

Applications mobiles avec **Cordova** et **PhoneGap**

Sébastien Pittion
Bastien Siebman



Applications mobiles avec Cordova et PhoneGap



Un seul développement pour un déploiement multiple

Maintenus par Adobe et la fondation Apache, Cordova et sa distribution PhoneGap sont des outils open source qui permettent de créer facilement et simultanément des applications mobiles pour plusieurs plates-formes du marché, comme iOS, Android, Windows Phone, et bien d'autres encore. En effet, il suffit aujourd'hui d'un seul développement pour assurer un déploiement multiple vers les différents magasins d'applications (App Store, Google Play Store, etc.). Utilisant les langages HTML, CSS et JavaScript, ces outils à la popularité croissante offrent une excellente alternative au code natif, notamment pour les développeurs web.

Concret et accessible, cet ouvrage est un recueil de bonnes pratiques et d'astuces pour réussir la création d'applications mobiles avec Cordova et PhoneGap, jusqu'à leur soumission sur les principaux magasins en ligne. Il comporte en outre une étude de cas complète, qui détaille la conception d'une application multi-plate-forme. Tout le code source des exemples est disponible en ligne sur <https://github.com/siebmanb/rappelle-toi>.

Au sommaire

PARTIE I : PREMIERS PAS AVEC CORDOVA ET PHONEGAP. Cordova ou PhoneGap ? Un peu d'histoire • Cordova et PhoneGap en bref • **Prérequis et installation.** Phase « terminal » • Node.js • Kits de développement • Commande Line Interfaces • **PARTIE II : DÉVELOPPEMENT D'UNE APPLICATION.** Création d'un projet. Réfléchir avant de se lancer • Ne pas réinventer la roue • Architecture et structure • Versionner ses sources • **Conception et architecture d'une application.** Étapes de conception • Architecture de l'application • Bonnes pratiques et astuces • **Ajout de fonctionnalités spécifiques.** Installation et fonctionnement des plug-ins • Utilisation du répertoire merges • Utilisation des hooks • **Debug du code.** Dans un navigateur • Dans un simulateur • Sur un appareil • Accéder à un serveur local • Répercuter les changements sans compiler • **PARTIE III : DIFFUSION D'UNE APPLICATION. Les magasins d'applications.** Publication sur l'App Store • Publication sur le Google Play Store • **PhoneGap Build.** Configuration via le fichier config.xml • Configuration via ConfiGAP • Utilisation.

S. Pittion

Autodidacte évoluant dans le développement web depuis 15 ans, **Sébastien Pittion** est actuellement ingénieur développeur front-end chez Viseo. Suivant le projet Cordova/PhoneGap depuis le début, il a conçu et développé diverses applications mobiles avec ces outils, notamment dans le domaine de la télémedecine. Il est par ailleurs l'administrateur du compte Twitter @PhoneGapFR.
@fingerproof

B. Siebman

Ingénieur et entrepreneur, **Bastien Siebman** est le fondateur de l'agence ButterflyEffect, spécialisée dans le Web et le mobile solidaires, qui a récemment rejoint le groupe de communication CitizenRepublic. Il est également le cocréateur de l'outil Templana.com, bibliothèque de templates pour Asana.
@siebmanb

À qui s'adresse ce livre ?

- À tous les acteurs d'un projet d'application mobile : décideurs, chefs de projet, développeurs, webdesigners...
- À ceux qui souhaitent concevoir, publier et vendre une application mobile pour iOS, Android, etc.



Sur le site <https://github.com/siebmanb/rappelle-toi>

– Téléchargez le code source de tous les exemples du livre

Applications mobiles
avec Cordova
et PhoneGap

CHEZ LE MÊME ÉDITEUR

F. NEUMAN. – **Le guide pratique iPhone et iOS8.**

N°14112, 2014, 160 pages.

Y. GARRETT. – **Le guide pratique des tablettes Android.**

N°14070, 2014, 160 pages.

J.-M. LACOSTE et T. SARLANDIE. – **Programmation iOS 6 pour iPhone et iPad.**

N°13639, 2013, 352 pages.

J. STARK. – **Applications iPhone avec HTML, CSS et JavaScript.**

N°12745, 2010, 190 pages.

F. DAoust, D. HAZAËL-MASSIEUX. – **Relever le défi du Web mobile. Bonnes pratiques de conception et de développement.**

N°12828, 2011, 300 pages.

J. CHABLE, D. GUIGNARD, E. ROBLES, N. SOREL. – **Programmation Android, 2^e édition.**

N°13303, 2012, 520 pages.

E. SARRION. – **jQuery mobile. La bibliothèque JavaScript pour le Web mobile.**

N°13388, 2012, 602 pages.

E. MARCOTTE. – **Responsive Web Design.**

N°13331, 2011, 160 pages.

L. WROBLEWSKI. – **Mobile first.**

N°13406, 2012, 144 pages.

K. McGRANE. – **Stratégie de contenu mobile.**

N°13675, 2013, 172 pages.

