



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 débuter 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 : Sensa 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.