

Formation Mobile Hybride

Ionic Cordova

Qui suis-je?

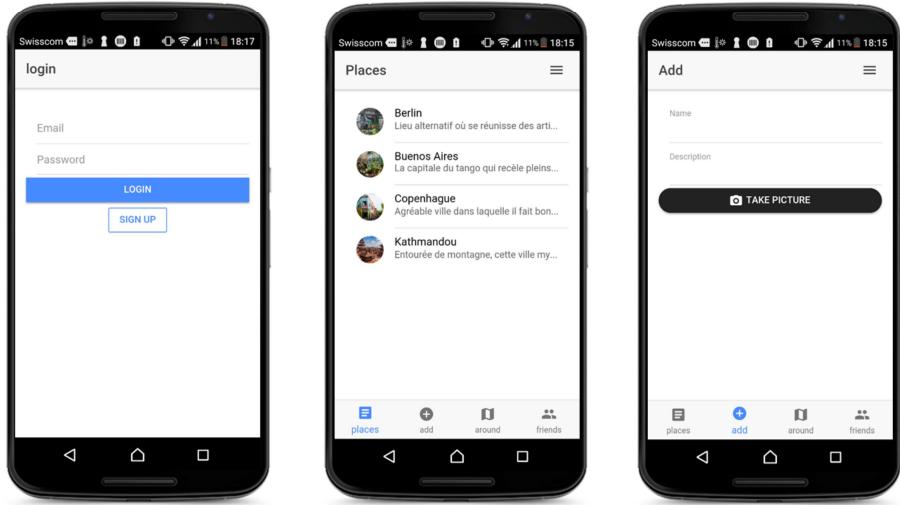
Fredrik Lahode



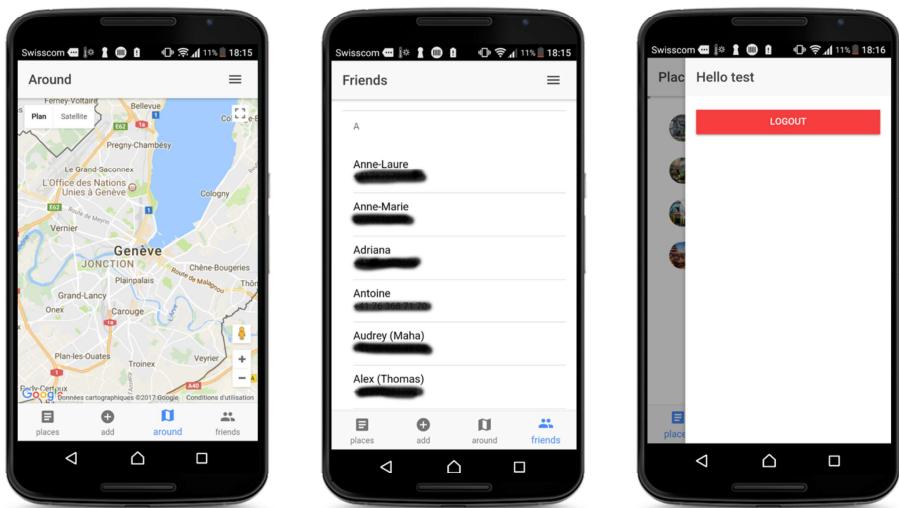
Développeur full stack

Ionic
Angular
TypeScript / Javascript
PHP
Drupal
NodeJS

Objectif



Objectif



Ionic?!?

C'est quoi?
Pour qui?



Ionic?!?

C'est quoi?
Pour qui?

Définition Wikipedia

Ionic est un **framework open-source** créée en 2013 permet de créer un code **multisupport** en utilisant des **outils Web** pour générer des **applications Ios, Android, etc.**



Techno mobile

Site web

App mobile

Responsive
design

App native

Site mobile

App native
cross-platform

Webapp

App hybride

Techno mobile

Site web



Accessible depuis son navigateur web
Fonctionne avec une connexion Internet
Développé avec les langages du web (HTML, CSS, Javascript, PHP, ...)

App mobile



Application pouvant être téléchargée sur les stores (Apple store, Google Play, etc.)
Peut pleinement utiliser les technologies de device
Développé soit en langage natif du téléphone, soit en combinant diverses technologies

Techno mobile

SITE RESPONSIVE DESIGN

Élaboration d'un site offrant une expérience de lecture optimale sur tous les devices.

