



# Ionic Framework

SUPPORT DE COURS PAR NICOLAS FAZIO

# 1. Introduction



## 1.1. C'est quoi Ionic??

Ionic est un framework open-source créé en 2013 par Max Lynch, Ben Sperry, et Adam Bradley. Il permet de développer des applications Cross-Platforms en utilisant des outils de développement et des technologies proche de celles que l'on utilise pour le web (HTML, SCSS, JS).

Cross-Platforms???

Oui! Ionic permet de créer des applications qui ont la spécificité de pouvoir fonctionner, donc être installé, sur quasiment n'importe quel type support informatique muni d'une interface graphique et d'outils pour interagir avec elle.

On peut même imaginer développer des applications pour piloter les appareils ménagers de son appartement, les lumières ou autres éléments d'un logement, les jouets des ces enfants ...

Link: <https://github.com/ionic-team>

Link: <https://codeburst.io/i-saw-a-dinosaur-or-how-i-built-a-real-life-version-of-chrome-t-rex-game-92175f6cad2c>

## 1.2. L'écosystème : Ionic Core, Ionic Angular, Ionic React

Il existe trois versions distinctes de Ionic qui sont disponibles, elles sont plus ou moins compatibles entre elles :

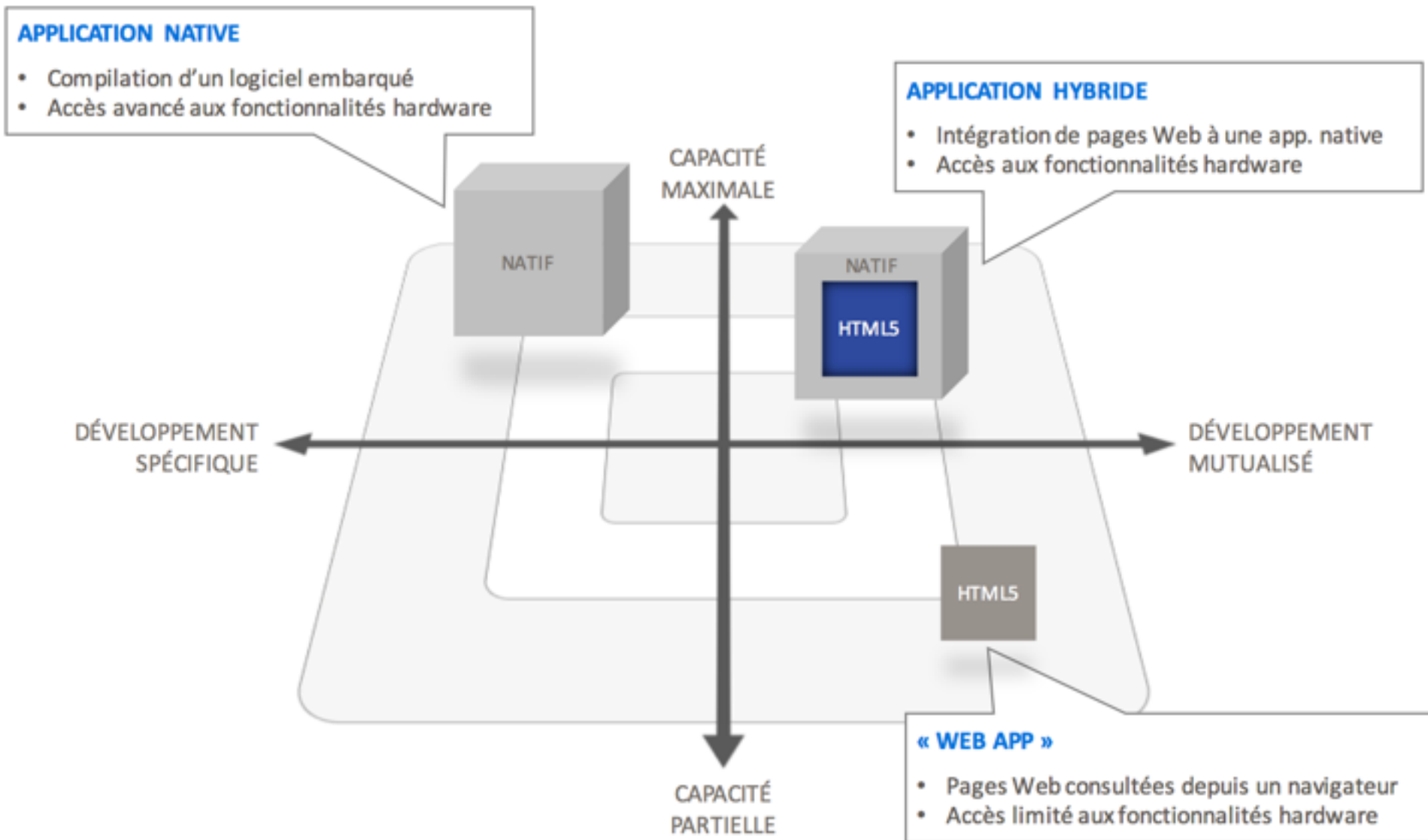
- Ionic V1: se base sur **AngularJS**
- Ionic V2 / V3: se base sur **Angular 4 et TypeScript**.
- Ionic V4: se base sur les **Web Component** et est utilisable avec n'importe quel framework

