



# HMIN309 - DÉVELOPPEMENT LOGICIEL POUR L'EMBARQUÉ

---

## TUTORIEL SUR FRAMEWORK DE DÉVELOPPEMENT MULTI-PLATEFORME IONIC

---

*Auteurs :*

BEYA NTUMBA Joel - N° 215 11 843

*Encadrant :*

Abdelhak-Djamel SERIAI



30 janvier 2020

## Résumé

Le marché du smartphone est dominé principalement par les Google-Android avec leurs propres systèmes d'applications (principalement basé sur JAVA et Kotlin), Apple-IOS (principalement basé sur Objective-C et Swift) et jusqu'à pas longtemps Microsoft-Windows (se reposant sur C# et XAML).

Pendant longtemps, le résultat de cette concurrence était que si on voulait développer une application pour les trois plateformes, il fallait la coder trois fois, en trois langages et de trois manières différentes.

L'arrivée du framework Cordova a tout changé. Ce framework a permis aux développeurs web de créer des applications multiplateformes avec une seule base de code en HTML, CSS et JavaScript, donnant accès aux différents API natifs des téléphones : l'appareil photo ou la géolocalisation, par exemple.

Ionic vient s'inscrire par-dessus Cordova, permettant l'utilisation d'Angular, avec tous ses avantages, et intégrant beaucoup de composants très utiles pour l'UX : Ionic vous permet de créer des applications dynamiques multiplateformes avec vos connaissances en Angular !

# Sommaire

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Introduction</b>                                  | <b>3</b> |
| 1.1      | IONIC . . . . .                                      | 3        |
| 1.2      | Hybride . . . . .                                    | 3        |
| 1.3      | Natif . . . . .                                      | 3        |
| 1.4      | Cross-Platform . . . . .                             | 3        |
| 1.5      | Apache Cordova . . . . .                             | 3        |
| 1.6      | Gulp . . . . .                                       | 3        |
| <b>2</b> | <b>Prérequis</b>                                     | <b>4</b> |
| <b>3</b> | <b>Installation</b>                                  | <b>4</b> |
| 3.1      | Node Js . . . . .                                    | 4        |
| 3.2      | Ionic Apache Cordova . . . . .                       | 4        |
| 3.3      | Android SDK . . . . .                                | 4        |
| 3.4      | Emulateur . . . . .                                  | 4        |
| <b>4</b> | <b>Création d'un projet</b>                          | <b>4</b> |
| 4.1      | Blank . . . . .                                      | 5        |
| 4.2      | SideMenu . . . . .                                   | 5        |
| 4.3      | Tabs . . . . .                                       | 5        |
| <b>5</b> | <b>Plateforme de déploiement d'application Ionic</b> | <b>5</b> |
| 5.1      | Android . . . . .                                    | 5        |
| 5.2      | IOS . . . . .  | 6        |
| 5.3      | Web . . . . .  | 6        |
| 5.4      | Mobile multiplateforme . . . . .                     | 6        |
| <b>6</b> | <b>Architecture du Framework</b>                     | <b>6</b> |
| <b>7</b> | <b>Conclusion</b>                                    | <b>6</b> |
| 7.1      | Avantages . . . . .                                  | 6        |
| 7.2      | Inconvénients . . . . .                              | 7        |
| 7.3      | Lien git d'une application Démo . . . . .            | 7        |

# 1 Introduction

Quelques définitions sur les éléments du domaine

## 1.1 IONIC

est un framework de création d'applications hybrides. En effet, Ionic permet de créer une application Web ayant l'air d'une application native. Cette application partagera également certaines capacités comme l'accès aux éléments systèmes, ce qui la rend hybride.

En résumé, Ionic est un Framework permettant la création d'applications multiplateforme rapidement et facilement en utilisant des technologies web (JavaScript, HTML, CSS). Il s'appuie sur Angular pour la partie web (construction des vues) et sur Cordova pour la partie native.

## 1.2 Hybride

On parle d'application hybride pour les applications utilisant le navigateur intégré de la plateforme et les technologies du web en accédant aux ressources de l'appareil (Caméra, Géolocalisation ...).

## 1.3 Natif

On parle d'application native lorsque celle ci est entièrement développée dans le langage natif de la plateforme.

## 1.4 Cross-Platform

On parlera d'application cross-platform lorsqu'un seul code est exécuté par plusieurs plateforme.

## 1.5 Apache Cordova

Est une plateforme servant à construire des applications natives mobiles en utilisant HTML/CSS/-Javascript

## 1.6 Gulp

Est un outil destiné à l'organisation et l'exécution des tâches composant un build. Il est utilisé pour l'automatisation des compilations.

## 2 Prérequis

- Notions en **HTML/CSS** pour les vues
- Notions en **TypeScript/Javascript** pour les modèles
- Notions en **Angular/React**(en developpement) pour la gestion de la structure de l’application
- Notions sur **NPM**(Node Package Manager)/**NodeJs** pour les installations des modules développés par la communauté.

## 3 Installation

### 3.1 Node Js

Comme cité précédemment, Ionic s’appui sur la plateforme Node JS.

Voici le lien de téléchargement et d’installation de Node Js ici

**Bon à savoir :** Sous Windows, la commande “npm” ne fonctionnera pas sans les variables d’environnement de système JAVA\_HOME, ANDROID\_HOME et NPM.