SITE MOBILE

Version allégée d'un site Internet, on y accède via son navigateur pour une consultation sur appareil mobile (m.tpg.ch)

WEBAPP

Comme un site mobile mais plus complet et axé application (ex. Outils Google)

Techno mobile

APP NATIVE

Application développée dans un langage spécifique à un OS.

iOS et Android => 2 applications distincts Objectif C (iOS) et Java (Android)

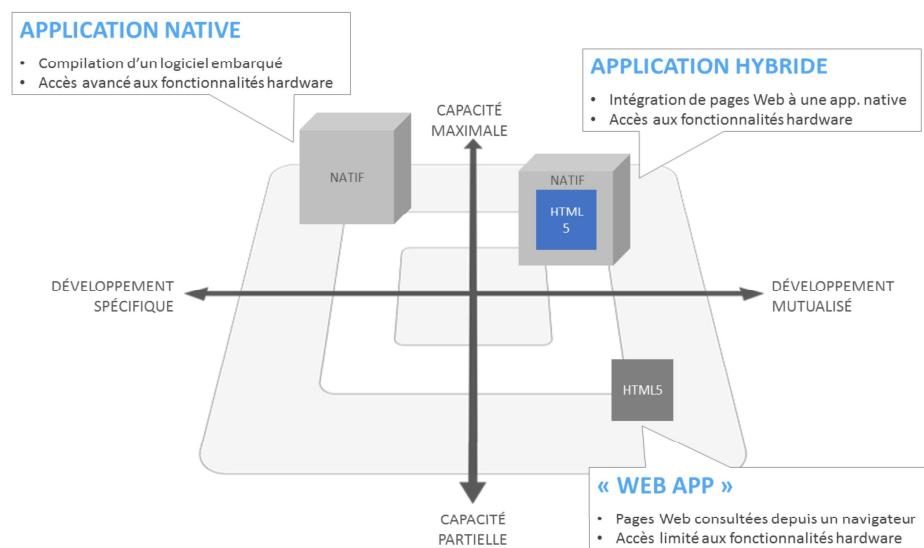
APP NATIVE CROSS-PLATFORM

Framework permettant de concevoir un socle commun aux applications avec du langage C#, Ruby, Javascript, et.c (ex Xamarin, Titanium,...)

APPLICATION HYBRIDE

Basé sur un contenu web et encapsulé dans une « sur-couche » adaptée au différents système.

Que choisir?



Application
native



« je préfère le natif ! »

Application
hybride



« C'est lent mais c'est moins cher »

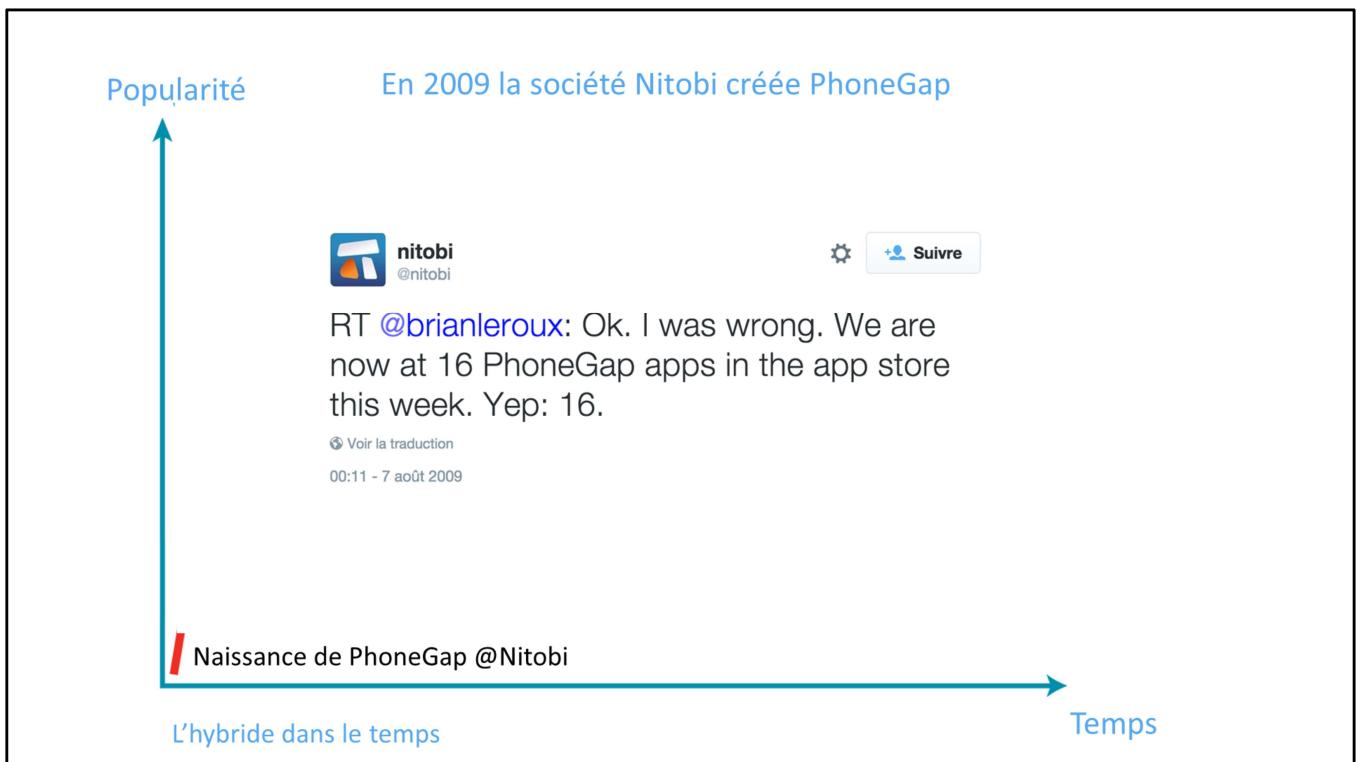
Pourquoi les gens
pensent ça Doc ?

