Apprendre Ionic



D’après Randriamanarina Franito elgissio,   
étudiant en informatique à l’université CNTEMAD  
et stagiaire chez DDNSI au sein de la CUA

Edition 2023

**Avant-propos**

# Introduction

Ionic, qu’est-ce que c’est ?  
 C’est un framework de développement qui permet de concevoir des post plateforme. Développer une application mobile post plateforme n’a jamais été une tache à réaliser d’une manière classique parce qu’il existe plusieurs modèle de téléphone et plusieurs système d’exploitation dont les plus connus sont : Android (google) et iOS (apple). C’est là que la difficulté se présente parce qu’il faut développer sur diffèrent système et tenir compte de leur différente version. Il faut également tester votre application sur énormément de téléphone. Au final on se retrouve avec quasi autant de version de l’application que de nombre de téléphone existant qui veut dire beaucoup de temps et de l’argent dépensé.   
 C’est pourquoi Ionic a vu le jour (en 2013). C’est un framework openSource pour le développement d’application pour smartphone. C’est un ensemble de composant et d’outils pour pouvoir developper olus facilement des applications mobiles fonctionnant uniquement sur google chrome. Il va juste utiliser les outils HTML/CSS/JS et il intègre d’autres framework comme angular, ReactJS et VueJS.  
 Pour utiliser les composants natifs comme appareil photo, gps ou le sistème de vibration de téléphone, Ionic utilise des librairies et des plugins compatibles qui sont apache CARDOVA et CAPACITOR.  
 Par exemple, si on veut utiliser l’appareil photo du téléphone, il suffit d’installer le plugin de la camera dans la code source et appeler la fonction relationnelle qui permettra aux utilisateurs de prendre une photo sans besoin de savoir le système d’exploitation (c’est l’application Ionic qui s’en charge en appelant la fonction qui fera appelle au plugin correspondant).

Ionic permet de fonctionner sur différent OS, il suffira d’un seul code source pour pouvoir fonctionner notre application sur android et iOS. Il est également possible de déployer notre application sur le téléphone afin de débuguer.

Malheureusement, Ionic a tendance à être mois rapide qu’une application native et parfois il n’est pas simple de trouver un plugin pour utiliser une fonctionnalité spécifique.

Néanmoins, il permet de gagner un temps considérable au développement de notre application et de façon dynamique. C’est ce qu’on va voir et apprendre ensemble dans ce livre.   
 Dans le chapitre suivant, on faire directement de la pratique donc je vous invite à apprendre quelque notion en ces prérequis avant de suivre : HTML/CSS, JS et TypeScript, Angular, npm, VSCode, ligne de commande.

# Le projet « to-do-list »

## Ce que nous-devons comprendre

Il ne faut pas avoir peur, car si vous avez des minimums de connaissance des prérequis précédemment mentionnés, ce chapitre va vous amuser car c’est juste un avant-gout de cet framework. Donc ne vous inquiétez pas, on va procéder pas à pas.

Pour la réalisation de ce projet, il faut savoir :

1. Créer un projet Ionic

* D’abord, installer nodeJS (<https://nodejs.org/en/download>)
* Installer ionic depuis la ligne de commande : npm install –g ionic
* Créer un projet ionic : ionic start nomProjet
* Compléter les configurations
  + Quel framework : angular ou react (angular)
  + Quel template de dapart : tabs, sidemenu ou blank (blank)
* Lancer le serveur ionic (dans le dossier nomProjet) : ionic serve

1. Ce qui est un composant Ionic

* Ce sont des balises ou directive qui sont propre à ionic, se nomment avec « ion-… » au début, par exemple :
  + <ion-toolbar> : créer la page avec un ensemble de fonctionnalité (ion-header, ion content, ion-footer,…)
  + <ion-title> : insérer un titre à la toolbar
  + <ion-card> : organiser les données d’une page
  + <ion-button> : créer un bouton

1. Lire la documentation

Ça ne sert à rien d’apprendre les composants par cœur mais il faut bien savoir les utiliser. Et pour se faire, y a pas mieux qu’une documentation très bien fournit et très explicite.

La documention officielle de Ionic est se truve sur l’adresse « <http://ionicframework.com/docs/> »

1. Incorporer du style (CSS)

C’est ce qui va nous permettre d’avoir une vue plus beau en design.