I. CANIVET-BOURGAUX. – **Référencement mobile.**

N°13667, 2013, 456 pages.

T. BAILLET. – **Créer son premier thème WordPress pour mobile.**

N°13441, 2012, 128 pages.

Applications mobiles avec **Cordova** et **PhoneGap**

Sébastien Pittion
Bastien Siebman

EYROLLES



ÉDITIONS EYROLLES
61, bd Saint-Germain
75240 Paris Cedex 05
www.editions-eyrolles.com

En application de la loi du 11 mars 1957, il est interdit de reproduire intégralement ou partiellement le présent ouvrage, sur quelque support que ce soit, sans l'autorisation de l'Éditeur ou du Centre Français d'exploitation du droit de copie, 20, rue des Grands Augustins, 75006 Paris.
© Groupe Eyrolles, 2015, ISBN : 978-2-212-14052-1

Préface

Depuis l'explosion du marché des smartphones et tablettes en 2007, les développeurs d'applications n'ont eu cesse de chercher des solutions techniques leur permettant de mutualiser leurs efforts de développement pour pouvoir adresser un maximum d'appareils du marché, tout en limitant la complexité inhérente au multi-plates-formes (*cross-platform*). Grâce à la naissance du projet PhoneGap en 2009, puis au rachat de l'équipe technique par Adobe fin 2011, et enfin à la publication du cœur du projet sous licence libre avec pour nom de code Cordova, de nombreuses entreprises et indépendants ont misé sur cette technologie dite « hybride », en s'unissant pour faire de Cordova une plate-forme solide et pérenne pour leur stratégie mobile multi-plates-formes.

En choisissant les standards du Web pour créer vos applications et en vous basant sur Cordova pour les intégrer dans les écosystèmes natifs des différentes plates-formes, vous allez pouvoir former des équipes polyvalentes et performantes, capables d'intervenir aussi bien sur le développement web desktop que mobile. Vous pourrez ainsi capitaliser sur vos projets en mutualisant les compétences et le code source, et serez prêt par ailleurs pour adresser les futurs appareils (télévisions connectées, montres, lunettes...).

Pour ce faire, cet ouvrage va vous permettre d'aborder tous les aspects du développement d'applications mobiles multi-plates-formes avec Cordova. Vous y découvrirez que l'outil Cordova lui-même est relativement simple, l'essentiel du travail étant de produire des applications JavaScript en mode SPA (*Single Page Applications*, applications 100 % JavaScript, souvent créées avec des frameworks de type Backbone, AngularJS, Ionic, ReactJS...) bien intégrées, utiles pour vos utilisateurs et agréables à prendre en main.

Je vous souhaite une bonne découverte de la solution Cordova, et espère découvrir très rapidement vos applications sur les différents App Stores ☺.

Julien Bouquillon
Compte Twitter : @revolunet
Développeur, contributeur Cordova et organisateur de meetups

Table des matières

AVANT-PROPOS	1
PREMIÈRE PARTIE	
Premiers pas avec Cordova et PhoneGap	3
CHAPITRE 1	
Cordova ou PhoneGap ?	5
Un peu d'histoire	5
HTML 5	5
L'App Store et consorts	6
HTML 5, le retour.	7
Cordova en bref	7
Pourquoi ?	7
Comment ?	8
Pour qui ?	8
En pratique	9
PhoneGap en bref	11
Pourquoi ?	11
Quels avantages ?	12
En pratique	12
CHAPITRE 2	
Prérequis et installation	15
Phase « terminal »	15
Shell.	15
Bash.	16
Émulateur de terminal.	17
Entrer des commandes	19

Aide et manuel d'utilisation	20
sudo	22
Node.js	23
Présentation	23
Installation de Node.js	24
Plusieurs installations en parallèle	25
Kits de développement	27
iOS	27
Android	30
Command Line Interfaces	37
Cordova CLI	38
PhoneGap CLI	39
DEUXIÈME PARTIE	
Développement d'une application	41
CHAPITRE 3	
Création d'un projet	43
Réfléchir avant de se lancer	43
Choisir les plates-formes cibles	43
Des étapes importantes avant le développement	44
Les plug-ins Cordova	46
Plug-in Cordova vs solution web	48
Ne pas réinventer la roue	50
Frameworks, bibliothèques et autres outils	50
Un choix important car structurant	52
Architecture et structure	53
Architecture d'un projet Cordova par défaut	53
Fichier de configuration	53
Application de démonstration	56
Utiliser un squelette personnalisé	57
Versionner ses sources	57
Pourquoi faire du versioning ?	57
Outils de versioning	58
Fonctionnement de Git	58
Git : commandes de base	59
Utiliser GitHub	60

