

# Ionic

Développez des applications  
mobiles multiplateformes  
avec Cordova et AngularJS

→ Informatique technique

Fichiers complémentaires  
à télécharger



  
Collection

epsilon

Cédric MILLAURIAUX

Les éléments à télécharger sont disponibles à l'adresse suivante :

**<http://www.editions-eni.fr>**

Saisissez la référence de l'ouvrage **EPION** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

## Avant-propos

1. Le marché des applications mobiles. . . . . 13
2. Pourquoi développer une application mobile aujourd'hui ? . . . . . 14
3. À qui s'adresse cet ouvrage ? . . . . . 15
4. Approche de cet ouvrage . . . . . 15
5. Le projet fil rouge . . . . . 16
6. Remerciements . . . . . 16

## Chapitre 1

### Applications mobiles multiplateformes

1. Qu'est-ce qu'une application mobile multiplateforme ? . . . . . 17
2. Pourquoi utiliser une technologie multiplateforme ? . . . . . 20
  - 2.1 Avantages du développement multiplateforme . . . . . 20
  - 2.2 Types de technologies . . . . . 21
    - 2.2.1 webview . . . . . 22
    - 2.2.2 Métalangages . . . . . 23
    - 2.2.3 OpenGL ES . . . . . 24
    - 2.2.4 Hybride . . . . . 25
3. Différences fondamentales entre développement natif et multiplateforme . . . . . 26

## Chapitre 2

### Architecture multiplateforme

1. Architecture multiplateforme webview .....	29
1.1 Principe de base .....	29
1.2 Navigateur web intégré .....	30
1.3 Système de plugins .....	31
1.4 Packaging .....	33
2. Ionic et le choix d'AngularJS .....	34
3. Architecture AngularJS .....	36
3.1 AngularJS et MVC .....	36
3.2 Application AngularJS .....	37

## Chapitre 3

### Découverte de Cordova

1. Présentation de Cordova .....	39
2. Installation de Cordova .....	41
2.1 Installation de NodeJS .....	41
2.2 Installation de GIT .....	42
2.3 Installation de Cordova .....	42
3. Commandes usuelles .....	43
3.1 Créer un projet .....	44
3.2 Ajouter une plateforme .....	44
3.3 Démarrer le serveur web .....	46
3.4 Démarrer l'application sur une plateforme native .....	48
3.5 Nettoyer le projet .....	48
4. Déployer sur Android .....	49
4.1 Installation de Java .....	49
4.1.1 Mise à jour des variables d'environnement Windows ..	49
4.1.2 Mise à jour des variables d'environnement Linux et OSX .....	50

4.2	Installation du SDK Android	51
4.2.1	Mise à jour des variables d'environnement Windows	51
4.2.2	Mise à jour des variables d'environnement Linux et OSX	52
4.2.3	Installation des packages Android	52
4.3	Ajout d'un émulateur	54
4.4	Ajout de la plateforme	55
4.5	Lancement de l'application	56
5.	Déployer sur iOS	56
5.1	Installation de Xcode	57
5.2	Ajout de la plateforme	57
5.3	Ouverture du projet dans Xcode	57
5.4	Création des certificats	58
5.5	Compilation de l'application	58
5.6	Lancement de l'application	59
6.	Déployer sur Windows	61
6.1	Installation de Visual Studio	61
6.2	Ajout de la plateforme	63
6.3	Lancement de l'application depuis la ligne de commande	63
6.3.1	Lancement d'une application de bureau	63
6.3.2	Lancement d'une application smartphone	63
6.4	Lancement de l'application depuis Visual Studio	64

## Chapitre 4

### Découverte de Ionic

1.	Présentation de Ionic	67
2.	Comparaison de Ionic et de ses challengers	69
2.1	Onsen	69
2.2	Intel XDK	70
2.3	Mobile Angular UI	70
2.4	TouchstoneJS	71
2.5	Frameworks commerciaux	71