Par exemple :

* + Les écarts de remplissage (padding)
  + Le type d’affichage (display)
  + Taille de fonte de texte (font-size)
  + Couleur de fond (background-color)
  + Espacement entre les marges (margin)

Le projet est dans ce lien git : …

## Création de notre premier composant

* Il faut créer un dossier dans src/app pour notre nouveau composant (ex : user)
* Et dans le dossier, on crée des fichiers (html, scss et ts) qui suit cette convention de nom : user.component.ts
* Dans user.component.html : (template)
  + On a un html classique (div,…) avec les directives propre en ionic (ion-content, …)
* Dans user.component.scss : (style)
  + On a les style css (background-color,…) classique et les styles propre en ionic(--ion-background-color,…)
* Dans user.component.ts : (contrôleur)
  + On export la classe : export class UserComponent{ }
  + On doit met le décorateur component pour que angular le traite comme un composant :

@Component({

selector : ‘app-user’, // le balise permettant de l’utiliser dans app-root (app.component.html)

templateUrl : ‘user.component.html’, // lier avec son template

styleUrls : [‘user.component.scss’] // lier avec ses styles

}) // sans « ; »

* + Importer les modules correspondant :

import { Component } from ‘angular/core’ ;

* Dans src/app/app.module.ts :
  + Mettre dans le tableau ‘declaration’ le nom du composant avant de l’utiliser : declaration : [AppComponent, UserComponent],
  + bien importer : import { UserComponent } from ‘./user/user.component’ ;

## Les composants Ionics

Ce sont des balises utilisé comme un élément HTML classique à l’emplacement ou on souhaitait l’utiliser. Ils peuvent avoir des attributs avec des propriétés qui dépendront du composant qu’on utilise (voir documentation) et peuvent aussi émettre des évènements.

NB : Ces composants ionics peuvent fonctionner même sans framework (sans angular mais juste un html simple). Il faut juste inclure un CDN qui permet d’utiliser Ionic sans framework (depuis la documentation officielle)

## Utilisation des composants Ionics

Il faut toujours se référer dans la documenation. Dans « get started », on a à droite des noms de composants Ionics qu’on peut utiliser.

Par exemple :

* ion-app : la doc nous informe que <ion-app> doit envelopper notre application ionic dans son intégralité ce qui garantit des styles et comportements prédéfini. (Pour les frameworks single page, on ne doit l’utiliser qu’une seule fois)
* ion-header : mettre un entête à notre application qui se combine bien avec ion-toolbar
* ion-toolbar : généralement placé dans ion-header et ion-footer pour le fixé dans leur position respective.
* ion-title : généralement placé dans ion-header dans ion-toolbar pour mettre un titre à l’entête de l’application ou de page.
* ion-content : C’est le corps de l’application.

Quand on clique sur un composant à droite, toute la description et utilisation de ce composant s’affiche à gauche. Comme :

* + comment l’utiliser (en angular, en js,…)
  + les contenus qu’on peut mettre dedans
  + les attributs et propriétés qu’on peut y associé
  + un exemple d’utilisation
  + un affichage simulé dans un téléphone

## Les catégories des components

* Output : qui nous aide à la sortie des données, qui affiche qlq chose à l’utilisateur
  + ion-img
  + ion-badge
  + ion-loading
  + ion-label
  + ion-title
  + ion-thumbnail
  + ion-toolbar
  + ion-alert
  + ion-toast
  + ion-modal, …
* Layout : qui nous aide à la mise en page, à structurer le contenu
  + ion-grid
  + ion-row
  + ion-col
  + ion-list
  + ion-card
  + ion-infinite-scroll
  + ion-tabs, …
* Input : qui nous aide à récupérer les entrées de l’utilisateur
  + ion-button
  + ion-input
  + ion-textarea
  + ion-menu
  + ion-select
  + ion-datetime
  + ion-fab
  + ion-toggle, …

## Utilisation de « ion-grid »

(Voir documentation)

C’est un composant qui va nous permettre à faire des grilles invisibles (comme des Css-grid) qui divise l’écran en 12 colonnes égales pour redimensionner des Layout qu’on a vu précédemment.

Pour bien utiliser le « ion-grid », on doit construire des lignes (ion-row) et des colonnes (ion-col) et mettre les layouts dedans comme ceci :

<ion-content> // dans le corps  
 <ion-grid> // mettre des grilles  
 <ion-row> // d’abord, construire une ligne  
 <ion-col size= ‘’6‘’ offset= ‘‘3’’ > // ensuite une colonne est sa taille (1 à 12)  
 // offset : le marge à gauche  
 <ion-card> // enfin utiliser un layout  
 <ion-card-title>…

On peut définir le « size » par la grandeur de l’écran :

* size-xs= « 6 » : prend la moitié de l’écran extra-small (très petite écran)
* size-md= « 6 » : prend la moitié de l’écran medium (smartphone moyen)
* size-lg= « 6 » : prend la moitié de l’écran large (tablette)
* size-xl= « 6 » : prend la moitié de l’écran extra-large (ordi)

## Utilisation de « ion-icon »

Ionic possède ses propres icônes, il suffit de cliquer sur le composant ion-icon dans la documentation et cliquer sur l’icône qu’on veut utiliser pour avoir le code d’utilisation.

Ex : <ion-button fill=’’outline’’ > // fill :outline => enlever la couleur de font du btn   
 <ion-icon name= ‘’trash-outline’’ slot= ‘’start’’> </ion-icon>  
 // fill : remplissage, slot : alignement  
 </ion-button>