



Framework Mobile

Ionic cross-platform

Ionic qu'est ce que c'est ?

IONIC EST UN FRAMEWORK EN PARTIE OPEN-SOURCE

Permettant de construire des applications natives en utilisant les technologies WEB.

Par natives on doit comprendre : disponibles via les Stores et pouvant utiliser les fonctionnalités natives des OS. Créé en 2013, et après presque 12 000 commits et 128 releases différentes, il en est à sa version stable "4"

Sous le capot, on va construire un site web mobile que l'on va packager dans une application Cordova.

Actuellement ces plateformes sont supportées :

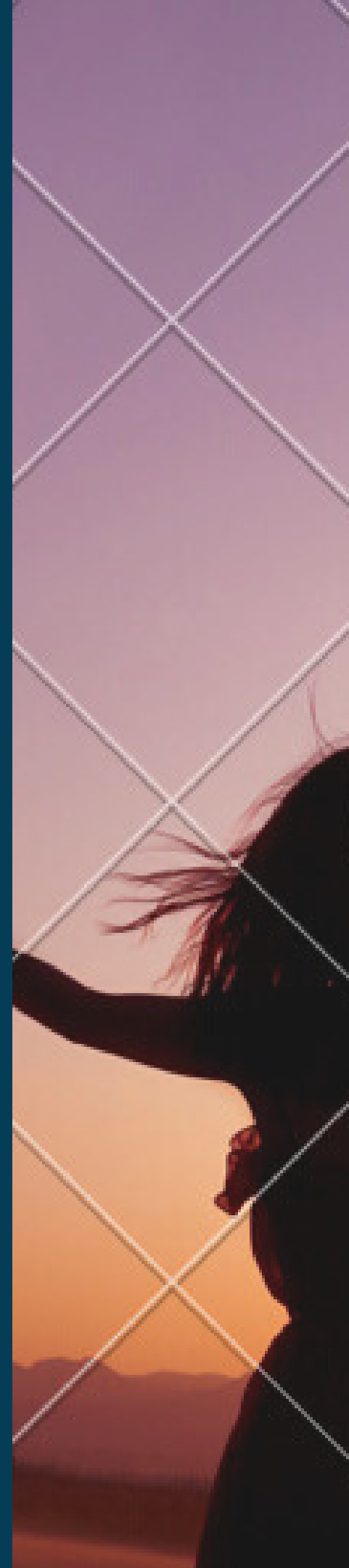
iOS 8.0+

Android 4.4+

UWP apps

L'objectif de Ionic est de faciliter ce processus de plusieurs manières :

- Utiliser les derniers outils de développement pour rendre cela fun.
- Fournir un framework graphique clef en main.
- Faciliter l'intégration avec Cordova et ses plugins.
- Proposer une série d'outils "pros" (mais payants).



LES OUTILS DE DÉVELOPPEMENT DU WEB

Le développement Ionic doit être fun et cela est réalisé (en tout cas prôné de cette manière par ses créateurs)

via l'utilisation de dernières versions des frameworks web les plus connus.

Nous sommes donc sur un projet Angular 5.X couplé à du typescript pour nous assurer d'une bonne qualité de code. Les feuilles de style, elles, sont générées via SASS. Il faudra donc s'assurer d'avoir la dernière version de node.js sur sa machine (par ici pour le télécharger).

Pour faciliter votre expérience de développeur, ionic propose une CLI très similaire à celle que d'angular : la ionic-cli que l'on installe sur sa machine avec cette ligne de commande :

```
1 | npm install -g ionic
```



Pour créer un projet vide nommé "myApp", on utilise la ligne de commande suivante :