CHAPITRE 4

Conception et architecture d'une application	63
Étapes de conception	63
Étape 1 : définir le concept	63
Étape 2 : choisir les plates-formes cibles	64
Étape 3 : créer les wireframes	64
Étape 4 : identifier les fonctionnalités	65
Étape 5 : identifier les plug-ins	66
Étape 6 : créer le repository	66
Étape 7 : créer le projet Cordova	69
Architecture de l'application	70
config.xml, le fichier de configuration	70
Les fichiers CSS	71
Les polices d'écriture	71
Le fichier index.html	71
Le fichier main.js	74
Le fichier app.js	74
Ajouter une géolocalisation	75
Ajouter une photo	75
Ajouter une vidéo	76
Bonnes pratiques et astuces	77
Un projet bien planifié est à moitié fait	77
Tester, tester, tester	77
Être prêt à reconstruire le projet à tout moment	78
Chercher à mutualiser et éviter les processus manuels	78

CHAPITRE 5

Ajout de fonctionnalités spécifiques	79
Installer les plug-ins	79
Depuis un repository Cordova	79
Depuis un repository GitHub	79
Depuis un dossier en local	80
Avec plugman	80
Manuellement	80
Fonctionnement des plug-ins	80
Device	81

SplashScreen	81
Camera	82
Geolocation	84
Dialogs	86
Network information	88
Battery Status	90
Status Bar	90
InAppBrowser	91
Utilisation du répertoire merges	94
Utilisation des hooks	95
Exemples de hooks	97
 CHAPITRE 6	
Debug du code	105
Dans un navigateur	105
Utiliser Chrome Dev tools	107
Émuler un appareil mobile	111
Dans un simulateur	113
Sur un appareil.	123
Accéder à un serveur local	124
Répercuter les changements sans recompiler	125
 TROISIÈME PARTIE	
Diffusion d'une application	127
 CHAPITRE 7	
Les magasins d'applications	129
Publication sur l'App Store	129
Gestion des certificats sur Member Center	130
Publication via iTunes Connect	133
Déploiement ad hoc	139
Publication sur le Google Play Store.	140
Création de l'APK	140
Création et mise en ligne de l'application	143

CHAPITRE 8

PhoneGap Build	149
Présentation	149
Configuration via le fichier config.xml	151
Paramètres généraux	151
Personnalisation avancée	154
Icônes et splashscreens	156
Schémas d'URL	158
Fonctionnalités	158
Plug-ins	159
Sécurité	161
Configuration via ConfiGAP	162
Paramètres généraux	162
Paramètres avancés	163
Icônes et splashscreens	164
Permissions et informations	165
Vue Plugins	166
Utiliser PhoneGap Build	167
Via l'interface web	168
Debugging et Hydration	176
Via l'API REST	178
Via le terminal	178
Support et distribution	180
 CONCLUSION	 181
 INDEX	 183

Avant-propos

Coder, c'est d'abord savoir parler un langage qui possède des règles propres, à l'instar de tous les langages. Qu'il s'agisse de mots, de ponctuation ou bien d'espacements, tous ces éléments syntaxiques s'avèrent incontournables pour tout développeur qui se respecte.

Car coder, c'est construire quelque chose, un programme destiné à une machine. Et pour ce faire, chaque développeur devra apprendre à maîtriser les outils dont il dispose, ainsi que, bien souvent, en découvrir de nouveaux. Compléter et dompter son vocabulaire sont donc deux des devoirs les plus importants d'un développeur. Après quoi, ce dernier pourra utiliser au mieux les outils mis à sa disposition. Il produira alors inmanquablement de magnifiques poèmes emplis de figures stylistiques.

S'il est déjà relativement difficile de maîtriser un langage en particulier, l'univers mobile tel qu'on le connaît aujourd'hui fait pourtant appel à bon nombre d'entre eux. La fragmentation du marché (constructeurs, modèles, systèmes d'exploitation et bien évidemment langages de programmation différents) en est la cause directe. D'autant plus qu'il est en perpétuel changement. Alors, comment trouver son chemin, notamment quand on n'est encore qu'un développeur novice ? Pas de panique, vous découvrirez grâce à cet ouvrage qu'il est tout à fait possible de créer des applications mobiles destinées aux principaux acteurs du marché (Apple et Google) sans forcément se disperser. Et ce, grâce à ce formidable outil qu'est Cordova/PhoneGap.

Les plates-formes mobiles telles qu'Android ou iOS sont en pleine expansion depuis plusieurs années déjà. À mesure de leur évolution, notamment en termes de performances, le spectre des applications pouvant être développées avec Cordova/PhoneGap s'est considérablement élargi. Là où le développement d'un « simple » Tetris relevait auparavant du défi, mettre au point un jeu en 3D est maintenant à portée de main, avec de nombreux outils pour vous y aider. Certes, la technologie Cordova/PhoneGap n'est pas réellement destinée à produire des graphismes performants en trois dimensions, mais elle offre tout de même déjà beaucoup de possibilités.

Que vous souhaitiez réaliser un utilitaire, une version mobile d'un service web existant, ou même un jeu en 2D, Cordova/PhoneGap vous donnera rapidement accès à un résultat plus que satisfaisant et vous permettra de mettre celui-ci à disposition du plus grand nombre. Car c'est là la force de cet outil : développer pour plusieurs plates-formes dont la liste ne cesse de s'allonger : iOS, Android, Windows Phone, Firefox OS...

