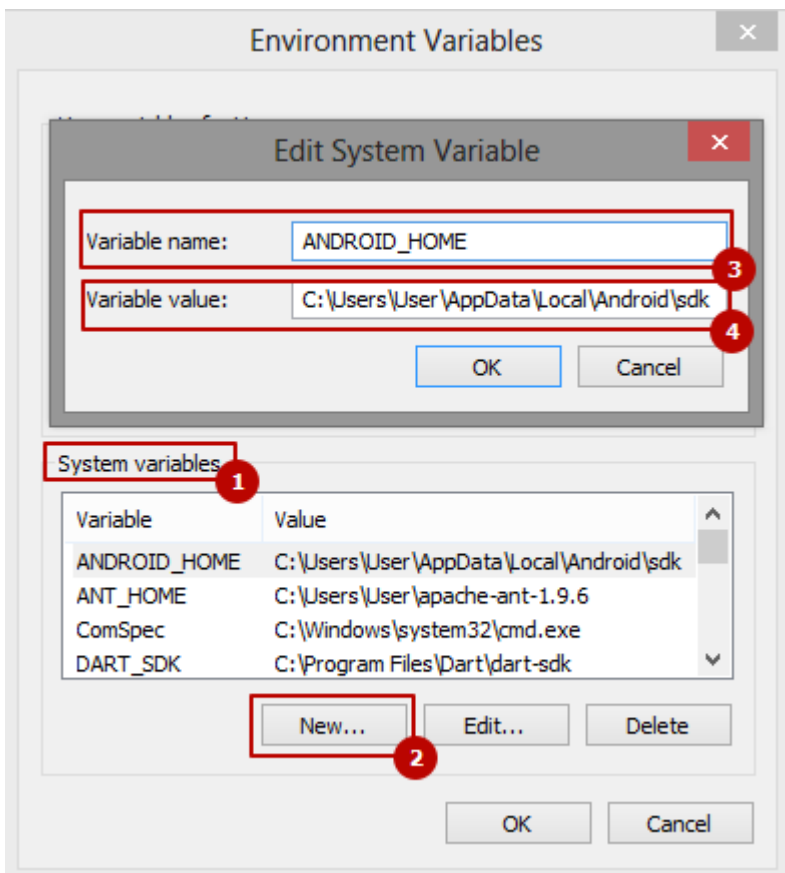
- Kit de développement Android (SDK Android, Gestionnaire de SDK Android, Outils de plateforme SDK Android, Outils de création de SDK Android)
- Émulateur Android avec un grand nombre de configurations Android
- IDE (pour le développement Android sur Java)
- Gradle
- Ce serait très utile si vous apprenez Java et que, dans le futur, vous voulez commencer à développer pour Android sur Java

Alors, téléchargez Android Studio depuis le site officiel [developer.android.com](https://developer.android.com)

L'installation d'Android Studio est très simple et il vous suffit de suivre les instructions. Mais vous devriez prendre note de l'emplacement d'installation du SDK Android

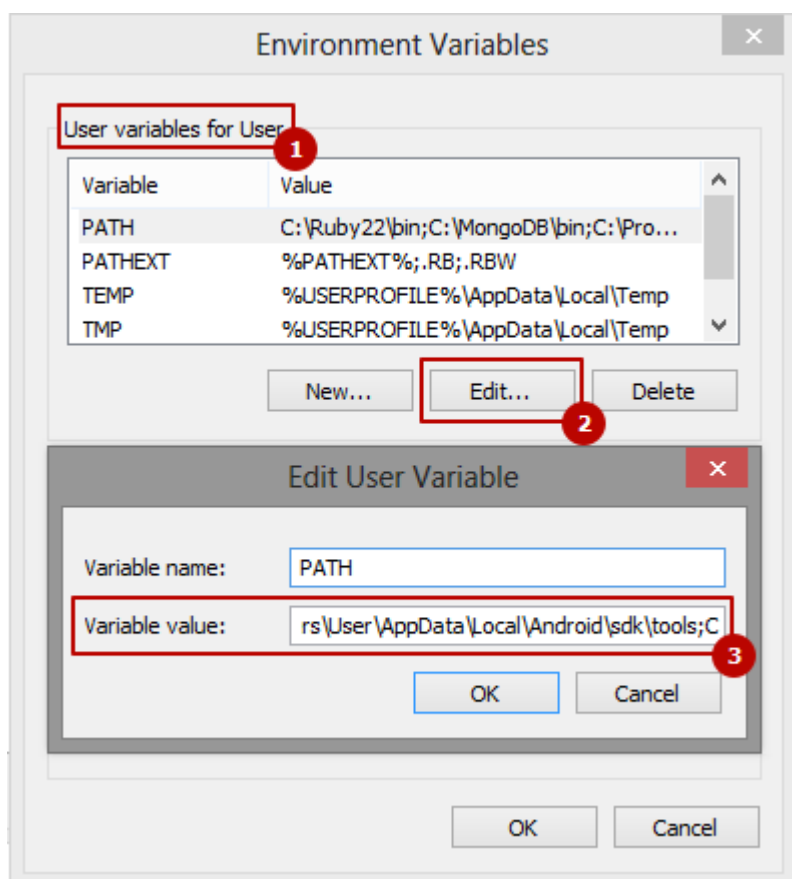


Une fois l'installation d'Android Studio terminée, vous devez ajouter une nouvelle variable système `ANDROID_HOME` avec le chemin d'accès à votre Android SDK



Vous devez maintenant ajouter les outils Android SDK et Android SDK à la variable système `PATH`. Dans la liste Variables utilisateur, sélectionnez `PATH` et cliquez sur le bouton Modifier. A la fin du champ Valeur de la variable, ajoutez un point-virgule et suivez les chemins:

```
C:\Users\User\AppData\Local\Android\sdk;C:\Users\User\AppData\Local\Android\sdk\tools;C:\Users\User\AppData\Local\Android\sdk\tools;
```



Maintenant, vous pouvez tester l'installation. Ouvrir l'invite de commande et utiliser la commande

```
adb version
```

Cela devrait afficher la version du pont de débogage Android. Si vous voyez un numéro de version, vous avez tout fait correctement!

Maintenant, ouvrez de nouveau l'invite de commandes et utilisez la commande








































```
android
```

pour ouvrir Android SDK Manager

Packages Tools

SDK Path: C:\Users\User\AppData\Local\Android\sdk

Packages

 Name	API	Rev.	Status
<input type="checkbox"/>  Tools			
<input type="checkbox"/>  Android SDK Tools		24.4.1	 Installed
<input type="checkbox"/>  Android SDK Platform-tools		23.1	 Installed
<input type="checkbox"/>  Android SDK Build-tools		23.0.3	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Android SDK Build-tools		23.0.2	 Installed
<input type="checkbox"/>  Android SDK Build-tools		23.0.1	 Installed
<input type="checkbox"/>  Android SDK Build-tools		22.0.1	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Android SDK Build-tools		21.1.2	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Android SDK Build-tools		20	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Android SDK Build-tools		19.1	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Tools (Preview Channel)			
<input type="checkbox"/>  Android SDK Tools		25.0.1...	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Android SDK Platform-tools		24 rc1	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Android SDK Build-tools		24 rc2	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Android N (API 23, N preview)			
<input checked="" type="checkbox"/>  SDK Platform Android N Preview	N	1	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Android TV Intel x86 Atom System Image	N	1	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Intel x86 Atom_64 System Image	N	1	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Intel x86 Atom System Image	N	1	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Android 6.0 (API 23)			
<input type="checkbox"/>  Documentation for Android SDK	23	1	 Installed
<input type="checkbox"/>  SDK Platform	23	2	 Installed
<input type="checkbox"/>  Samples for SDK	23	2	 Installed
<input checked="" type="checkbox"/>  Android TV ARM EABI v7a System Image	23	2	 Update available: rev. 3
<input checked="" type="checkbox"/>  Android TV Intel x86 Atom System Image	23	2	 Update available: rev. 3
<input type="checkbox"/>  Android Wear ARM EABI v7a System Image	23	2	 Installed
<input type="checkbox"/>  Android Wear Intel x86 Atom System Image	23	2	 Installed

Show:  Updates/New  Installed [Select New or Updates](#) Obsolete[Deselect All](#)

Done loading packages.