La dernière version de Ionic (v4) n'est pas encore officiellement release, mais elle est déjà disponible pour les développeurs. Cette version utilise une approche complètement différente de celle qu'ils avaient jusqu'alors adoptée.

En effet, Ionic V4 se compose désormais en plusieurs éléments:

- **Ionic Core** (cœur du framework)
- **Ionic Angular** (wrapper pour Angular)
- **Ionic React** (wrapper pour React)

Cela permet de pouvoir utiliser Ionic plus facilement qu'avant car il s'intègre à présent parfaitement à votre environnement de développement sans être obligé de réapprendre toute la logique d'un framework.



L'écosystème ne serait pas complet si nous n'avions pas les outils permettant de « builder » notre application sur les divers platforms.

Voici les autres outils:

- **Angular 6** Framework pour programmer des applications web et mobile-web
- **Typescript** Language permettant de typer le code
- **SCSS** Outil permettant d'étendre les fonctionnalités CSS par la compilation
- **Capacitor** Outils permettant d'accéder aux ressources natives de l'OS et de « builder » des applications en mode Cross-Platform
- **Stencil** Framework permettant de créer des WebComponent Cross-Platform

## 1.3. Quel usage ? Natif vs Hybrid vs Cross-Platform vs PWA

Pourquoi utiliser Ionic plutôt que de développer des applications en utilisant le langage cible du device (natif) comme Java ou Objectif C ?

Les applications Natif sont réputées plus performantes et plus fiables que celles codées en HTML, SCSS, JS (hybrid)

C'est quoi la différence entre Natif et Hybrid, Multiplatform, Cross-Platform et PWA ???

- **Natif** application développée dans un langage spécifique à un OS (Android => Java; IOS => Objectif C)
- **Hybrid** Application basée sur un contenu web et encapsulé dans une « sous-couche » qui s'adapte aux différent OS
- **Cross-Platform** Framework permettant de concevoir un socle commun pour des applications sur divers OS
- **PWA** Application similaire au Hybrid mais destinée à être distribuée par le web, offrant la possibilité d'être installée sur les appareil mobile.

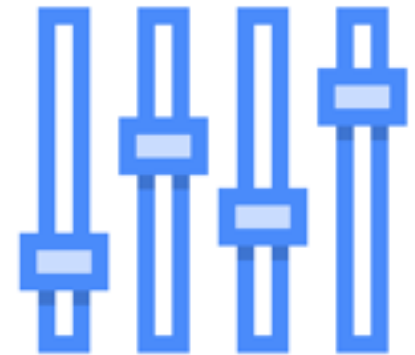
Attribute	Native Approach	Cross Platform Hybrid
Developer Skill Set Needed	Objective-C, iOS SDK, Java, Android SDK	HTML, CSS, Javascript
Distribution Method	App Stores	App Stores Desktop Browser Desktop App (e.g. Electron) Mobile Browser Progressive Web App
Speed to Develop	Slow	Fast
Development Cost	High	Low
Maintenance Cost	High	Moderate/low
Graphical Performance	High	Moderate
App Performance	High	Driven by use case
Access to native functionality	Full native library	Full native library (requires third-party plugins)
UX consistency across platforms and devices	Requires separate apps	Yes



## 1.4. Les outils Ionic

- Ionic CLI
- Ionic Lab
- Ionic Creator
- Ionic View App
- Ionic Deploy
- Ionic Package
- Ionic Monitor
- Ionic Icons

## 2. Installation de Ionic + configuration des postes de travail



## 2.1.Installation de l'environnement de développement

- <https://github.com/creationix/nvm>
- \$ nvm install 9
- \$ npm install -g typescript

## 2.2.Installation de Ionic dans l'écosystème Angular

- \$ npm install -g @angular/cli
- ng new MyApp --style=scss
- \$ npm install --save @ionic/angular@latest @ionic/schematics-angular@latest
- \$ npm install --save @ionic-native/core@beta @ionic-native/splash-screen@beta @ionic-native/status-bar@beta

## 2.3.Installation de Capacitor

- \$ npm install --save @capacitor/core @capacitor/cli

## 2.4.Installation de Ionic

- \$ npm install -g ionic@canary
- \$ ionic start MyApp

## 2.5.Créer un projet Ionic avec Angular

Pour démarrer un projet Ionic V4, il y a deux possibilités:

- Utiliser Ionic CLI et créer un nouveau projet avec ``$ ionic start myApp``
- Ajouter Ionic à un projet Angular 6. Créer un projet avec Angular CLI: ``$ ng new myApp`` puis importer Ionic dans le projet via la configuration du fichier ``angular.json`` puis ajouter tous les autres éléments de configuration.