2.6	Frameworks n'utilisant pas Cordova . . . . .	71
2.7	Ionic, le choix du roi ? . . . . .	71
3.	Installation de Ionic . . . . .	72
3.1	Prérequis . . . . .	72
3.2	Installation de la ligne de commande . . . . .	72
4.	Apports de Ionic à Cordova . . . . .	73
4.1	Création du projet . . . . .	74
4.2	Exécution de l'application sur un serveur web . . . . .	74
4.3	Rechargement en temps réel : le livereload . . . . .	75
4.4	Gestion des plugins . . . . .	76
4.5	Gestion de la configuration . . . . .	76
4.5.1	Sauvegarde de la configuration . . . . .	77
4.5.2	Restauration de la configuration . . . . .	78
4.5.3	Remise à zéro de la configuration . . . . .	78
4.6	Ionic Cloud . . . . .	79
4.7	Commandes diverses . . . . .	79

## Chapitre 5

### Mise en place de l'application

1.	Initialisation du projet . . . . .	81
1.1	Création de l'application . . . . .	81
1.2	Arborescence du projet . . . . .	83
1.2.1	Répertoire hooks . . . . .	83
1.2.2	Répertoire platform . . . . .	83
1.2.3	Répertoire plugins . . . . .	84
1.2.4	Répertoire resources . . . . .	84
1.2.5	Répertoire SCSS . . . . .	85
1.2.6	Répertoire www . . . . .	85
1.2.7	Fichiers divers . . . . .	86
1.3	Environnement de développement . . . . .	87

2.	Découverte du projet par défaut .....	88
2.1	Fichier index.html .....	89
2.1.1	Section Head .....	89
2.1.2	Section Body .....	90
2.2	Fichier app.js .....	91
3.	Création du premier couple contrôleur/service .....	93
3.1	Création du contrôleur .....	93
3.2	Création du service .....	97
3.3	Injection de dépendances .....	99
3.4	Affichage d'une liste .....	101
4.	Vues .....	105
4.1	Installation de ui-router .....	106
4.2	Création d'une vue .....	107
4.3	Injection de la vue dans la page principale .....	107
5.	Routes .....	108
5.1	Création d'une route .....	108
5.2	Choix d'une route par défaut .....	109
5.3	Navigation entre les routes .....	110
5.3.1	Création d'une deuxième route .....	110
5.3.2	Récupération de paramètres dans le contrôleur .....	113
5.3.3	Redirection avec un lien HTML .....	113
5.3.4	Redirection avec la directive ui-sref .....	114
5.3.5	Redirection depuis le contrôleur .....	116

## Chapitre 6

### Habiller l'application

1.	Aperçu des composants CSS .....	119
1.1	Header .....	120
1.1.1	Ajout d'un header .....	120
1.1.2	Utilisation de la directive ion-header-bar .....	121
1.1.3	Choix de la couleur .....	122

1.1.4	Ajouter des boutons	122
1.1.5	Sous-header	124
1.2	Footer	124
1.3	Boutons	126
1.3.1	Utilisation d'un bouton	127
1.3.2	Taille des boutons	127
1.3.3	Largeur des boutons	129
1.3.4	Liens et boutons détourés	130
1.3.5	Groupe de boutons	131
1.3.6	Icônes	132
1.4	Listes	135
1.4.1	Liste simple	135
1.4.2	Séparateur	137
1.4.3	Icônes	138
1.4.4	Boutons	140
1.4.5	Images	142
1.5	Carte	143
1.5.1	Déclaration d'une carte	144
1.5.2	Titre et séparateur	147
1.5.3	Image	148
1.5.4	Contenu	149
1.5.5	Liste	151
1.6	Formulaires	151
1.6.1	Formulaire simple	152
1.6.2	Champs de saisie	153
1.6.3	Mise en forme	155
1.6.4	Combinaison de champs	159
1.6.5	Saisie HTML5	160
1.6.6	Case à cocher	163
1.6.7	Switch	164
1.6.8	Boutons radio	166
1.6.9	Liste déroulante	168
1.6.10	Slider	170