Dans le gestionnaire de SDK Android, sélectionnez pour installer

- Outils du SDK Android
- Outils de plate-forme SDK Android
- Outils de création de SDK Android
- Outils de création de SDK Android
- Android 6.0 (API 23)
- Android 5.1.1 (API 22)
- Android 5.0.1 (API 21)
- Android 4.2.2 (API 17)
- Outils de débogage GPU

- Référentiel de support Android
- Bibliothèque de support Android
- Services Google Play
- Google Repository
- Pilote USB Google
- Intel x86 Emulator Accelerator (programme d'installation HAXM)

et cliquez sur le bouton Installer.

Remarque:

[Niveaux d'API pris en charge par Cordova Android](#)

[Comprendre les niveaux d'API Android](#)

[Plate-forme Android / API Version Distribution](#)

## Installer Cordova

Ouvrez l'invite de commandes et installez Cordova en utilisant la commande

```
npm install -g cordova
```

## Des versions

Dernière version de Cordova:

Cordova 6.1.0 - <https://cordova.apache.org/news/2016/03/23/tools-release.html> Cordova 6.0.0 - <https://cordova.apache.org/news/2016/01/28/tools-release.html>

Dernière plateforme Android et plate-forme iOS

Cordova Android 5.2.2 - <https://cordova.apache.org/announcements/2016/07/02/android-5.2.0.html> Cordova iOS 4.2.1 - <https://cordova.apache.org/announcements/2016/07/11/cordova-android-5.2.1.html>

**Lire Commencer avec Cordova en ligne:** <https://riptutorial.com/fr/cordova/topic/884/commencer-avec-cordova>

---

# Chapitre 2: Commencer avec Cordova

## Introduction

Développement d'applications mobiles utilisant Cordova

Apache Cordova est un framework de développement mobile open-source. Il vous permet d'utiliser des technologies Web standard - HTML5, CSS3 et JavaScript pour le développement multi-plateforme.

## Exemples

### Créer une version Android (.apk)

Installez cordova en utilisant la commande suivante **npm install -g cordova** .

Utilisez `cordova -version` pour vérifier la **version cordova** .

Définissez les variables de chemin `ANDROID_HOME` et `JAVA_HOME`.

Exemple:

```
export ANDROID_HOME = / home / geethu / android-sdk-linux
```

```
export PATH = $ PATH: $ ANDROID_HOME / tools: $ ANDROID_HOME / platform-  
tools
```

```
export JAVA_HOME = / usr / lib / jvm / java-7-openjdk-amd64
```

```
export PATH = $ PATH: $ JAVA_HOME / bin
```

Créez un projet Cordova vide à l'aide de l'outil de ligne de commande. Accédez au répertoire où vous souhaitez créer votre projet et tapez.

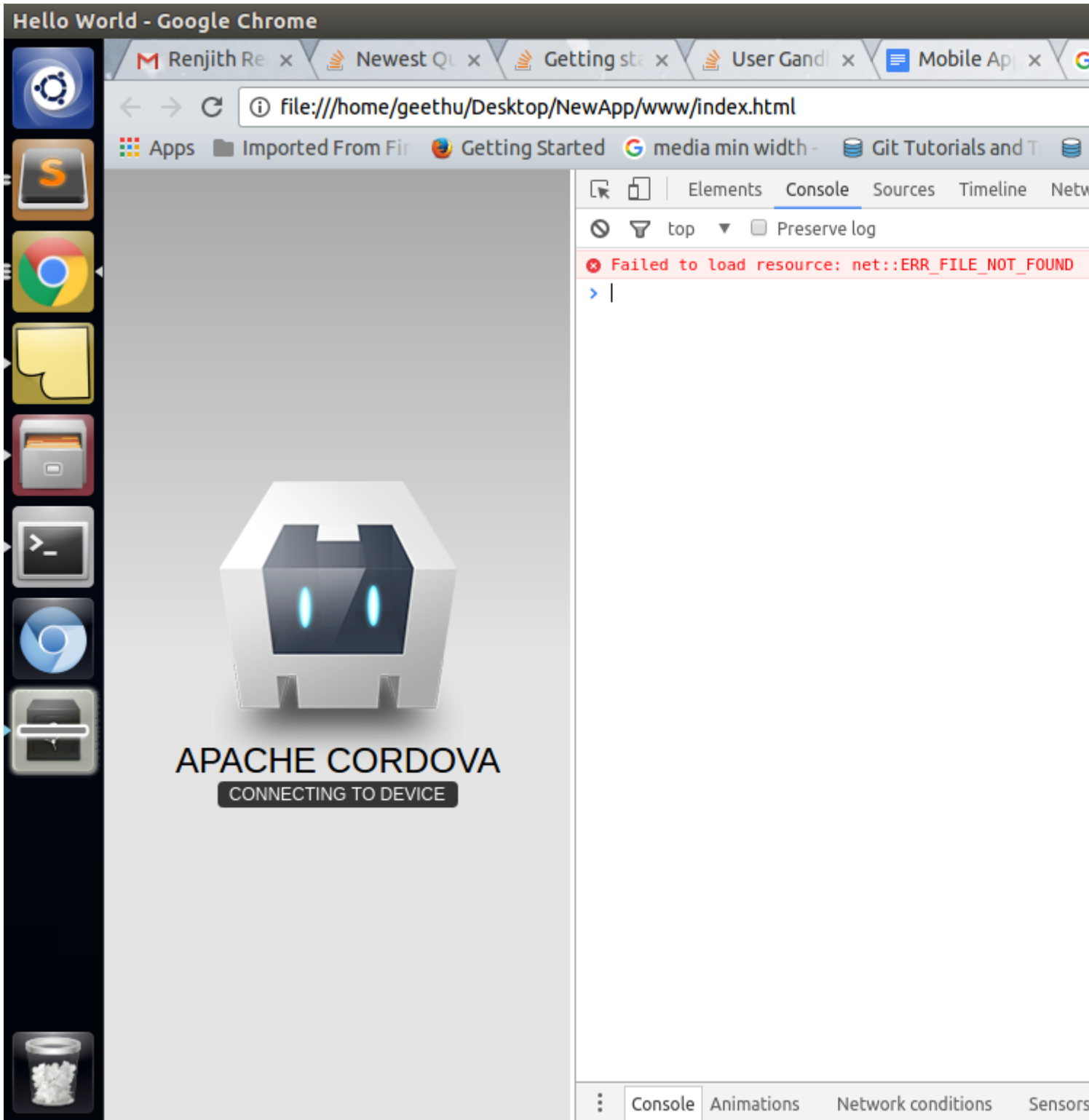
**cordova créer atelier com.yourname.workshop atelier**

Cordova CLI, créez un projet Cordova nommé Workshop dans un répertoire nommé workshop.

