

# Formation Mobile Hybride

Cordova, Ionic

# Présentation du cours



## POUR QUI ?

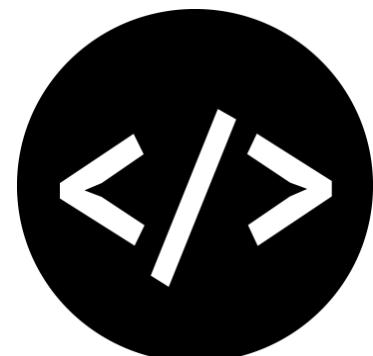
Des développeurs, avec une connaissance Typescript et Angular 2 intéressés par le développement hybride. La connaissance de Sass est un plus.



## APPLICATION HYBRIDE

Nous allons parcourir ensemble le fonctionnement et l'utilisation des technologies : Ionic/Cordova

ionic



## TRAVAUX PRATIQUES

Les parties théoriques seront accompagnées de travaux pratiques tout au long de la formation pour mener à bien un prototype que vous pourrez finaliser par vous même.

# Qui suis je ?

Mickael Dumand



Développeur Front End  
Freelance

#JavaScript #Angular #React #Redux #Webpack  
#Babel #TypeScript #Cordova #Gulp/Grunt  
#Sass #Ionic #Git

**@MeKimak**

# Agenda

1. Contexte mobile hybride
2. Installation des outils
3. Les composants Ionic
4. Ionic navigation
5. Mise en place d'un prototype

# Contexte Mobile Hybride

# Créer un service mobile



## SITE MOBILE

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

Ce site peut être « Responsive », Dédié au canal mobile ou Adaptatif



## APPLICATION HYBRIDE

Un moteur natif qui encapsule du HTML5/JS fortement mutualisé entre les plateformes.

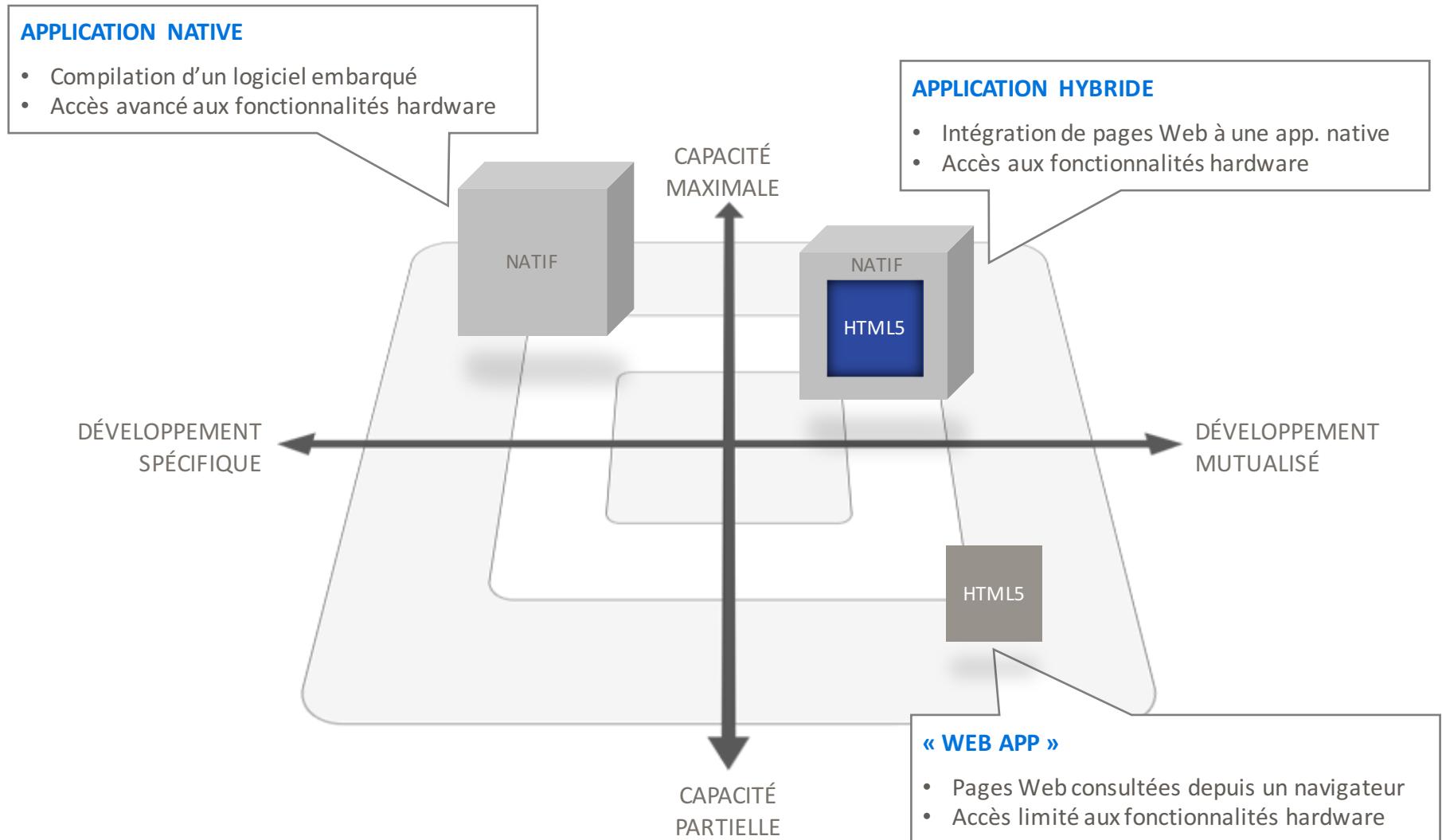
Permet d'utiliser les fonctionnalités natives du mobile et une distribution par les stores.



## NATIF

Application développée dans un langage spécifique à un OS et distribuée par les stores.

# Comment 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...**



→  
Temps

**Monte Marty, je  
vais t'expliquer.**



→  
Temps

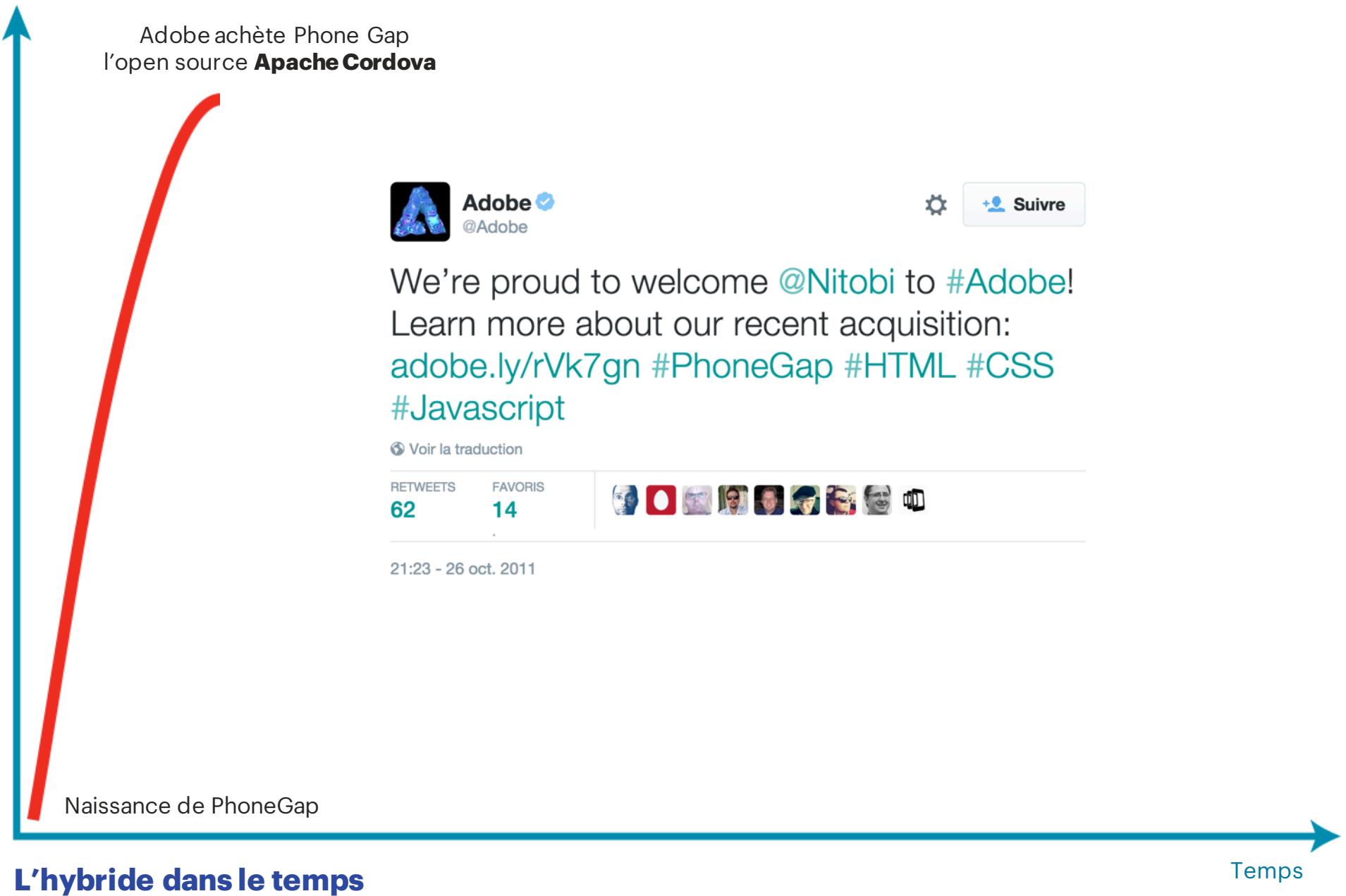
## En 2009 la société Nitobi crée PhoneGap

Popularité



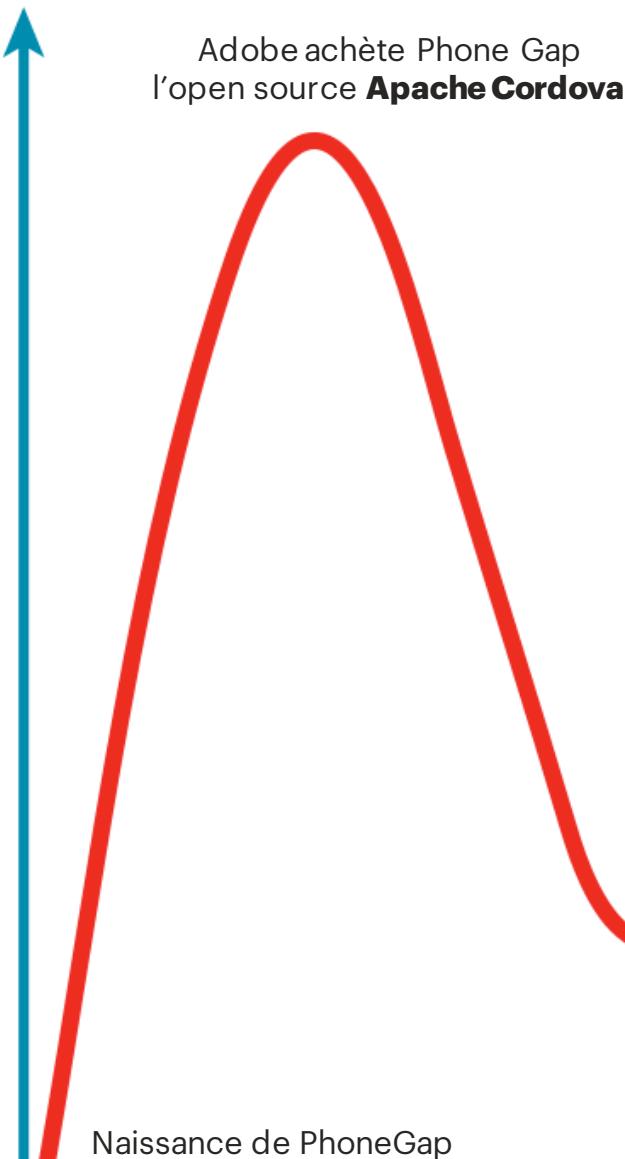
## En 2011 Adobe rachète Nitobi et open source le projet