Note

On parle dans ce cas de développement *cross-platform*, ce terme désignant le fait de ne maintenir qu'une seule base de code exécutable sur différents systèmes d'exploitation et machines. Vous verrez bien entendu dans cet ouvrage comment cela est possible.

Nous avons organisé ce livre en trois parties : histoire, présentation et installation de Cordova/PhoneGap, développement de votre première application, et mise à disposition auprès du grand public. Pour illustrer au mieux chaque étape de ce processus, nous avons conçu et développé, spécifiquement pour ce livre, une application volontairement simple, dont le code source, ouvert à tous, est hébergé sur GitHub (<https://github.com/siebmanb/rappelle-toi>). Vous pourrez ainsi vous y référer au cours de la lecture de cet ouvrage ou bien même participer à son amélioration !

Partie I

Premiers pas avec Cordova et PhoneGap

Un nouveau monde s'ouvre à vous ! En effet, vous avez décidé de vous lancer dans le développement d'applications mobiles et vous avez choisi Cordova/PhoneGap pour vous aider dans cette voie.

Cette première partie va vous permettre tout d'abord de vous familiariser avec ces deux outils, qui ne sont pas synonymes comme vous allez le voir. Nous brosserons ensuite le tableau d'un environnement de développement idéal, en vous accompagnant dans les diverses étapes d'installation et de configuration nécessaires au bon déroulement de votre projet. Que ce soit le terminal, Node JS, les kits de développement iOS ou Android, ou encore la fameuse CLI Cordova, vous aurez ainsi toutes les cartes en mains pour vous lancer dans le développement, ce qui fera l'objet de la deuxième partie de cet ouvrage.

1

Cordova ou PhoneGap ?

Si vous êtes un développeur web intéressé par la création d'applications pour appareils mobiles, tels les smartphones et tablettes, il y a de grandes chances pour que vous ayez déjà entendu parler de Cordova ou PhoneGap. Peut-être vous demandez-vous alors, comme bon nombre de développeurs web, quelle est la différence entre les deux ?

Celle-ci semble à première vue assez mince, voire floue, tant certaines ressources en ligne ont tendance à mélanger ces termes. Depuis l'arrivée de Cordova, une réelle confusion s'est installée, jusque dans l'esprit des habitués de PhoneGap. Alors qu'en est-il vraiment ? Doit-on utiliser Cordova ou PhoneGap ? En réalité, vous allez voir, c'est très simple...

Un peu d'histoire

HTML 5

Le 9 janvier 2007, Steve Jobs lève le voile sur l'iPhone, ce smartphone révolutionnaire que le monde entier connaît aujourd'hui. L'appareil inclut alors quelques programmes, dont Safari Mobile, navigateur Internet extrêmement performant et innovant. Il est toutefois impossible à l'époque d'installer de nouveaux programmes. La seule solution proposée par Apple est l'ajout de *WebApps*, autrement dit de sites Internet améliorés, via Safari Mobile.

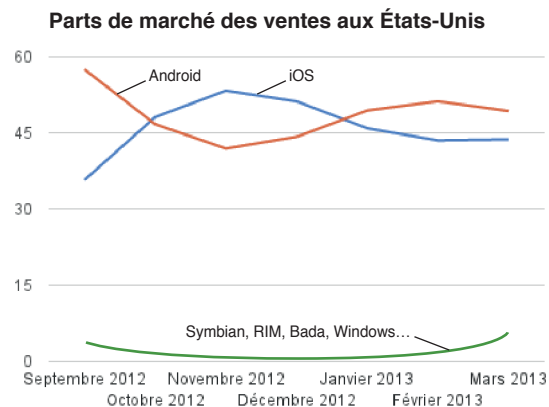
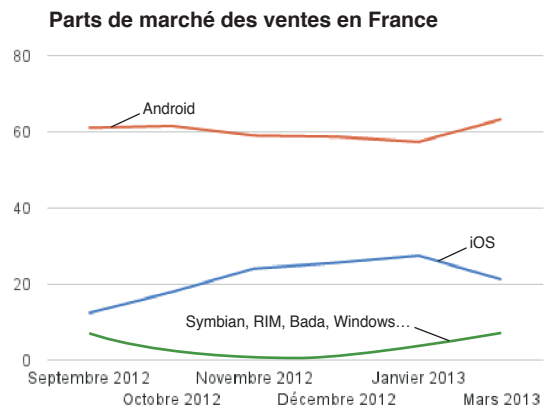
Steve Jobs lui-même vante alors les mérites d'HTML 5 (*Hypertext Markup Language version 5*). La nouvelle version du format de documents universellement reconnu sur la Toile. Avec l'HTML 5, bien que le standard ne soit pas encore totalement finalisé, les développeurs auront accès à de nombreux nouveaux outils prometteurs. L'ambition est belle : faire évoluer le Web depuis un modèle de documents vers de vrais programmes complexes.