Accédez au répertoire de `cd atelier` du projet.

Ajouter la plate-forme Android comme plate-forme **Cordova ajouter Android** .

[



] 1

Ajoutez ces fichiers de script à votre index.html dans l'ordre suivant.

```
<script type = "text/javascript" src = "cordova.js"></script>
```

Pour modifier le nom et l'icône de l'application, modifiez le fichier config.xml dans le dossier www.

Pour construire le projet dans le dossier atelier / plates-formes / Android

### Cordova construire Android

Exécutez-le sur un appareil Android connecté à votre ordinateur à l'aide d'un câble USB, tapez:

## Cordova courir Android

---

### 1. > Installation de Cordova .apk sur le périphérique

---

---

En outre, je souhaiterais ajouter plus de détails sur la façon d'installer le fichier .apk sur votre mobile Android, tout en déployant votre application sur le serveur.

```
cordova plugin add cordova-plugin-mfp
```

Exécutez la commande ci-dessus, puis essayez d'ajouter le serveur sur lequel vous souhaitez déployer l'application.

```
mfpdev server add
```

puis suivez les instructions de CLI. exécutez la commande ci-dessous pour modifier vos paramètres de configuration.

```
mfpdev app config server <profile Name of server>  
  
cordova build
```

il va créer un fichier .apk dans le

/ plates-formes / dossier Android

téléchargez l'apk sur votre mobile, installez-le et jouez avec votre application.

Lire Commencer avec Cordova en ligne: <https://riptutorial.com/fr/cordova/topic/9287/commencer-avec-cordova>



# Chapitre 3: Comment détecter l'état de la connexion réseau

## Exemples

### Utilisation du plug-in cordova-plugin-network-information

La détection de l'état actuel de la connexion réseau et la réponse aux modifications éventuelles peuvent être effectuées à l'aide de l'un des plug-ins. Cet exemple concerne le [plug-in cordova-plugin-network-information](#).

Ajoutez le plugin au projet:

```
cordova plugin add cordova-plugin-network-information
```

Après l'événement d'événement Cordova, un objet de connexion est disponible via `navigator.connection`. La propriété `type` contient l'état actuel du réseau:

```
document.addEventListener("deviceready", function() {  
    var networkState = navigator.connection.type;  
}, false);
```

`networkState` contient maintenant l'une des constantes suivantes:

```
Connection.UNKNOWN // Unknown connection  
Connection.ETHERNET // Ethernet connection  
Connection.WIFI // WiFi connection  
Connection.CELL_2G // Cell 2G connection  
Connection.CELL_3G // Cell 3G connection  
Connection.CELL_4G // Cell 4G connection  
Connection.CELL // Cell generic connection  
Connection.NONE // No network connection
```

La détection d'une modification de la connexion réseau peut être effectuée en connectant une fonction à l'événement en `online` ou `offline`:

```
document.addEventListener("online", function() {  
    // device went online  
    var networkState = navigator.connection.type; // Get new network state  
    ...  
}, false);  
  
document.addEventListener("offline", function() {  
    // device went offline  
    var networkState = navigator.connection.type; // Get new network state  
    ...  
}, false);
```

Lire Comment détecter l'état de la connexion réseau en ligne:

<https://riptutorial.com/fr/cordova/topic/3615/comment-detecter-l-etat-de-la-connexion-reseau>

---

# Chapitre 4: Comment installer / désinstaller un plugin Cordova personnalisé

## Exemples

Utilisez la commande plugman pour installer / désinstaller le plug-in cordova

Vous pouvez utiliser la commande plugman pour installer / désinstaller des plugins Cordova personnalisés.

Pour installer plugman

```
npm install -g plugman
```

Installer la syntaxe de la commande plugin:

```
plugman <install|uninstall> --platform <ios|android|blackberry10|wp8> --project <directory> --  
plugin <name|url|path>
```

Exemple:

```
plugman install --platform ios --project platforms/ios/ --plugin plugins/cordova-plugin-test/
```

Pour plus d'options de plugman, cochez [ce lien](#)

Lire [Comment installer / désinstaller un plugin Cordova personnalisé en ligne:](#)

<https://riptutorial.com/fr/cordova/topic/7506/comment-installer---desinstaller-un-plugin-cordova-personnalise>

---

# Chapitre 5: Comment personnaliser le dossier www spécifique à la plate-forme dans cordova

## Exemples

### créer des css / js spécifiques à une plateforme (android / ios)

Disons que vous voulez créer un fichier css / js spécifique à une plateforme. Pour cela, vous devez créer un dossier de **fusion** dans le dossier racine de votre ordinateur. Dans le dossier de fusion crée un répertoire pour chaque plate-forme (Android / iOS ..). puis, dans un dossier de plate-forme spécifique, créez un dossier css / js et placez votre fichier css / js spécifique au dossier de la plate-forme. Voilà, une fois que vous avez exécuté la commande de **compilation cordova** , tous les fichiers js / css correspondant à chaque plate-forme seront placés dans un dossier de plate-forme respecté

**Remarque:** Assurez-vous que votre racine www / index.html aura les mêmes css / js définis. Pour cela, assurez-vous que vous avez le même nom de fichier correspondant à chaque plate-forme dans le dossier de fusion.

```
//let say you are in CordovaMergesExample folder
cd CordovaMergesExample

//create test folder with com.test id and TestApp as name
cordova create test com.test TestApp

//add platform android and ios
cordova platform add android ios

-----

//create merges/android/css/override.css and merges/ios/css/override.css

-----

//In root www/index.html add this stylesheet
<link rel="stylesheet" type="text/css" href="css/override.css" />

-----

cordova build

---> cordova build engine automatically identify the platform in merges folder and add files
in respective folders. Check platforms/android/assets/www/css and platforms/ios/www/css
```

Lire Comment personnaliser le dossier www spécifique à la plate-forme dans cordova en ligne:  
<https://riptutorial.com/fr/cordova/topic/7498/comment-personnaliser-le-dossier-www-specifique-a-la-plate-forme-dans-cordova>

---

# Chapitre 6: Cordova Crop Image Plugin

## Exemples

Recadrer l'image après avoir cliqué en utilisant l'appareil photo ou en sélectionnant l'image.

Il récolte les images en forme carrée.

Ce projet cordova utilise deux plugins:

1. Cordova Camera Plugin - <https://github.com/apache/cordova-plugin-camera>
2. Cordova Crop Image Plugin - <https://github.com/jeduan/cordova-plugin-crop>

Le plug-in Camera est associé au plug-in Crop Image en plaçant le code du plug-in Cop Image dans le rappel de succès du code du plug-in de caméra.

