

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.

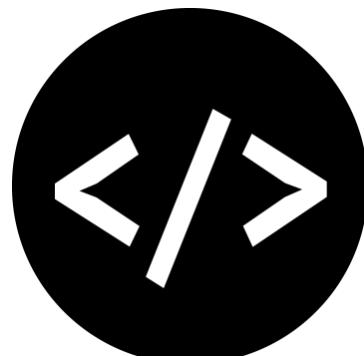


ionic



APPLICATION HYBRIDE

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



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 ?

Matthieu Gayon



Développeur Full Stack JS
Freelance

#JavaScript #ES6 #TypeScript #Sass
#Angular #Ionic
#React #React-native #Redux
#NodeJS #Git

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.

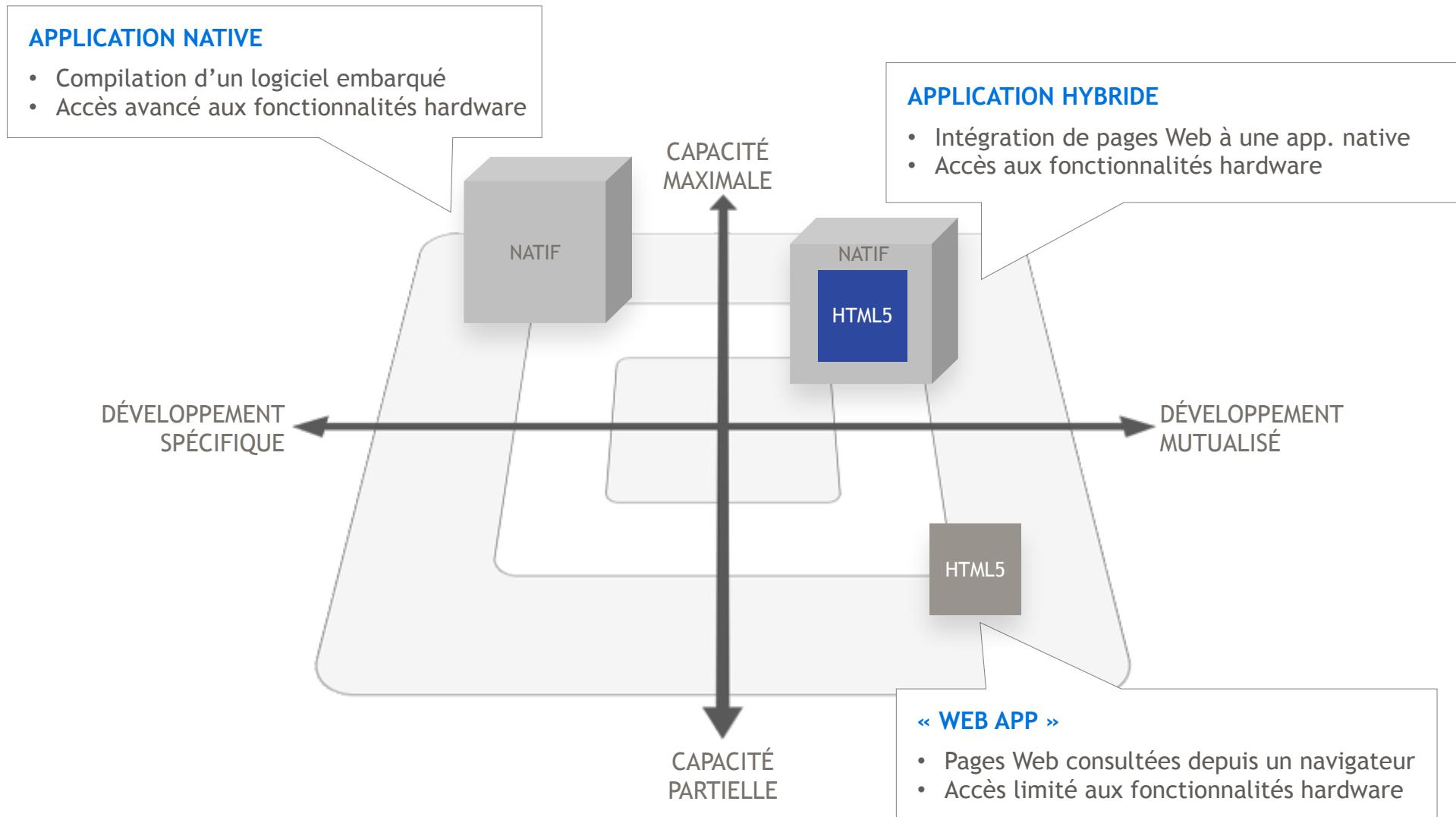
Permet d'utiliser les fonctionnalités natives du mobile et une distribution par les stores.



NATIF

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

Comment choisir ?



Application native



« C'est ultra fluide »



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

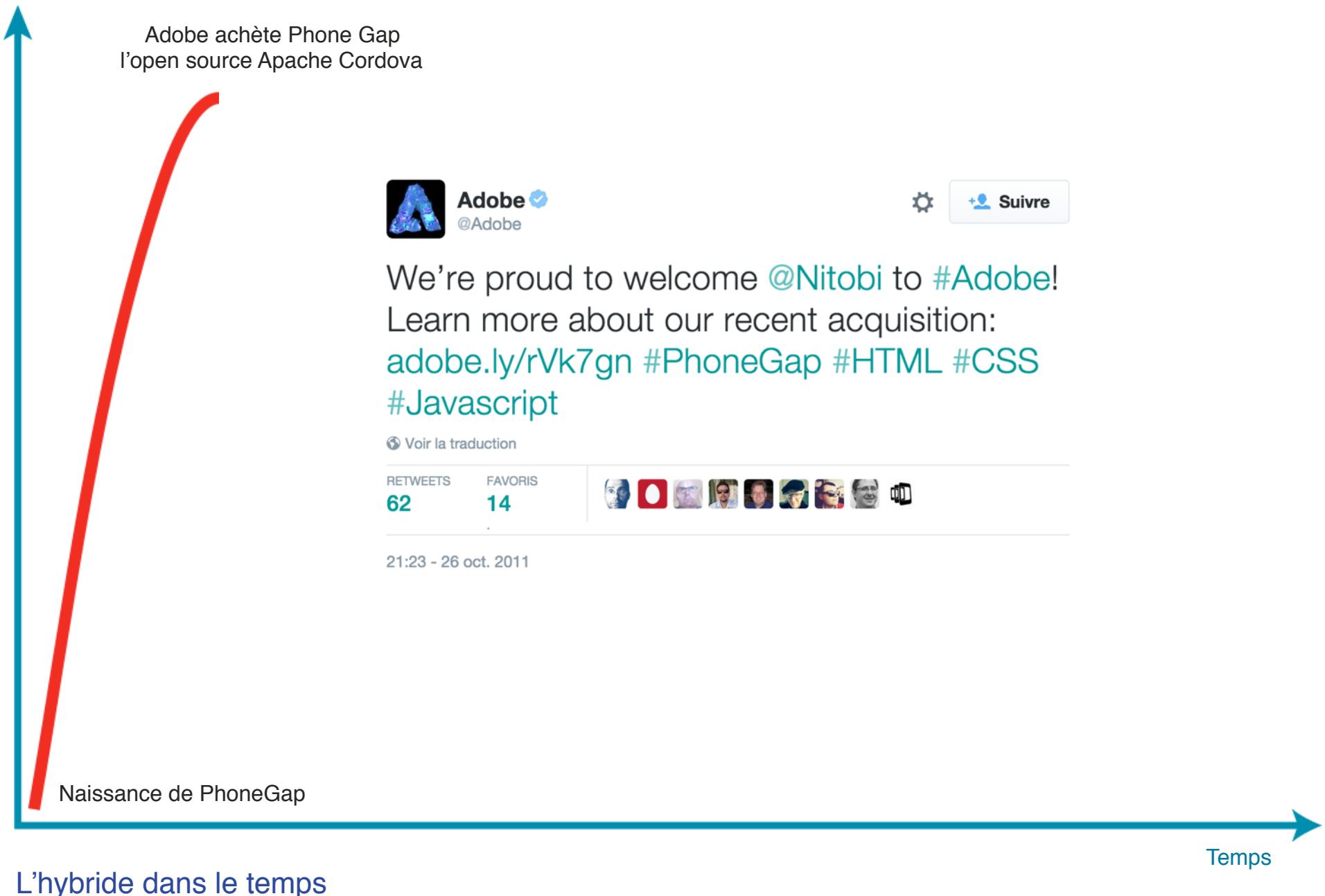
En 2009 la société Nitobi publie le framework PhoneGap