Je trouve ça cool
l'hybride moi...



Monte Marty, je
vais t'expliquer.





Popularité

En 2011 Adobe rachète Nitobi et open source le projet

Rachat de PhoneGap par Adobe



Adobe @Adobe



Suivre

We're proud to welcome [@Nitobi](#) to #Adobe!
Learn more about our recent acquisition:
adobe.ly/rVk7gn #PhoneGap #HTML #CSS
#Javascript

[Voir la traduction](#)

RETWEETS
62

FAVORIS
14



21:23 - 26 oct. 2011

Naissance de PhoneGap @Nitobi

L'hybride dans le temps

Temps

Popularité

En 2012 Facebook abandonne HTML5 pour ses app mobiles

Rachat de PhoneGap par Adobe

Mark Zuckerberg: Our Biggest Mistake Was Betting Too Much On HTML5

Posted Sep 11, 2012 by Drew Olanoff (@drew)



Today, Mark Zuckerberg revealed that Facebook's mobile strategy relied too much on HTML5, rather than native applications.

Naissance de PhoneGap @Nitobi

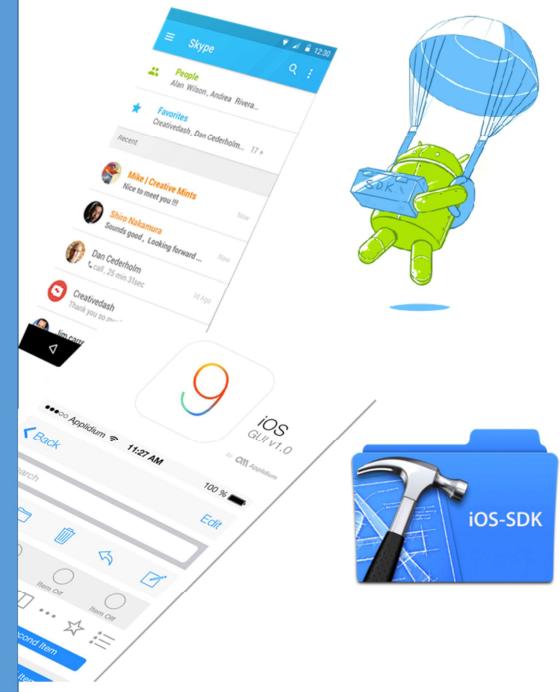
L'hybride dans le temps

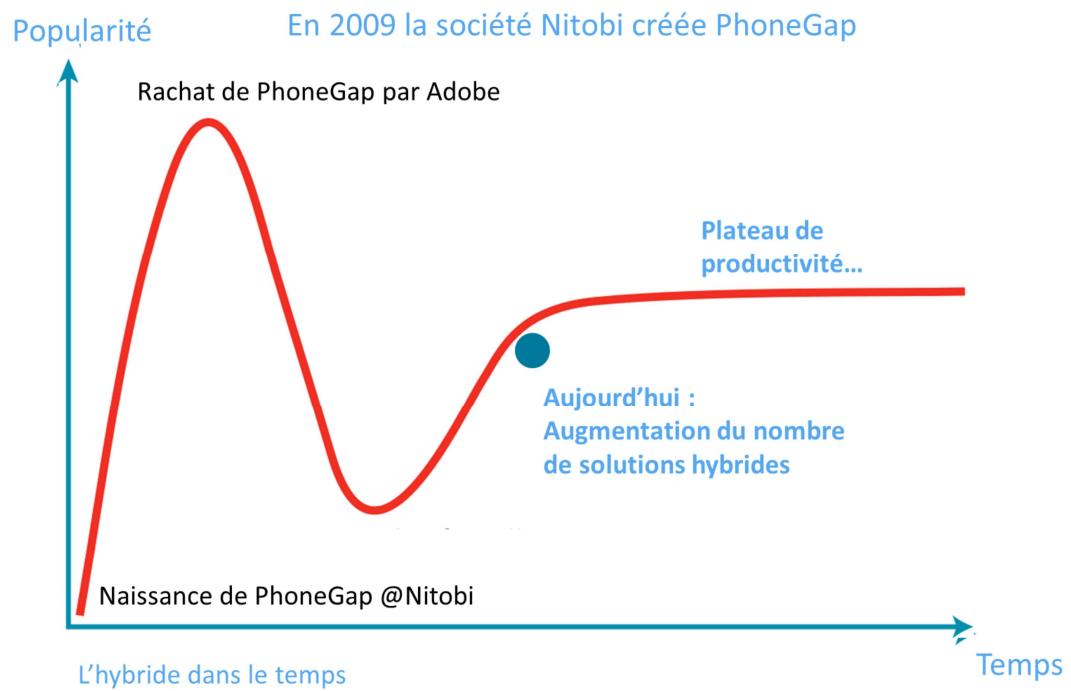
Temps

FIN DE l'HYBRIDE

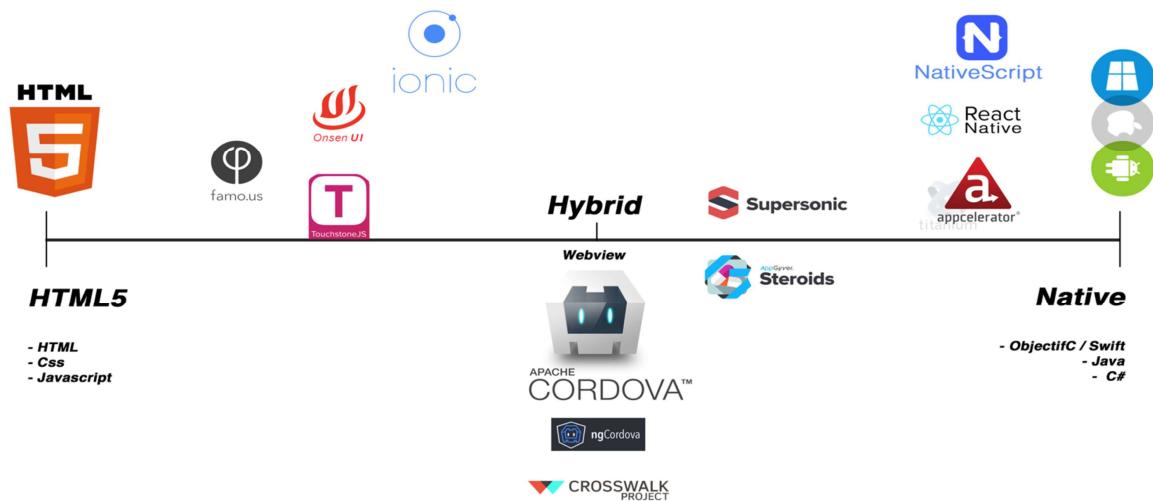
Quel était le problème?

Aucun SDK HTML5
« convenable » pour
mobile





Aperçu des solutions hybrides et alternatives



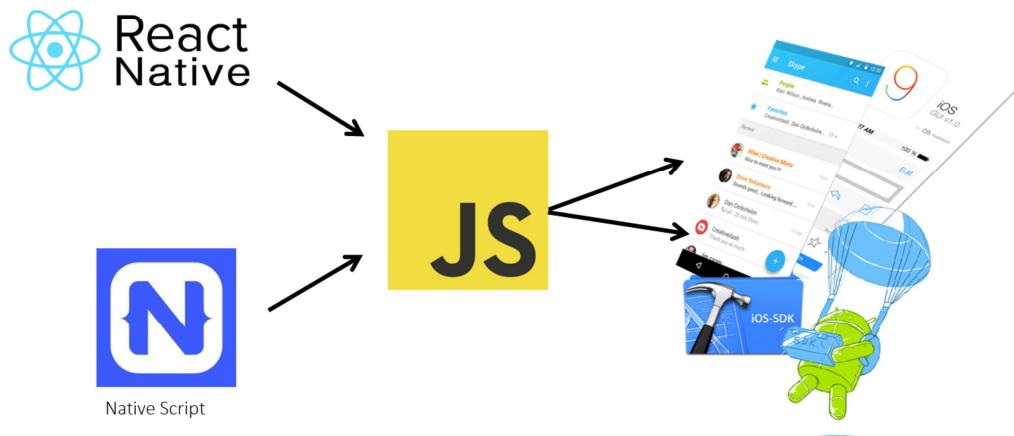


**LE MONDE SE DIVISE EN
DEUX CATÉGORIES...**

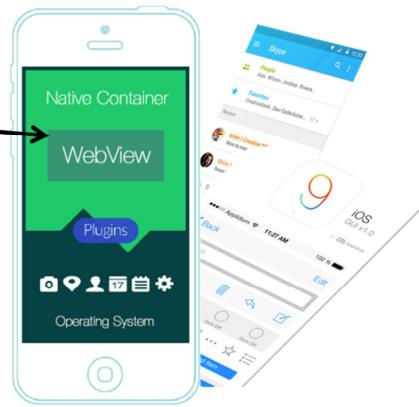
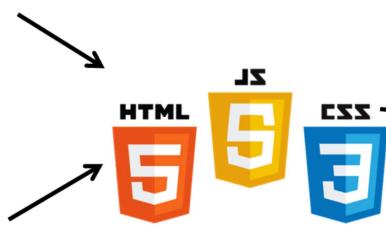
egen.fr