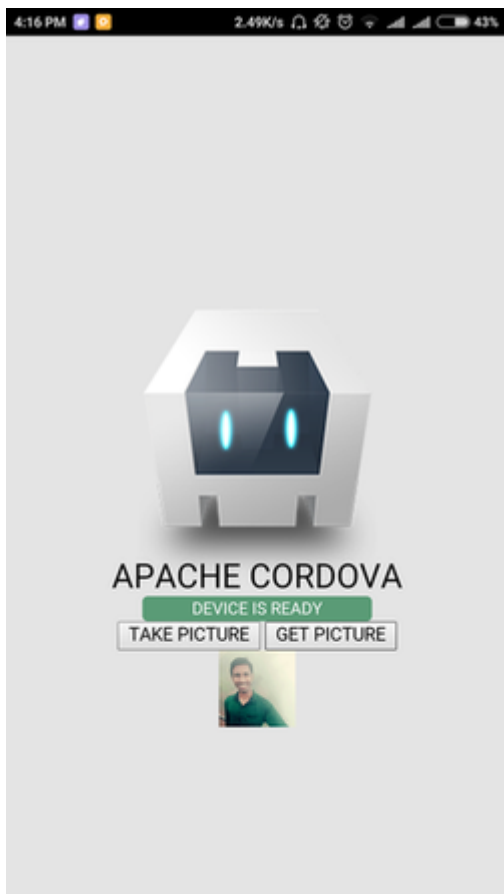
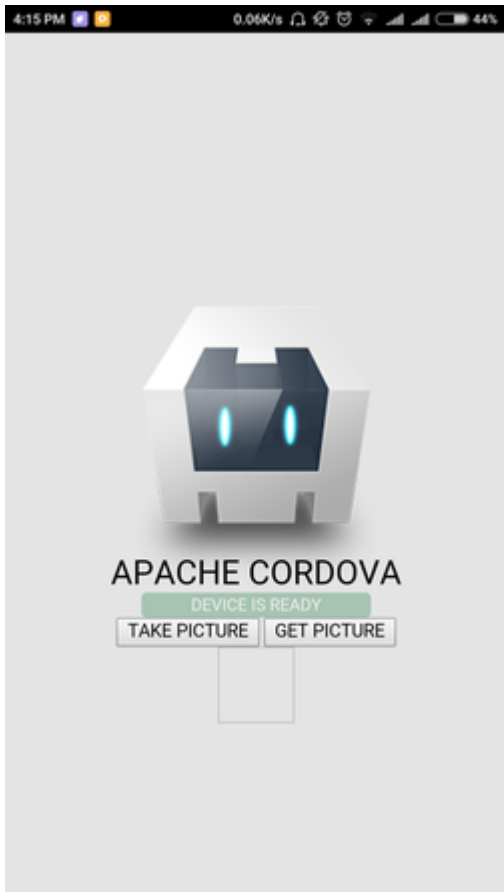
```
/*Camera Plugin Code*/
navigator.camera.getPicture(onSuccess, onFail, {
    quality: 50,
    destinationType: Camera.DestinationType.FILE_URI
});

function onSuccess(imageData) {
    console.log(imageData);

    /*Crop Image Plugin Code*/
    plugins.crop(function success (data) {
        console.log(data);
        var image = document.getElementById('myImage');
        image.src = data;
    },
    function fail () {

    }, imageData, {quality:100});
}

function onFail(message) {
    alert('Failed because: ' + message);
}
```



Lire Cordova Crop Image Plugin en ligne: <https://riptutorial.com/fr/cordova/topic/7078/cordova-crop-image-plugin>

---

# Chapitre 7: Cordova ios construire

## Introduction

J'espère que vous êtes familier avec les bases de Cordova. Essayons de construire une version de Cordova ios, la version de ios est un peu différente de celle de la construction Android, nous avons besoin de mac machine pour effectuer cette tâche. Non, vous pouvez également préparer la version en ligne d'ios, mais pour tester et déboguer votre application sur un simulateur Mac, vous devez avoir la machine Mac avec vous.

Commençons par un exemple.

## Remarques

Si vous rencontrez un problème lors de la préparation de la version de ios, n'hésitez pas à me contacter, j'essaierai de vous aider.

## Exemples

### Cordova HelloWorld Project

Pour préparer le build ios, nous devons d'abord créer le projet cordova. permet de créer le projet par l'outil de ligne de commande.

```
cordova create hello com.example.hello "HelloWorld"
```

Allez dans le répertoire du projet par `cd hello` .

nous sommes maintenant dans le répertoire du projet, permet de vérifier quelles plates-formes sont à notre disposition

```
cordova platform ls
```

Comme nous devons préparer la version ios, nous ajouterons la plateforme ios au projet.

```
cordova platform add ios@version    => put the desired version you want to add.  
cordova platform add ios@latest.    => add the latest version available.
```

Maintenant, nous avons besoin de **Xcode** sur mac machine pour préparer la version ios.

Téléchargez la dernière version de Xcode disponible via l'App Store. Veuillez ignorer les versions bêta de xcode car elles ont parfois un problème de compatibilité.

Après l'installation de Xcode, installez le clex Xcode à l'aide du terminal par la commande ci-dessous.



```
xcode-select --install
```

Il fera apparaître la fenêtre, veuillez suivre les instructions et installer le cli correctement. Maintenant, nous avons besoin d'un outil supplémentaire pour déployer la version d'ios, installez-la également.

```
npm install -g ios-deploy
```

Pendant l'installation de l'outil de déploiement, si vous obtenez l'erreur de refus d'autorisation, veuillez essayer avec la commande sudo.

```
sudo npm install -g ios-deploy
```

permet de préparer la construction ios en utilisant les commandes

```
cordova platform add ios
```

Si vous n'avez pas ajouté la plate-forme auparavant, ajoutez-la.

```
cordova prepare => it will move the all required files to platform folder and create a .xcworkspace file in the platform.ios folder.
```

```
cordova build ios => to build the ios application.
```

Mais attendez, nous n'avons pas lancé le simulateur de construction, faisons-le aussi.

vous avez deux façons de le faire

1. en utilisant cli
2. en utilisant Xcode

**Tout d'abord, faisons-le avec CLI.**

```
corodva run ios => it will run the application on the default simulator available.
```

Si vous voulez jouer avec les simulateurs, alors explorez la commande `cordova emulator`

**en utilisant Xcode**

sauter le dossier du projet et déplacez-vous dans le dossier `/platform/ios` folder de votre projet. Ouvrez le fichier `.xcworkspce` en utilisant la commande `open <ProjectName>.xcworkspcae` par exemple, `open MyApp.xcworkspce`.

il vous redirigera vers la fenêtre Xcode, sur la partie supérieure gauche de la fenêtre, vous pourrez voir votre projet, cliquez sur le projet, et en haut de l'écran, vous pourrez voir le bouton Exécuter, cliquez dessus et jouez avec votre application.

Je vous remercie.

Lire Cordova ios construire en ligne: <https://riptutorial.com/fr/cordova/topic/10666/cordova-ios-construire>

---

# Chapitre 8: Créer votre première application avec Cordova

## Remarques

En cas d'exécution de `cordova run android` échoue. Assurez-vous que votre appareil Android est connecté à votre ordinateur et exécutez les `adb devices` pour vous assurer que les outils de développement Android (ADT) peuvent détecter votre appareil.

