

# Formation Mobile Hybride

Cordova, Ionic, Firebase

# Présentation du cours



## POUR QUI ?

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



## APPLICATION HYBRIDE

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



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



#JavaScript #Angular #React  
#Cordova #Gulp/Grunt #Sass #Ionic

@MeKimak

# Agenda

1. Contexte mobile hybride
2. Installation des outils
3. Les composants Ionic
4. Système de 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.

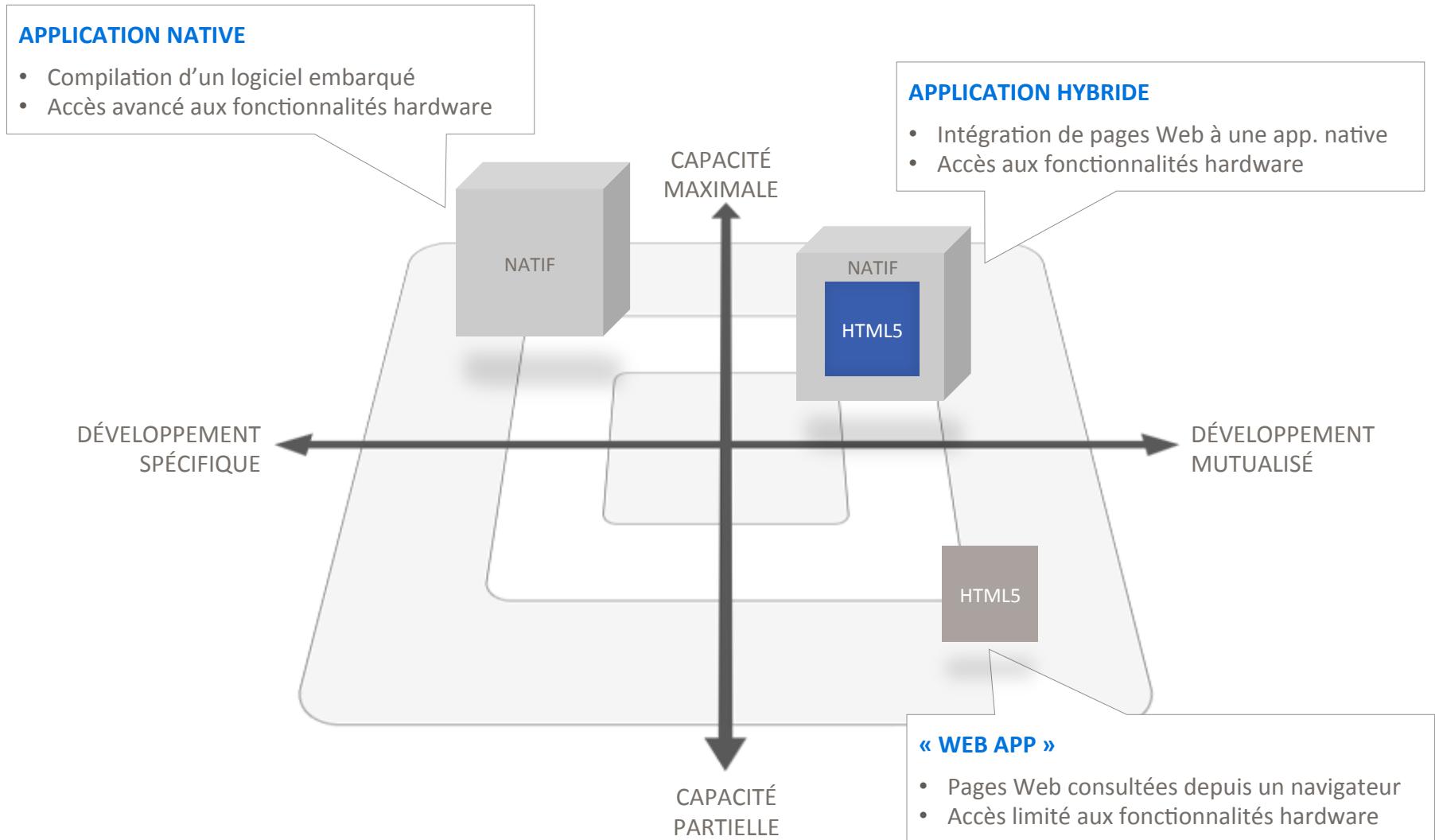
Permet d'utiliser les fonctionnalités natives du mobile et une distribution par les stores.