Popularité



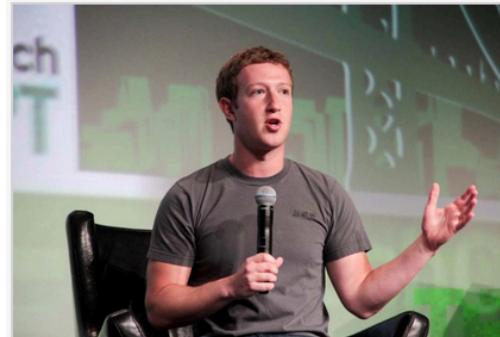
## En 2012 Facebook abandonne HTML5 pour ses applications mobile

Popularité



### 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.

L'hybride dans le temps

Temps

## En 2012 Facebook abandonne HTML5 pour ses applications mobile

Popularité

Adobe achète Phone Gap  
l'open source **Apache Cordova**



GOD HAS SPOKEN

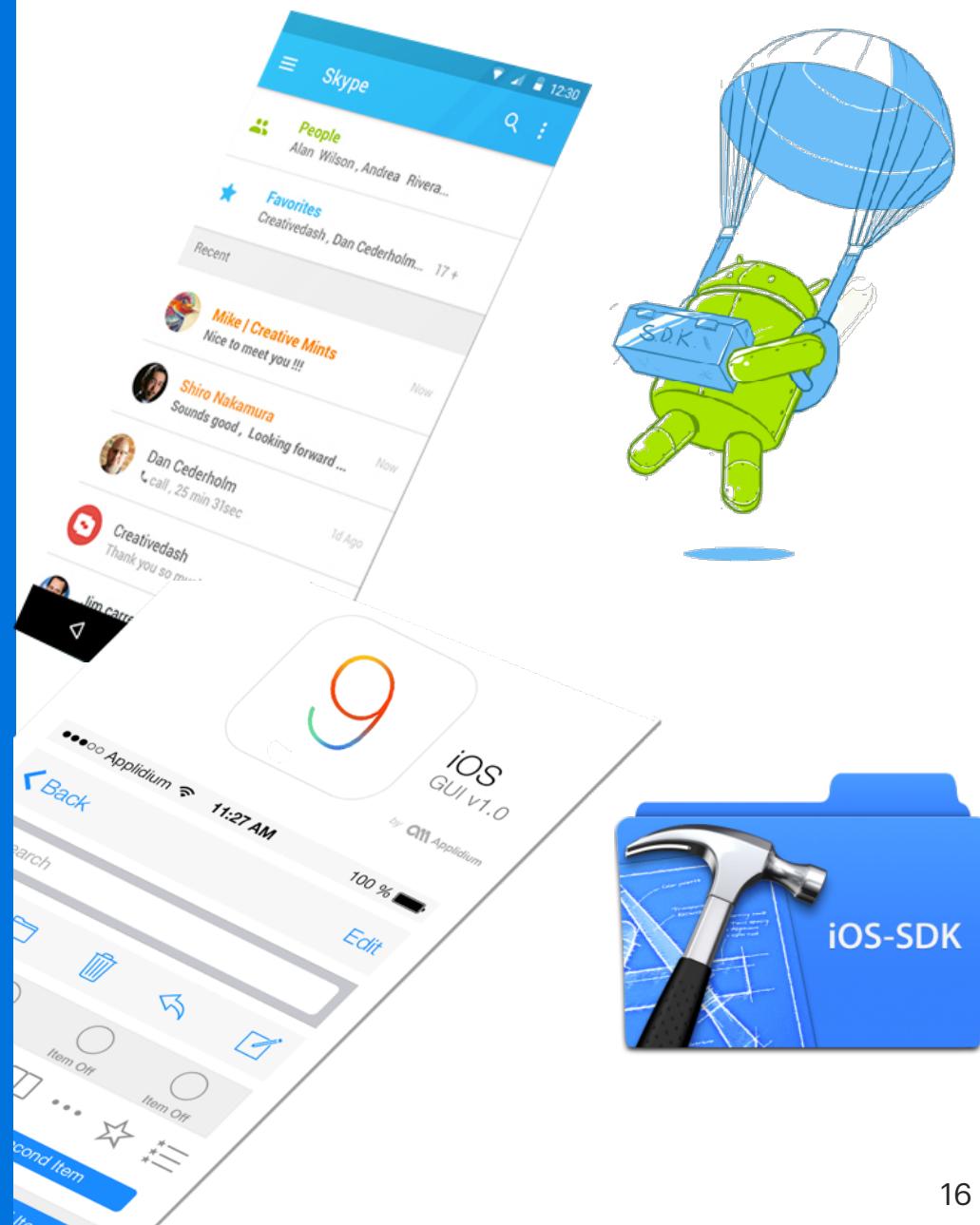
Il devient possible de développer des applications mobiles pour du natif