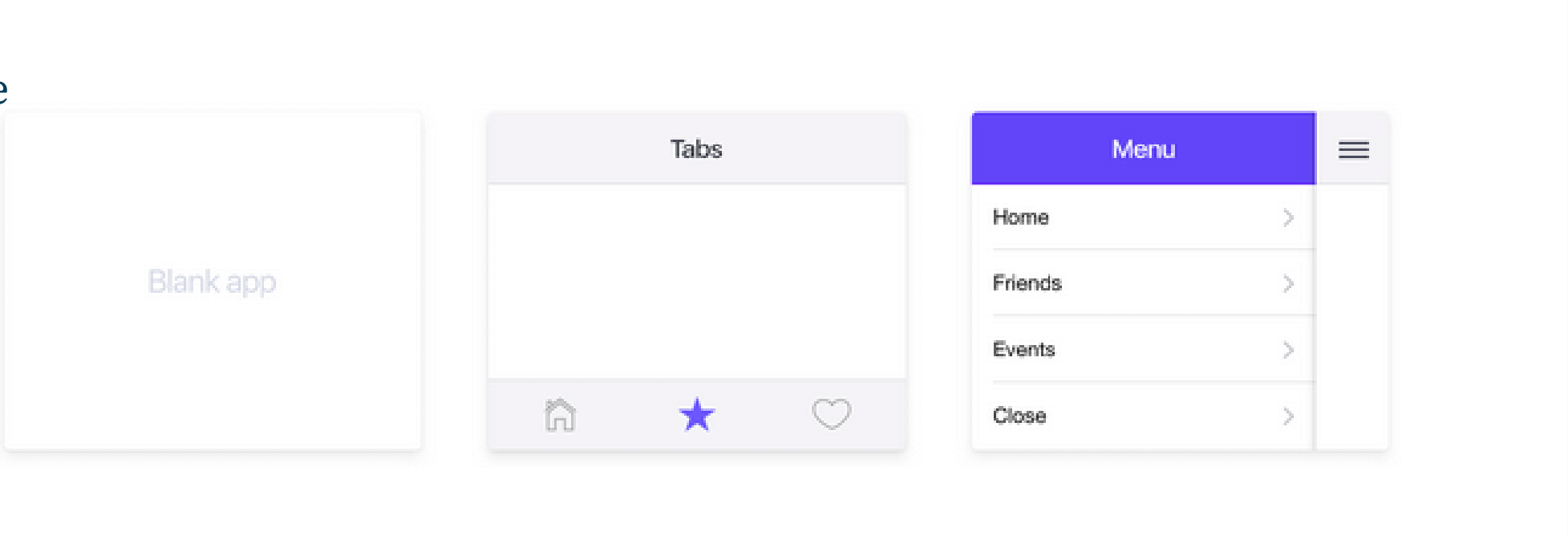
```
1 | ionic start myApp blank
```

Pour lancer votre projet pour la première fois, on va encore une fois utiliser une commande "classique" pour un développeur web : serve. Le navigateur par défaut de votre machine de développement va alors s'ouvrir sur votre projet.

```
[16:34:55] dev server running: http://localhost:8100/  
  
[OK] Development server running!  
Local: http://localhost:8100  
External: http://192.168.1.17:8100, http://169.254.117.108:8100
```

La création d'un projet se fait elle aussi en une seule ligne de commande.

Ionic vous propose de partir d'un projet "vide", d'un projet de type "tab bar" (navigation à base d'onglets), d'un projet de type "hamburger menu", ou d'autres types d'applications plus complets encore listés sur cette page.





Un framework graphique

À cet instant, nous avons maintenant un site web "classique" et l'idée est de le développer dans l'optique d'un affichage "mobile".

La puissance de ionic est de nous proposer une librairie complète de composants graphiques à utiliser dans notre application.

Ces composants auront plusieurs avantages non négligeables lorsque l'on connaît la difficulté de maintenir ce genre d'éléments sur une gamme de navigateurs :

- Leur design correspond automatiquement à celui "natif" de la plateforme d'exécution.

Un toggle switch aura un "design" Android sous Android et un design "ios" sur un iphone. À ce niveau il sera difficile de faire la différence entre du natif et du ionic sur un simple coup d'oeil.