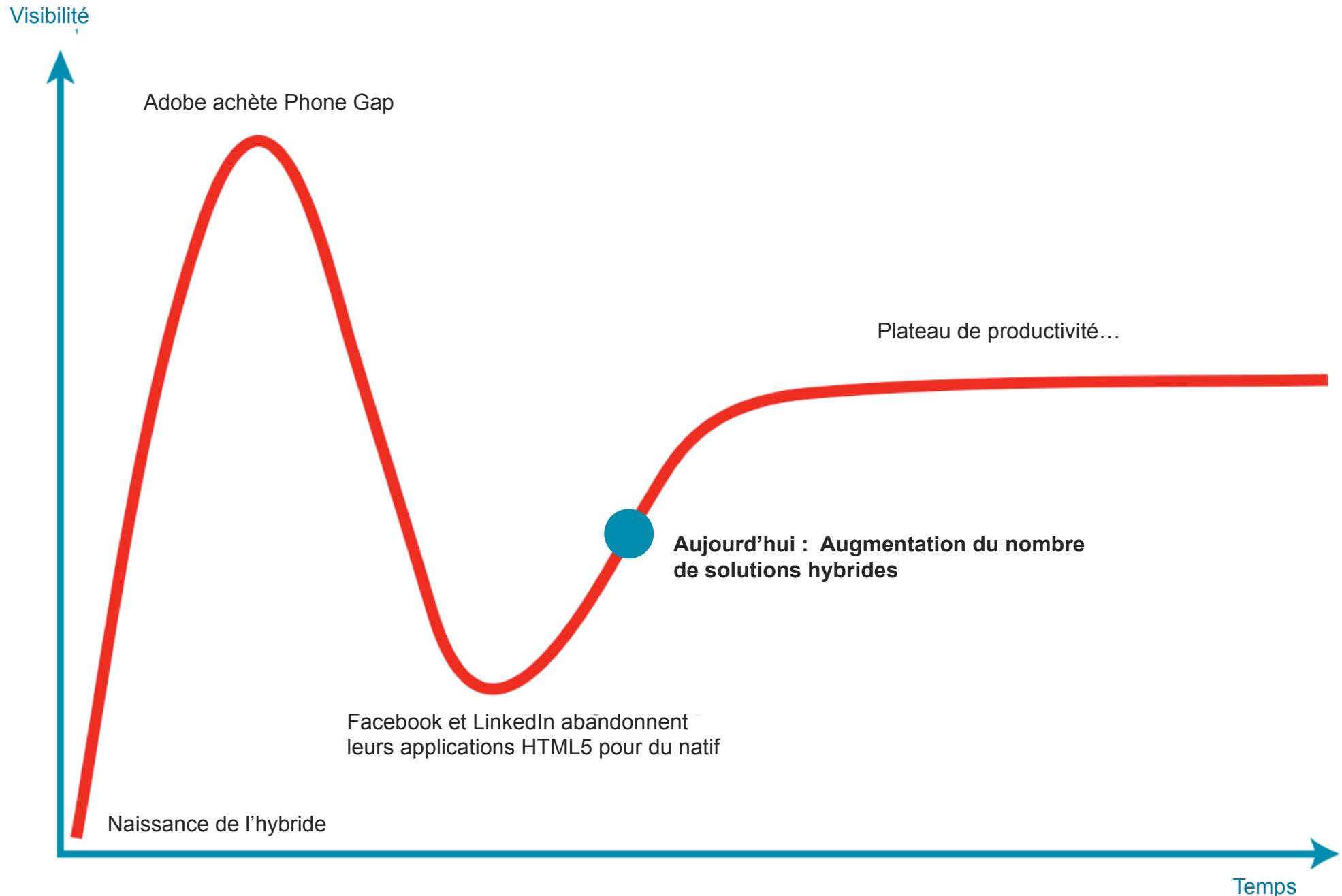
## NATIF

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

# Comment choisir ?



# Cycle de vie de l'hybride (Gartner Hype Cycle)



# Aperçu des solutions hybrides et alternatives



# Comparaison

Mobile Hybrid				 APACHE CORDOVA™			 doc		
 ionic	✓	✓	~	✓	✓	~	+++	free	open-source
 NativeScript	✓	✓	~	✓	✓	✗	++	free	open-source
 React Native	✓	✗	✗	✗	✗	✓	+	free	open-source
 appcelerator® titanium	✓	✓	✗	✗	✗	✗	++	\$259/mo Team	owner
 Onsen UI	✓	✓	✗	✓	✓	~	++	free	open-source
 famous.us	✓	✓	~	~	✓	~	-	free	open-source
 TouchstoneJS	✓	~	✗	✓	✗	✓	-	free	open-source
 Supersonic	~	~	✗	✗	✓	✓	---	free	open-source