Sous linux, les variables d’environnement JAVA\_HOME et ANDROID\_HOME sont nécessaires.

### 3.2 Ionic Apache Cordova

Ensuite, il suffit d’exécuter la commande suivante (en ligne de commande)

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

### 3.3 Android SDK

Pour développer une application pour appareils Android il est indispensable de disposer des outils Android tel que le SDK Android. Si l’on souhaite cibler l’application pour d’autres systèmes d’exploitations, il est là aussi indispensable de disposer du SDK de la plateforme visée. Ceux-ci serviront à la compilation et à la construction de l’application.

### 3.4 Emulateur

L’utilisation d’un émulateur pour tester l’application lors de la phase de développement est une option viable. En effet, utiliser les différents terminaux virtuels proposés par un émulateur pour tester, par exemple, son adaptation aux différents écrans est recommandé.

Une alternative est de lancer l’application sur un serveur web local (localhost) et utiliser les outils de visualisation du navigateur pour simuler un appareil précis.

## 4 Création d’un projet

Une fois les outils installés, et qu’on dispose d’un environnement de travail opérationnel, il est maintenant temps de créer notre projet que l’on nommera ‘monProjet’.

Ionic Framework propose trois Templates prédéfinis.

## 4.1 Blank

Le premier et le plus basique est la création d'une application vide. Pour cela, tapez la commande suivante (en ligne de commande) :

```
$ ionic start "NomDuProjet" blank
```

Il est recommandé de commencer par un projet vide lorsque l'on a une certaine expérience de la technologie. Pour des débutants, il est conseillé d'utiliser un des deux autres Templates.

## 4.2 SideMenu

Le second et certainement le plus utile, est la création d'une application avec un menu latéral intégré. Pour cela taper la commande suivante dans le terminal :

```
$ ionic start "NomDuProjet" blank
```

## 4.3 Tabs

Enfin, le dernier Template proposé qui est celui par défaut lorsqu'on ne spécifie pas le Template est la création d'une application avec des onglets. Pour cela tapez la commande suivante (en ligne de commande) :

```
$ ionic start "NomDuProjet" tabs
```

# 5 Plateforme de déploiement d'application Ionic

Ionic est un framework permettant la création d'applications hybrides. De ce fait il est possible de cibler les plateformes (systèmes d'exploitations) que l'on souhaite viser.

## 5.1 Android

Pour configurer votre application pour les appareils Android. Il suffit d'exécuter la suivante commande à la racine de votre projet (en ligne de commande) :

```
$ npm install -native-run  
$ ionic platform add android
```

Ensuite, pour compiler puis lancer et déployer votre application sur un appareil connecté ou depuis un émulateur, il suffit d'exécuter la commande suivante (en ligne de commande) :

```
$ ionic cordova run android
```

## 5.2 IOS

Pour configurer votre application pour les appareils IOS. Il suffit d'exécuter la suivante commande à la racine de votre projet (en ligne de commande) :

```
$ ionic cordova build ios
$ ionic cordova emulate ios
```

Ensuite, pour compiler puis lancer et déployer votre application sur un appareil connecté ou depuis un émulateur, il suffit d'exécuter la commande suivante (en ligne de commande) :

```
$ ionic cordova prepare ios
```

## 5.3 Web

Il est par ailleurs possible de déployer l'application sur un serveur local de test. Pour cela exécuter à la racine de votre projet la commande suivante (en ligne de commande) :

```
$ ionic serve
```

## 5.4 Mobile multiplateforme

Cette fois ci possibilité de déployer l'application sur un serveur local de test pour les applications multi-plateformes(IOS, Android, Windows Phone). Pour cela exécuter à la racine de votre projet la commande suivante (en ligne de commande) :

```
$ ionic serve --lab
```

# 6 Architecture du Framework

Le projet Ionic basé sur AngularJS a une architecture **MVC**. Cette architecture 3-tier est intéressante de part la séparation des données (le modèle), de la gestion de l'interface (la vue) et des actions (le contrôleur)

# 7 Conclusion

## 7.1 Avantages

- Le cross-plateforme. A partir d'un socle de développement pouvoir déployé (en une seule ligne de commande) une application sur plusieurs environnements (IOS, Android, Windows Phone). Cela permet de réduire les couts et délais de développement pour proposer plusieurs applications natives pour les différents systèmes d'exploitation mobile
- Il est basé sur Apache Cordova et AngularJS qui disposent d'une grande communauté et par conséquence de nombreux plugins de développement.
- En quelques lignes de commande (trois) il est possible de créer une application multiplateforme.
- Les documentations des technologies AngularJS et Apache Cordova sont complètes.

- Le Framework s’appuie sur AngularJS, gage de stabilité puisque le Framework est maintenant largement utilisé et surtout développé par Google.
- Le déploiement de l’application vers un appareil mobile connecté se fait en une ligne de code (agréable pour les tests).
- La prise en main des technologies est relativement rapide, même pour des développeurs ayant peu d’expérience avec les technologies Web.

## **7.2 Incovénients**

L’appréhension de l’architecture MVC (Modèle Vue Controller) proposé par AngularJS et qui est la structure de l’application, peut-être un inconvénient pour les développeurs peu à l’aise avec celle-ci.

## **7.3 Lien git d’une application Démo**

Vous trouverez ici le dépôt git des sources d’une petite application démo de ce tutoriel ici