2. Grille et responsive design . . . . .	171
2.1 Grille simple . . . . .	172
2.2 Taille de colonne imposée . . . . .	174
2.3 Espacement entre colonnes . . . . .	175
2.4 Alignement vertical . . . . .	176
2.5 Grille responsive . . . . .	176
3. Aperçu des composants JS . . . . .	179
3.1 Vue principale et scroll . . . . .	179
3.2 Animation de chargement . . . . .	182
3.3 Animation de chargement et blocage des actions utilisateur . . . . .	185
3.4 Rafraîchissement de l'écran . . . . .	188
3.5 Modale . . . . .	191
3.6 Pop-up . . . . .	196
3.6.1 Alerte . . . . .	197
3.6.2 Saisie utilisateur . . . . .	199
3.7 Listes . . . . .	201
3.7.1 Options . . . . .	201
3.7.2 Bouton de suppression . . . . .	203
3.7.3 Ordonner la liste . . . . .	205
3.8 Clavier . . . . .	207
3.9 Carrousel . . . . .	208
3.10 Zone de défilement . . . . .	213
4. Navigation . . . . .	219
4.1 Vues . . . . .	219
4.2 Menu latéral . . . . .	230
4.3 Onglets . . . . .	234
5. Configuration générale . . . . .	242
5.1 Cache . . . . .	243
5.2 Navigation . . . . .	243
5.2.1 Barre de navigation . . . . .	243
5.3 Formulaire . . . . .	245
5.4 Onglets . . . . .	246



5.5	Vues	247
6.	Gestes	248
6.1	Directives gestuelles	248
6.2	Service gestuel	252

## Chapitre 7

### Communiquer avec un serveur

1.	Présentation de l'architecture REST	253
1.1	Protocole HTTP	253
1.2	Architecture REST	256
2.	Création d'un service de communication HTTP	258
2.1	Service \$http	258
2.2	Traitement du retour	260
3.	Promesses et appels asynchrones	262
3.1	Appels asynchrones et callback	262
3.2	Promesses	263
3.2.1	Création d'une promesse	264
3.2.2	Retour de résultat	265
3.2.3	Utilisation du retour	266
3.2.4	Avantage des promesses	266
4.	Échanges avec l'API OMDb	268
4.1	OMDb API	268
4.1.1	Recherche d'une liste de films	268
4.1.2	Détails d'un film	277
5.	Utilisation du scroll infini	282

## Chapitre 8 Développement générique

1. Pourquoi développer des composants génériques. . . . .	287
2. Modules. . . . .	288
2.1 Présentation du principe des modules. . . . .	289
2.2 Application de l'architecture modulaire . . . . .	292
2.2.1 Module starter . . . . .	292
2.2.2 Création du module films . . . . .	294
3. Directives . . . . .	297
3.1 Concept . . . . .	297
3.2 Création d'une directive . . . . .	297
3.2.1 Types de directives . . . . .	298
3.2.2 Passage de paramètres . . . . .	300
3.2.3 Renvoi d'un template . . . . .	301
3.2.4 Contrôleur interne . . . . .	302
3.2.5 Manipulation de l'élément parent . . . . .	303
3.3 Mise en application : la directive film . . . . .	304
3.3.1 Création de la directive . . . . .	304
3.3.2 Implémentation du contrôleur . . . . .	305
3.3.3 Adaptation du template . . . . .	306
3.3.4 Injection de la directive . . . . .	306
3.3.5 Spécialisation de la directive . . . . .	307
4. Filtres. . . . .	312
4.1 Filtre simple . . . . .	312
4.1.1 Utilisation de filtres. . . . .	312
4.1.2 Mise en place d'un filtre . . . . .	313
4.2 Filtre personnalisé. . . . .	315
4.2.1 Déclarer un filtre . . . . .	315
4.2.2 Implémenter un filtre . . . . .	315
4.2.3 Implémenter un tri . . . . .	316
5. Fichier de configuration et constantes . . . . .	316

**Chapitre 9****S'adapter à la plateforme**

1. Pourquoi décliner l'interface pour une plateforme ? .....	319
2. Ionic et l'adaptation à la plateforme .....	320
2.1 Marqueurs de la plateforme .....	320
2.1.1 Spécifications de la plateforme dans la vue .....	320
2.1.2 Spécifications de la plateforme dans le code JS .....	322
2.1.3 Routage en fonction de la plateforme .....	324
2.1.4 Autres fonctionnalités de la classe platform .....	325
2.2 Événements système .....	327
3. Adaptations pour Android .....	329
3.1 Rappel des adaptations natives .....	329
3.2 Bouton retour .....	330
4. Adaptations pour iOS .....	333
5. Fichier config.xml .....	335
5.1 Structure du fichier config.xml .....	336
5.2 Informations sur l'application .....	336
5.3 Page de démarrage .....	337
5.4 Restriction des requêtes multidomaines .....	337
5.5 Préférences générales .....	338
5.6 Plateformes .....	338
5.7 Plugins .....	339