Piloter des composants natifs en JavaScript

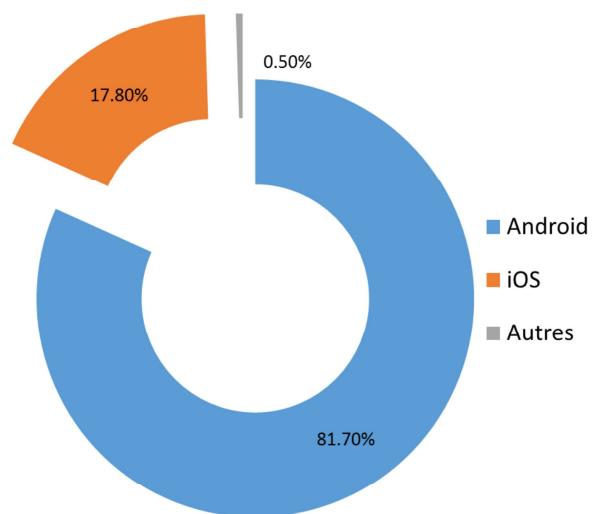


Utiliser HTML5 dans un « navigateur » embarqué



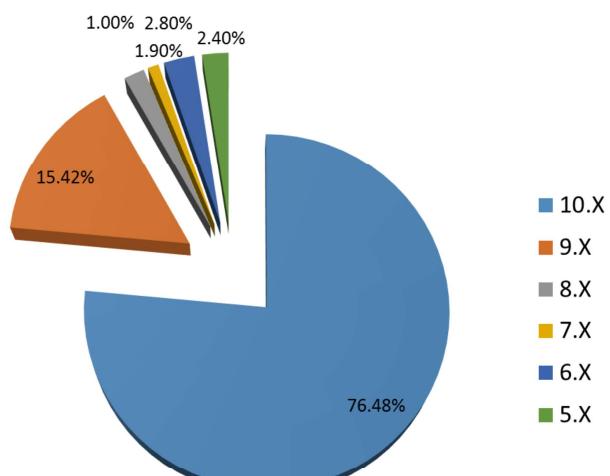
Vision actuelle du marché

Android et iOS dominent le monde



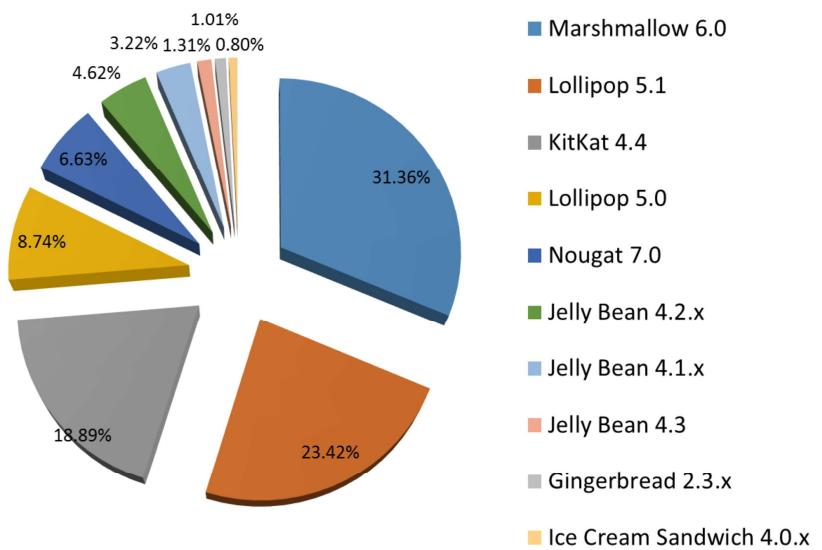
Vision actuelle du marché

Version iOS utilisée



Vision actuelle du marché

Version Android utilisée



Pourquoi Ionic ?

- Documentation complète
- Une large communauté, des tutoriaux et beaucoup de ressources disponibles
- Offre d'emploi sur Ionic
- Après avoir levé \$1 million à ses débuts, la société lève \$2.6 million en 2015 et \$8.5 million en 2016.
- Certaines offres d'emplois parlent d'ionic.
- Basé sur Angular et Typescript
- Ecosystème complet, build, application de test, extensions, etc.

Outils Ionic

Ionic CLI



Ionic View



Ionic deploy



Ionicons



Ionic auth



Ionic push



Ionic native



Ionic package

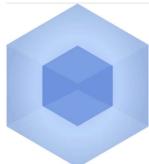


Outils Ionic



Apache Cordova

Plateforme pour construire des applications natives en utilisant les technologies HTML,CSS et JavaScript.



Webpack

Outil de build. il permet de packager les assets javascript, css, images, en des fichiers « bundle » prêt à être insérés côté navigateur.



Angular & Typescript

Framework JavaScript pour programmer des applications Web SPA (Single Page Application)

Sass

Outil permettant d'étendre les fonctionnalités CSS par la compilation.