dans le temps

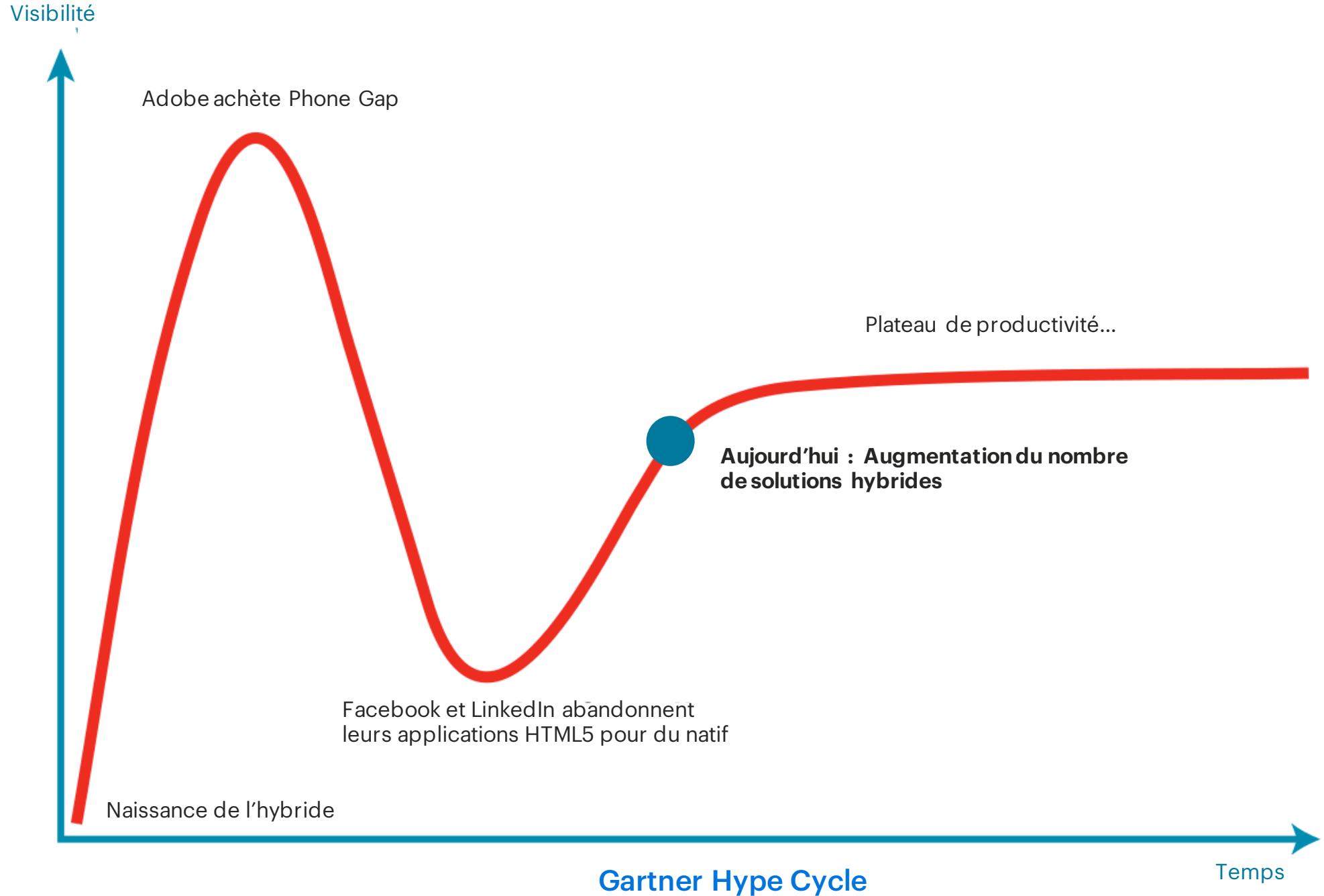
Temps

**Le vrai problème ?**

# Aucun SDK HTML5 pour mobile !



## Mais on est en 2016 : les choses changent !



# Aperçu des solutions hybrides et alternatives



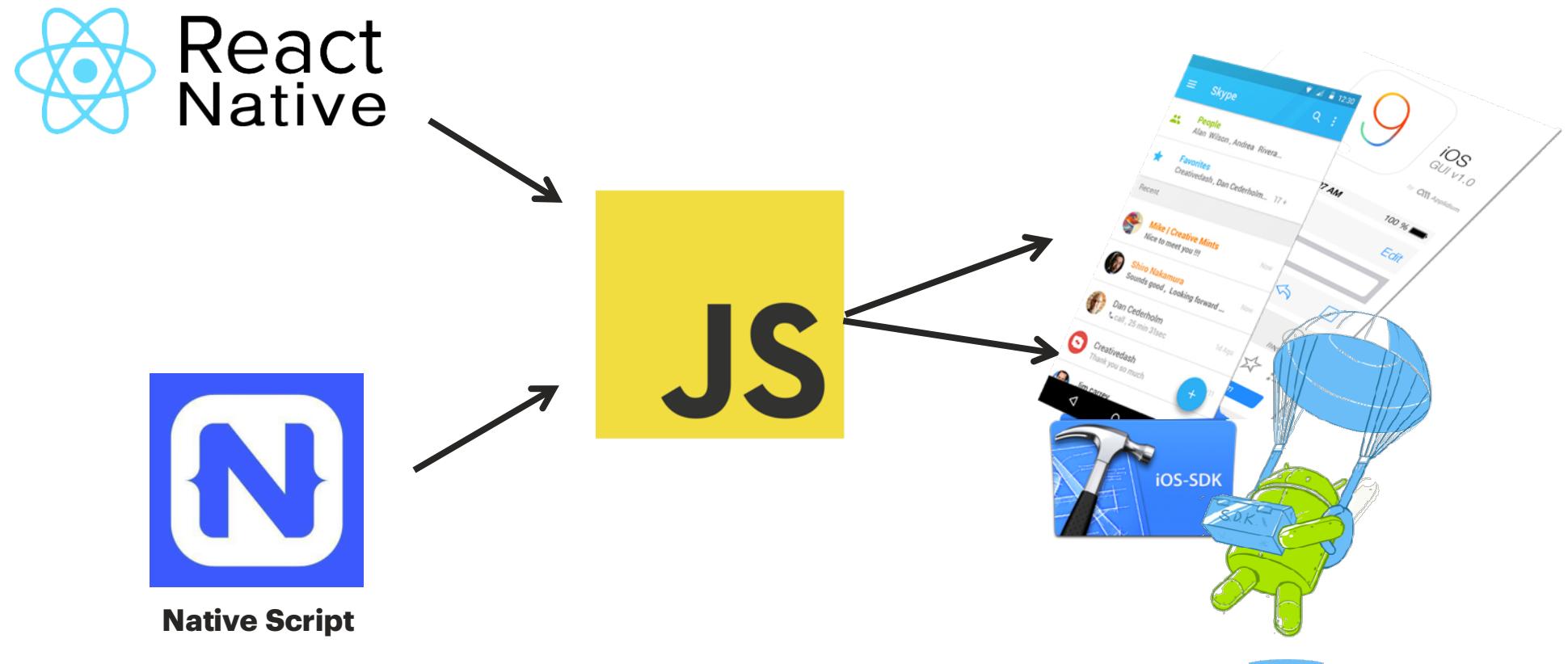


**LE MONDE SÉDIVISE EN  
DEUX CATÉGORIES...**

# Panorama des solutions hybrides



# Piloter des composants natifs en JavaScript

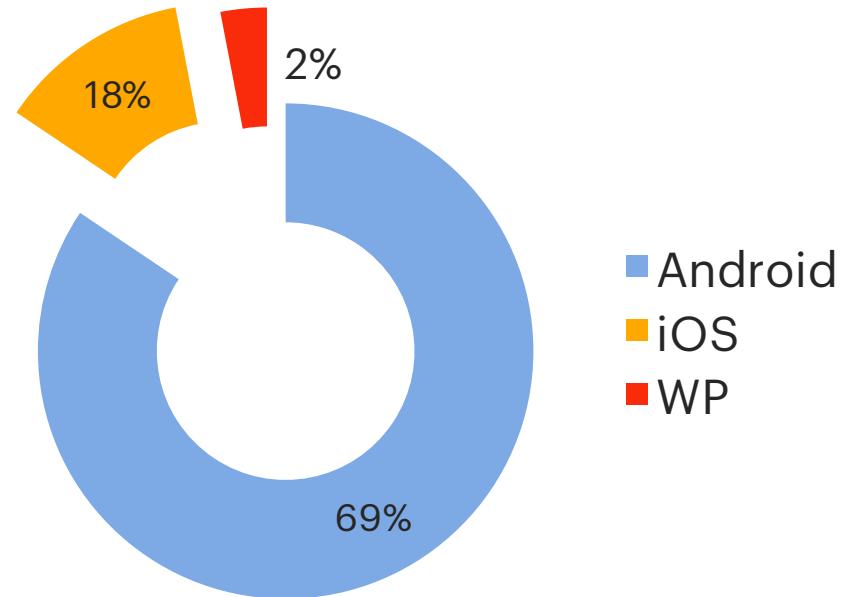


# Utiliser HTML5 dans un « navigateur » embarqué

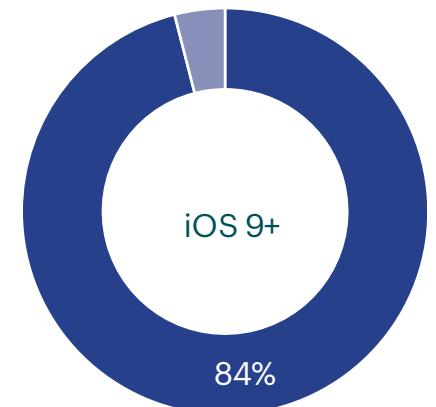
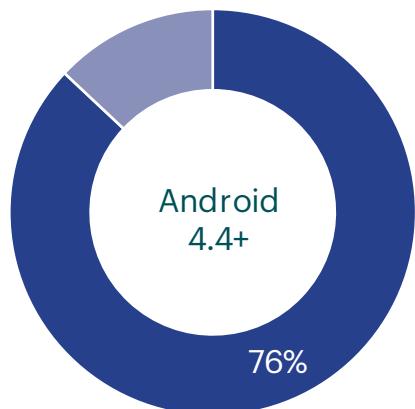


# Vision actuel du marché mobile mondial

Android et iOS dominent le monde



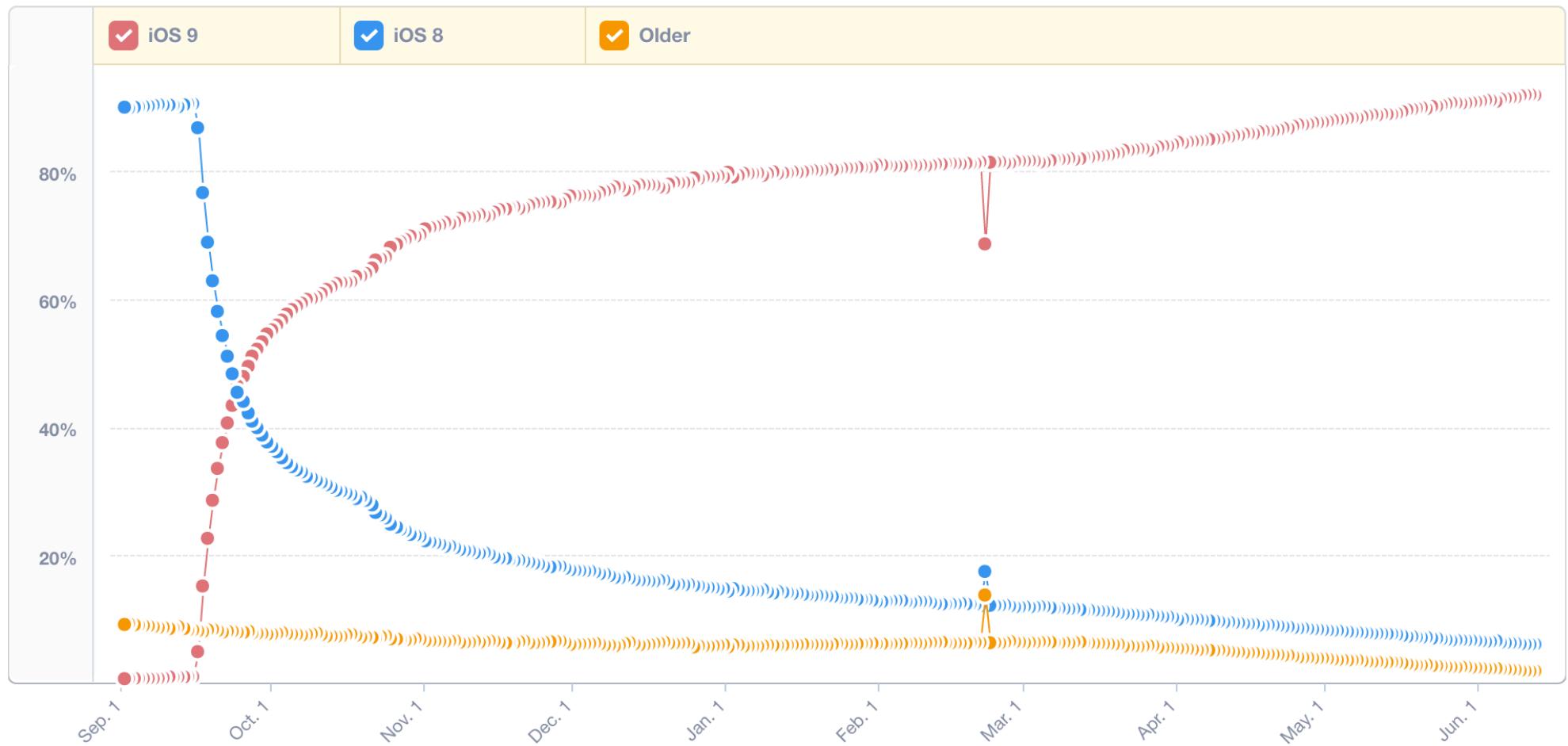
Fragmentation des OS majoritaires



## iOS 9 adoption ▾

Sep 1, 2015 - Jun 13, 2016 ▾

Hour Day



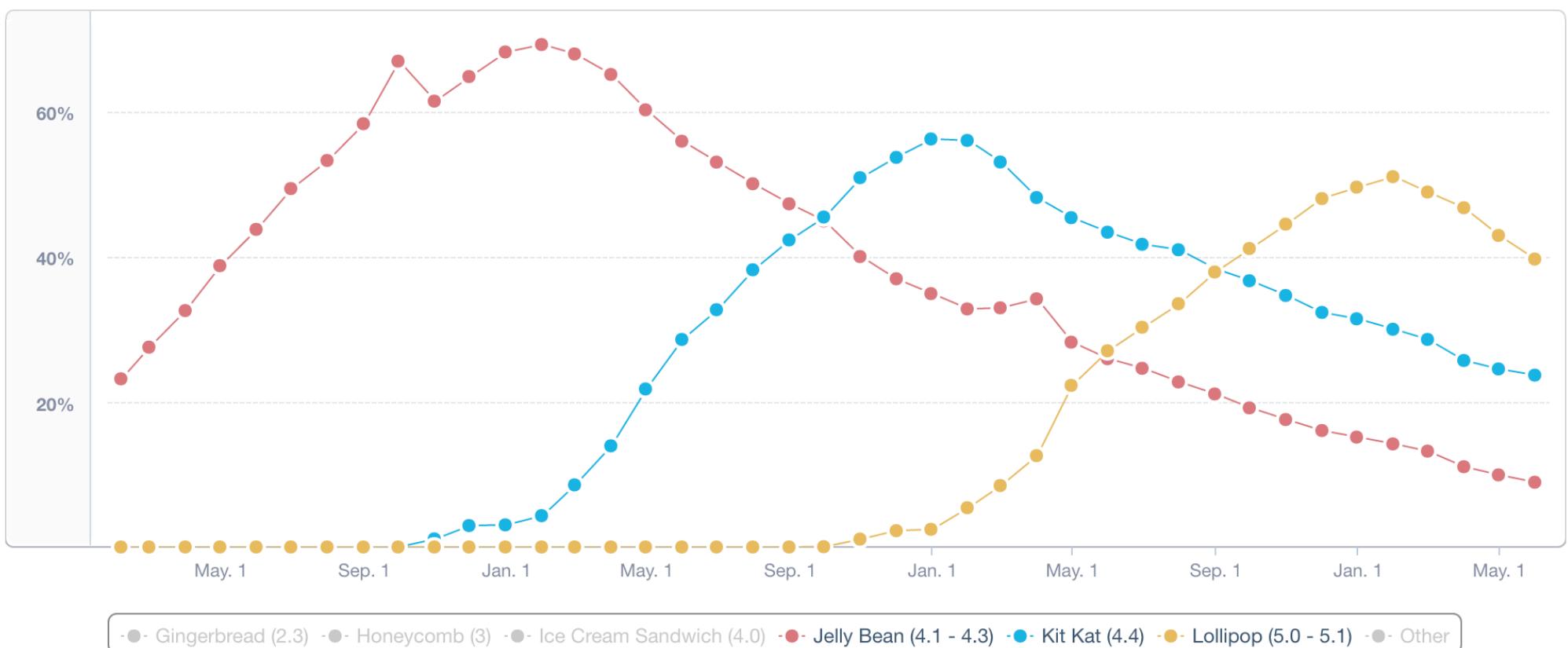
# Android

## Android OS Adoption

Feb 5th, 2013

Jun 13th, 2016

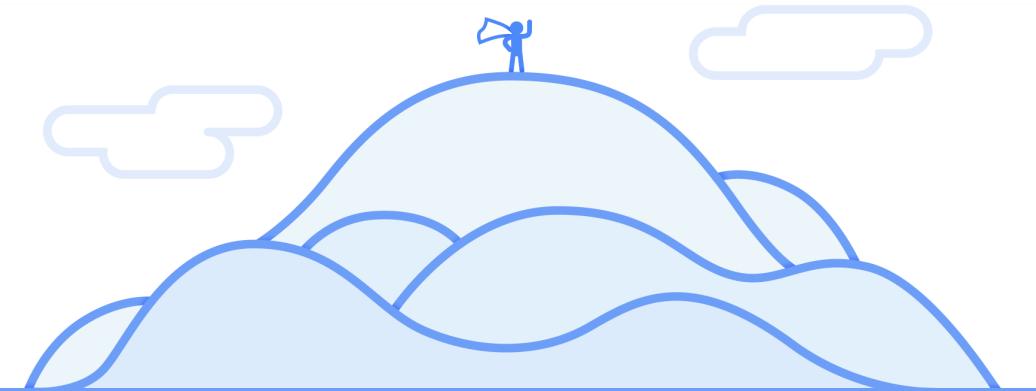
DONE



# Pourquoi choisir Ionic ?



ionic



# Ionic n'est pas seulement un framework, c'est un SDK.

- Documentation complète
- Une large communauté, des tutoriaux et beaucoup de ressources disponibles.