L'App Store et consorts

Le monde n'est semble-t-il pas encore prêt pour les WebApps, Apple pivote alors et annonce l'App Store le 6 mars 2008. Enfin, l'iPhone s'ouvre : on ne parle désormais plus de programmes, mais d'applications. L'infrastructure est inédite, le succès immédiatement et indéniablement au rendez-vous. Les développeurs peuvent maintenant créer et distribuer/monétiser « facilement » de nouvelles fonctionnalités pour le smartphone *à la pomme*. Il leur faudra juste maîtriser un langage propriétaire : l'Objective-C – extension du C.

Bien entendu, la réussite rencontrée par l'App Store fit des émules et très vite le *Google Play Store* (anciennement *Google Marketplace*) et le *Windows Phone Store* apparaissent, respectivement pour Android et Windows Phone. Notons tout de même qu'au niveau mondial les trois systèmes d'exploitation pour mobiles les plus utilisés actuellement sont Android, iOS et Windows Phone (toutefois assez loin derrière pour le moment).

Figure 1-1
Statistiques des OS mobiles
pour mars 2013
Source : d'après Kantar Worldpanel
ComTech



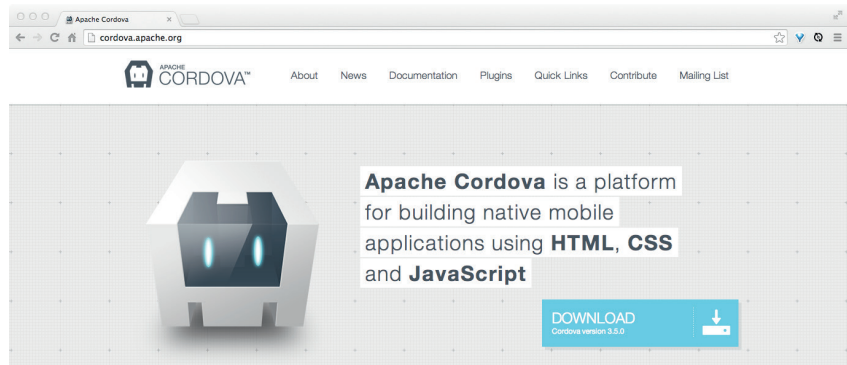
HTML 5, le retour

S'il est nécessaire de parler à iOS en Objective-C, Android ne comprend que le Java, et Windows Phone, quant à lui, accepte idéalement le C++. C'est face à cette diversité de langages que PhoneGap a vu le jour. En réalité, le *framework* a été créé par la société Nitobi Software, lors d'un *iPhone-DevCamp* à San Francisco en 2009, dans le but de proposer aux développeurs iOS une alternative plus simple à l'Objective-C et sa syntaxe déroutante. Cette alternative n'est ni plus ni moins que l'HTML accompagné de CSS (*Cascading Style Sheets*) et JavaScript... les langages du Web.

Cette fois, l'idée de développer des applications pour iPhone en HTML plaît. Adobe rachète Nitobi fin 2011 et PhoneGap (désignant à l'époque le projet open source) est donné à la fondation Apache Software afin de demeurer un composant libre (sous licence Apache 2.0). Le projet devient alors Apache Cordova, Cordova étant le nom de la rue où sont situés à l'époque les locaux de Nitobi à Vancouver. Aujourd'hui, PhoneGap fait référence aux solutions estampillées Adobe utilisant techniquement Apache Cordova.

Cordova en bref

Figure 1-2
Le site officiel
d'Apache Cordova



 About Apache Cordova™

Pourquoi ?

Cordova permet donc de développer des applications pour iOS (c'est-à-dire destinées aux smartphones et tablettes et distribuées via l'App Store) avec les langages du Web. Celles-ci, contrairement aux WebApps traditionnelles, ont la particularité de s'exécuter dans un contexte différent leur offrant (du moins pour le moment) davantage de libertés. En effet, le futur d'HTML 5 laisse entrevoir de belles choses jusque-là encore compliquées, voire impossibles à réaliser dans un navigateur Internet. Citons par exemple la gestion d'une caméra pour la prise de photos et/ou de vidéos, ou encore l'accès au répertoire de l'appareil.

Là où HTML 5 n'est pas encore totalement présent, Cordova excelle en proposant immédiatement l'accès à des API (*Application Programming Interface*), ailleurs encore changeantes et donc

relativement peu implémentées par les éditeurs de navigateurs Internet. Le framework joue même la carte de la transparence, car les contributeurs du projet s'assurent de suivre scrupuleusement les spécifications W3C (*World Wide Web Consortium*) et WHATWG (*Web Hypertext Application Technology Working Group*). Chose importante, car dans quelques années, lorsque HTML 5 sera finalisé et disponible en totalité dans nos navigateurs Internet, Cordova cessera simplement d'exister, ce qui est d'ailleurs officiellement son but à long terme.

Comment ?