**Chapitre 10****Plugins**

1. Tirer parti des capacités des smartphones .....	341
2. Fonctionnement du système de plugins .....	343
2.1 Rappels sur l'architecture des plugins .....	343
2.2 Rechercher un plugin .....	344
2.3 Installer un plugin .....	345
2.4 Désinstaller un plugin .....	347

3. Social : partager sur les réseaux sociaux . . . . .	347
4. Camera : prendre des photos . . . . .	353
5. Geolocation : géolocaliser l'utilisateur . . . . .	363
6. LocalStorage : stocker des données . . . . .	369
7. NgCordova : utiliser les plugins à la manière d'AngularJS. . . . .	374

## Chapitre 11 Ionic et performances

1. Performances de Ionic et AngularJS sur mobile . . . . .	377
2. Conseils pratiques pour optimiser les performances . . . . .	379
2.1 Généralités . . . . .	379
2.1.1 Appels asynchrones et promesses. . . . .	379
2.1.2 Librairies . . . . .	381
2.1.3 Animations . . . . .	381
2.1.4 Manipulation du DOM. . . . .	383
2.1.5 Taille et poids des images . . . . .	384
2.1.6 Privilégier les maps . . . . .	385
2.1.7 Minification JS et CSS. . . . .	386
2.1.8 Les outils de diagnostic . . . . .	386
2.2 AngularJS . . . . .	388
2.2.1 Binding unique. . . . .	388
2.2.2 Identifiant d'itération . . . . .	389
2.2.3 ng-if vs ng-show . . . . .	390
2.2.4 Enchaînement de filtres . . . . .	390
2.2.5 Mode débogage . . . . .	390
2.2.6 Pour aller plus loin. . . . .	391
2.3 Ionic. . . . .	391
2.3.1 Scroll natif . . . . .	391
2.3.2 Collection repeat . . . . .	392
2.3.3 Infinite scroll . . . . .	392
2.3.4 Cache . . . . .	393

2.3.5 Manipulation du DOM pendant les transitions . . . . .	393
2.3.6 Crosswalk . . . . .	394

## **Chapitre 12** **Publier son application**

1. Préparer son application . . . . .	397
1.1 Préparation logicielle . . . . .	397
1.2 Génération des icônes . . . . .	398
1.3 Génération du splash screen . . . . .	399
2. Publication sur Google Play Store . . . . .	401
3. Publication sur Itunes . . . . .	406
4. Mettre à jour l'application . . . . .	412

## **Chapitre 13** **Pour aller plus loin**

1. Enrichir l'application . . . . .	413
2. Ionic.io . . . . .	414
2.1 Ionic Cloud . . . . .	414
2.2 Ionic Creator . . . . .	420
2.3 Ionic View . . . . .	421
3. Ionic 2 . . . . .	421
3.1 Différences avec Ionic 1 . . . . .	422
3.2 Essayer Ionic 2 . . . . .	425
3.2.1 Création du projet . . . . .	426
3.2.2 Analyse du projet . . . . .	426
3.2.3 Mettre en place les premiers composants . . . . .	427
3.2.4 Ajouter une deuxième page . . . . .	429
3.2.5 Documentation et tutoriels . . . . .	433
Index . . . . .	435



## Chapitre 4

# Découverte de Ionic

### 1. Présentation de Ionic

Ionic est un framework reposant sur Cordova et qui a pour ambition de fournir au développeur les bases et l'outillage pour débiter rapidement et sereinement la création d'une application mobile multiplateforme.

Avec Cordova, les développeurs disposent d'un socle fiable et d'un écosystème imposant (utilitaires, documentation, exemples) pour développer des applications mobiles s'appuyant sur des technologies web. Néanmoins, lors de la création d'un projet Cordova, il reste à la charge du développeur de mettre en place tout un socle technique avant de commencer à créer le premier écran : framework JavaScript, style des éléments HTML de base, fonctions utilitaires et éléments récurrents (pop-up, message de confirmation, animation de chargement, etc.).