## 2.6. Lancer un projet dans le navigateur d'un serveur local de développement

Pour démarrer le mode de développement, il suffit de faire comme vous avez l'habitude avec un projet Angular: ``$ ng serve`` ou ``$ ionic serve``

Pour créer un build de production: ``$ ng build`` ou ``$ ionic build``



## 2.7. Structure d'une application Angular avec Ionic

La structure d'un projet Angular avec Ionic n'est pas très différente de celle d'un projet standard. Il y a que quelques fichiers à ajouter et d'autres qui doivent être modifiés.

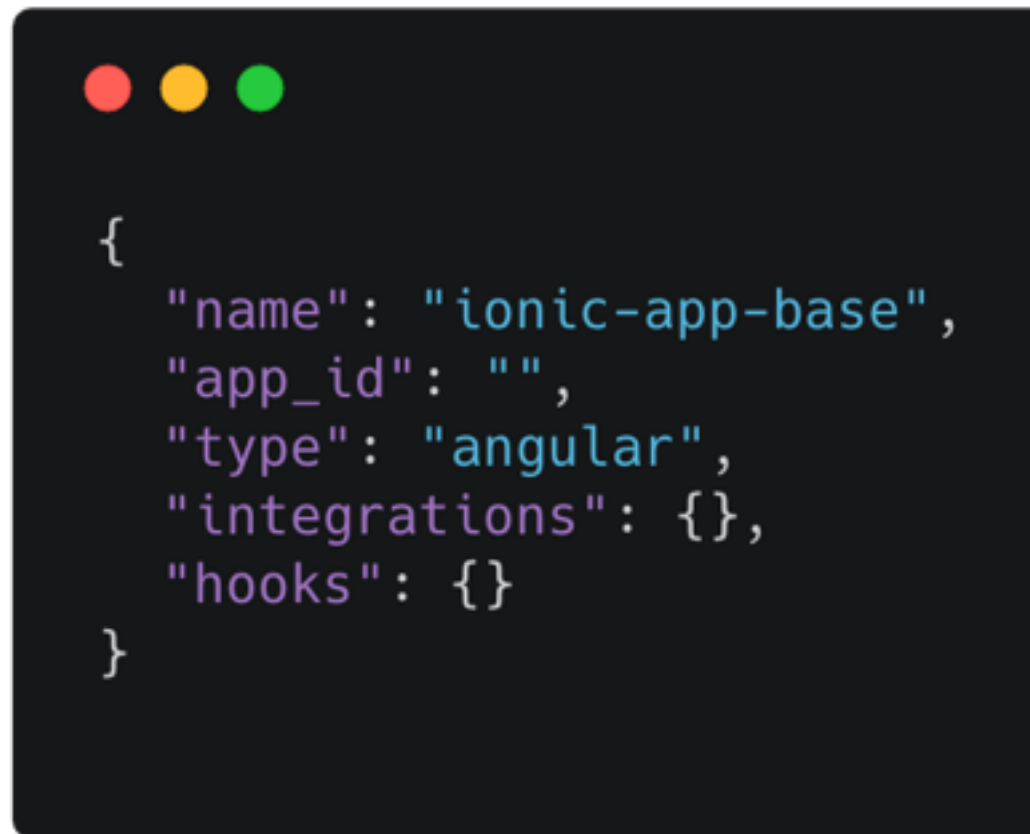
Voici déjà les modifications à apporter à votre projet Angular 6.

- Fichier `./angular.json`



```
"assets": [  
  "src/favicon.ico",  
  "src/assets",  
  {  
    "glob": "**/*",  
    "input": "node_modules/@ionic/angular/dist/ionic",  
    "output": "/ionic"  
  }  
]
```

- Fichier `../ionic.config.json`



```
{  
  "name": "ionic-app-base",  
  "app_id": "",  
  "type": "angular",  
  "integrations": {},  
  "hooks": {}  
}
```

- Fichier `./src/app/app.module.ts`

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { RouterModule, RouteReuseStrategy, Routes } from '@angular/router';

import { IonicModule, IonicRouteStrategy } from '@ionic/angular';
import { SplashScreen } from '@ionic-native/splash-screen/ngx';
import { StatusBar } from '@ionic-native/status-bar/ngx';

import { AppComponent } from './app.component';
import { AppRoutingModuleModule } from './app-routing.module';