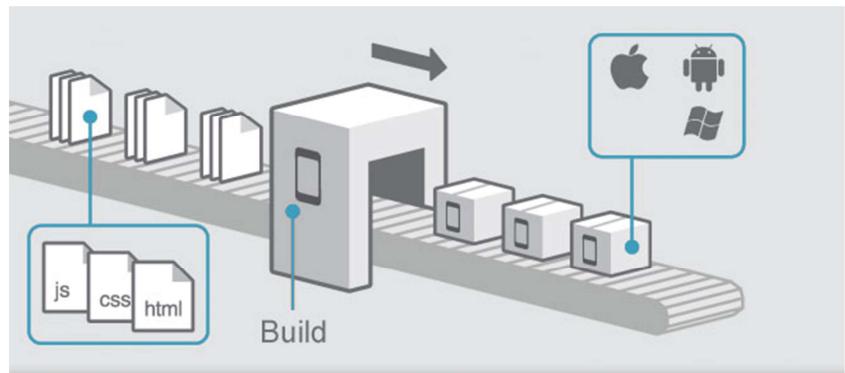
Comment fonctionne Ionic?

Une WebView* dans un conteneur natif



*WebView:
composant natif
pour afficher des
pages web.

Comment fonctionne Ionic?



REVISION

C'EST PARTI !

Installation

Pré-requis NodeJS et Typescript

Installation de nvm

<https://github.com/creationix/nvm>

Installation de node

```
nvm install node
```

Installation de typescript

```
npm install -g typescript
```

Lors de l'installation de node par nvm, il est possible de spécifier la version de node

Ex.: nvm install v7

N.B: Une fois installé, celui-ci switchera automatiquement sur la version nouvellement installée (manuellement, on ferait nvm use v7). Toutefois, cela ne définit pas la version par défaut et lors du redémarrage du terminal, on revient sur l'ancienne version de node par défaut. Pour palier à ceci:

```
nvm alias default v7
```

Installation

Installation de cordova

```
npm install -g cordova
```

Installation d'Ionic

```
npm install -g ionic
```

Vérifier l'installation et la version

```
ionic info
```

```
global packages:
  @ionic/cli-utils : 1.4.0
    Ionic CLI      : 3.4.0

System:
  Node    : v7.8.0
  OS      : OS X El Capitan
  Xcode   : Xcode 7.3.1 Build version 7D1014
  ios-deploy : not installed
  ios-sim   : not installed
  npm      : 4.2.0
```

Quelques commandes

Lister les commandes du CLI

```
ionic help
```

Lister les types de projets par défaut

```
ionic start --list
```

Installer son premier projet « tabs »

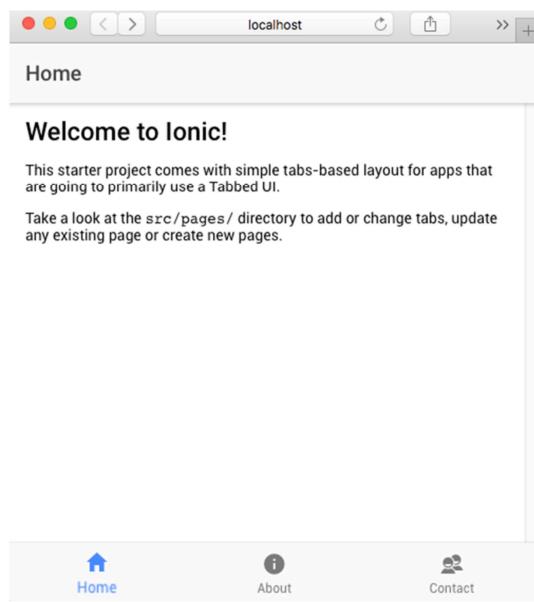
```
ionic start hello-world tabs
```

Lancer son projet sur le navigateur en local

```
ionic serve
```

Quelques commandes

`http://localhost:8100/`



Quelques commandes

[Afficher la documentation de Ionic](#)

`ionic docs`

[Générer des composants, pipes, etc.](#)

`ionic generate`

Structure d'un projet

```
1  <!DOCTYPE html>
2  <html lang="en" dir="ltr">
3  <head>
4      <meta charset="UTF-8">
5      <title>Ionic App</title>
6      <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7      <meta name="format-detection" content="telephone=no">
8      <meta name="msapplication-tap-highlight" content="no">
9
10     <link rel="icon" type="image/x-icon" href="assets/icon/favicon.ico">
11     <link rel="manifest" href="manifest.json">
12     <meta name="theme-color" content="#4e8ef7">
13
14     <!-- cordova.js required for cordova apps -->
15     <script src="cordova.js"></script>
16
17     <!-- un-comment this code to enable service worker -->
18     <script>
19         if ('serviceWorker' in navigator) {
20             navigator.serviceWorker.register('service-worker.js')
21                 .then(() => console.log('service worker installed'))
22                 .catch(err => console.error('Error', err));
23         }
24     </script>-->
25
26     <link href="build/main.css" rel="stylesheet">
27
28     </head>
29     <body>
30
31     <!-- Ionic's root component and where the app will load -->
```