15,872 stars

2,367 forks

21.8K followers

- 
- 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 AngularJs et Typescript

- Un écosystème complet, build, application de test, extensions etc.

# Les outils Ionic

Ionic CLI



Ionic View



Ionic lab



Ionic Creator



Ionic Box



Ionicons



Ionic push



Ionic native

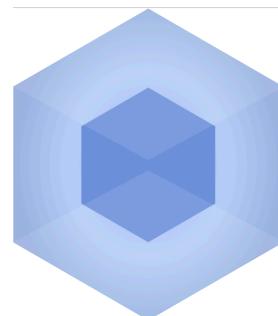


# L'écosystème



## 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.



## AngularJs 2 + Typescript

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



## Sass

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

# Rappel

# ECMAScript 6 (ES2015)

## Classes

```
class Shape {  
    constructor (id, x, y) {  
        this.id = id  
        this.move(x, y)  
    }  
    move (x, y) {  
        this.x = x  
        this.y = y  
    }  
}
```

---

## Modules

```
// lib/math.js  
export function sum (x, y) { return x + y }  
export var pi = 3.141593  
  
// someApp.js  
import * as math from "lib/math"  
console.log("2π = " + math.sum(math.pi, math.pi))  
  
// otherApp.js  
import { sum, pi } from "lib/math"  
console.log("2π = " + sum(pi, pi))
```

---

## Promises

```
doSomething().then((response) => {  
    console.log(response);  
});
```

---

## Arrow functions

```
someFunction((response) => {  
    console.log(response);  
});
```

rather than:

```
someFunction(function(response){  
    console.log(response);  
});
```

```
function add(x: number, y :number):number {  
    return x + y;  
}  
add('a', 'b'); // compiler error
```

---

### Example

```
export class MyApp {  
  
    rootPage: any = HomePage;  
  
    constructor(platform: Platform) {  
  
        platform.ready().then(() => {  
  
            });  
    }  
}
```

## Binding a Property to a Value

```
<input [value]="firstName">
```

---

## Calling a Function on an Event

```
<button (click)="someFunction($event)">
```

---

## Rendering Expressions

```
<p>Hi, {{name}}</p>
```

---

## Two Way Data Binding

```
<input [(ngModel)]="name">
```

```
<input [value]="name" (input)="name = $event.target.value">
```

# Angular 2

## Creating a Local Variable

```
<p #myParagraph></p>
```

```
<button (click)="myParagraph.innerHTML = 'Once upon a time...'">
```

---

## Embedded Templates

```
<p *something="someExpression"></p>
  =
<template [something]="someExpression">
  <p></p>
</template>
```

---

## Directives

```
<section *ngIf="showSection"></section>
<li *ngFor="let item of items"></li>
```

---

## Decorators

```
@Component({
  selector: 'my-component',
  services: [MyService]
})
```

## « Information »

**Ionic 2 est en  
Release  
Candidate.**

Très stable  
pour une utilisation  
avancée mais pas  
encore release.

# Installation

# Pré-requis

## NODEJS + NPM



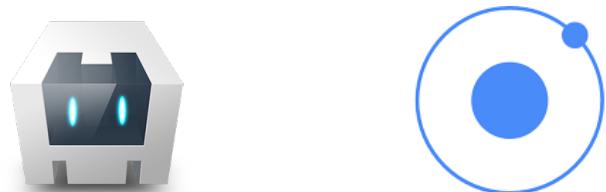
---

## RUBY + SASS



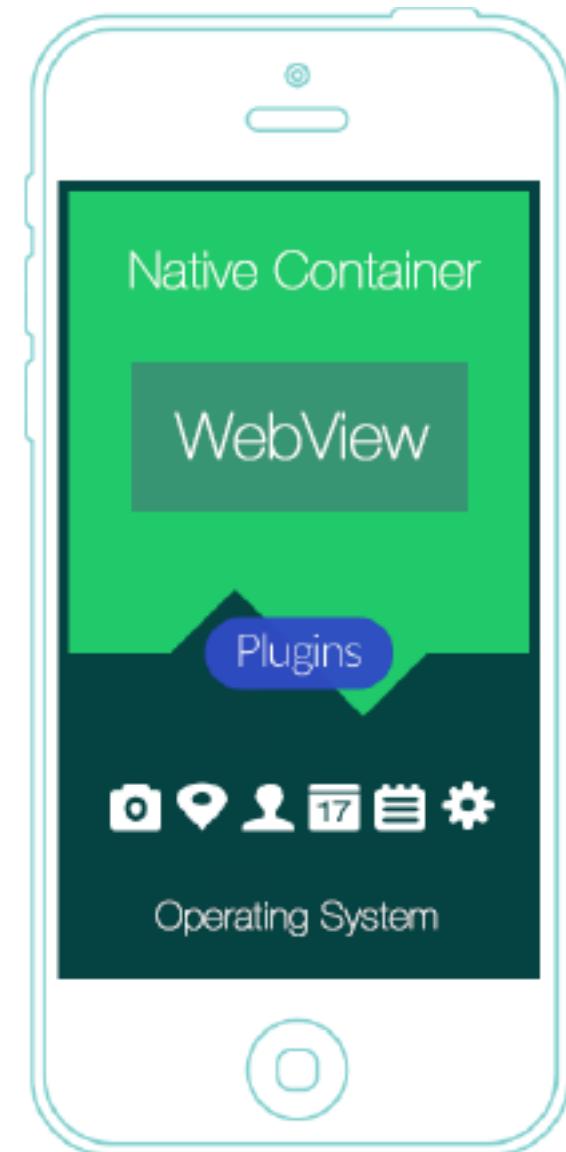
---

## CORDOVA + IONIC



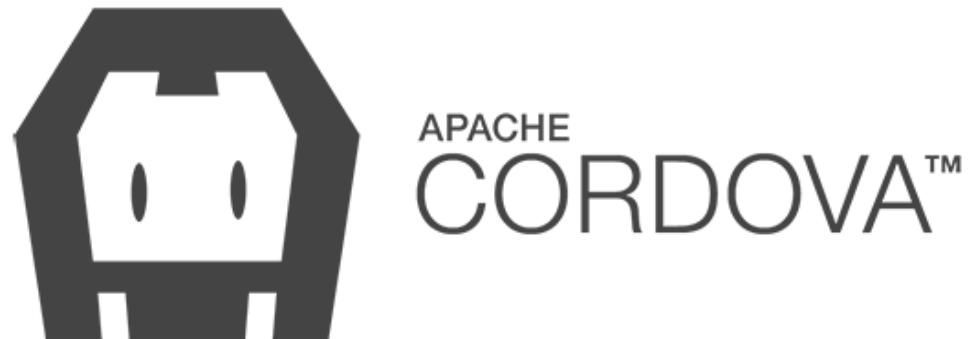
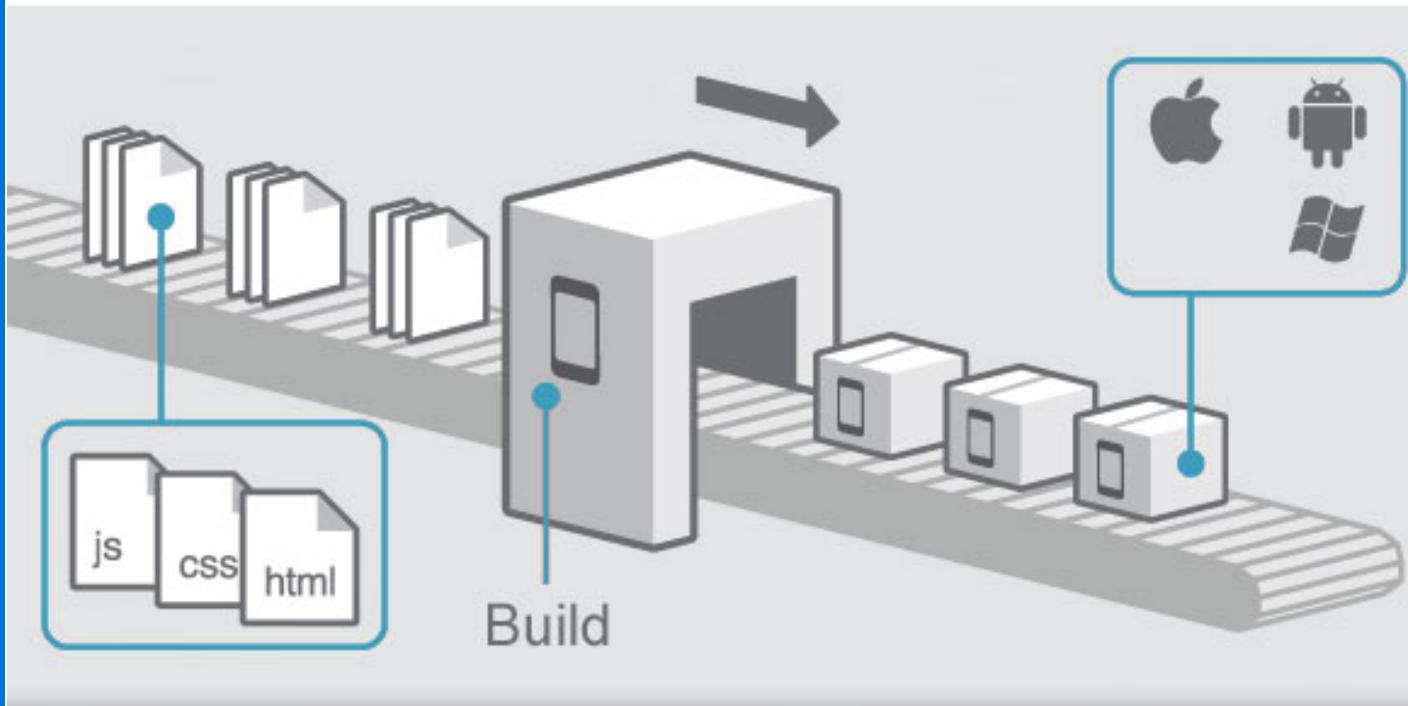
# Comment fonctionne l'hybride ?

Une WebView\* dans un conteneur natif



\*WebView: composant natif pour afficher des pages web.

# Outil de build multi-plateforme



# Cordova CLI

- **Installation**

```
$ sudo npm install -g cordova
```

---

- **Ajouter une nouvelle plateforme**

```
$ cordova platform add ios
```

```
$ cordova platform add android
```

```
$ cordova platform add wp8
```

---

- **Compiler**

```
$ cordova build android
```

```
$ cordova emulate android
```

```
$ cordova run android
```

# Ionic CLI

- **Installation**

```
$ sudo npm install -g ionic
```

---

- **Ajouter une nouvelle plateforme**

```
$ ionic platform add ios
```

```
$ ionic platform add android
```

```
$ ionic platform add wp8
```

---

- **Compiler**

```
$ ionic build android
```

```
$ ionic emulate android
```

```
$ ionic run android
```

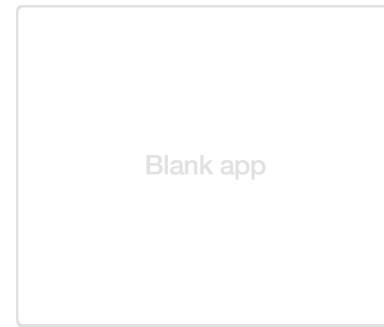
```
$ ionic help
```

Available tasks: (use --help or -h for more info)