Popularité



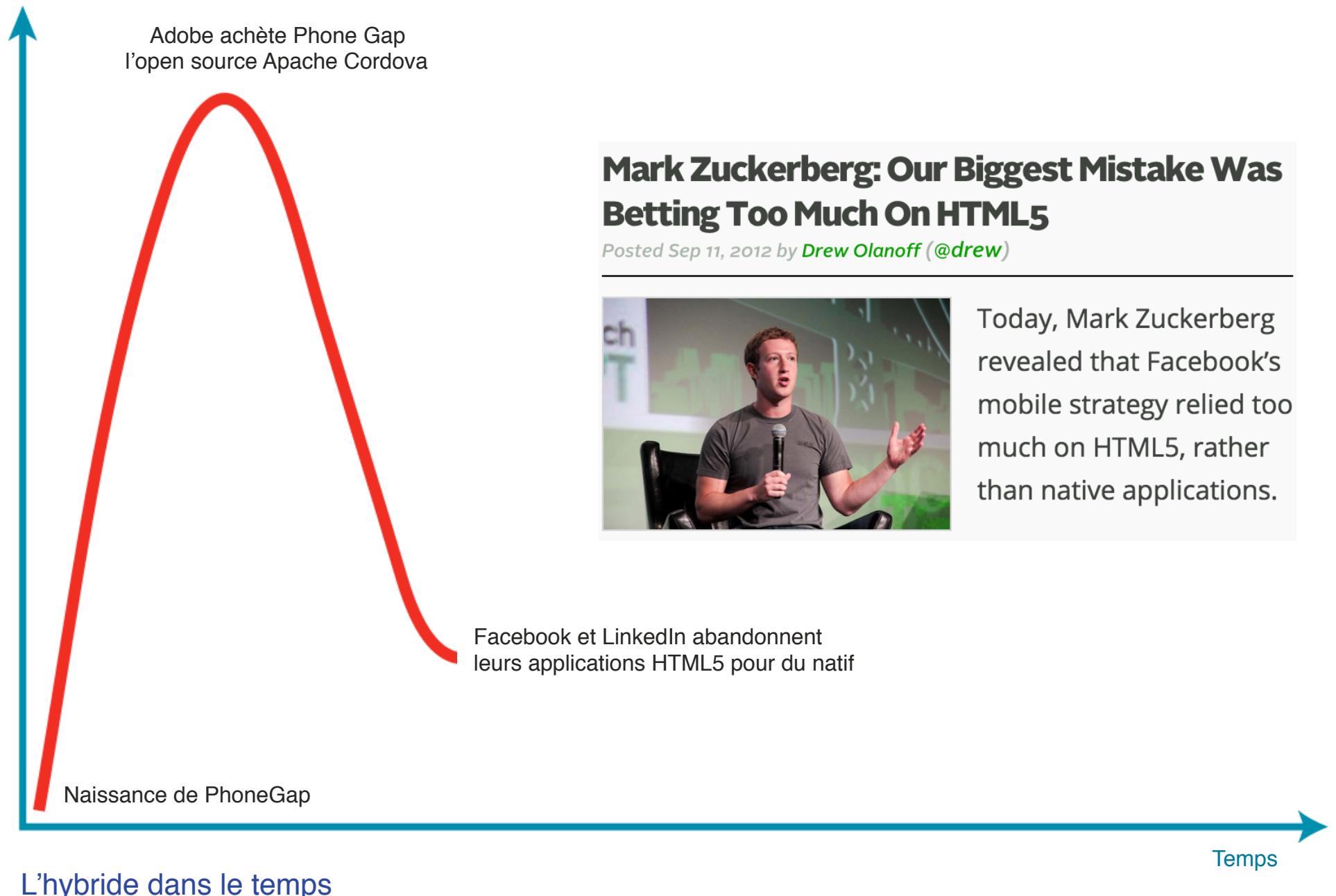
En 2011 Adobe rachète Nitobi et open source le projet

Popularité



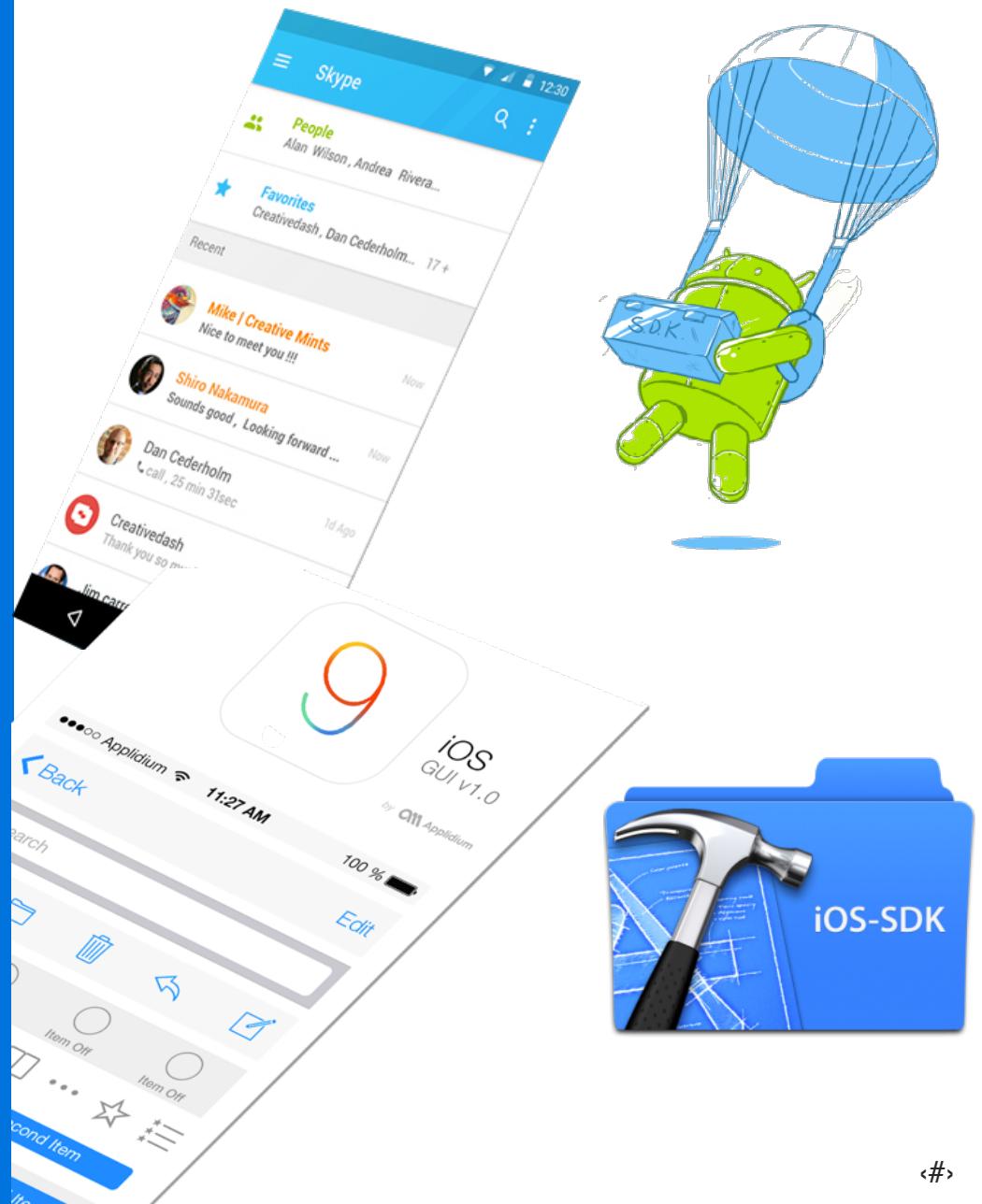
En 2012 Facebook abandonne la technologie HTML5 pour ses applications mobile