hooks = Contient tous les “hooks” relativ à Cordova. En principe, il n'y a pas besoin d'intervenir dans ce dossier

node_modules = Identique à Angular, celui contient tous les modules “contrib” du projet, y compris angular et ionic lui-même

resources = Contient les icônes et splashscreen pour les différents mobiles (par défaut ios et android)

src = Contient toutes les sources du projet

app = Dossier par défaut de l'application

assets = Emplacement où sont stockés les médias ou fichiers json appelés par le projet

pages = Pages de l'application

theme = Fichier de style du projet

www = Contient le build du projet

Autres fichiers importants:

config.xml

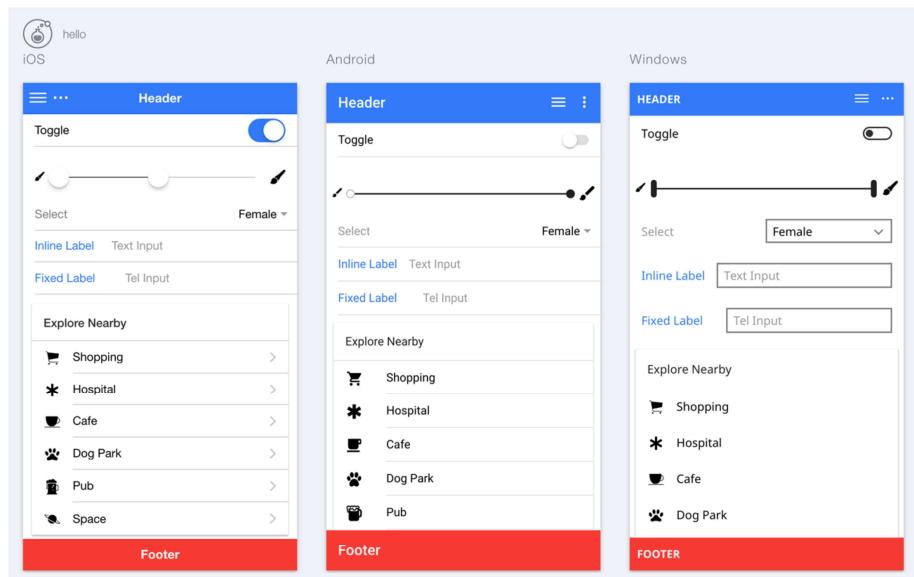
Composants Ionic

<https://ionicframework.com/docs/components>

Pour générer un nouveau composant « custom »

```
ionic generate component
```

Composants Ionic



Pour sélectionner ses icônes: <http://ionicons.com>

Pages et
Navigation

Ionic 2 navigation



Angular 2 Router

Structure d'une page

home.ts

```
import { Component } from '@angular/core';
import { NavController } from 'ionic-angular';

@Component({
  selector: 'page-home',
  templateUrl: 'home.html'
})
export class HomePage {

  constructor(public navCtrl: NavController) {
  }
}
```

home.html

```
<ion-header>
<ion-navbar>
  <button ion-button menuToggle>
    <ion-icon name="menu"></ion-icon>
  </button>
  <ion-title>Home</ion-title>
</ion-navbar>
</ion-header>

<ion-content padding>
  <h3>Ionic Menu Starter</h3>

  <p>
    If you get lost, the <a href="http://ionicframework.com/docs/v2">docs</a> will show you the way.
  </p>

  <button ion-button secondary menuToggle>Toggle Menu</button>
</ion-content>
```

home.scss

```
page-home {  
}
```

Pages et Navigation

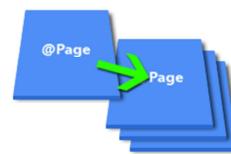
La logique d'empilement de page (stack)



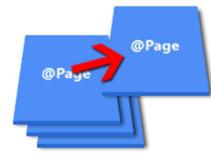
Navigation Stack



Push



Pop

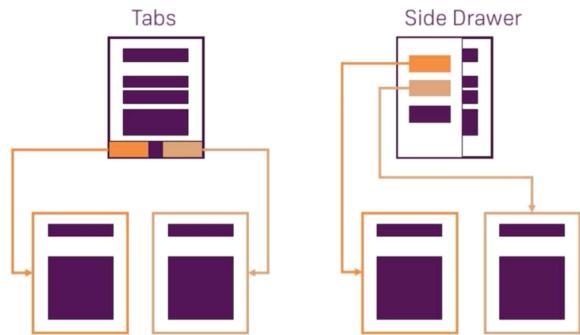


Pages et Navigation

Single Stack



Multiple Stacks



Pages et Navigation

Empiler dans la stack

```
this.navCtrl.push("nextPage");
```

Dépiler dans la stack

```
this.navCtrl.pop();
```

Dépiler entièrement la stack

```
this.navCtrl.popToRoot();
```

Change le root de la stack

```
this.navCtrl.setRoot("otherPage");
```

Attention, il existe également des directives navPush et navPop mais ne fonctionnent pas avec le LazyLoading