Ionic propose donc de mettre en place automatiquement une architecture de base, composée des éléments suivants :

- Un framework JavaScript : AngularJS
- Une feuille de style surchargeant les principaux éléments HTML (boutons, cases à cocher, champs de formulaire, etc.) avec pour but de les faire ressembler aux composants des applications natives Android et iOS

- Des styles applicables à des conteneurs HTML pour gérer la spécificité de l'affichage sur mobile : listes, cartes, boutons occupant toute la largeur de l'écran, etc.
- Des composants JavaScript permettant de mettre en place rapidement et simplement des fonctionnalités avancées comme l'affichage de fenêtres de dialogue, la gestion des gestes de l'utilisateur ou encore la gestion d'un menu latéral
- Des outils complémentaires à ceux offerts par Cordova pour faciliter le développement (rechargement automatique des sources par exemple) ou exécuter des tâches annexes (génération des icônes, déploiement de l'application, etc.)

Avec Ionic, le développeur peut immédiatement commencer à travailler sur son application mobile et pourra se reposer tout au long du développement sur divers composants standards, que ce soit pour maîtriser l'affichage, les interactions utilisateur, les interactions avec les composants bas niveau des smartphones ou encore le métier de l'application, plutôt qu'avoir à les développer lui-même. En résumé, Ionic offre un cadre pour le développement d'applications mobiles multiplateformes, permettant de développer plus rapidement et plus efficacement qu'avec Cordova seul en s'appuyant sur des composants et outils standards, de manière à pouvoir se concentrer sur le développement métier, l'affichage et l'ergonomie.

Ionic est un framework open source et gratuit, dont la première version est sortie en 2013. Depuis, une large communauté s'est formée autour de ce projet, améliorant et enrichissant le framework grâce aux retours d'expérience de ses très nombreux utilisateurs. L'équipe Ionic revendique fièrement avoir permis à plus d'un million d'applications de voir le jour grâce à sa technologie, depuis sa sortie. Fort de son succès mais aussi des leçons apprises tout au long du développement de la première version, l'équipe a annoncé travailler sur une version 2 en repartant de la feuille blanche et en s'appuyant sur Angular 2 comme framework JavaScript.

## 2. Comparaison de Ionic et de ses challengers

Il est difficile de comparer Ionic à la concurrence : il existe effectivement de nombreuses initiatives visant à équiper le développeur Cordova d'outils et de composants lui permettant de gagner du temps tout en se rapprochant du visuel et de l'ergonomie des applications natives. Mais Ionic n'est pas qu'une feuille de style CSS adaptée au mobile, un ensemble de fonctions JavaScript ou d'outils permettant de générer une application plus efficacement : c'est un peu tout cela à la fois. C'est d'ailleurs l'un des gros avantages de Ionic : il offre une véritable boîte à outils faite d'un seul tenant au développeur Cordova qui n'a plus le besoin d'assembler diverses technologies entre elles pour arriver à ses fins.

Il est important de noter que les projets présentés dans ce chapitre reposent tous sur Cordova. L'objectif de tous ces frameworks n'est pas de se substituer à cette technologie qui fait la quasi-unanimité dans l'univers du développement d'applications multiplateformes du type webview, mais plutôt de venir se reposer sur celui-ci et d'enrichir la partie logicielle, inexistante lors de la création d'un projet Cordova.

### ■ Remarque

*La liste présentée ci-dessous est loin d'être exhaustive : des frameworks de ce type sont publiés quasiment tous les jours et revendiquent chacun des spécificités ou une philosophie qui font leur force comparé à la concurrence. Les technologies présentées ci-dessous sont simplement celles rassemblant le plus de développeurs.*

### 2.1 Onsen

Site web : <https://onsen.io/>

Onsen expose les mêmes ambitions que Ionic : devenir un framework tout en un offrant au développeur des composants visuels et ergonomiques prêts à l'emploi, des fonctions JavaScript et un outillage permettant de gagner en efficacité et en stabilité lors du développement d'applications mobiles multiplateformes.



Là où Ionic a fait le choix d'AngularJS comme framework JavaScript, Onsen laisse la liberté au développeur de choisir entre les frameworks suivants : AngularJS, Angular 2, React et JavaScript. La communauté et les ressources sont néanmoins moins développées que celles de Ionic.