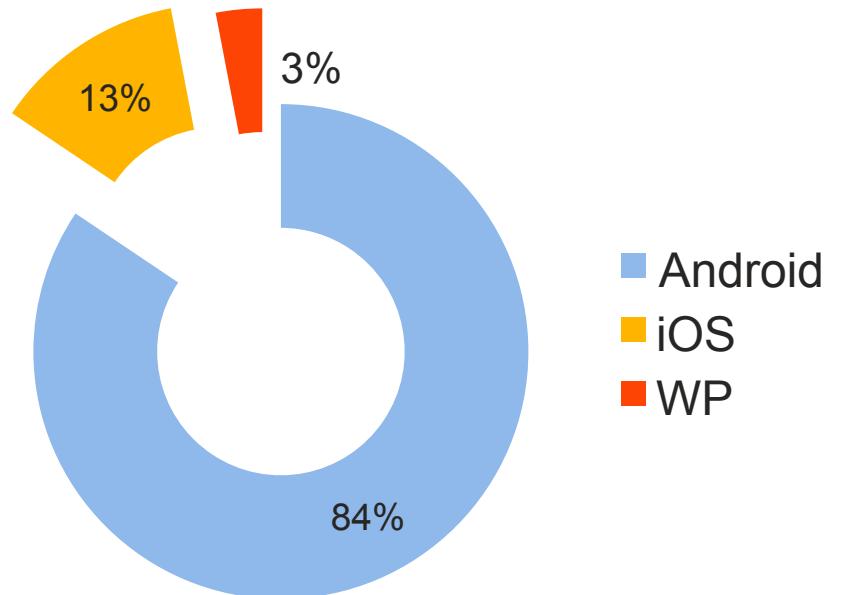
Operating System

Tooling

Community

# Vision actuel du marché mobile mondial

Android et iOS dominent le monde



Fragmentation des OS majoritaires



# IOS adoption

Sep 17th, 2014 – Feb 5th, 2015 **DONE** Hour Day



Time/Date in UTC

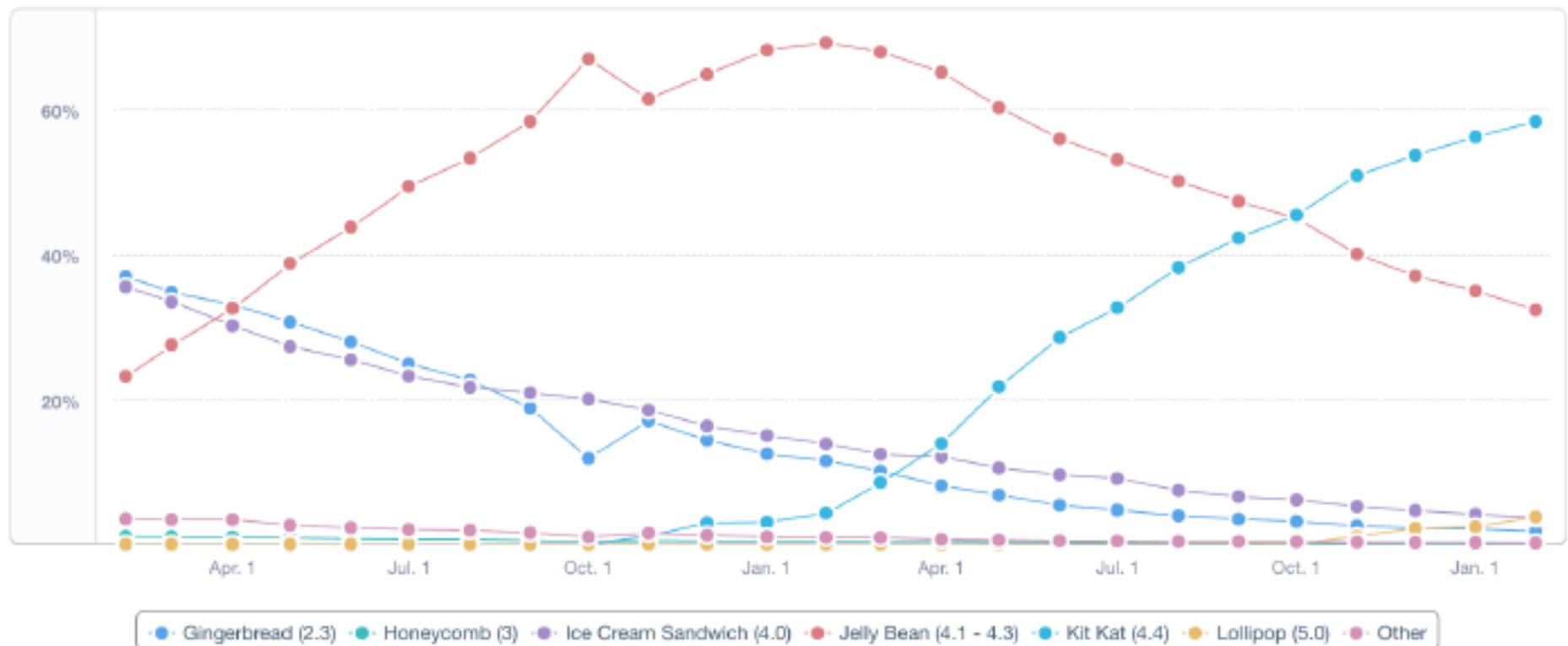
THIS REPORT WAS GENERATED FROM 210,511,475,074 RECORDS.