<code>start</code>	Starts a new Ionic project in the specified PATH
<code>serve</code>	Start a local development server for app dev/testing
<code>platform</code>	Add platform target for building an Ionic app
<code>run</code>	Run an Ionic project on a connected device
<code>emulate</code>	Emulate an Ionic project on a simulator or emulator
<code>build</code>	Locally build an Ionic project for a given platform
<code>plugin</code>	Add a Cordova plugin
<code>resources</code>	Automatically create icon and splash screen resources ( <b>beta</b> ) Put your images in the ./resources directory, named splash or icon. Accepted file types are .png, .ai, and .psd. Icons should be 192x192 px without rounded corners. Splashscreens should be 2208x2208 px, with the image centered in the middle.
<code>package</code>	Package an app using the Ionic Build service ( <b>beta</b> )
<code>upload</code>	Upload an app to your Ionic account
<code>share</code>	Share an app with a client, co-worker, friend, or customer
<code>lib</code>	Gets Ionic library version or updates the Ionic library
<code>setup</code>	Configure the project with a build tool ( <b>beta</b> )
<code>browser</code>	Add another browser for a platform ( <b>beta</b> )
<code>service</code>	Add an Ionic service package and install any required plugins
<code>add</code>	Add an Ion, bower component, or addon to the project
<code>remove</code>	Remove an Ion, bower component, or addon from the project
<code>list</code>	List Ions, bower components, or addons in the project
<code>ions</code>	List available ions to add to your project
<code>templates</code>	List available Ionic starter templates
<code>info</code>	List information about the users runtime environment
<code>help</code>	Provides help for a certain command
<code>link</code>	Sets your Ionic App ID for your project
<code>hooks</code>	Manage your Ionic Cordova hooks
<code>state</code>	Saves or restores state of your Ionic Application using the package.json file
<code>docs</code>	Opens up the documentation for Ionic

# Let's go ionic !

## Créer un projet



Blank app

**\$ ionic start hello blank --v2**

---

Your system information:

```
Cordova CLI: 6.2.0
Ionic Framework Version: 2.0.0-beta.10
Ionic CLI Version: 2.0.0-beta.25
Ionic App Lib Version: 2.0.0-beta.15
ios-deploy version: Not installed
ios-sim version: Not installed
OS: Mac OS X El Capitan
Node Version: v4.4.7
Xcode version: Xcode 7.3.1 Build version 7D1014
```

**\$ ionic info**

---

## Lancer le projet dans le navigateur



**\$ ionic serve**

# Structure d'un projet

```
ionic2-places
  .git
  .tmp
  hooks
  node_modules
  platforms
  plugins
  resources
  src
    app
    assets
    components
    pages
    theme
    declarations.d.ts
  index.html
  manifest.json
  service-worker.js
  www
    .editorconfig
    .gitignore
    config.xml
    ionic.config.json
    package.json
    tsconfig.json
    tslint.json
```

```
src/index.html
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=1.0">
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.log('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 -->
32   <ion-app></ion-app>
33
34   <!-- The polyfills js is generated during the build process -->
35   <script src="build/polyfills.js"></script>
36
37   <!-- The bundle js is generated during the build process -->
38   <script src="build/main.js"></script>
39
```

# Structure d'un projet

The image shows a file explorer window and a code editor side-by-side.

**File Explorer:** The left pane displays the project structure under a folder named "hello". The "src" directory contains an "app" folder which includes files like "app.component.ts", "app.module.ts", "app.scss", "main.dev.ts", and "main.prod.ts". It also contains folders for "assets", "components", "pages", and "theme". A "package.json" file is also present at the root level.

**Code Editor:** The right pane shows the content of the "index.html" file. The code is a standard HTML document structure with DOCTYPE, head, and body sections. The head section includes meta tags for title, charset, viewport, and msapplication-tap-highlight. It also links to CSS files for different platforms: "app.ios.css" for iOS, "app.md.css" for MD (likely Material Design), and "app.wp.css" for WP (Windows Phone). The body section contains an "ion-app" element. The code is annotated with line numbers from 1 to 29.

```
<!DOCTYPE html>
<html dir="ltr" lang="en">
<head>
  <title>Ionic</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <meta name="format-detection" content="telephone=no">
  <meta name="msapplication-tap-highlight" content="no">
  <link ios-href="build/css/app.ios.css" rel="stylesheet">
  <link md-href="build/css/app.md.css" rel="stylesheet">
  <link wp-href="build/css/app.wp.css" rel="stylesheet">
</head>

<body>
  <ion-app></ion-app>

  <!-- cordova.js required for cordova apps -->
  <script src="cordova.js"></script>
  <!-- Polyfill needed for platforms without Promise -->
  <script src="build/js/es6-shim.min.js"></script>
  <!-- Zone.js and Reflect-metadata -->
  <script src="build/js/Reflect.js"></script>
  <script src="build/js/zone.js"></script>
  <!-- the bundle which is built from the app -->
  <script src="build/js/app.bundle.js"></script>
</body>
```

# Ionic generators

\$ ionic generate --list

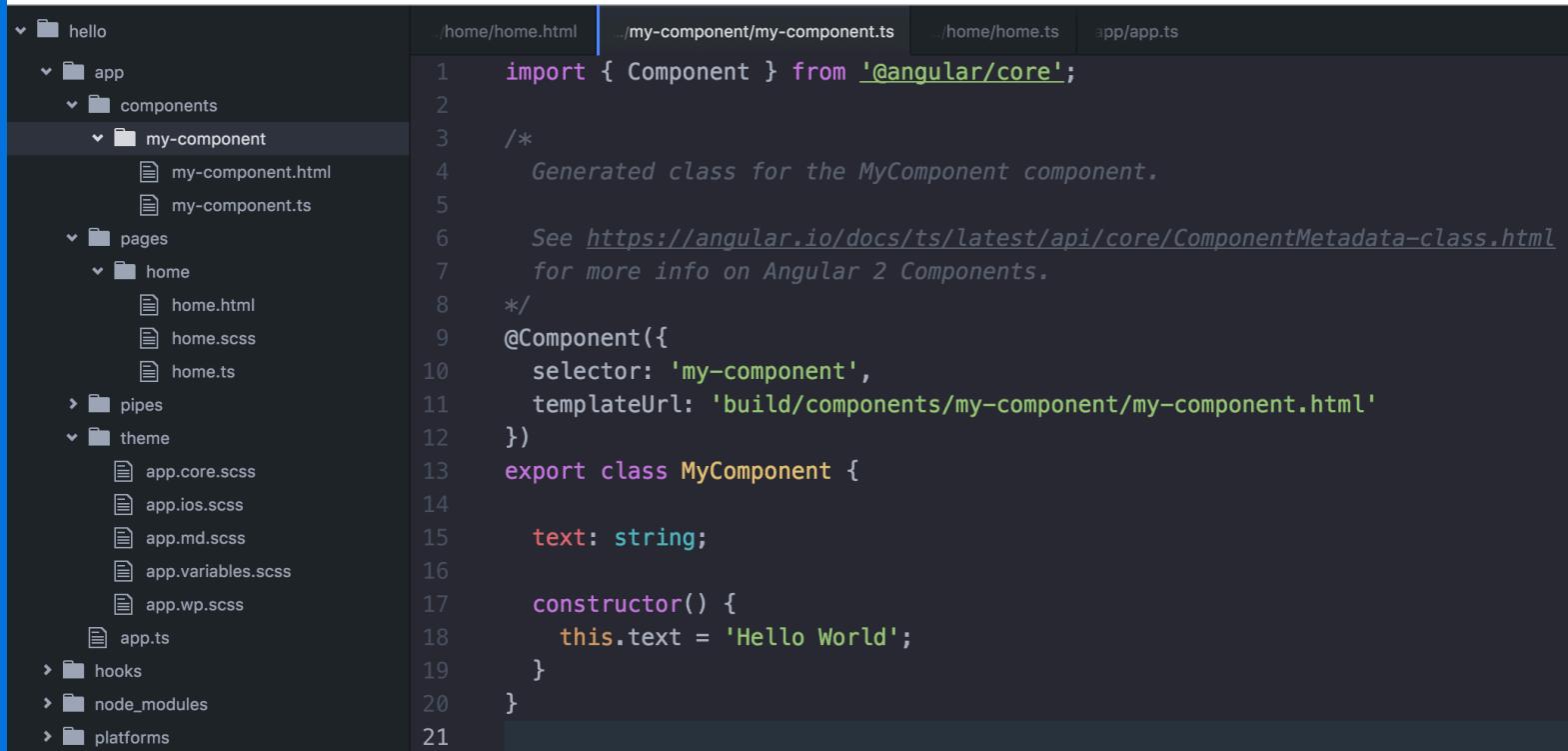
## Angular:

- \* component
- \* directive
- \* provider
- \* pipe

## Ionic:

- \* page
- \* tabs

\$ ionic g component MyComponent



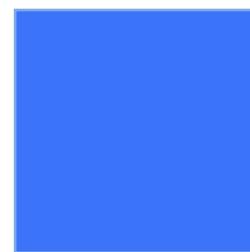
The screenshot shows a file explorer on the left and a code editor on the right. The file explorer displays a project structure with a 'hello' folder containing an 'app' folder. Inside 'app' are 'components', 'pages', and 'theme' folders. The 'components' folder contains a 'my-component' folder with 'my-component.html' and 'my-component.ts' files. The 'pages' folder contains a 'home' folder with 'home.html', 'home.scss', and 'home.ts' files. The 'theme' folder contains 'app.core.scss', 'app.ios.scss', 'app.md.scss', 'app.variables.scss', 'app.wp.scss', and 'app.ts' files. The 'hooks', 'node\_modules', and 'platforms' folders are also visible. The code editor shows the 'my-component.ts' file with the following content:

```
1 import { Component } from '@angular/core';
2
3 /*
4   Generated class for the MyComponent component.
5
6   See https://angular.io/docs/ts/latest/api/core/ComponentMetadata-class.html
7   for more info on Angular 2 Components.
8 */
9 @Component({
10   selector: 'my-component',
11   templateUrl: 'build/components/my-component/my-component.html'
12 })
13 export class MyComponent {
14
15   text: string;
16
17   constructor() {
18     this.text = 'Hello World';
19   }
20 }
```

# Les composants Ionic

# Ionic UI scaffolding

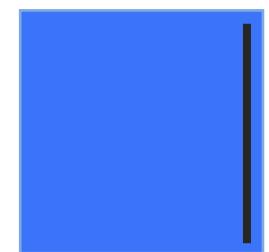
<ion-app>



<ion-nav>



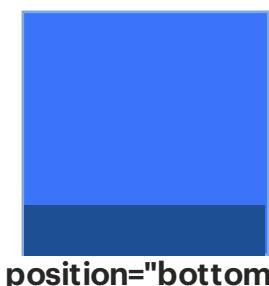
<ion-content>



<ion-toolbar>



<ion-toolbar>



<ion-navbar>



<ion-buttons>





## UI Components

A comprehensive preview of our mobile UI components - everything you'll need.