## Exemples

### Utilisation de l'outil de ligne de commande

Vous créez d'abord un nouveau projet Cordova:

```
cordova create HelloWorld my.application.identifiant AppName
```

Cela va créer un projet Cordova vierge

- dans le dossier *HelloWorld*
- avec l'identifiant *my.application.identifiant* (qui devrait être unique pour chaque application)
- avec le nom *AppName*

Ensuite, vous ajoutez les plates-formes souhaitées:

```
cordova platform add android
// and/or
cordova platform add browser
// and/or
cordova platform add ios // On macOS only
// etc...
```

Construisez votre application pour générer un fichier exécutable:

```
cordova build // Build project for all platforms
cordova build ios // Build project only for iOS platform
cordova build android // Build project only for Android platform
```

Une fois construit, vous pouvez exécuter l'application sur l'une des plates-formes que vous avez ajoutées:

```
cordova run android --emulator // Run Android app in emulator
cordova run android --device // Run Android app on physical connected device
cordova run browser // Will run the app in the browser
```

Si vous souhaitez créer l'application pour Eclipse, Xcode, Visual Studio, etc.:

```
cordova prepare [platform_name] // Prepare copies of www folder and any plugins into the appropriate platform folder
```

Lire Créer votre première application avec Cordova en ligne:

<https://riptutorial.com/fr/cordova/topic/2396/creer-votre-premiere-application-avec-cordova>

---

# Chapitre 9: Déboguer l'application

## Remarques

Important: lorsque vous déboguez des applications Cordova, si un événement et un code `OnDeviceReady` s'exécutent, au moment du lancement de l'application, votre débogueur ne sera toujours pas connecté (contrairement au débogage Visual Studio C # où l'application attend le processus de débogage). attacher avant de continuer le lancement du programme).

Cela signifie que les messages de console ou les points d'arrêt initiaux ne seront pas capturés.

La solution à ce problème peut être une configuration différée ou un enregistrement différé de la console avec `setTimeout` lorsque l'événement `DeviceReady` est déclenché.

## Exemples

### Déboguer sur un appareil Android via USB

Une application Cordova s'exécute en tant que site Web sur un composant `WebView` de la plateforme mobile native. Le débogage d'une application Cordova peut donc être fait en utilisant les outils de développement de votre navigateur favori. Les étapes suivantes sont nécessaires pour connecter l'application, exécutée sur le périphérique, au navigateur Chrome sur un ordinateur de développement:

1. Activer le débogage USB sur votre appareil ( [vous pouvez suivre ce guide](#) )
2. Installez `adb` Android Debug Bridge (non requis sur les versions récentes de Chrome) ( [guide pour OSX](#) )
3. Connectez votre téléphone et exécutez des `adb devices` sur votre terminal (non requis sur les versions récentes de Chrome), puis sélectionnez `ok` dans la fenêtre contextuelle de votre téléphone. `Allow USB debugging?` .
4. Ouvrir Chrome
5. Accédez à `chrome://inspect` ou choisissez [Autres outils => Inspecter les périphériques ...](#)
6. Sélectionnez votre appareil et déboguez en utilisant les outils de développement de Chrome

Selon votre appareil, vous devrez peut-être d'abord télécharger les pilotes USB.

Vous devez également activer les «sources inconnues» sous Sécurité dans les paramètres si vous souhaitez charger l'application sur votre téléphone.

### Déboguer les applications Cordova utilisant GapDebug

<https://www.genuitec.com/products/gapdebug/>

GapDebug est un outil de débogage mobile complet qui comble le vide laissé par les autres options de débogage. Fonctionnant sur les plates-formes Windows et Mac, GapDebug permet le débogage des applications mobiles hybrides, telles que PhoneGap et Cordova, sur les appareils

iOS et Android modernes. Et, GapDebug est toujours gratuit pour le débogage local.

Étape pour la première configuration sont donnés dans ce lien suivant:

<https://www.genuitec.com/products/gapdebug/learning-center/configuration/>

## Déboguer sur un appareil iOS avec USB

### 1. Désactiver la navigation privée

Ouvrez les paramètres Safari de votre appareil et assurez-vous que **la navigation privée est désactivée** . Le débogage à distance ne fonctionnera pas si *la navigation privée* est activée.

### 2. Activer l'inspecteur Web

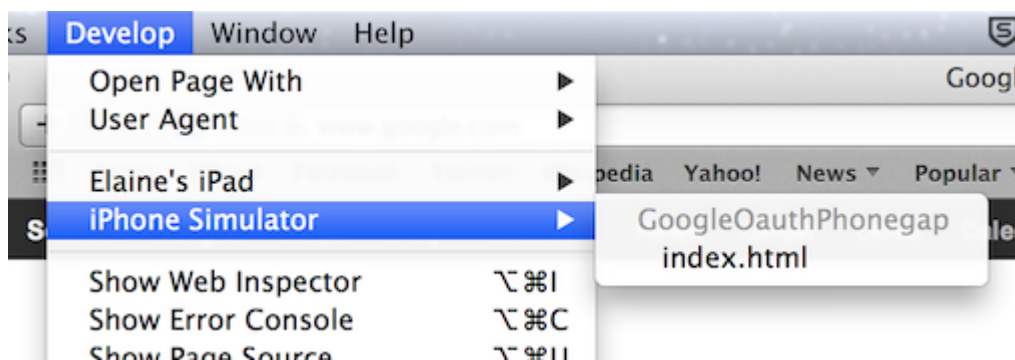
Appuyez sur l'onglet *Avancé des paramètres* Safari de votre appareil et assurez-vous que **Web Inspector est activé** .

### 3. Activer le menu de développement de Safari

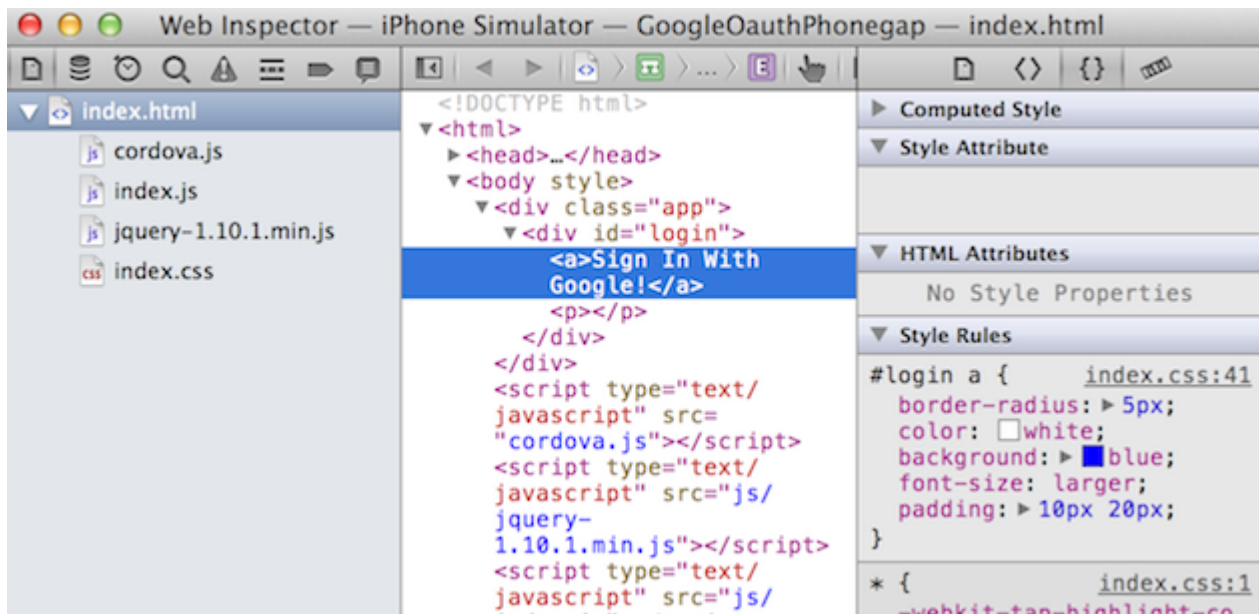
Sur votre bureau ou ordinateur portable, ouvrez les préférences de Safari et cliquez sur l'onglet Avancé. Cochez la case **Afficher le menu Développer dans la barre de menus** .