Pages et Navigation

Deeplinking et Lazyloading

=> @IonicPage() (Ionic v3)

```
import { Component } from '@angular/core';
import { IonicPage, NavController } from 'ionic-angular';

@IonicPage()
@Component({
  selector: 'page-home',
  templateUrl: 'home.html'
})
export class HomePage {

  constructor(public navCtrl: NavController) {
  }
}
```

Diminution de taille des bundles (car les composants et les pipes ne sont PAS «pré-chargés» par le module racine de l'application)

Modularité accrue - Les composants et les pipes sont organisés en modules spécifiques

Diminution du temps de lancement de l'application - Les modules de composants et de tuyaux sont UNIQUEMENT chargés quand et où c'est nécessaire

Facilite le “deeplinking”

Deep linking pour Mobile apps équivalent au lien sur site web – Le contenu peut être accessible depuis différentes sources.

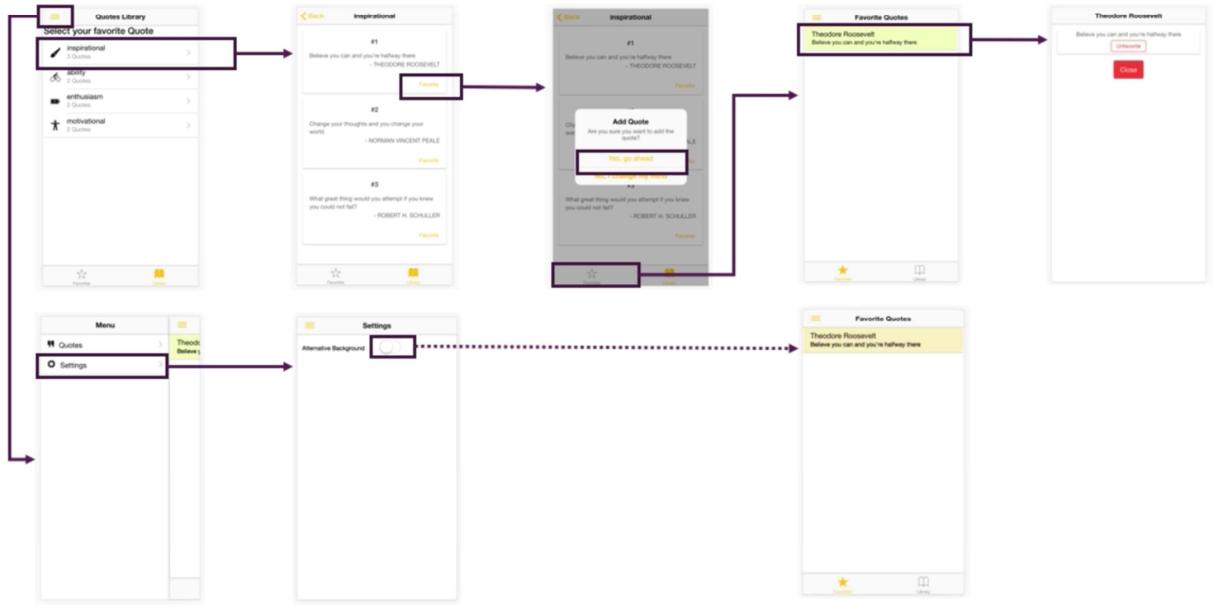
Pages et Navigation



Les pipes et les composants doivent être encapsulés par un module et importés directement dans le module de la page

https://docs.google.com/document/d/1vGokwMXPQItZmTHZQbTO4qwj_SQymFhRS_nJmiH0K3w/edit

Comment structurer la navigation de son projet



Pages et Navigation

Récupérer le projet

```
git clone https://github.com/lahode/cours-ionic3
```



Installer

```
npm install
```

Lancer

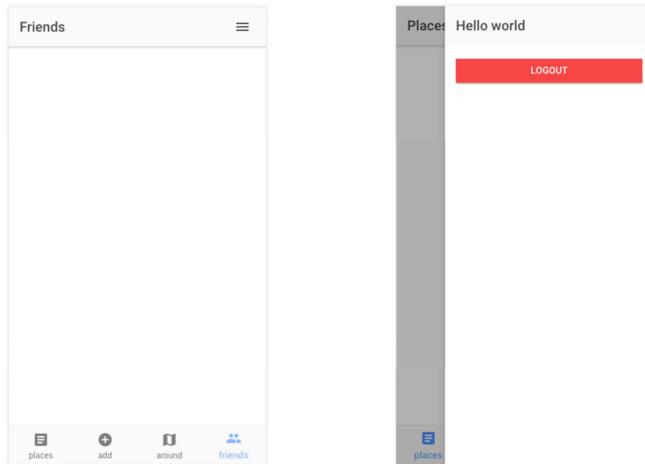
```
ionic serve
```

Exercice

Consignes

Mettre en place une tabulation vers 4 pages différentes ainsi qu'un menu à droite qui affiche le bouton « Logout ».

Le barre du haut doit être comprise dans un composant séparé



Merci de votre attention