### Icon Buttons

[Demo Source](#)

To add icons to a button, add an icon component inside of it:

```
<!-- Float the icon left -->
<button>
  <ion-icon name="home"></ion-icon>
  Left Icon
</button>

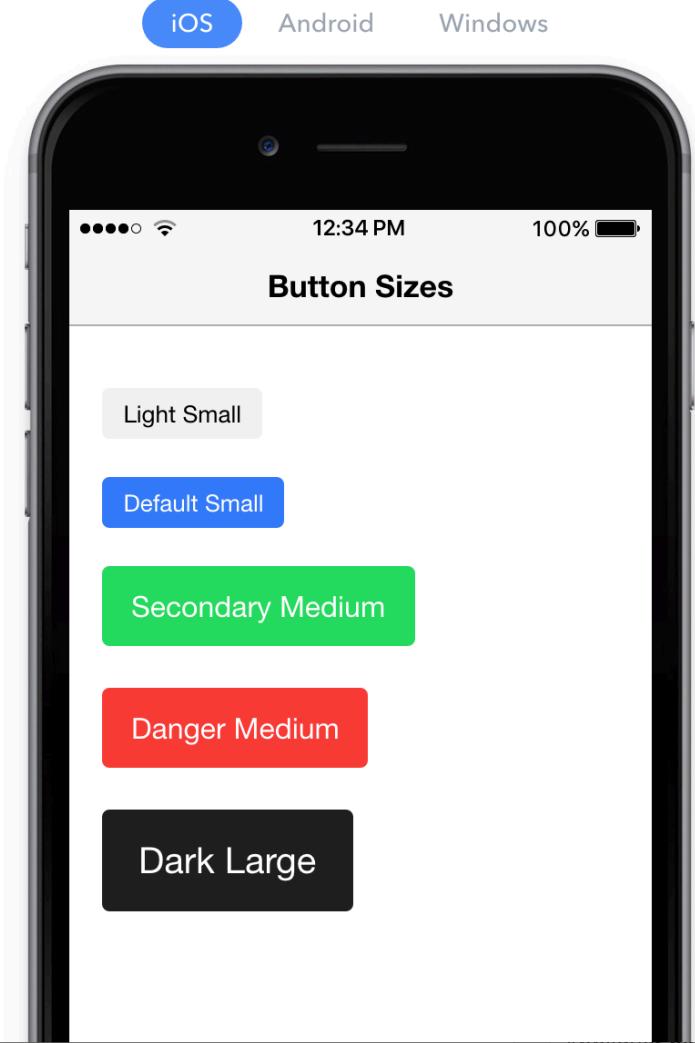
<!-- Float the icon right -->
<button>
  Right Icon
  <ion-icon name="home"></ion-icon>
</button>

<!-- Only icon (no text) -->
<button>
  <ion-icon name="home"></ion-icon>
</button>
```

iOS

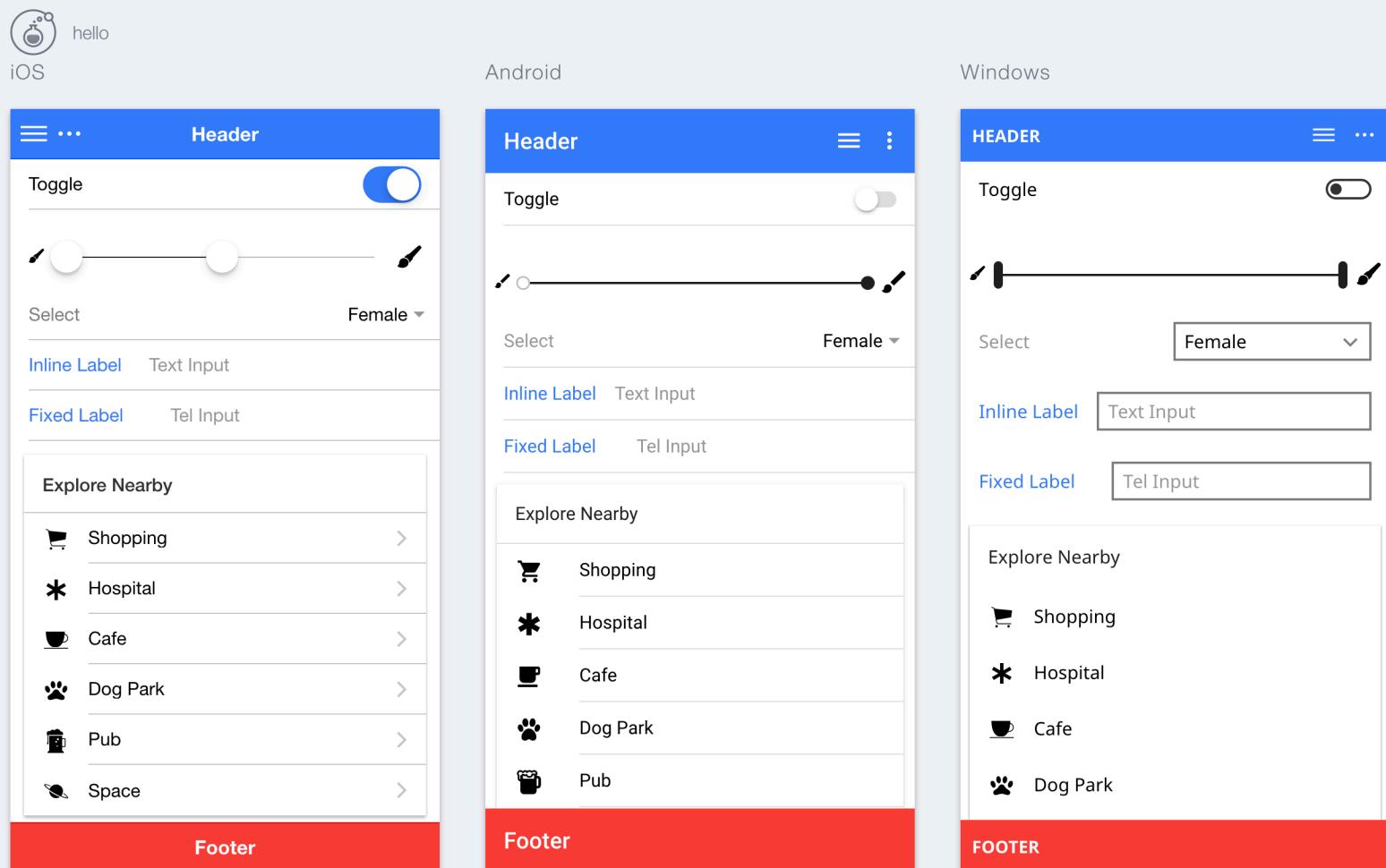
Android

Windows



# Platform continuity

\$ ionic serve --lab



A vous de jouer !

# Platforms config

The image shows a code editor with two panes. The left pane displays the project file structure:

- hello (selected folder)
  - app
    - pages
      - home
        - home.html
        - home.scss
        - home.ts
      - theme
        - app.core.scss
        - app.ios.scss
        - app.md.scss
        - app.variables.scss
        - app.wp.scss
        - app.ts
    - hooks
    - node\_modules
    - platforms
    - plugins
    - resources
    - typings
    - www
      - build
        - index.html
      - .gitignore
      - config.xml
      - gulpfile.js
      - ionic.config.json
      - package.json

# Ionic Navigation

# Ionic 2 navigation != Angular 2 Router

driftyco / ionic

Watch 1,516 Star 24,544 Fork 4,974

Code Issues 428 Pull requests 17 Pulse Graphs

## Update to new Angular 2 Router #6563

New issue

Open dheran opened this issue on 18 May · 5 comments

dheran commented on 18 May

Ionic 2

Update ionic to new Angular 2 Router

5

Ionitron added the v2 label on 18 May

adambadley commented on 18 May

Ionic member +

This will still be a work in progress as the new Angular 2 router still has some holes that does not allow us to update yet. The next release will not be integrated with the ng2 router yet, and we're working with the Angular team to solve this.

adambadley self-assigned this on 18 May

Labels

v2

Milestone

No milestone

Assignees

adambadley

6 participants

adambadley, Ionitron, Driftyco, Adam Bradley, Ionic Team, Ionic Community

Notifications

Unsubscribe

You're receiving notifications because you commented.

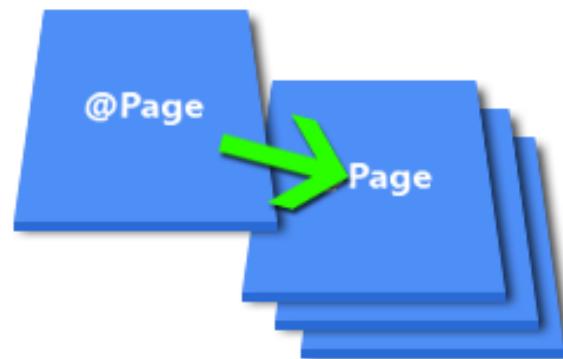
Pour le moment...

# Ionic Navigation Stack

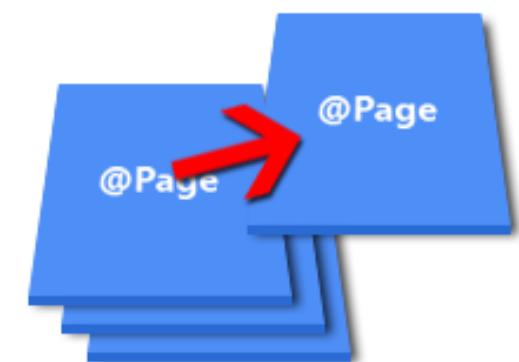
Navigation Stack



Push



Pop

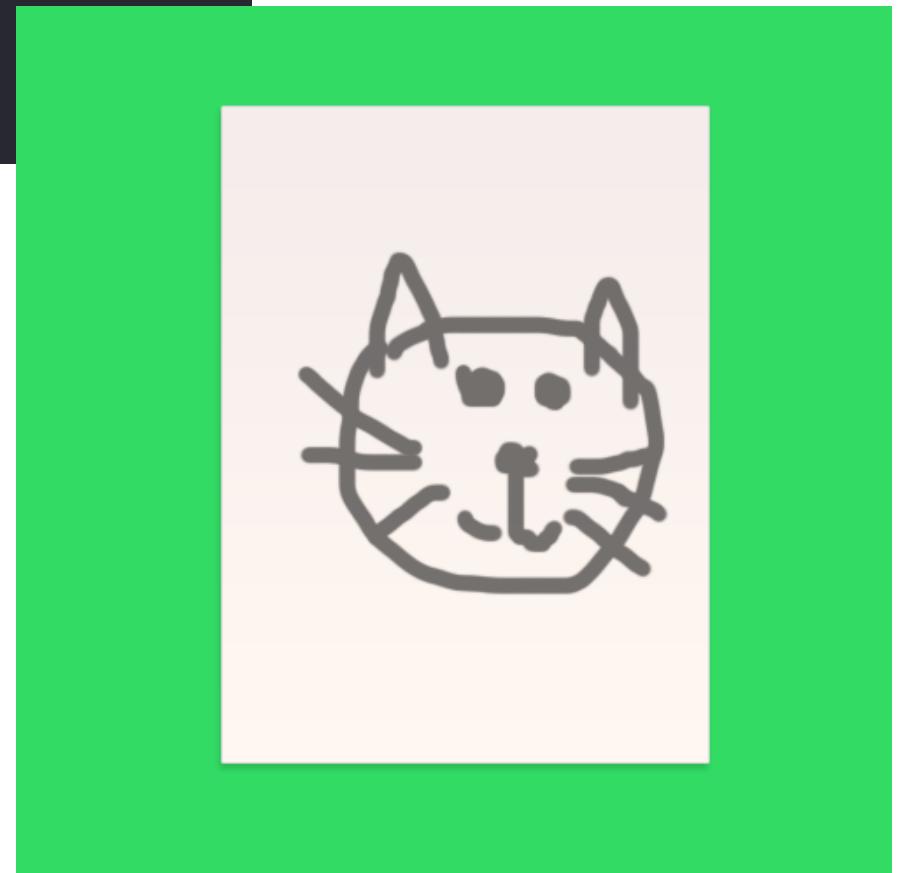


# rootPage = CatsPage

```
@Component({
  template: '<ion-nav [root]="rootPage"></ion-nav>'
})
export class MyApp {

  private rootPage:any = CatsPage;

  constructor(private platform:Platform) {
```



# this.nav.push(DogsPage);

```
import {DogsPage} from '.../.../pages/dogs/dogs';

@Component({
  templateUrl: 'build/pages/cats/cats.html',
})
export class CatsPage {
  constructor(public nav: NavController) {

  }
  next(): void {
    this.nav.push(DogsPage);
  }
}
```

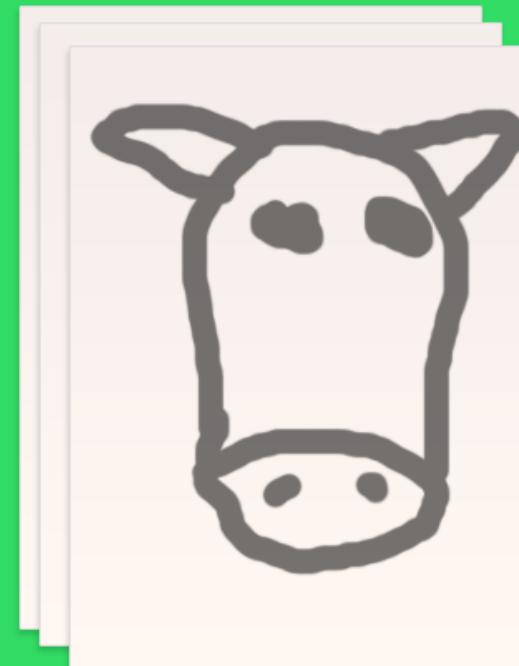


# this.nav.push(CowsPage);

```
import {CowsPage} from '.../.../pages/cows/cows';

@Component({
  templateUrl: 'build/pages/dogs/dogs.html',
})
export class DogsPage {
  constructor(public nav: NavController) {

  }
  next(): void {
    this.nav.push(CowsPage);
  }
}
```