Popularité

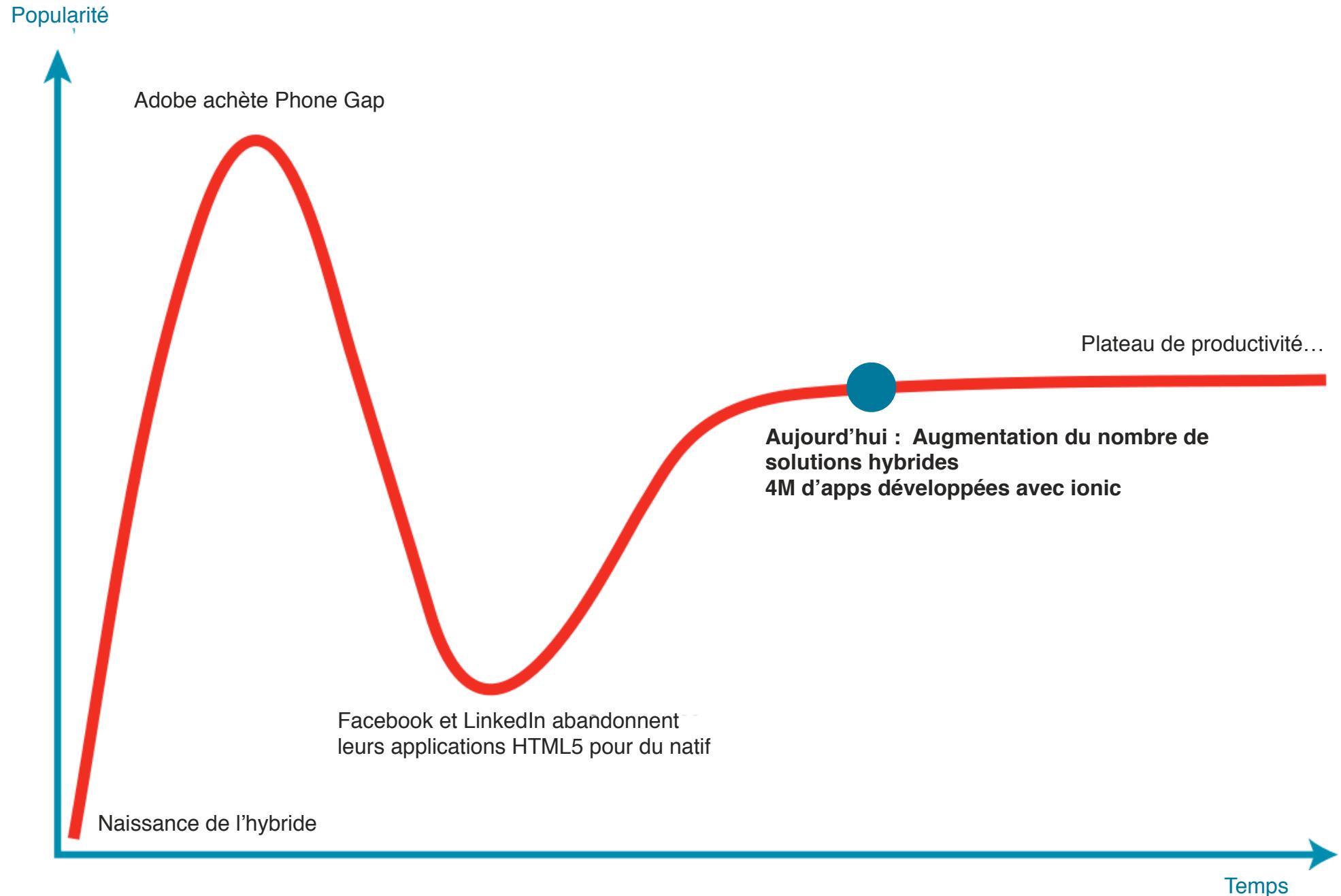


Le vrai problème ?

Aucun SDK HTML5 pour mobile !



Mais on est en 2017 : les choses ont changé !



Aperçu des solutions hybrides et alternatives



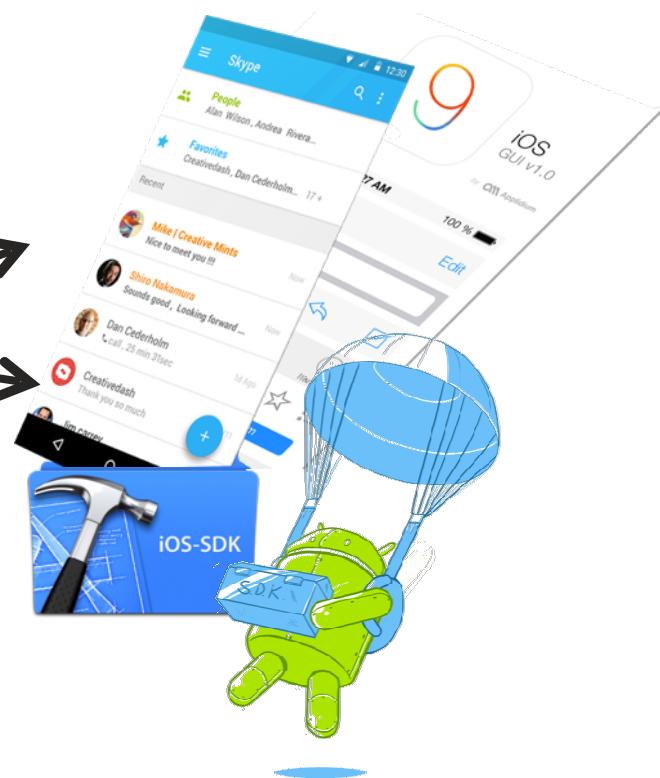
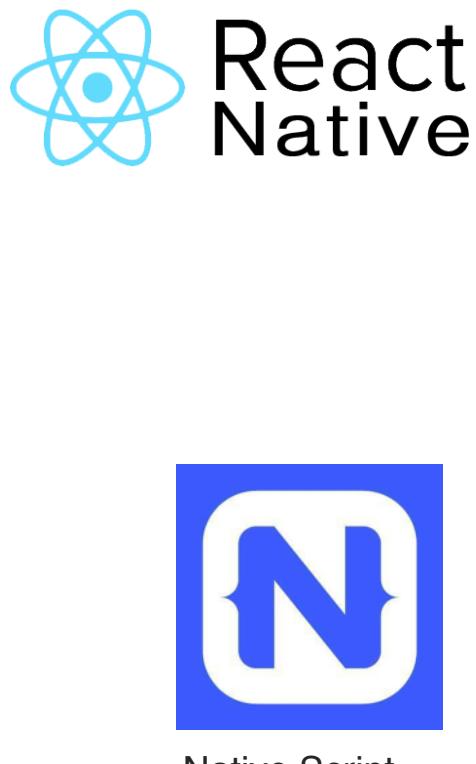


**LE MONDE SE DIVISE EN
DEUX CATÉGORIES...**

Panorama des solutions hybrides



Piloter des composants natifs en JavaScript

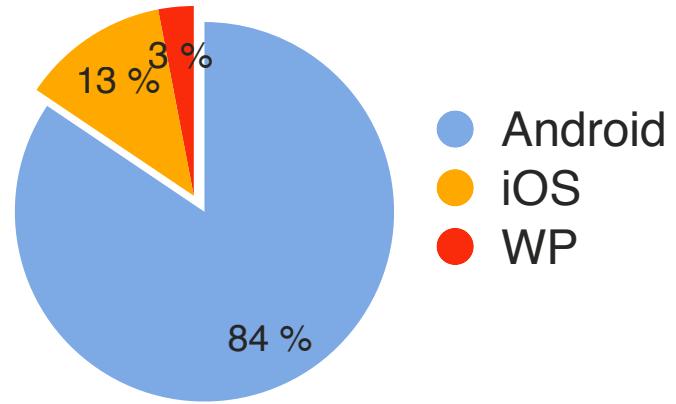


Utiliser HTML5 dans un « navigateur » embarqué



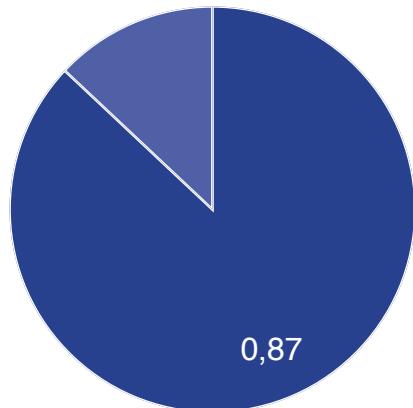
Vision actuel du marché mobile mondial

Android et iOS dominent le marché

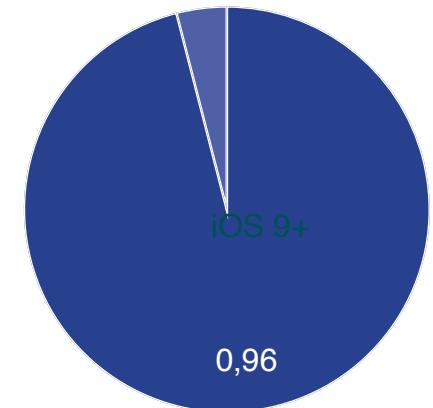


Fragmentation des OS majoritaires

Android
4.4+



iOS 9+



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.

