

# Structure d'un projet Apache Cordova

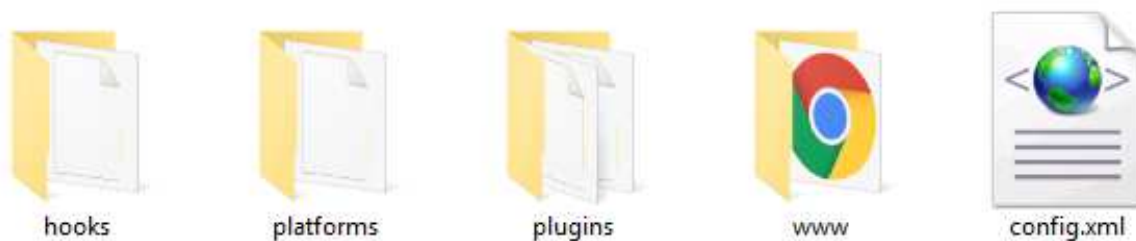
## Introduction

Apache Cordova permet de créer des applications mobiles hybrides. Ces applications sont développées en utilisant les langages du Web et compilées pour un ou plusieurs environnements mobiles.

La création d'un projet Apache Cordova via la commande « cordova create » entraîne la création d'un répertoire qui porte le nom du projet. Celui-ci contient toute une arborescence et un ensemble de fichiers qui sont utilisés par Apache Cordova pour créer l'application mobile hybride.

## Structure d'un projet Cordova

La création d'un nouveau projet Apache Cordova va donner naissance à une arborescence dont la racine est le répertoire qui porte le nom du projet. Celui-ci à son tour contient initialement quatre sous-répertoires : `www`, `plugins`, `platforms` et `hooks`. Egalement, dans ce répertoire racine on va trouver un fichier intitulé « `config.xml` » qui permet de configurer l'application mobile.



## Répertoire `www`

Le développement de l'application mobile se fait essentiellement au sein du répertoire « `www` ». En effet, celui-ci est destiné à accueillir le code source de l'application. Il contient par défaut trois sous-répertoires : `css`, `img` et `js`. En plus, il contient le fichier « `index.html` » qui est par défaut le point d'entrée de l'application. En effet, c'est lui qui sera exécuté par le Webview lorsque l'application est lancée.



Le répertoire « js » est destiné à accueillir tous les fichiers JavaScript relatifs à l'application mobile. Par défaut, il contient le fichier « index.js » qui est invoqué dans le fichier « index.html ».

Le répertoire « img » est destiné à accueillir toutes les images utilisées dans l'application mobile. Par défaut, il contient le fichier « logo.png » qui contient le logo d'apache cordové utilisé dans l'application créée par défaut lors de la création d'un nouveau projet.

Le répertoire « css » est destiné à accueillir tous les fichiers de style css. Il contient par défaut le fichier « index.css » utilisé dans le fichier « index.html » de l'application générée par défaut. Toutes les feuilles de styles de l'application mobile hybrides doivent résider dans ce répertoire.

Respecter cette arborescence permet d'avoir un code source organisé et facile à maintenir. L'emplacement des fichiers est ainsi standardisé ainsi que les noms des fichiers principaux (index.html, index.js et index.css).

D'autres répertoires et fichiers peuvent être créés et ajoutés par la suite par le développeur. C'est le cas du répertoire « res » par exemple qui va contenir les icônes et les splash screens. C'est également le cas des fichiers de certains Frameworks qui peuvent être utilisés dans le projet à l'instar des fichiers du framework « JQuery Mobile ».

## Répertoire plugins

Pour pouvoir profiter pleinement des fonctionnalités natives d'un dispositif mobile il faut utiliser des plugins. Selon l'application mobile hybride il se peut que le développeur ait besoin d'un plugin pour utiliser une fonctionnalité supportée par celui-ci. A ce moment il doit utiliser la commande « cordova plugin add » suivie par le nom du plugin. L'exécution de cette commande va permettre l'ajout du plugin au projet en cours dans le répertoire réservé aux plugins intitulé « plugins ».



Le développeur ne doit pas toucher au contenu du répertoire plugin. En effet, la gestion des plugins (tels que l'ajout ou la suppression) doit se faire à travers les commandes Cordova.

## Répertoire platforms

Une application mobile hybride a la particularité de pouvoir être compilée pour plusieurs plateformes mobiles à partir d'un même code source. Toutefois, pour pouvoir compiler l'application pour un environnement mobile particulier il faut tout d'abord installer son SDK. Puis, il faut ajouter le ou les plateformes ciblées via la commande « cordova platform add ». Celle-ci va permettre d'ajouter les fichiers nécessaires pour pouvoir compiler l'application pour l'environnement souhaité. Ces fichiers résident dans un répertoire qui porte le nom de l'environnement sous le répertoire « platforms ».



Ce répertoire peut être utile au développeur dans plusieurs cas. Les deux cas les plus fréquents sont la récupération de l'exécutable et la récupération du code source. Pour le premier cas, si le développeur souhaite récupérer l'exécutable (fichier d'extension APK pour le cas de la plateforme Android) pour le tester, pour le partager ou pour le publier. Le deuxième cas c'est lorsqu'il n'est pas possible de compiler le code source sur la même machine. C'est le cas si le développement s'effectue sur un PC et que la plateforme cible est « iOS ». Dans ce cas, il faut récupérer le code source et le compiler sur un ordinateur de la marque Apple avec le système d'exploitation « Mac OS X » avec l'environnement de développement « Xcode ».

## Répertoire hooks

Ce répertoire est destiné à accueillir des scripts appelés « hooks ». Il s'agit de script qui sont développés par les développeurs des applications mobiles ou des développeurs de plugins pour personnaliser les commandes Cordova. Par exemple, vous pouvez définir un « hook » qui permet de vérifier systématiquement le formatage du code dans vos fichiers JavaScript avant chaque appel à la commande « cordova build » pour générer l'application. Pour automatiser l'exécution de ce script il suffit d'indiquer son type (Exemples : before\_build, after\_build, before\_run et after\_run).

Cependant, l'usage du répertoire « hooks » pour créer ce type de script est devenu obsolète du moment qu'il est possible de faire la même chose dans les fichiers « conig.xml » et « plugin.xml ». Il existe encore pour des raisons de compatibilité avec des anciens plugins qui utilisent ce répertoire.