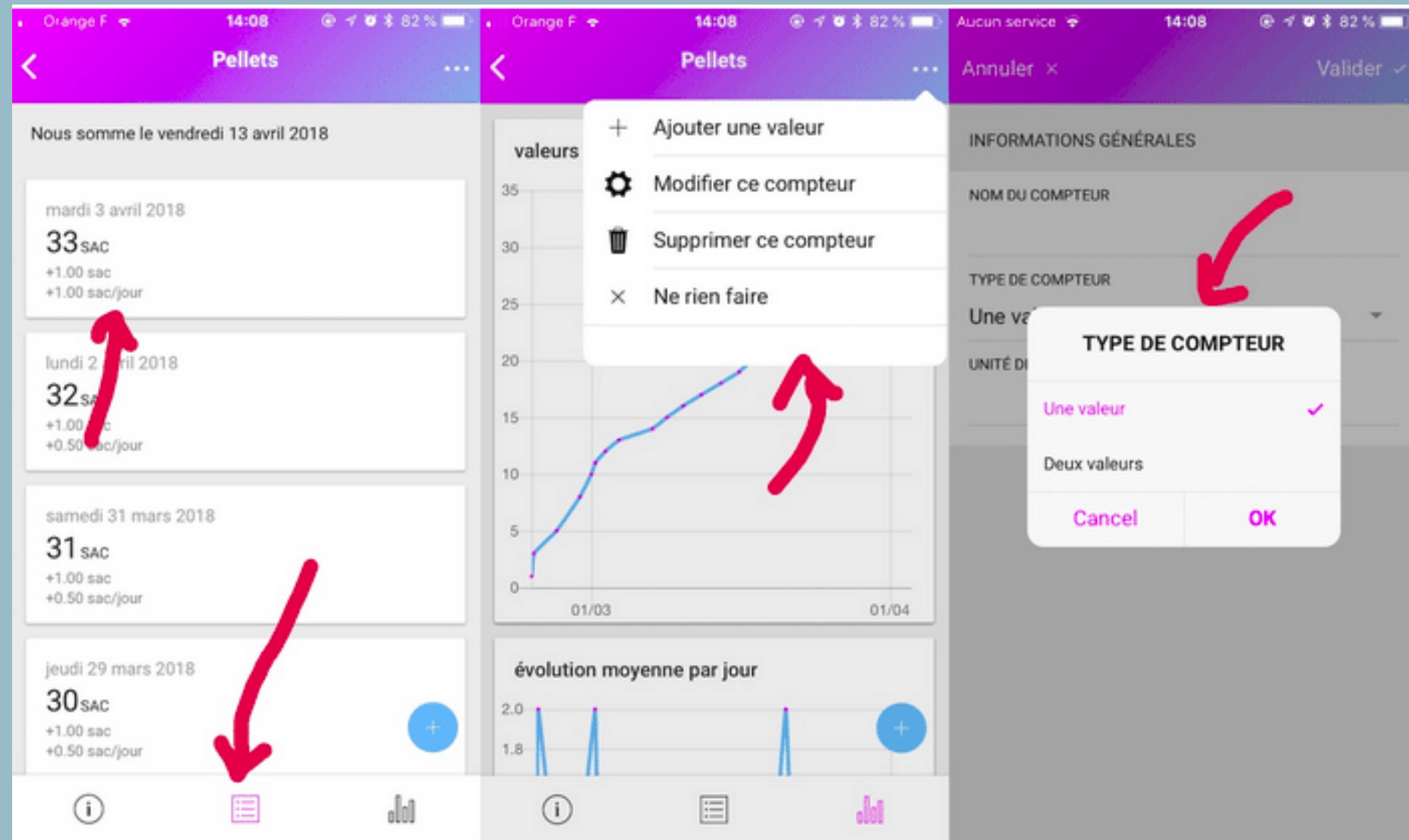
- Leur comportement est aussi celui de la plateforme d'exécution.

L'intégration avec la plateforme est complète : il est par exemple possible de naviguer vers la page précédente avec un slide depuis le bord gauche de l'écran sur ios (la gesture classique sur cette plateforme).