# this.nav.pop();

```
@Component({
  templateUrl: 'build/pages/cows/cows.html',
})
export class CowsPage {
  constructor(public nav: NavController) {

  }
  next(): void {
    this.nav.pop();
  }
}
```



# This.nav.setRoot(CatsPage)

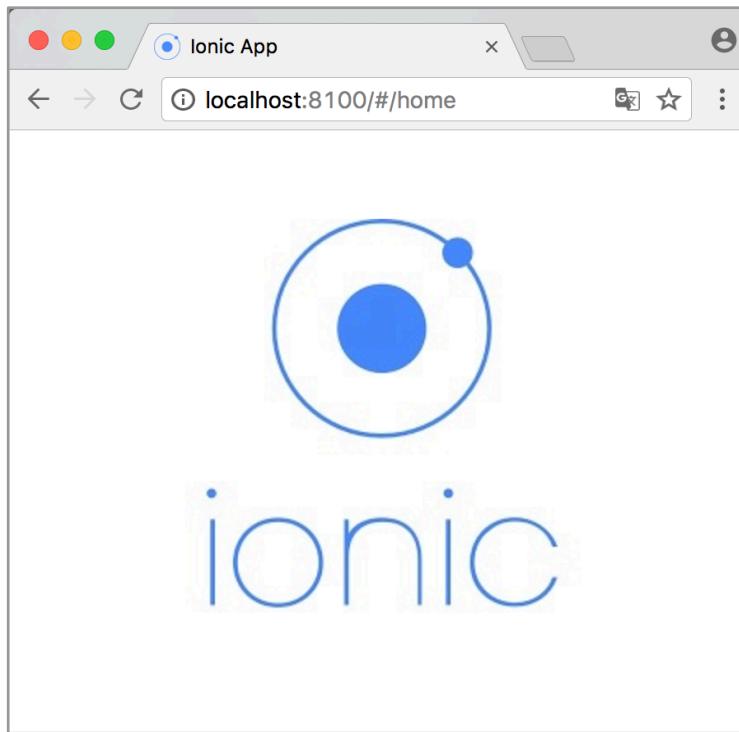
```
import {CatsPage} from '.../.../pages/cats/cats';

@Component({
  templateUrl: 'build/pages/dogs/dogs.html',
})
export class DogsPage {
  constructor(public nav: NavController) {

  }
  next(): void {
    this.nav.setRoot(CatsPage);
  }
}
```



# Gestion d'URL grâce au « DeepLinker »



DeepLinker implements `DeepLinkerConfig`, which is an object with an array of links. So for basic example based on the blank starer, a link setup like so:

```
imports: [
  IonicModule.forRoot(MyApp, {}, {
    links: [
      { component: HomePage, name: 'Home', segment: 'home' }
    ]
  })
]
```

# Tabs

\$ ionic start test-tabs tabs --lab

iOS

Home

## Welcome to Ionic!

This starter project comes with simple tabs-based layout for apps that are going to primarily use a Tabbed UI.

Take a look at the app/ directory to add or change tabs, update any existing page or create new pages.

Android

Home



HOME



ABOUT



CONTACT

## Welcome to Ionic!

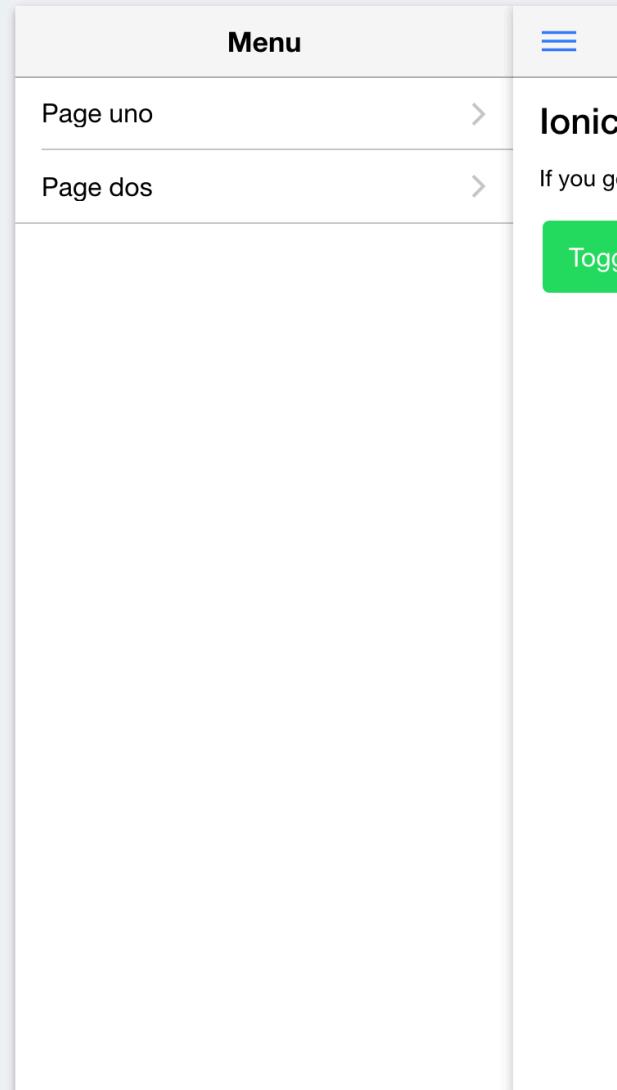
This starter project comes with simple tabs-based layout for apps that are going to primarily use a Tabbed UI.

Take a look at the app/ directory to add or change tabs, update any existing page or create new pages.

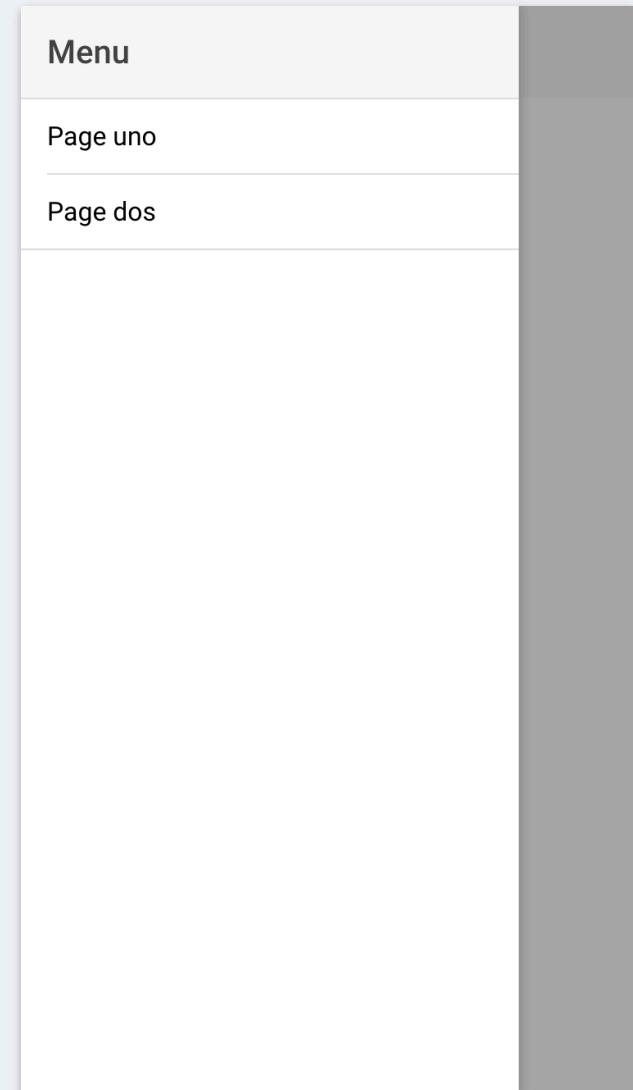
# Sidemenu

\$ ionic start test-sidemenu sidemenu --lab

iOS

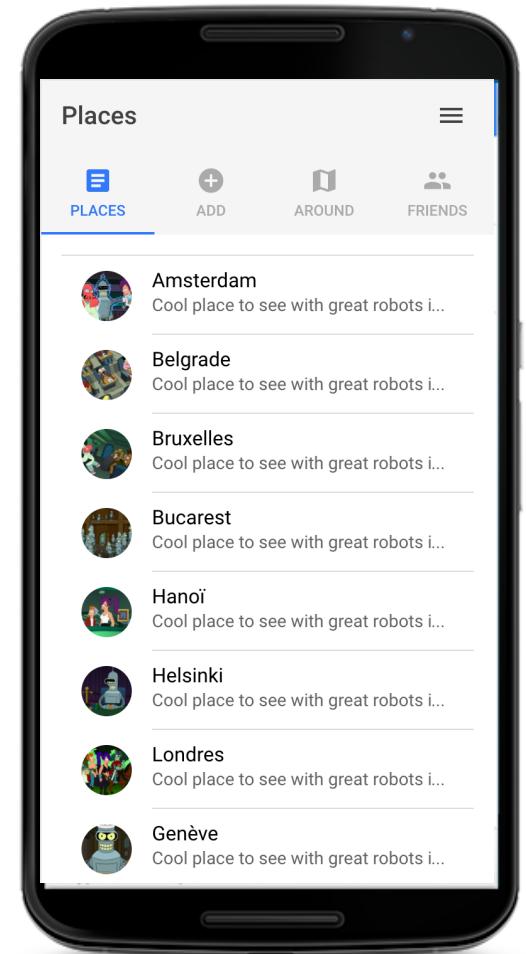
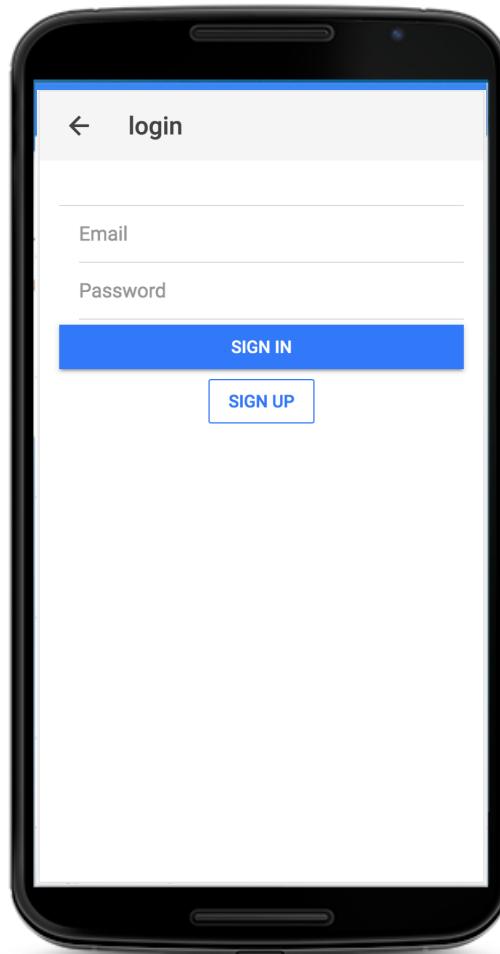
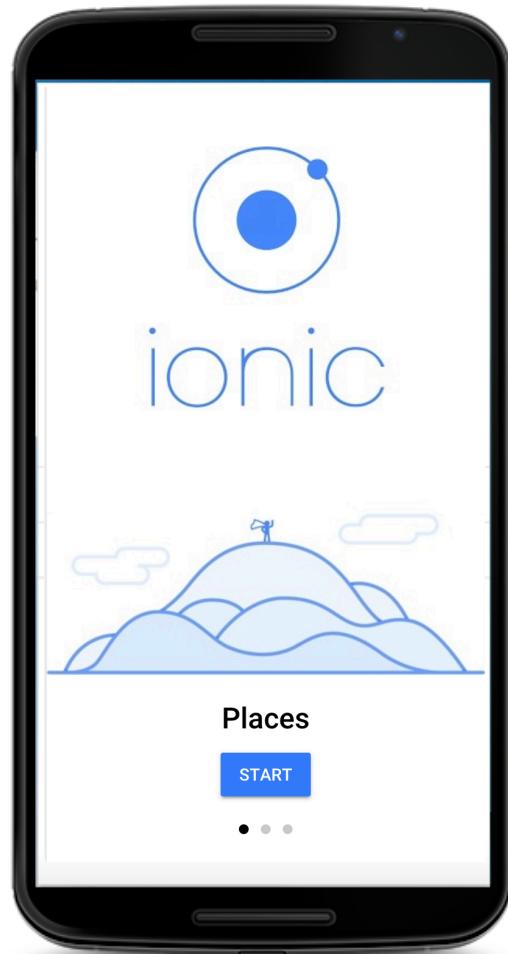