Question fonctionnement, Cordova est ce que nos chers collègues anglophones appellent couramment un *wrapper* : un enrobage ou le fait d'utiliser de manière créative les outils mis à disposition par Apple afin d'en concevoir de nouveaux. En effet, le SDK (*Software Development Kit*) iOS peut facilement s'accommoder des idées les plus folles en ouvrant l'accès aux capteurs et autres fonctionnalités innovantes de l'iPhone/iPad. Et si on souhaite afficher une page HTML sur son bel écran tactile, on s'intéressera à un composant spécialement prévu pour cela : la *WebView* (plus ou moins une fenêtre de Mobile Safari dépourvue de barre d'URL et autres menus).

Sans toutefois rentrer dans le détail, Cordova exploite donc la *WebView* (en plein écran) en augmentant cette dernière avec certaines API HTML 5 très utiles, lesquelles se chargeront alors de communiquer avec la machine en Objective-C via JavaScript. Cordova est une sorte de traducteur JavaScript vers Objective-C (et vice versa) reposant sur les standards du Web, fonctionnant de façon extrêmement simple et quasi transparente. Une application web exécutée dans ce contexte, dialoguant avec du code natif, est nommée « application hybride ».

Pour qui ?

Un des principes fondamentaux du Web est qu'il s'agit d'une plate-forme mondiale libre, ouverte et relativement facile d'accès. C'est ce qui fait son succès, ce qui explique qu'on le retrouve aujourd'hui partout sur nos smartphones et tablettes, quels que soient leur marque et système d'exploitation. Oui, chaque appareil inclut son navigateur Internet... et par conséquent sa *WebView*. Peu importe que cette dernière soit écrite en Java, C++ ou Objective-C, elle aura toujours pour finalité l'affichage de fichiers HTML.

En partant de ce constat simple, l'équipe à l'origine de la création de Cordova a eu l'intelligence de ne pas s'arrêter à l'iPhone. Elle a travaillé d'arrache-pied à prendre en charge la concurrence de manière totalement transparente et cohérente. Ainsi, iOS, Android, BlackBerry, Fire OS, Ubuntu, Windows Phone, Windows 8 et Tizen sont aujourd'hui officiellement compatibles – Cordova faisant abstraction de leurs différences dans la plupart des cas, autant que faire se peut.

Remarque

Il s'agit là d'une liste de plates-formes en constante remise en cause. En effet, lorsque l'une d'entre elles n'est plus jugée comme suffisamment présente sur le marché, sa compatibilité n'est simplement plus maintenue dans les versions suivantes de Cordova.

En pratique

Malgré la vaste panoplie de systèmes d'exploitation pris en charge par Cordova, nous ne traiterons dans cet ouvrage que d'iOS et Android, du fait de leurs parts de marché dominantes (se référer à la figure 1-1). Pour ces deux plates-formes, l'ensemble des fonctionnalités offertes par Cordova est pleinement accessible comme le montre le tableau 1-1, tiré de la documentation officielle. Citons, par exemple, la gestion du système de fichiers, l'accès aux différents capteurs présents sur les appareils, ainsi qu'à certaines données utilisateur, mais aussi l'ajout d'événements permettant d'agir à différents moments du cycle de vie des applications et de la connexion réseau.

Tableau 1-1. Fonctionnalités offertes par Cordova pour les plates-formes prises en charge

	Amazon FireOS	Android	BlackBerry 10	Firefox OS	iOS	Ubuntu	Window Phone 8	Windows 8	Tizen
CLI Cordova	✓ (Mac, Windows, Linux)	✓ (Mac, Windows, Linux)	✓ (Mac, Windows)	✓ (Mac, Windows, Linux)	✓ (Mac)	✓ (Ubuntu)	✓ (Windows)	✓	
WebView embarquée	✓	✓			✓	✓			
Ajout de plug-ins	✓	✓	✓		✓	✓	✓	✓	
Fonctionnalités officielles									
Accéléromètre	✓	✓	✓	✓	✓	✓	✓	✓	✓
Appareil photo	✓	✓	✓	✓	✓	✓	✓	✓	✓
Capture Audio/Vidéo	✓	✓	✓		✓	✓	✓		
Boussole	✓	✓	✓		✓ (3Gs+)	✓	✓	✓	✓
État de la connexion	✓	✓	✓		✓	✓	✓	✓	✓
Répertoire de contacts	✓	✓	✓	✓	✓	✓	✓		
Informations sur l'appareil	✓	✓	✓	✓	✓	✓	✓	✓	✓
Événements spécifiques	✓	✓	✓		✓	✓	✓	✓	✓
Système de fichiers	✓	✓	✓		✓	✓	✓	✓	
Géolocalisation	✓	✓	✓	✓	✓	✓	✓	✓	✓
Internationalisation	✓	✓			✓	✓	✓		
Navigateur Internet intégré	✓	✓	✓		✓	✓	✓	✓	
Lecteur multimédia	✓	✓	✓		✓	✓	✓	✓	✓
Notifications	✓	✓	✓		✓	✓	✓	✓	✓
Écrans de lancement	✓	✓	✓		✓	✓	✓	✓	
Stockage interne	✓	✓	✓		✓	✓	✓ (LocalStorage et IndexedDB)	✓ (LocalStorage et IndexedDB)	✓