- Modernes et à jour : certains composants que l'on voit dernièrement dans les tendances designs sont disponibles. Les cards de matériel en sont un très bon exemple.

- Personnalisables : nous sommes sur du développement web et de base ces composants sont donc facilement personnalisables à coup de CSS mais les composants sont pensés personnalisables dans la façon dont ils sont construits.

VOICI QUELQUES EXEMPLES DE CES CONTRÔLES MISE EN OEUVRE DANS UNE APPLICATION :



Chacun des composants que l'on va utiliser sera sous le capot un composant angular. Il sera ainsi possible de les configurer via des propriétés (@Input) et ils exposeront aussi des événements correspondant à ce que l'on attend d'eux.

Un framework de layout

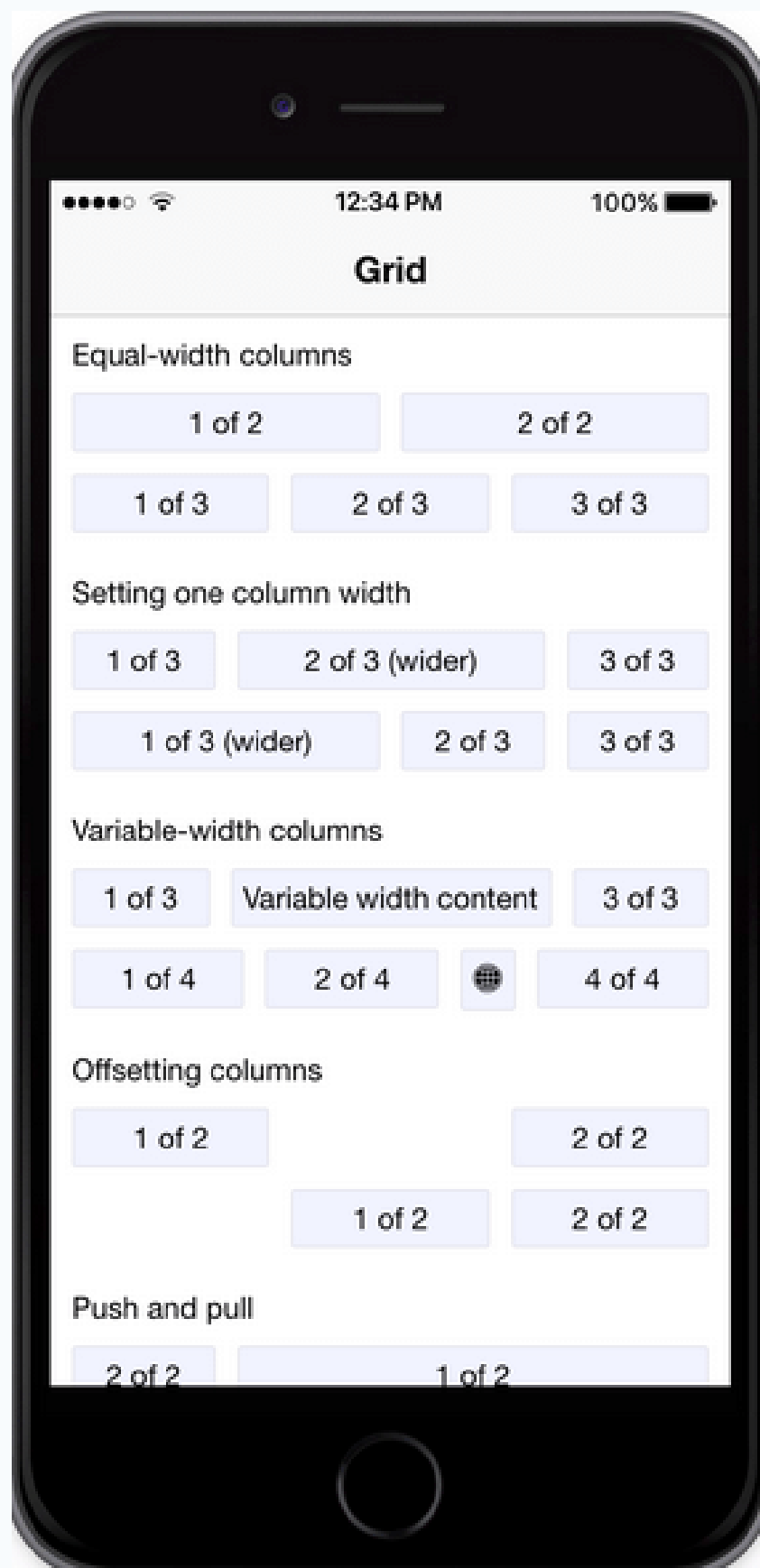
Pour nous faciliter la vie, Ionic propose aussi un système de disposition par grille qui ressemble assez à ce dont on peut avoir l'habitude avec bootstrap. Rien ne nous oblige de passer par lui mais cela devient simple de créer des layouts à colonne/lignes.