Android



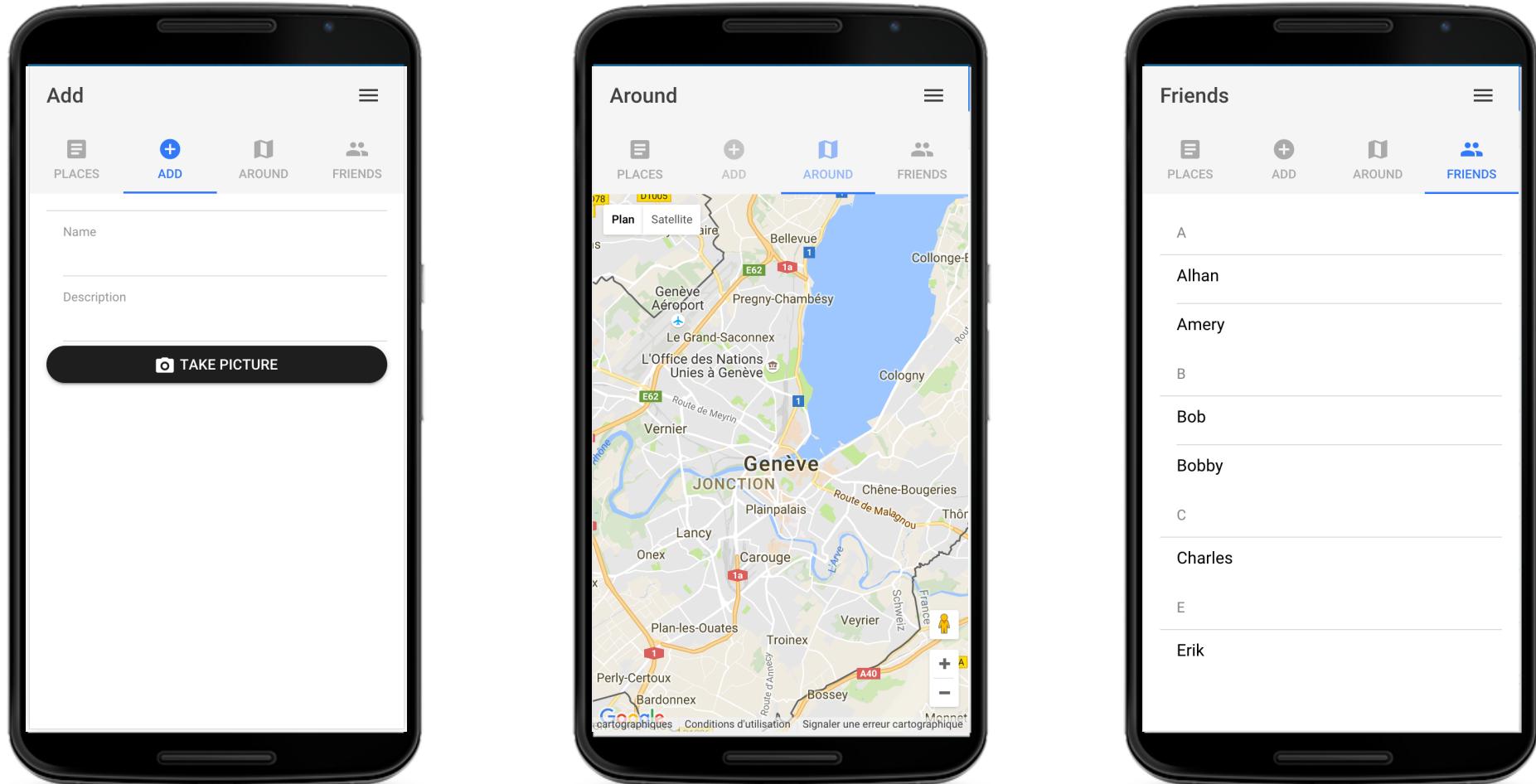
# Démarrons un nouveau projet

# Objectif:



**Prototype « Ionic Places »**

# Objectif:



**Prototype « Ionic Places »**

# Installation

## Récupérer le projet

```
$ git clone https://github.com/kimak/ionic2-places
```

A l'aise avec git et github ? « Fork me. »



---

```
$ ionic serve
```

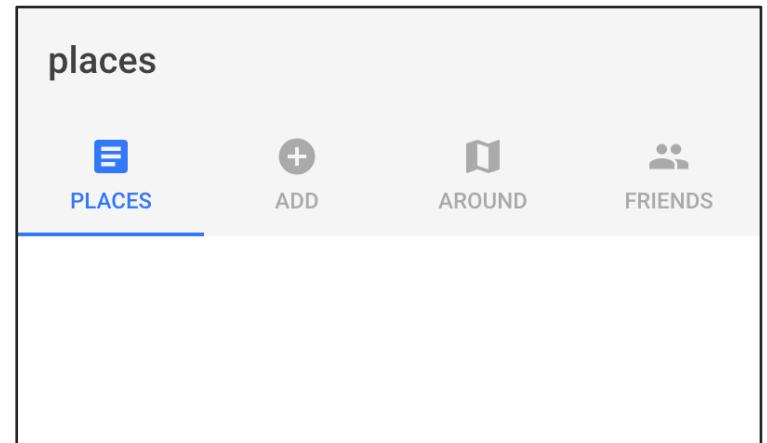
### Ionic Blank

The world is your oyster.

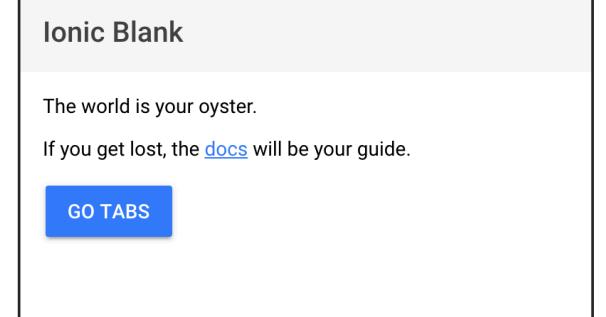
If you get lost, the [docs](#) will be your guide.

# Pratique

## 1) Mettre en place un « tabs »



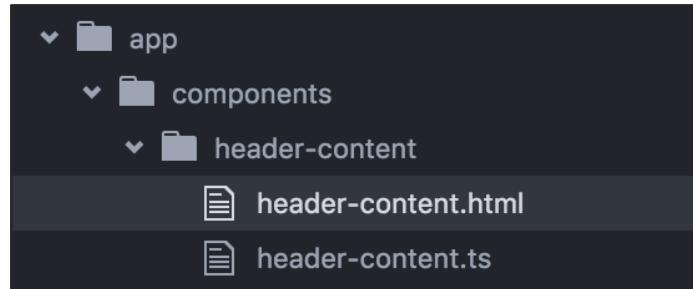
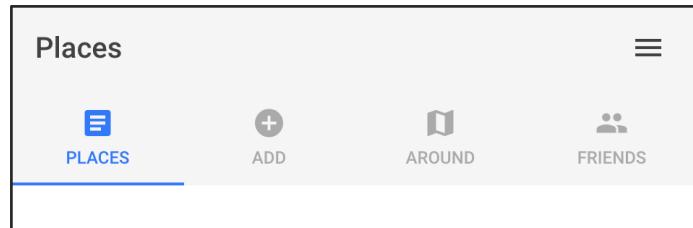
## + Mettre en place un bouton vers les tabs



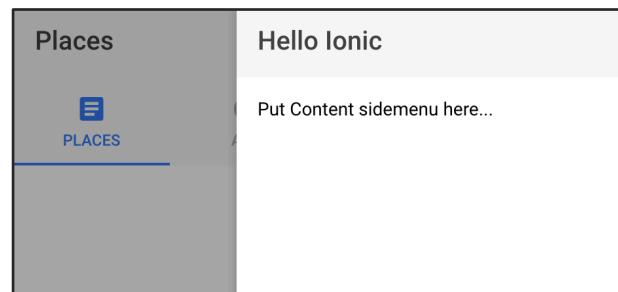
```
goTabs(){  
  this.navCtrl.push(TabsPage)  
}
```

# Pratique

## 2) Mettre en place un composant header



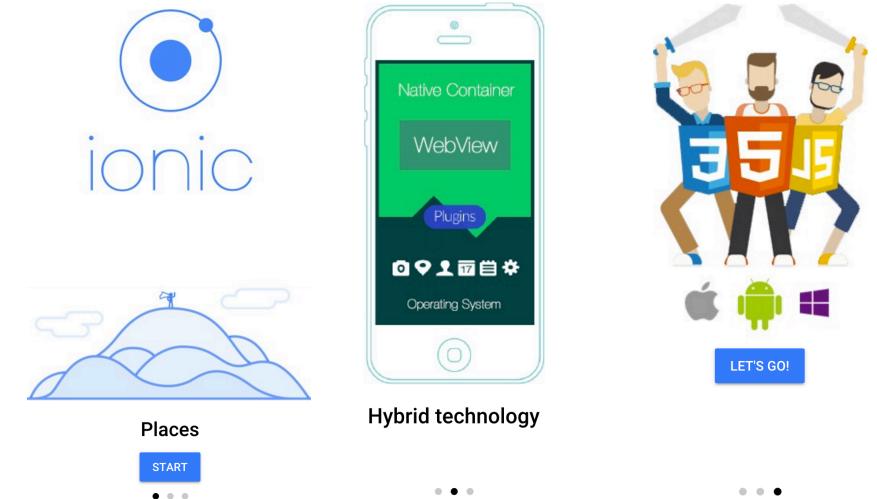
## + Mettre en place un sidemenu



```
<ion-menu [content]="content" side="right">
  <ion-content>
    <ion-toolbar>
      <ion-title>Hello Ionic</ion-title>
    </ion-toolbar>
    <div padding>
      Put Content sidemenu here...
    </div>
  </ion-content>
</ion-menu>
```

# Pratique

## 3) Mettre en place un slider sur la home »



## 4) Mettre en place une page login

A template for a login page. At the top left is a back arrow icon and the word 'login'. Below this is a light gray input field labeled 'Username'. Underneath is another light gray input field labeled 'Password'. At the bottom is a large blue rectangular button with the white text 'SIGN IN'.

# Testons sur nos téléphones



**Ionic View App**

The screenshot shows a mobile application interface titled "MY APPS". At the top, there is a small icon of an eye, the text "Carrier WiFi", the time "3:14 PM", and a battery icon. Below this, there is a list of four applications:

- Facebook**  
ID: A5D64ED3  
SIZE: 2.9 MB  
MODIFIED: FEB 18, 2014 3:45:25 PM
- Snapcat**  
ID: AFF7CB5B  
SIZE: 112.3 KB  
MODIFIED: FEB 17, 2014 2:12:32 PM
- Grumblr**  
ID: D37DD3A5  
SIZE: 8.4 MB  
MODIFIED: FEB 11, 2014 11:07:06 AM
- Slappy Bird**  
ID: F3DF48C8  
SIZE: 2.7 MB  
MODIFIED: FEB 18, 2014 10:38:01 AM