Source : Cordova

Avec sa version 3, Cordova a subi un changement d'architecture profond. Celle-ci s'articule désormais autour d'un modèle cœur/plug-ins dans lequel chacune des API listées devient optionnelle et devra être ajoutée selon les besoins de l'application à construire. Pour l'occasion, modularité rime avec simplicité et légèreté. Toutes les API composant alors historiquement Cordova sont autant de plug-ins publiés maintenant dans un catalogue au même niveau que ceux créés par la communauté (<http://plugins.cordova.io/>). Nous ne détaillerons pas davantage le fonctionnement des versions antérieures, de plus amples informations à leur sujet sont disponibles dans d'autres publications, ainsi que sur la Toile.

Figure 1-3
*L'infrastructure de Cordova
pré et post-version 3*
Source : PhoneGap

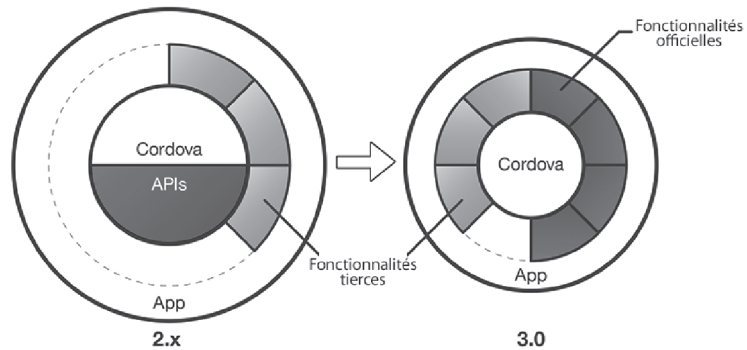
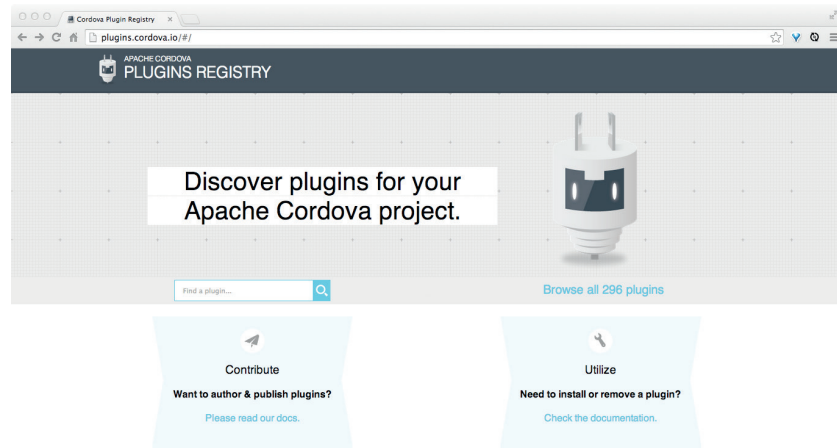
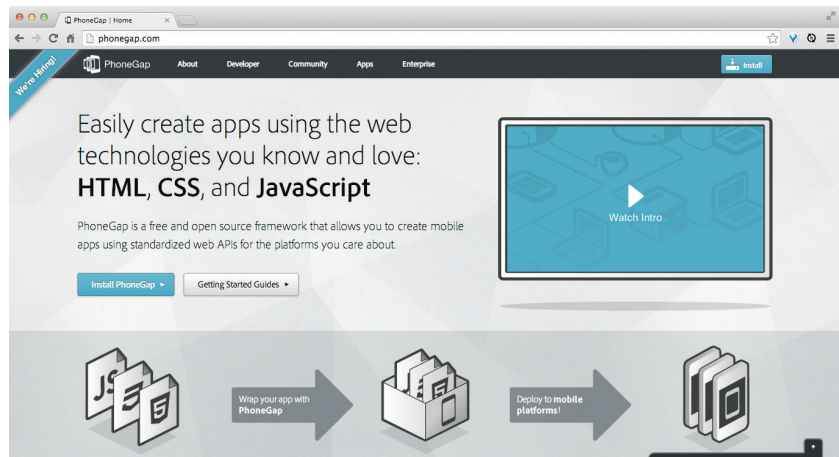


Figure 1-4
*Le catalogue officiel
de plug-ins
pour Cordova*



PhoneGap en bref

Figure 1-5
Le site officiel d'Adobe
PhoneGap

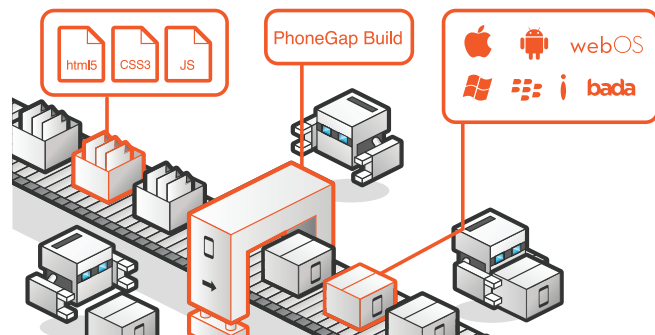


Pourquoi ?