@NgModule({
  declarations: [AppComponent],
  entryComponents: [],
  imports: [BrowserModule, IonicModule.forRoot(), AppRoutingModuleModule],
  providers: [
    StatusBar,
    SplashScreen,
    { provide: RouteReuseStrategy, useClass: IonicRouteStrategy }
  ],
  bootstrap: [AppComponent]
})
export class AppModule {}
```

- Fichier `./src/app/app.routing.ts`

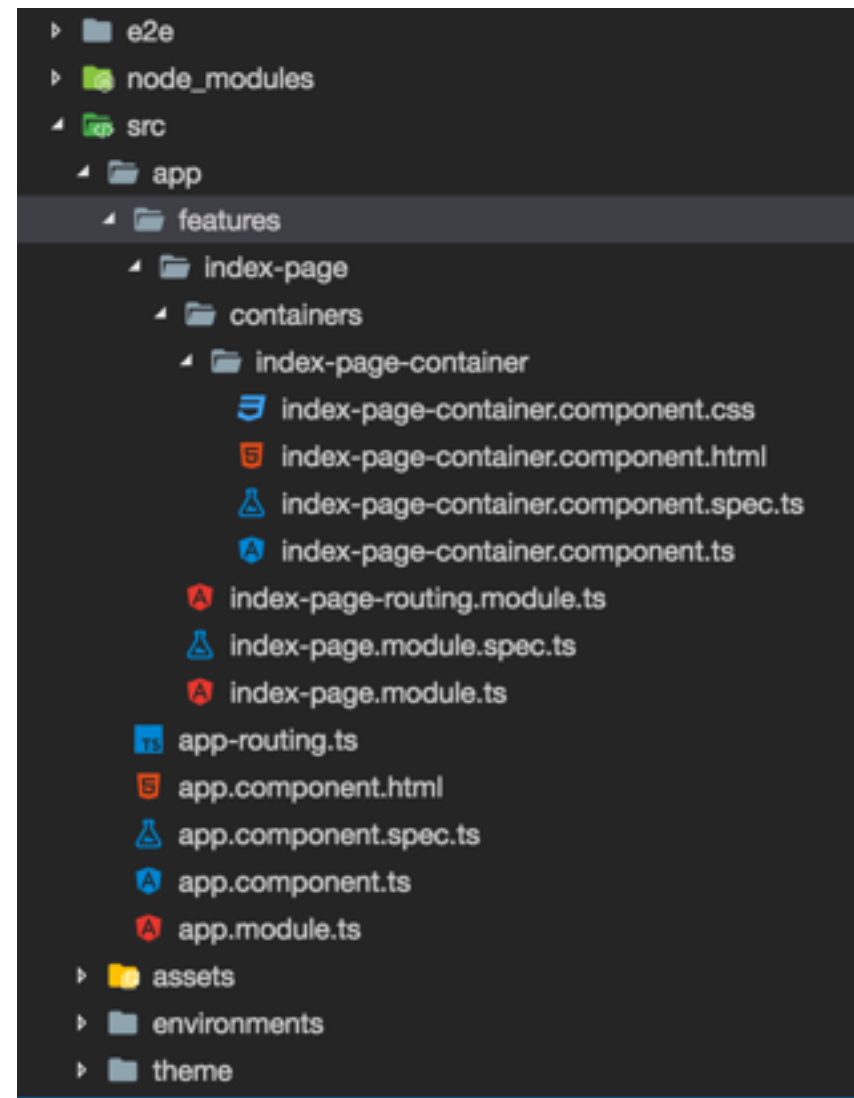


```
import { ModuleWithProviders } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

const routes: Routes = [];

// create lightway module routing
export const routing: ModuleWithProviders = RouterModule.forRoot(routes);
```

La structure des dossiers est légèrement différentes que celle d'Angular. Libre à vous de vous y faire mais n'oubliez pas que vous être libre d'organiser votre projet comme bon vous semble.  
Je vous conseil de choisir une structure Angular Pattern Design.



### 3. Présentation des composants Ionic/Stencil



## 3.1.UI Component overview

Ionic permet de créer de vue HTML très facilement en proposant à la manière d'une librairie CSS, un large choix de composant et de pattern design pour concevoir votre application.

Doc Ionic v3: <https://ionicframework.com/docs/components/>

Doc Ionic v4: <https://github.com/ionic-team/ionic/tree/master/core/src/components>

Dev Resources: <https://ionicframework.com/docs/developer-resources/>

## 3.2. Documentation des composants, ionic icons

Ionic met également à disposition une bibliothèque d'icons

Doc: <https://ionicframework.com/docs/ionicons/>



## 3.3.Navigation

La navigation de page en page se fait comme pour Angular

Doc: <https://angular.io/guide/router>

## 3.4. Gestures

Les événements DOM ionic:

- click
- press
- swipe
- pane
- pinch
- rotate