## 2.2 Intel XDK

Site web : <https://software.intel.com/en-us/intel-xdk>

Intel XDK est un projet aux ambitions plus larges que celles de Ionic : étendre les fonctionnalités de Cordova et proposer une plateforme adaptée à de multiples cas d'usage (Internet des objets, applications mobiles et même jeux vidéo). Le parti pris de Intel XDK est de ne pas imposer de framework JS ou de composants visuels, mais plutôt d'offrir de nombreux outils pour développer plus rapidement et de manière plus industrielle des applications mobiles multiplateformes. Par exemple, un éditeur graphique permet de dessiner les écrans, tandis que des API spécifiques permettent d'utiliser la technologie WebGL pour dessiner directement à l'écran des formes et animations complexes tout en privilégiant les performances.

## 2.3 Mobile Angular UI

Site web : <http://mobileangularui.com/>

Mobile Angular UI est un framework aux ambitions plus modestes : offrir une boîte à outils pour développeurs d'applications mobiles, composée de feuilles de style CSS (inspirées par le célèbre Bootstrap), du framework AngularJS et de fonctions et services permettant de gérer des éléments complexes comme des fenêtres modales, les gestes de l'utilisateur, les formulaires, les onglets, etc. Mobile Angular UI ne propose donc pas d'étendre les fonctionnalités de Cordova ou d'apporter des outils externes comme le font Ionic et ses concurrents directs. Néanmoins, il s'agit d'un framework simple à prendre en main et permettant de prototyper rapidement une application mobile.

## 2.4 TouchstoneJS

Site web : <http://touchstonejs.io/>

TouchstoneJS est un jeune framework ayant à terme l'ambition de proposer une alternative sérieuse à Ionic. La philosophie de départ est néanmoins différente puisque le framework JavaScript intégré est React.js et non AngularJS. Les débuts sont prometteurs mais il reste beaucoup de travail à fournir autour de la documentation et des exemples fournis.

## 2.5 Frameworks commerciaux

D'autres frameworks peuvent être également cités, ils possèdent de nombreuses qualités mais présentent l'inconvénient d'être payants : Senza Touch, Kendo UI, monaca, etc.

## 2.6 Frameworks n'utilisant pas Cordova

Il existe également d'autres frameworks se démarquant de la philosophie de Ionic : c'est-à-dire proposant une alternative à Cordova de manière à se rapprocher davantage des librairies natives à la plateforme. Dans certains cas, il ne s'agit plus de s'appuyer sur une webview et de mettre en œuvre des technologies web pour développer l'application, mais de proposer un métalangage pilotant directement des composants natifs. Les deux frameworks les plus connus présentant ces caractéristiques sont NativeScript et Titanium.

## 2.7 Ionic, le choix du roi ?

En résumé, Ionic est loin d'être le seul à jouer dans la catégorie des frameworks visant à enrichir l'expérience Cordova. Certains de ses concurrents affirment être plus performants, d'autres mieux outillés, alors que d'autres jouent la carte d'une esthétique plus travaillée. Ionic reste néanmoins un des frameworks les plus populaires et les mieux documentés, ce qui en fait un choix intéressant pour intégrer le cercle des développeurs d'applications webview.

Une fois cette technologie maîtrisée, il ne peut être que conseillé de s'essayer à certaines de ces alternatives pour y trouver de bonnes idées et pourquoi pas, un outil mieux adapté à son propre besoin.

### 3. Installation de Ionic

#### 3.1 Prérequis

Avant d'installer et de commencer à utiliser Ionic, il est impératif de procéder à l'installation de Cordova et d'en comprendre le fonctionnement, ainsi décrits au chapitre précédent.

À l'issue de l'installation de Cordova, les éléments suivants sont donc installés sur le poste de développement :

- NodeJS
- GIT
- Cordova
- Optionnel : une plateforme mobile (Android, iOS ou Windows)

#### 3.2 Installation de la ligne de commande

Ionic a fait les mêmes choix technologiques que Cordova, à savoir reposer sur NodeJS pour ses utilitaires. L'installation de Ionic s'effectue donc par l'intermédiaire du gestionnaire de dépendances de NodeJS : NPM.

■ Dans un terminal, tapez la commande suivante :

```
■ npm install -g ionic
```

Cette commande peut mettre plusieurs minutes à s'exécuter en fonction des ressources du poste de développement et de la qualité de la connexion Internet.