ionic

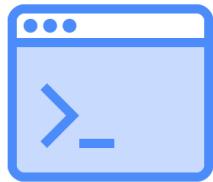
Build amazing native and progressive web apps with Angular and open web technologies. One app running on everything 

TypeScript ★ 28.8k 7.2k

- Beaucoup d'offre d'emplois et d'appels d'offres basé sur ionic
- Après avoir levé \$1 millions à ses début la société lève \$2.6 millions en 2015 et \$8.5 millions en 2016.
- Basé sur AngularJS et Typescript
- Un écosystème complet : build, application de test, extensions, plugins

Les outils Ionic

Ionic CLI



Ionic View



Ionic lab



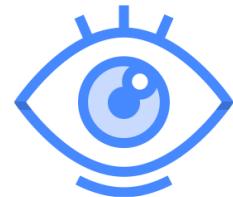
Ionic Creator



Ionic Box



Ionicons



Ionic push



Ionic native

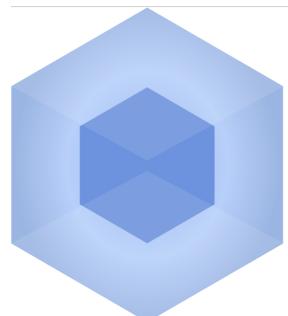


L'écosystème



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);  
});
```

TypeScript

Type compilation

```
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(() => {  
  
            });  
    }  
}
```

Angular2

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">
```

Angular2

Creating a Local Variable

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

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

Directives

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

Decorators

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

Depuis peu :)

Ionic 2 est en
version release

V2.2.2

Installation

Pré-requis

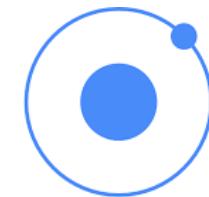
NODEJS + NPM



RUBY + SASS



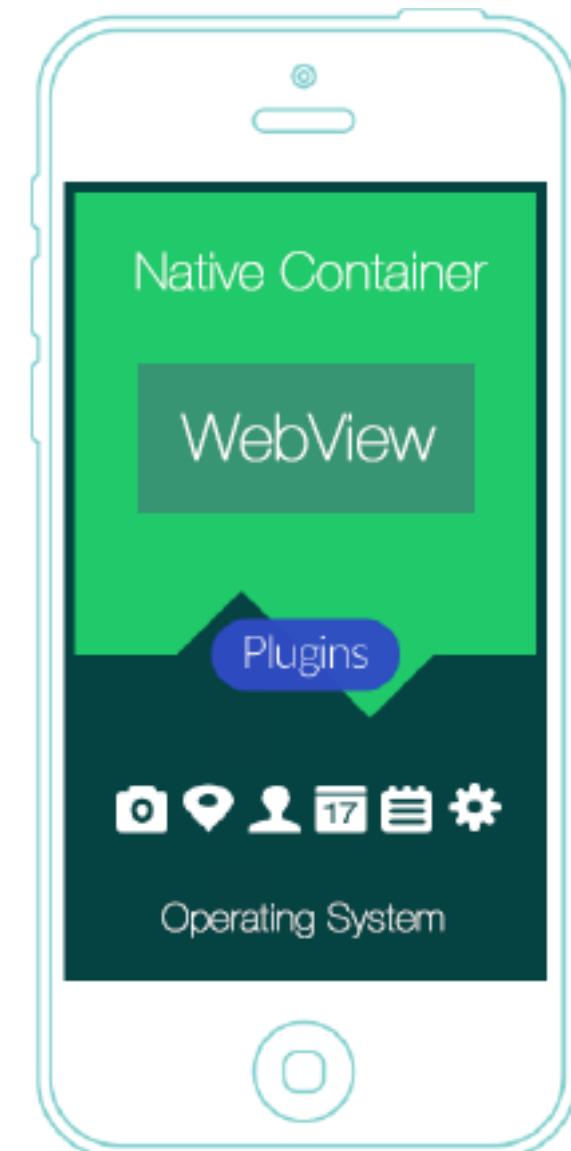
CORDOVA + IONIC



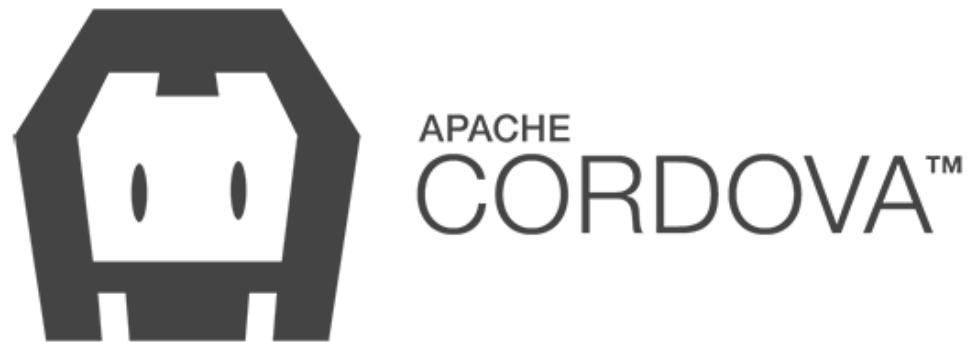
Comment fonctionne l'hybride ?

Une WebView* dans un conteneur natif

*WebView: composant natif pour afficher des pages web.



Outil de build multi-plateforme



Cordova CLI

The command line tool to build, deploy and manage Cordova-based applications.