Plus d'informations sont disponibles dans la documentation :
<https://ionicframework.com/docs/components/#grid>.

Un framework de navigation

Une application mobile c'est aussi une navigation de page en page et ionic propose ainsi son propre modèle de navigation pour nous faciliter la vie.

Tout est basé sur un composant de navigation racine "ion-nav" qui propose les différentes APIs nécessaires à la gestion des pages : affichage, retour en arrière, etc. En général on place le composant ion-nav dans la page principale (comme le fait le template "navigation de base") et on le récupère par injection de dépendances (il est du type NavController) pour accéder à ses APIs.



Un intérêt de ces pages est aussi d'avoir leur propre cycle de vie (en plus de celui de base des composants angular) sous la forme de fonction (ionViewDidLoad, ionViewWillEnter, etc.) et même de conditionner l'affichage/sortie d'une page (ionViewCanEnter, ionViewCanLeave).

```
1 | ionViewDidLoad() {  
2 |     console.log("Chargement de la page");  
3 | }  
4 |  
5 | ionViewCanLeave() {  
6 |     // on ne quitte pas cette page !  
7 |     return Promise.resolve(false);  
8 | }
```

☐ *Basé sur Cordova*

Les applications natives finalement générées sont basées sur Cordova.

Il est donc tout d'abord nécessaire de rajouter une plateforme ciblée en utilisant la CLI ionic encore une fois.

Pour ajouter un projet android on utilisera par exemple :

```
1 | ionic cordova platform add android
```

Ionic Pro

Ionic propose aussi sous forme de services payants des fonctionnalités diverses. En général, ces fonctionnalités sont testables sous forme de version d'essai gratuite.

- **Creator : un éditeur d'interface graphique en ligne pour prototyper rapidement une application.**
- **Ionic view App : une application sur les Stores qui permet d'afficher une application "servie" sur un poste de développement et la tester en conditions réelles.**
- **Ionic deploy : un service de code push.**
- **usine de build.**
- **Monitor : un service de monitoring & analyse d'usage pour vos apps. interface graphique creator**



OSS FRAMEWORK



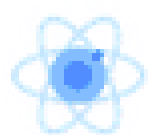
MOBILE DEVOPS



APP BUILDER



INTEGRATIONS



INTRODUCING IONIC REACT

One codebase. Any platform. Now in React

- ✓ 100+ mobile optimized React UI components
- ✓ Standard React tooling with `react-dom`
- ✓ iOS / Android / Electron / PWA

Try Ionic React >

AlbumList.jsx

```
<h4>Discover</h4>
<h2>Music we think you'll love.</h2>
<IonList>
  {albums.map(album =>
    <IonItem key={album.id}>
      <IonThumbnail slot="start">
        <img src={album.artwork} />
      </IonThumbnail>
      <IonLabel>
        <h2>{album.title}</h2>
        <p>{album.artist}, {album.year}</p>
      </IonLabel>
    </IonItem>
  )}
</IonList>
```

Sources :

<https://blogs.infinitesquare.com/posts/web/qu-est-ce-que-c-est-ionic>

<https://ionicframework.com>