Maintenant que vous êtes familiarisé avec Cordova, il est grand temps d'aborder PhoneGap. Détenue par la société Adobe, cette marque fait désormais plus référence à de multiples services qu'à une simple technologie. Forte de sa longue expérience dans le domaine de la création graphique et web, Adobe a dans un premier temps intégré la solution à des outils de la Creative Suite tels que Dreamweaver. Puis l'entreprise a développé une série de services autour de PhoneGap, comme PhoneGap Build (<https://build.phonegap.com/>) et plus récemment PhoneGap Enterprise (<http://enterprise.phonegap.com/>), un portail des solutions PhoneGap dédiées aux entreprises (support, marketing, etc.).

Si le projet PhoneGap est simplement basé sur Cordova, PhoneGap Build est en réalité un des produits phares de la marque. Avec l'aide de Cordova, le développement d'applications pour smartphones et tablettes est déjà indéniablement simplifié, car homogénéisé. Mais PhoneGap Build va encore plus loin car ce service en ligne ne promet rien de moins que la compilation des applications PhoneGap dans le Nuage (voir figure 1-6).

Figure 1-6
Fonctionnement de PhoneGap Build
Source : PhoneGap



Quels avantages ?

PhoneGap Build est aujourd'hui la principale réponse à la question : « Pourquoi choisir PhoneGap plutôt que Cordova ? » Cordova/PhoneGap libère les développeurs de la contrainte d'apprendre à maîtriser les langages de programmation et API propres aux différentes plates-formes. PhoneGap Build, quant à lui, les décharge de toutes les subtilités liées à la compilation des binaires destinés à être déposés sur les magasins d'applications en ligne. Pourtant, si PhoneGap Build est pour beaucoup un atout majeur, vous découvrirez au fil de cet ouvrage les avantages et inconvénients de ce service. Vous apprendrez aussi pourquoi nous préférons malgré tout bien souvent nous en passer et utiliser simplement Cordova plutôt que PhoneGap, bien que les deux outils soient interchangeables.

Enfin, le développement d'applications mobiles avec des technologies telles que HTML, CSS et JavaScript est, dans son ensemble, clairement une aubaine pour les développeurs web non particulièrement désireux d'apprendre d'autres langages.

L'idée d'un code unique pour plusieurs plates-formes est – bien que ce ne soit pas toujours réalisable – probablement le plus gros point positif de Cordova/PhoneGap. Cependant, ce type d'applications conviendra en général mieux aux projets relativement peu ambitieux, car certaines interfaces seront au final plus simples/rapides à coder en natif – natif qui fournira également toujours de meilleures performances que celles atteignables au sein d'une WebView.

Attention

Il est crucial de bien peser le pour et le contre avant de faire ce choix déterminant. Une application de montage vidéo n'est, par exemple, pas une bonne candidate pour Cordova/PhoneGap, tandis qu'un réseau social comme Twitter pourrait l'être.

En pratique

Si Adobe a accepté de donner Cordova à la fondation Apache, les contributeurs au projet restent majoritairement ses employés, aidés de développeurs travaillant pour des entreprises de renommée mondiale telles que Microsoft, IBM, Intel, Google et BlackBerry. Chacun peut également décider d'apporter son aide à titre bénévole en soumettant d'éventuels correctifs, voire en décrivant simplement tout problème rencontré sur un portail dédié dont l'accès est possible via le site officiel de l'outil.

Côté usage, d'après les quelques données statistiques publiquement disponibles, le framework Cordova/PhoneGap a été téléchargé plus d'un million de fois à ce jour par plus de 400 000 développeurs à travers le monde. Il s'agit donc bien là du leader de l'application mobile multi-plates-formes avec, en juillet 2011, plus de 500 000 visiteurs par mois sur son site officiel. Le service PhoneGap Build (payant) comptait, à la même période, plus de 10 000 utilisateurs avec un volume de 23 000 applications compilées et une croissance de 2 000 utilisateurs par mois. On peut donc facilement imaginer que ces chiffres ont doublé depuis lors.

Remarque

PhoneGap a remporté plusieurs prix prestigieux tels que celui de la technologie de l'année 2012 et celui du meilleur outil de développement *cross-platform*.

Aujourd'hui, plusieurs milliers d'applications hybrides, réalisées à l'aide de Cordova ou PhoneGap, contribuent à la bonne santé de l'App Store, du Google Play Store et du Windows Phone Store, pour ne citer qu'eux. Quelques-unes d'entre elles sont d'ailleurs très connues, pas très loin du haut des classements dans leurs catégories. Espérons qu'elles soient vite, grâce à cet ouvrage, rejointes par les vôtres !