- 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
```

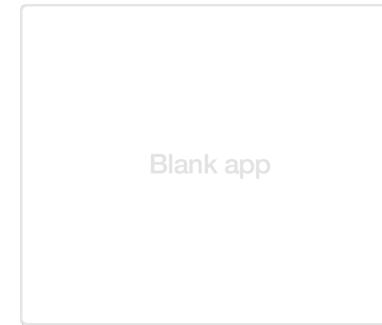
```
$ ionic help
```

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

start	Starts a new Ionic project in the specified PATH
serve	Start a local development server for app dev/testing
platform	Add platform target for building an Ionic app
run	Run an Ionic project on a connected device
emulate	Emulate an Ionic project on a simulator or emulator
build	Locally build an Ionic project for a given platform
plugin	Add a Cordova plugin
resources	Automatically create icon and splash screen resources (beta) 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.
package	Package an app using the Ionic Build service (beta)
upload	Upload an app to your Ionic account
share	Share an app with a client, co-worker, friend, or customer
lib	Gets Ionic library version or updates the Ionic library
setup	Configure the project with a build tool (beta)
browser	Add another browser for a platform (beta)
service	Add an Ionic service package and install any required plugins
add	Add an Ion, bower component, or addon to the project
remove	Remove an Ion, bower component, or addon from the project
list	List Ions, bower components, or addons in the project
ions	List available ions to add to your project
templates	List available Ionic starter templates
info	List information about the users runtime environment
help	Provides help for a certain command
link	Sets your Ionic App ID for your project
hooks	Manage your Ionic Cordova hooks
state	Saves or restores state of your Ionic Application using the package.json file
docs	Opens up the documentation for Ionic

Let's go ionic !

Créer un projet



\$ 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

The image shows a file explorer on the left and a code editor on the right. The file explorer displays the directory structure of an Ionic 2 project named 'ionic2-places'. The code editor shows the content of the 'src/index.html' file.

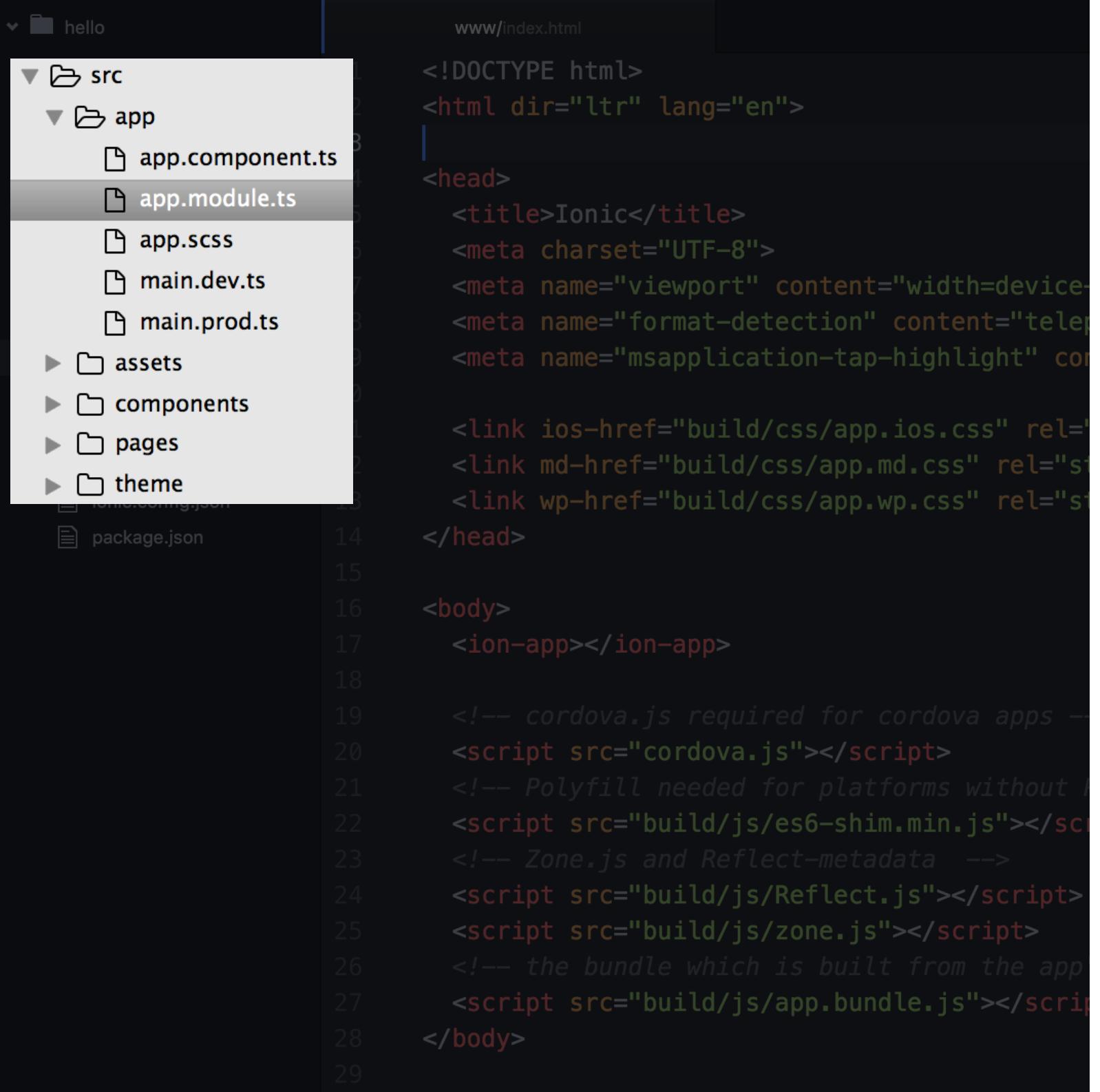
File Explorer (Left):

- 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 on the left and a code editor window on the right. The file explorer displays the directory structure of a project named 'hello'. The 'src' folder contains 'app', 'assets', 'components', 'pages', 'theme', and 'package.json'. Inside 'app', there are files: 'app.component.ts', 'app.module.ts', 'app.scss', 'main.dev.ts', and 'main.prod.ts'. The 'app.module.ts' file is currently selected. The code editor shows the content of 'index.html', which is a standard HTML template for an Ionic app. It includes the DOCTYPE declaration, HTML and head tags with meta and title elements, and a body containing an ion-app element. It also includes script tags for cordova.js, es6-shim.min.js, Reflect.js, zone.js, and app.bundle.js.

```
www/index.html
```

```
<!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 terminal window with a file tree on the left and code snippets on the right.

File Tree:

- hello
- app
 - components
 - my-component
 - my-component.html
 - my-component.ts
 - pages
 - home
 - home.html
 - home.scss
 - home.ts
 - pipes
 - theme
 - app.core.scss
 - app.ios.scss
 - app.md.scss
 - app.variables.scss
 - app.wp.scss
- hooks
- node_modules
- platforms

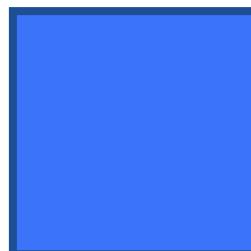
Les composants Ionic

Ionic UI scaffolding

<ion-app>



<ion-nav>



<ion-content>



<ion-toolbar>



position= "top"

<ion-toolbar>



position="bottom"

<ion-nav-bar>



<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>

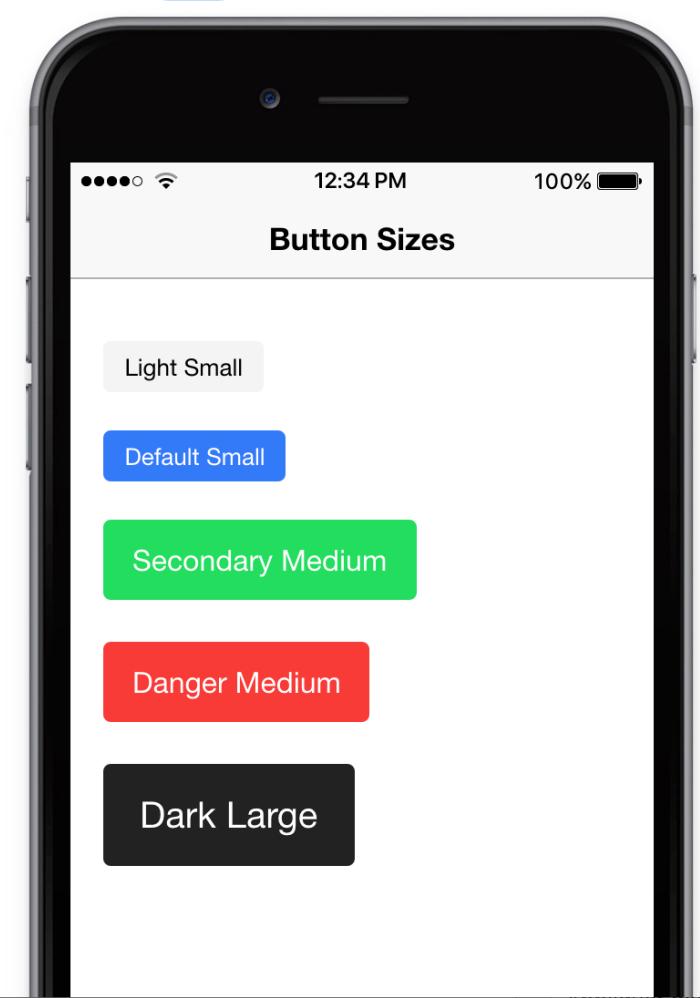
<!-- Float the icon right -->
<button>
  Right Icon
  <ion-icon name="home"></ion-icon>
</button>