### 4. Démarrer l'inspecteur Web

Lancez votre application sur le simulateur iOS ou sur un périphérique physique. Si vous utilisez un périphérique physique, vous devez le connecter à votre ordinateur de bureau ou à votre ordinateur portable avec le câble USB standard. Une fois l'application lancée, basculez vers Safari, sélectionnez l'élément de menu **Développer** , puis recherchez l'entrée correspondant à la page Web à déboguer.



Vous pouvez maintenant utiliser l'inspecteur Web comme vous le feriez pour déboguer une page Web.



Lire Débuguer l'application en ligne: <https://riptutorial.com/fr/cordova/topic/4004/debuguer-l-application>

---

# Chapitre 10: Firebase Push Notification Cordova

## Exemples

### Firebase Push Notification dans Cordova Android

#### Ajouter Firebase à votre projet Android

##### *Ajouter Firebase à votre application*

Pour ajouter Firebase à votre application, vous aurez besoin d'un projet Firebase et d'un fichier de configuration Firebase pour votre application.

1. Créez un projet Firebase dans la console Firebase, si vous n'en avez pas déjà un. Si un projet Google existant est déjà associé à votre application mobile, cliquez sur Importer un projet Google. Sinon, cliquez sur Créer un nouveau projet.
2. Cliquez sur Ajouter Firebase à votre application Android. Si vous importez un projet Google existant, cela peut se produire automatiquement et vous pouvez simplement télécharger le fichier de configuration.
3. Lorsque vous y êtes invité, entrez le nom du package de votre application. Il est important de saisir le nom du package utilisé par votre application. cela ne peut être réglé lorsque vous ajoutez une application à votre projet Firebase.
4. À la fin, vous allez télécharger un fichier google-services.json. Vous pouvez télécharger ce fichier à tout moment. Si vous ne l'avez pas déjà fait, copiez-le dans le dossier du module de votre projet, généralement app /.

## Cordova Firebase Push Notification Plugin

<https://www.npmjs.com/package/cordova-plugin-fcm>

Pour obtenir le jeton d'accès:

```
FCMPlugin.getToken(  
  function(token) {  
    alert(token);  
  },  
  function(err) {  
    console.log('error retrieving token: ' + err);  
  }  
);
```

Rappel pour recevoir une notification push:

```
FCMPlugin.onNotification(  
  function(data) {
```



```

    if(data.wasTapped) {
        //Notification was received on device tray and tapped by the user.
        alert( JSON.stringify(data) );
    }else{
        //Notification was received in foreground. Maybe the user needs to be notified.
        alert( JSON.stringify(data) );
    }
},
function(msg) {
    console.log('onNotification callback successfully registered: ' + msg);
},
function(err) {
    console.log('Error registering onNotification callback: ' + err);
}
);

```

Placez le jeton d'accès et le rappel pour recevoir la notification push dans le fichier index.js de la fonction receivedEvent

## Envoi d'une notification Push via l'API REST

```

//POST: https://fcm.googleapis.com/fcm/send
//HEADER: Content-Type: application/json
//HEADER: Authorization: key=AlzaSyAMMh0mdVIRXPcBejyatAtdZgmklepwoNs //key is server-key
{
  "notification":{
    "title":"Notification title", //Any value
    "body":"Notification body", //Any value
    "sound":"default", //If you want notification sound
    "click_action":"FCM_PLUGIN_ACTIVITY", //Must be present for Android
    "icon":"fcm_push_icon" //White icon Android resource
  },
  "data":{
    "param1":"value1", /Any data to be retrieved in the notification callback
    "param2":"value2"
  },
  "to":"eRImo7algBM:APA91bHSxS0dmgsOi9su_XytEtCbei0Zi00Dgm76VHvbqeb-
WPoZcLyNVpnaLWPLw7U1u93h00ZhtBxn_hVGxPAwXXfc-yNy6_kkfzUdTpcI2QPB0vzJBmOFzX3RRZ15wmFkCUFtyhc",
//Topic or single device
  "priority":"high", //If not set, notification won't be delivered on completely closed
iOS app
  "restricted_package_name":"com.zensar.fcm" //Optional. Set for application filtering
}

```

Configurez l'API REST ci-dessus à l'aide du client Postman Rest.

The screenshot shows the Postman REST client interface. At the top, the method is set to 'POST' and the URL is 'https://fcm.googleapis.com/fcm/send'. Below the URL bar, there are tabs for 'Authorization', 'Headers (2)', 'Body', 'Pre-request Script', and 'Tests'. The 'Headers (2)' tab is selected and highlighted in orange. Underneath, two headers are listed:

- Content-Type:** application/json
- Authorization:** key=AlzaSyAMMh0mdVIRXPcBejyatAtdZgmklepwoNs

Authorization Headers (2) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary Text

```
1 {
2   "notification":{
3     "title":"Notification title",
4     "body":"Notification body",
5     "sound":"default",
6     "click_action":"FCM_PLUGIN_ACTIVITY",
7     "icon":"fcm_push_icon"
8   },
9   "data":{
10    "param1":"value1",
11    "param2":"value2"
12  },
13  "to":"eRImo7alg8M:APA91bHSxS0dmgs0i9su_XytEtCbei0Zi00Dgm76VHvbqeb-WPoZcLyNVpnaLWPLw7U1u93h00ZhtBxn_h
14    -yNy6_kkfzUdTpcI2QP80vzJBmOFzX3RRZ15wmFkCUFtyhc",
15  "priority":"high",
16  "restricted_package_name":"com.zensar.fcm"
17 }
```

**Comment cela fonctionne** *Envoyez une notification push à un périphérique ou à un sujet unique.*

1.a L'application est au premier plan: l'utilisateur reçoit le message de notification dans sa barre de notification. L'utilisateur appuie sur la notification et l'application est ouverte. L'utilisateur reçoit les données de notification dans le rappel JavaScript '.

1.b L'application est en arrière-plan: l'utilisateur reçoit le message de notification dans sa barre de notification. L'utilisateur appuie sur la notification et l'application est ouverte. L'utilisateur reçoit les données de notification dans le rappel JavaScript '.