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

iOS

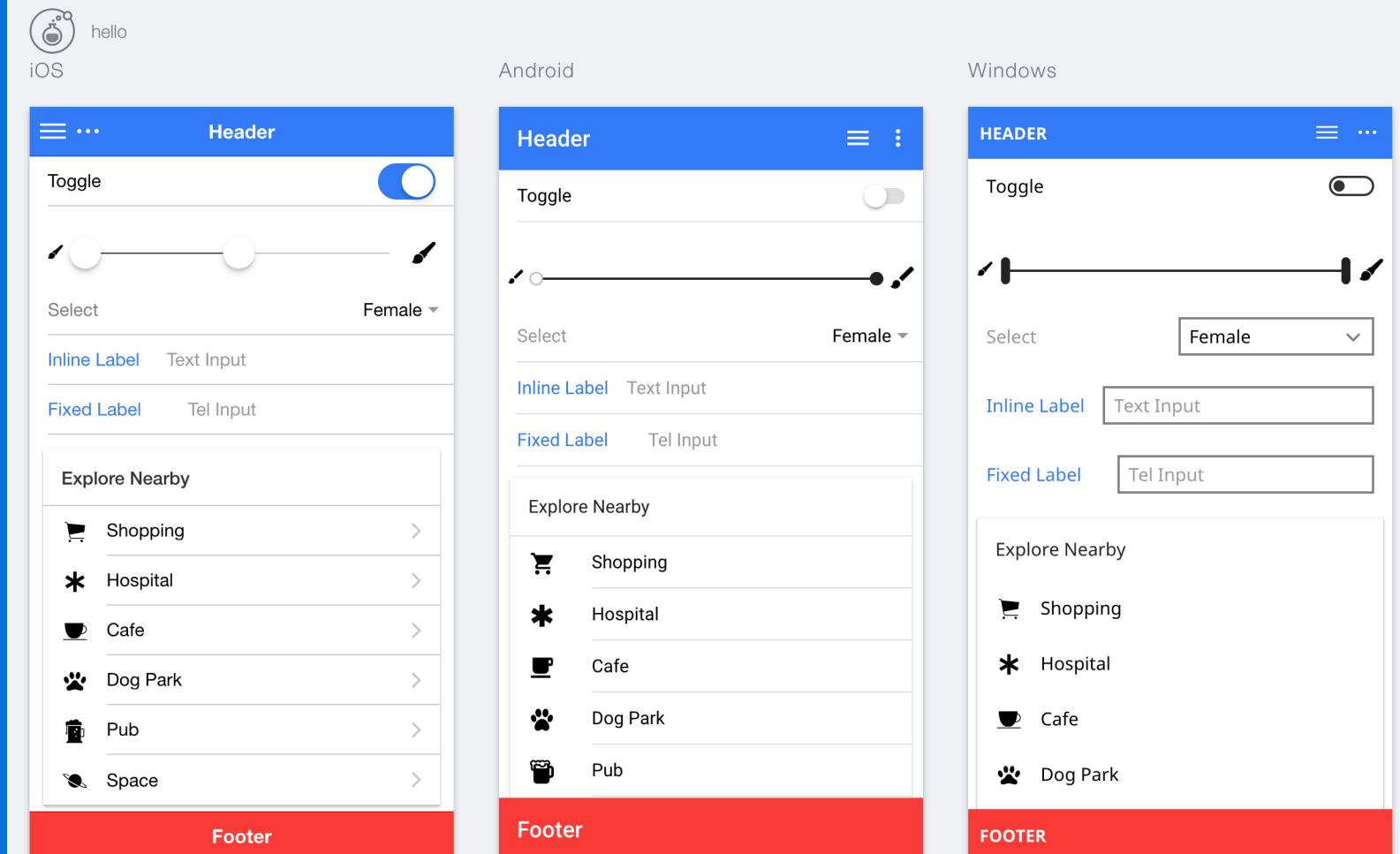
Android

Windows



Platform continuity

\$ ionic serve --lab



A vous de jouer !

Platforms config

The screenshot shows a code editor with a sidebar displaying the project structure of an Ionic application named "hello". The structure includes folders for "app", "pages", "home", "theme", "hooks", "node_modules", "platforms", "plugins", "resources", "typings", and "www". Inside "www", there are files for "index.html", ".gitignore", "config.xml", "gulpfile.js", "ionic.config.json", and "package.json". The main editor area is focused on the file "app/app.ts", which contains the following TypeScript code:

```
4 import {HomePage} from './pages/home/home';
5
6
7 @Component({
8   template: '<ion-nav [root]="rootPage"></ion-nav>'
9 })
10 export class MyApp {
11   rootPage: any = HomePage;
12
13 constructor(platform: Platform) {
14   platform.ready().then(() => {
15     // Okay, so the platform is ready and our plugins are available.
16     // Here you can do any higher level native things you need.
17     StatusBar.styleDefault();
18   });
19 }
20
21 imports: [
22   IonicModule.forRoot(MyApp, {
23     mode: 'md',
24     platforms: {
25       ios: {
26         tabsPlacement: 'top',
27       }
28     }
29   })
30 ]
```

The code defines a root component "MyApp" that sets the "HomePage" as the initial root page. It also includes a constructor that runs after the platform is ready, setting the status bar style to default. The "imports" section uses the "IonicModule.forRoot" method to configure the application, specifying "mode: 'md'" and defining "platforms" for "ios" with "tabsPlacement: 'top'".

Ionic Navigation

Ionic 2 navigation != Angular 2 Router

driftyco / ionic

Code Issues 428 Pull requests 17 Pulse Graphs

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

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

adamdbradley commented on 18 May

Ionic member

This will still be a work in progress as the new 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.

adamdbradley self-assigned this on 18 May

Labels

v2

Milestone

No milestone

Assignees

adamdbradley

6 participants



Notifications

Unsubscribe

You're receiving notifications because you commented.

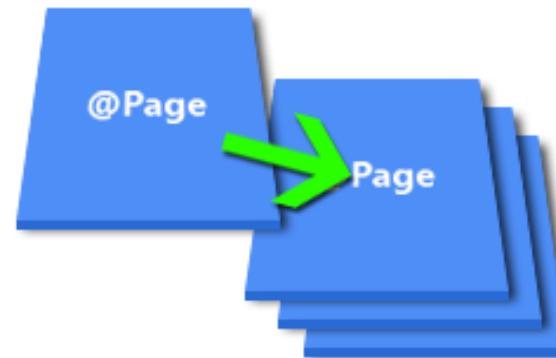
Pour le moment...

Ionic Navigation Stack

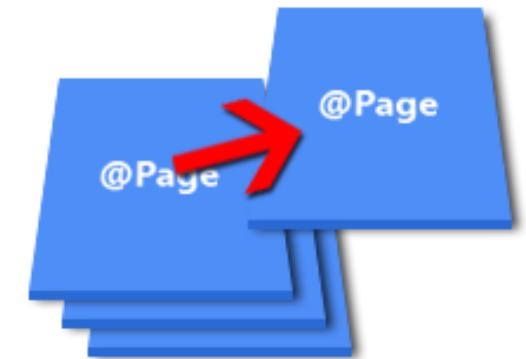
Navigation Stack



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)