Lire [Firebase Push Notification Cordova en ligne:](https://riptutorial.com/fr/cordova/topic/6892/firebase-push-notification-cordova)

<https://riptutorial.com/fr/cordova/topic/6892/firebase-push-notification-cordova>

# Chapitre 11: Google Analytics à Cordova

## Exemples

### Google Analytics en Cordova sans aucun plugin.

Insérer la fonction d'analyse dans index.js

```
function analytics(){

    (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
    (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
    m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
    })(window,document,'script','lib/analytics.js','ga');

    if(window.localStorage) {

        ga('create', 'UA-XXXXXXX-1', {
            'storage': 'none'
            , 'clientId': window.localStorage.getItem('ga_clientId') /*The tracker id
obtained from local storage*/
        });
        ga(function(tracker) {
            window.localStorage.setItem('ga_clientId', tracker.get('clientId'));
            /*The tracker id for each device is different and stored in local storage*/
        });
    }
    else {

        ga('create', 'UA-XXXXXXX-1', 'auto');
    }

}
```

Insérez chacun des tags de script ci-dessous dans chaque page HTML et modifiez le nom de la page

```
<script>
    (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
    (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
    m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
    })(window,document,'script','lib/analytics.js','ga');

    ga('set','checkProtocolTask',null);
    /*checkProtocal Task is set to null so that GA allows tracking other than http/https
*/

    ga('set', 'page', "Page Name");
    /*Page Name is name of each html page*/

    ga('send', 'pageview');
</script>
```

Lire Google Analytics à Cordova en ligne: <https://riptutorial.com/fr/cordova/topic/6909/google->



# Chapitre 12: Notification Push dans Android et iOS

## Exemples

### Utilisation du nouveau plug-in-phonegap

Pour envoyer des notifications push aux applications cordova. La première étape consiste à obtenir un jeton de périphérique. Un "jeton de périphérique" est spécifique à chaque périphérique et à chaque projet.

#### Pré-requis :

1. Numéro de projet Google Cloud Messaging

Pour cela, accédez à [Google Developer Console](#) et créez un nouveau projet. Sous Informations sur le projet, le numéro du projet

2. Clé de l'API Google Cloud Messaging pour le projet ci-dessus (nécessaire pour le serveur)

Allez dans Bibliothèque -> Google Cloud Messaging -> Activer. Accédez aux informations d'identification pour créer une clé API de type serveur.

### Credentials

## Add credentials to your project

- 1 Find out what kind of credentials you need

We'll help you set up the correct credentials.

If you wish to you can skip this step and create an [API key](#), [client ID](#) or [service account](#).

#### Which API are you using?

Determines what kind of credentials you need.

Google Cloud Messaging

#### Where will you be calling the API from?

Determines which settings you'll need to configure.

Web server (e.g. node.js, Tomcat)

What credentials do I need?

Ajout du push-plugin au projet:

```
cordova plugin add https://github.com/phonegap/phonegap-plugin-push --variable
```

```
SENDER_ID="XXXXXXX"
```

**SENDER\_ID** représente l'ID du projet

Placez le code suivant dans la fonction `receivedEvent` dans `index.js`

```
var push = PushNotification.init({
  android: {
    senderID: "XXXXXX"
  },
  ios: {
    alert: "true",
    badge: "true",
    sound: "true"
  },
  windows: {}
});

push.on('registration', function(data) {
  console.log("device token: " + data.registrationId);
});

push.on('notification', function(data) {
  console.log(data.message);
  console.log(data.title);
  console.log(data.count);
  console.log(data.sound);
  console.log(data.image);
  console.log(data.additionalData);
});

push.on('error', function(e) {
  console.log(e.message)
});
```

En exécutant le code ci-dessus à partir d'un appareil Android ou iOS, vous obtenez un jeton de périphérique.

**REMARQUE:** Le jeton de périphérique doit être généré uniquement sur un périphérique réel et non sur un périphérique virtuel.

Pour tester la notification push, accédez à ce lien [Test de notification en ligne](#)

**Pour Android:** entrez le jeton de périphérique, le message et la clé API

# GCM

**Device Token**

**Message**

**Api Key**

Submit

Lire Notification Push dans Android et iOS en ligne:

<https://riptutorial.com/fr/cordova/topic/6181/notification-push-dans-android-et-ios>

---

# Chapitre 13: Plugins Cordova: comment installer, comment ils fonctionnent, exemples

## Exemples

### Que sont les plugins Cordova?

Les plugins Cordova sont en termes simples une couche au-dessus de la plate-forme native respective.

Les plugins fournissent une interface pour accéder à la plate-forme native.

### Comment les plugins Cordova peuvent-ils être utiles?

Les plugins Cordova fournissent une interface commune pour interagir avec le code natif.

Chaque plug-in possède un fichier JavaScript intermédiaire qui donne accès à des fonctionnalités spécifiques à la plate-forme.

### Installation de Cordova Plugin

```
cordova plugin add <plugin-name>
```

Exemple: le plug-in cordova ajoute un appareil-plug-in Cordova

Le plugin doit être installé dans le répertoire racine du projet.

Remarque:

Avant d'ajouter le plug-in, remplacez le contenu du dossier www spécifique à la plate-forme par le dossier www externe du répertoire racine. En effet, lors de l'ajout d'un plugin, le contenu du dossier www externe est remplacé par le dossier www spécifique à la plateforme.

Remarque:

Lorsque vous ajoutez une nouvelle plate-forme et que vous avez des plug-ins installés sur votre projet, il n'est pas nécessaire d'installer les plug-ins existants. Cordova ajoute automatiquement les plug-ins installés pour la nouvelle plate-forme.

### Plugins les plus populaires

1. [cordova-plugin-battery-status](#) - utilisé pour surveiller l'état de la batterie de l'appareil.
2. [Cordova-plugin-camera](#) - fournit une API pour prendre des photos et pour choisir des



images de la bibliothèque d'images du système.

3. [cordova-plugin-contacts](#) - donne accès à la base de données des contacts de l'appareil.
4. [cordova-plugin-device](#) - décrit le matériel et le logiciel de l'appareil.
5. [cordova-plugin-device-motion](#) - accès à l'accéléromètre de l'appareil.
6. [cordova-plugin-file](#) - implémente une API de fichier permettant un accès en lecture / écriture aux fichiers résidant sur le périphérique.
7. [cordova-plugin-geolocation](#) - fournit des informations sur l'emplacement de l'appareil, telles que la latitude et la longitude.
8. [cordova-plugin-mondialisation](#) - obtient des informations et effectue des opérations spécifiques aux paramètres régionaux, à la langue et au fuseau horaire de l'utilisateur.
9. [cordova-plugin-inappbrowser](#) - affiche des articles utiles, des vidéos et des ressources Web dans votre application. Les utilisateurs peuvent afficher des pages Web sans quitter votre application.
10. [cordova-plugin-network-information](#) - fournit des informations sur la connexion cellulaire et wifi de l'appareil et indique si l'appareil dispose d'une connexion Internet.
11. [cordova-plugin-vibration](#) - fournit un moyen de faire vibrer l'appareil.
12. [Cordova-plugin-statusbar](#) - fournit des fonctions pour personnaliser iOS et Android StatusBar.
13. [cordova-plugin-whitelist](#) - implémente une stratégie de liste blanche pour naviguer dans la vue Web de l'application sur Cordova 4.0. **Plugin recommandé!**

Pour les plugins spécifiques à cordova, suivez le lien ci-dessous

<https://cordova.apache.org/plugins/>

Lire Plugins Cordova: comment installer, comment ils fonctionnent, exemples en ligne:

<https://riptutorial.com/fr/cordova/topic/7077/plugins-cordova--comment-installer--comment-ils-fonctionnent--exemples>

# Chapitre 14: Rendre l'application libérée de Cordova CLI

## Exemples

### Android

**Étape 1:** Accédez au répertoire racine du projet et ouvrez l'invite de la ligne de commande

```
cordova build --release android
```