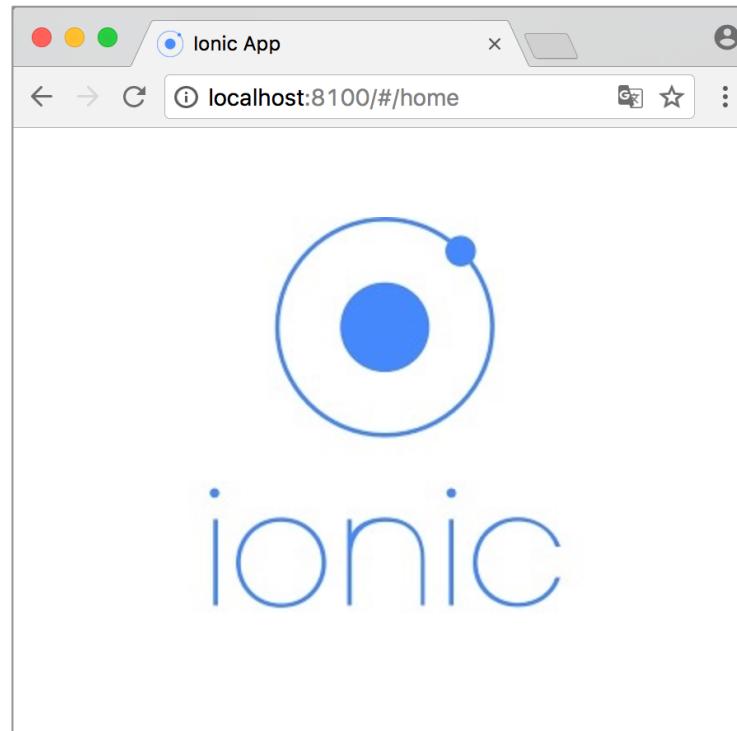
```
import {CatsPage} from '../../../../../pages/cats/cats';

@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!

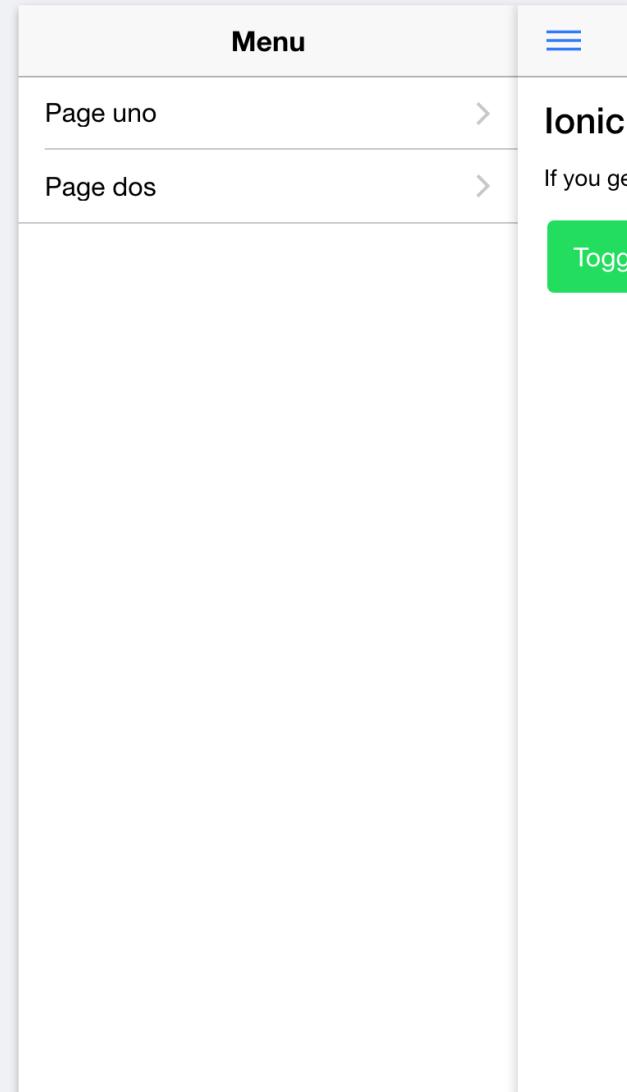
This starter project comes with simple tabs-based layout for apps that are going to primarily use a Tabbed UI.

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

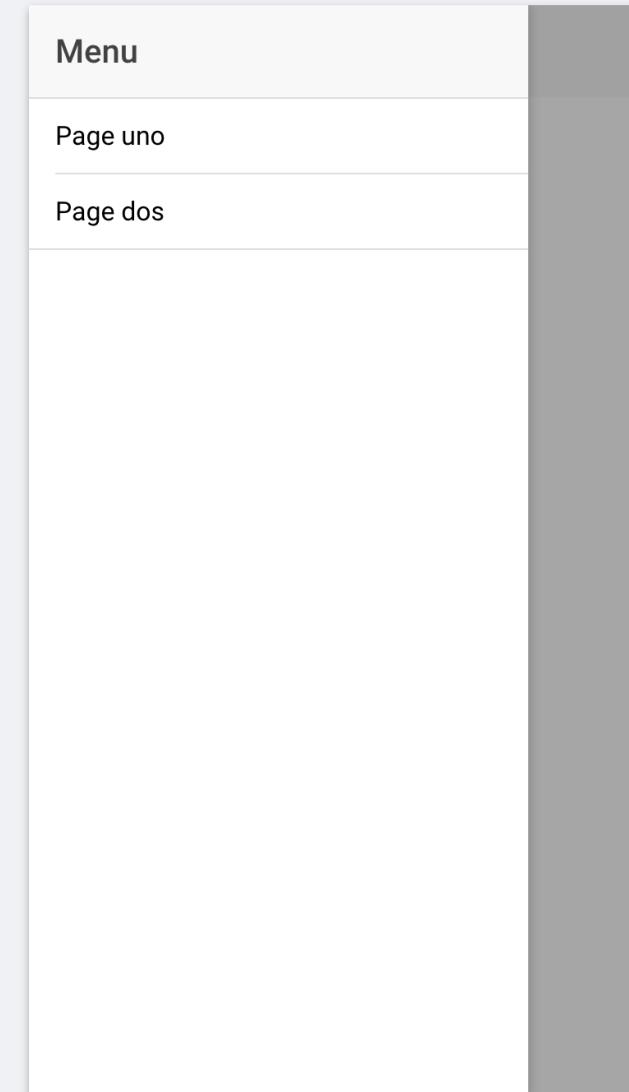
Sidemenu

\$ ionic start test-sidemenu sidemenu --lab

iOS

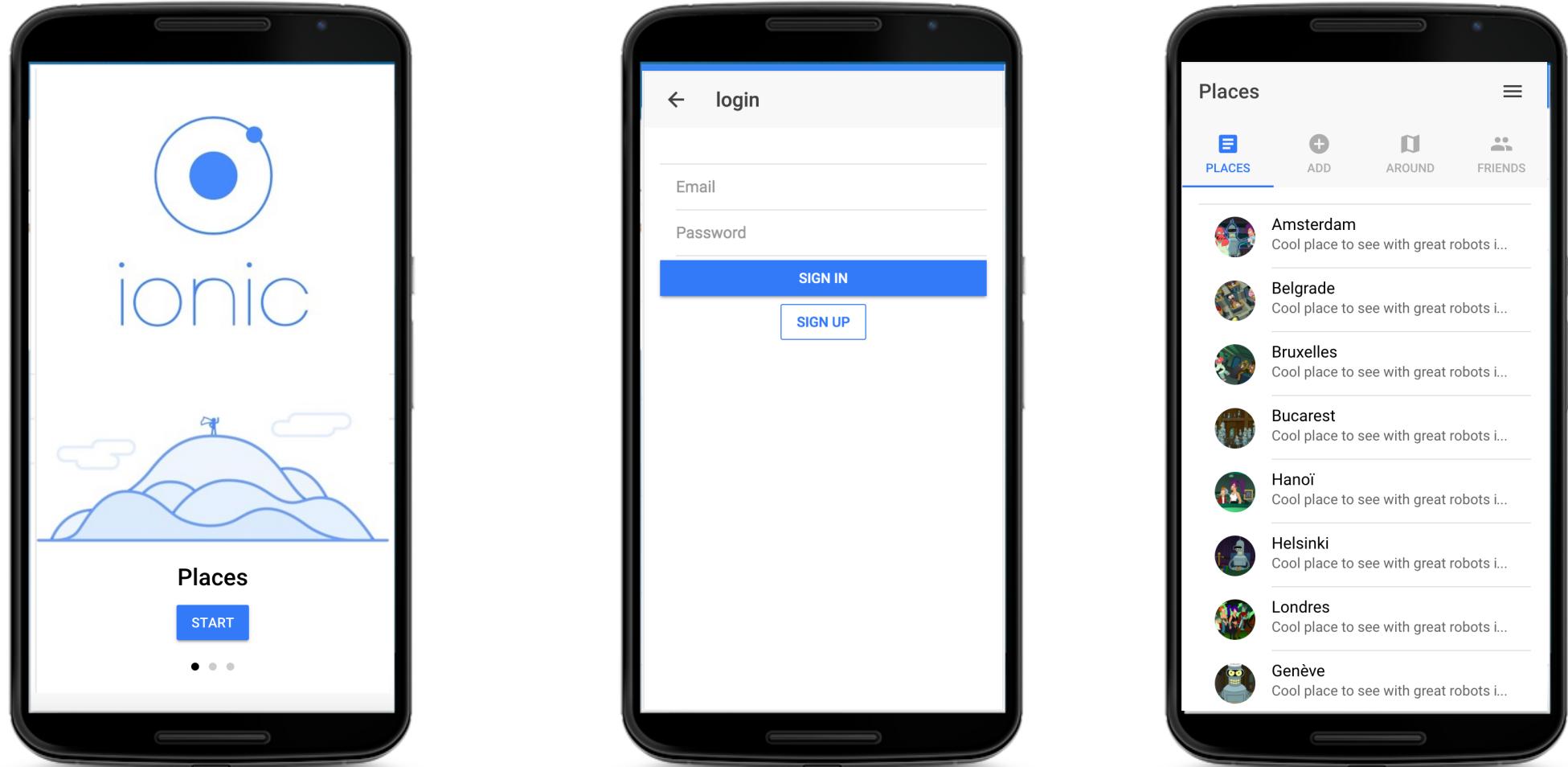


Android



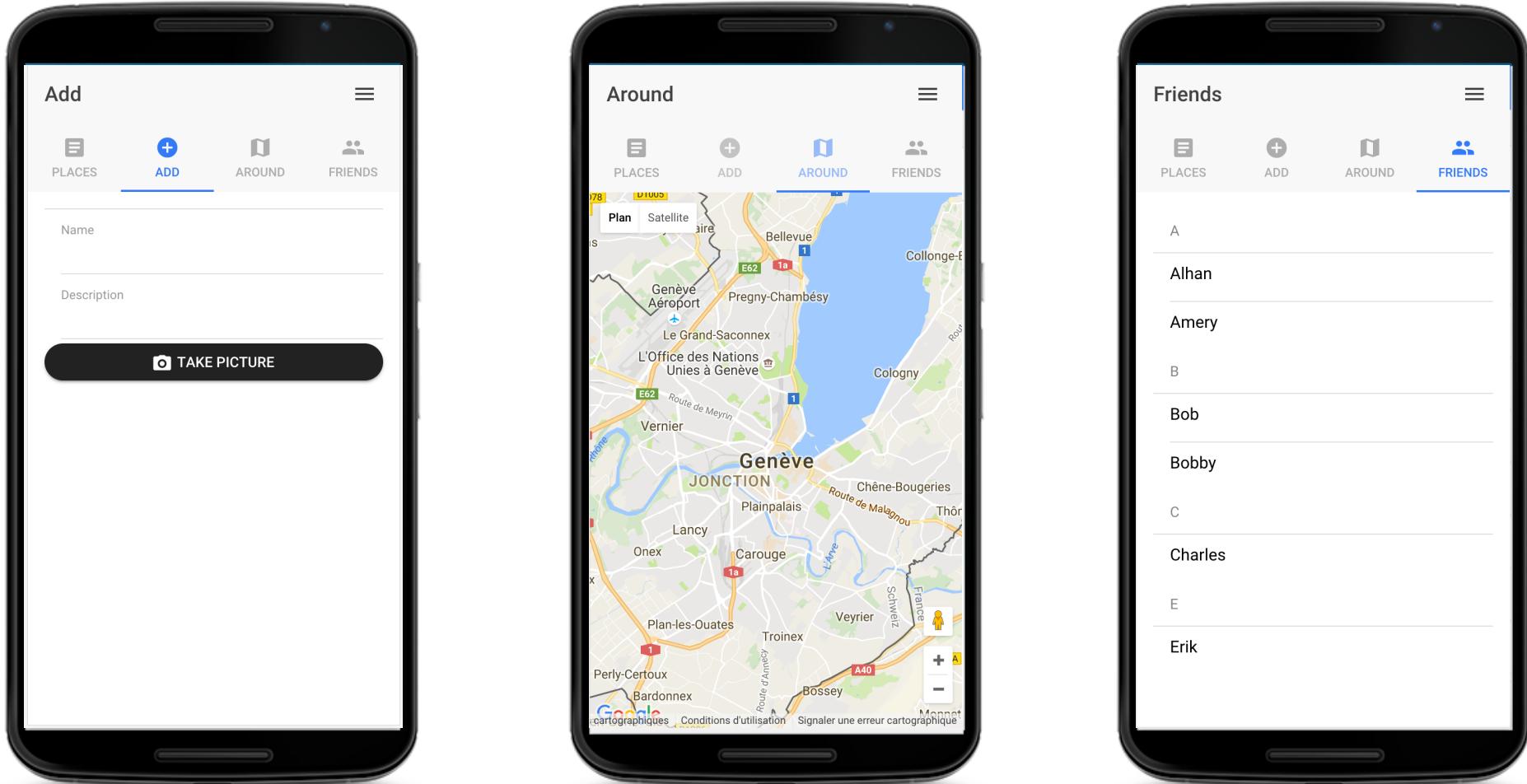
Démarrons un nouveau projet

Objectif:



Prototype « Ionic Places »

Objectif:



Prototype « Ionic Places »

Installation

Récupérer le projet

```
$ git clone git@github.com:matthieugayon/ionic-places.git
```

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



```
$ ionic serve
```

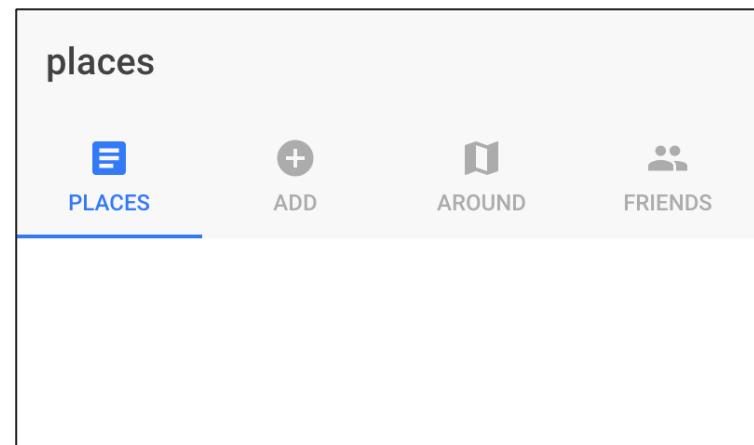
Ionic Blank

The world is your oyster.

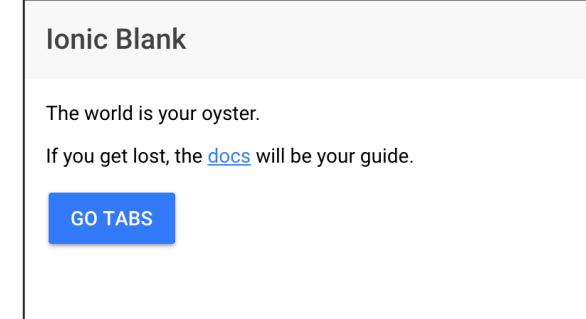
If you get lost, the [docs](#) will be your guide.

Pratique

1) Mettre en place un « tabs »



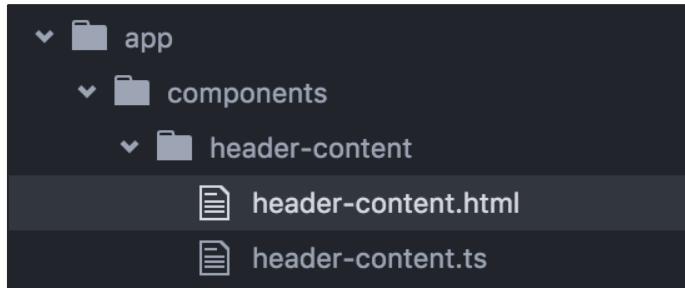
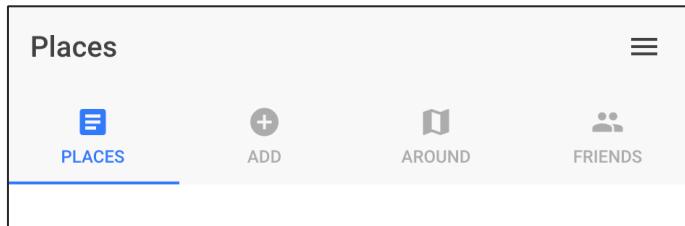
+ Mettre en place un bouton vers les tabs



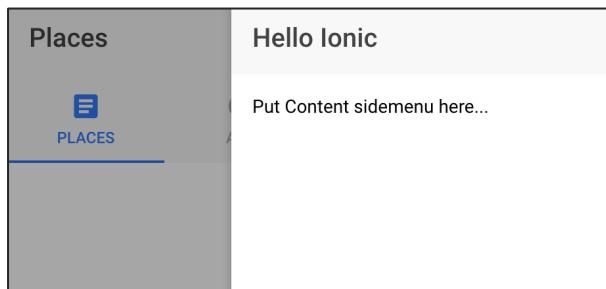
```
goTabs(){  
  this.navCtrl.push(TabsPage)  
}
```

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>
```