Cela génère un apk non signé sous `\plates-formes \ android \ build \ output \ apk` avec le nom **android-release-unsigned.apk**

**Étape 2:** Génération de clés pour obtenir un fichier apk signé

Syntaxe:

```
keytool -genkey -v -keystore <keystoreName>.keystore -alias <Keystore AliasName> -keyalg <Key algorithm> -keysize <Key size> -validity <Key Validity in Days>
```

Exemple:

```
keytool -genkey -v -keystore ExampleApp.keystore -alias TestExampleApp -keyalg RSA -keysize 2048 -validity 10000
```

```
keystore password? : xxxxxxxx
What is your first and last name? : xxxxxxx
What is the name of your organizational unit? : xxxxxxxx
What is the name of your organization? : xxxxxxxx
What is the name of your City or Locality? : xxxxxxx
What is the name of your State or Province? : xxxxxx
What is the two-letter country code for this unit? : xxx
```

Le fichier de clés est généré dans le même dossier sous le nom `ExampleApp.keystore`

**Étape 3:** Déplacer le fichier de clés généré vers `\plates-formes \ android \ build \ outputs \ apk`

Exécutez l'outil jarsigner dans l'invite de commande sous `\plates-formes \ android \ build \ output \ apk`

Syntaxe:

```
jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore <keystorename> <Unsigned APK file> <Keystore Alias name>
```

Exemple:

```
jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore ExampleApp.keystore android-release-unsigned.apk TestExampleApp
```

Cela génère l'apk signé avec le même nom.

#### **Étape 4:** outil d'alignement zip pour optimiser l'APK

```
zipalign -v 4 android-release-unsigned.apk android.apk
```

Le zipalign est situé sous \ Android \ sdk \ build-tools \ 23.0.3 \ zipalign

Cela génère un apk signé avec le nom android.apk qui peut maintenant être téléchargé sur app store

## iOS

**Étape 1:** Créez un fichier build.json dans le répertoire racine du projet.

Exemple de build.json

```
{
  "ios": {
    "debug": {
      "codeSignIdentity": "iPhone Developer",
      "provisioningProfile": "your-developer-provisioning-profile-UUID-here"
    },
    "release": {
      "codeSignIdentity": "iPhone Distribution",
      "provisioningProfile": "your-distribution-provisioning-profile-UUID-here"
    }
  }
}
```

Remarque: L'UUID peut être obtenu en ouvrant le fichier .mobileprovision dans un éditeur de texte et en recherchant «UUID».

**Étape 2:** Exécutez la commande suivante à partir du dossier racine du projet sur le terminal

```
cordova build ios --device --release
```

Lire Rendre l'application libérée de Cordova CLI en ligne:

<https://riptutorial.com/fr/cordova/topic/7092/rendre-l-application-liberee-de-cordova-cli>

# Chapitre 15: Signez Android avec Cordova 5

## Exemples

### Ajoutez la configuration de génération pour signer le fichier .apk

1. Ajoutez un magasin de clés à l'aide de:

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

Note: Ceci devrait être à la racine du projet. Bien que ce ne soit pas une exigence stricte, cela facilite le référencement des fichiers.

2. Ajoutez un build.json avec la configuration release / dev pour keystore, à la racine du projet:

```
{
  "android": {
    "debug": {
      "keystore": "..\android.keystore",
      "storePassword": "android",
      "alias": "mykey1",
      "password": "password",
      "keystoreType": ""
    },
    "release": {
      "keystore": "..\android.keystore",
      "storePassword": "",
      "alias": "mykey2",
      "password": "password",
      "keystoreType": ""
    }
  }
}
```

3. Ajoutez le commutateur --buildConfig à la commande de compilation Cordova / Ionic:

```
cordova build android --release --buildConfig=build.json
```

ou avec ionic comme

```
ionic build android --release --buildConfig=build.json
```

Le fichier signé sera généré sous la nouvelle structure de dossier à

```
/platforms/android/build/outputs/apk/android-release.apk
```

Lire Signez Android avec Cordova 5 en ligne: <https://riptutorial.com/fr/cordova/topic/3061/signez-android-avec-cordova-5>

---

# Chapitre 16: Visual Studio Tools pour Apache Cordova

## Exemples

### Obtenir les outils Apache Cordova dans Visual Studio

1. Ouvrez le Panneau de configuration -> Programmes et fonctionnalités, choisissez l'élément Visual Studio 2015, puis choisissez le bouton Modifier.
2. Dans l'assistant d'installation de Visual Studio, choisissez le bouton Modifier.
3. Dans la liste des fonctionnalités facultatives à installer, cochez la case HTML / JavaScript (Apache Cordova), choisissez le bouton Suivant, puis choisissez le bouton Mettre à jour.

### Mettre à jour les outils Apache Cordova dans Visual Studio

1. Dans Visual Studio, choisissez Outils-> Extensions et mises à jour.
2. Dans l'onglet Mises à jour de la boîte de dialogue Extensions et mises à jour, choisissez Mises à jour du produit. Si une mise à jour pour Visual Studio Tools pour Apache apparaît, sélectionnez-la, puis choisissez le bouton Mettre à jour.

Lire Visual Studio Tools pour Apache Cordova en ligne:

<https://riptutorial.com/fr/cordova/topic/2393/visual-studio-tools-pour-apache-cordova>

# Crédits

S. No	Chapitres	Contributeurs
1	Commencer avec Cordova	<a href="#">Akash Pal</a> , <a href="#">Community</a> , <a href="#">Hitesh Riziya</a> , <a href="#">Michiel</a> , <a href="#">Mikhail</a> , <a href="#">ModusPwnens</a>
2	Comment détecter l'état de la connexion réseau	<a href="#">Philip Bijker</a>
3	Comment installer / désinstaller un plugin Cordova personnalisé	<a href="#">Rahul Raghuvanshi</a>
4	Comment personnaliser le dossier www spécifique à la plateforme dans cordova	<a href="#">Rahul Raghuvanshi</a>
5	Cordova Crop Image Plugin	<a href="#">Akash Pal</a>
6	Cordova ios construire	<a href="#">sparrowTrajon</a>
7	Créer votre première application avec Cordova	<a href="#">Devid Farinelli</a> , <a href="#">grgarside</a> , <a href="#">Hristo Eftimov</a> , <a href="#">James Wong</a> , <a href="#">Philip Bijker</a> , <a href="#">ProllyGeek</a>
8	Déboguer l'application	<a href="#">Akash Pal</a> , <a href="#">Alex Filatov</a> , <a href="#">Alexus</a> , <a href="#">Devid Farinelli</a> , <a href="#">mike nelson</a> , <a href="#">Philip Bijker</a>
9	Firestore Push Notification Cordova	<a href="#">Akash Pal</a>
10	Google Analytics à Cordova	<a href="#">Akash Pal</a>
11	Notification Push dans Android et iOS	<a href="#">Akash Pal</a>
12	Plugins Cordova: comment installer,	<a href="#">Akash Pal</a> , <a href="#">Hristo Eftimov</a>

	comment ils fonctionnent, exemples	
13	Rendre l'application libérée de Cordova CLI	<a href="#">Akash Pal</a>
14	Signez Android avec Cordova 5	<a href="#">Aditya Singh</a>
15	Visual Studio Tools pour Apache Cordova	<a href="#">Charitha Goonewardena</a>