# Android adoption

Feb 5th, 2013

– Feb 5th, 2015

DONE



Time/Date in UTC

THIS REPORT WAS GENERATED FROM 755,825,497,354 RECORDS.

‘’



ionic

*Ionic is not a good solution if you need to support older generation devices. Our compatibility starts at iOS 6 and Android 4.1. We will never support versions earlier than those. This is a framework for the future.*

’’

# 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 début 2015.
  - Certaines offres d'emplois parlent d'Ionic.

- 
- **Basé sur AngularJS**

- **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



ngCordova



# L'écosystème



## Apache Cordova

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



## Gulp

Outil d'automatisation « task runner » il permet d'écrire des tâches récurrentes de développement en JavaScript.



## AngularJs

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



## Sass

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

# Installation

# Pré-requis

NODEJS + NPM



---

RUBY + SASS



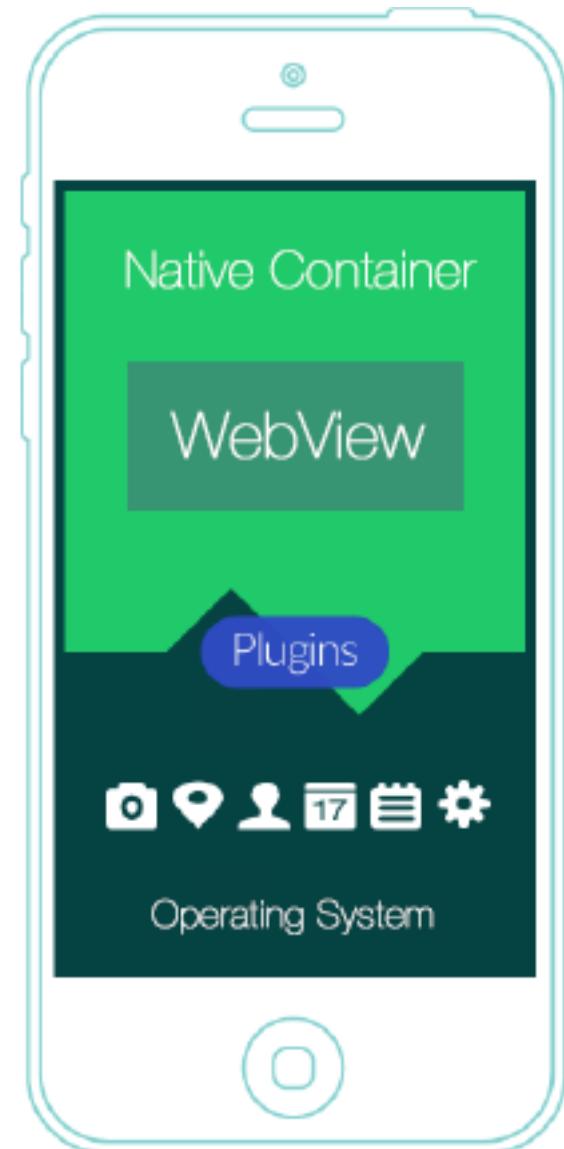
---

CORDOVA + IONIC



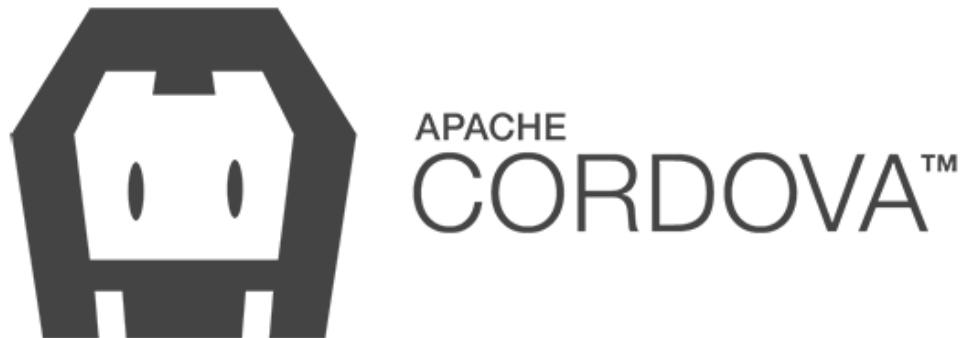
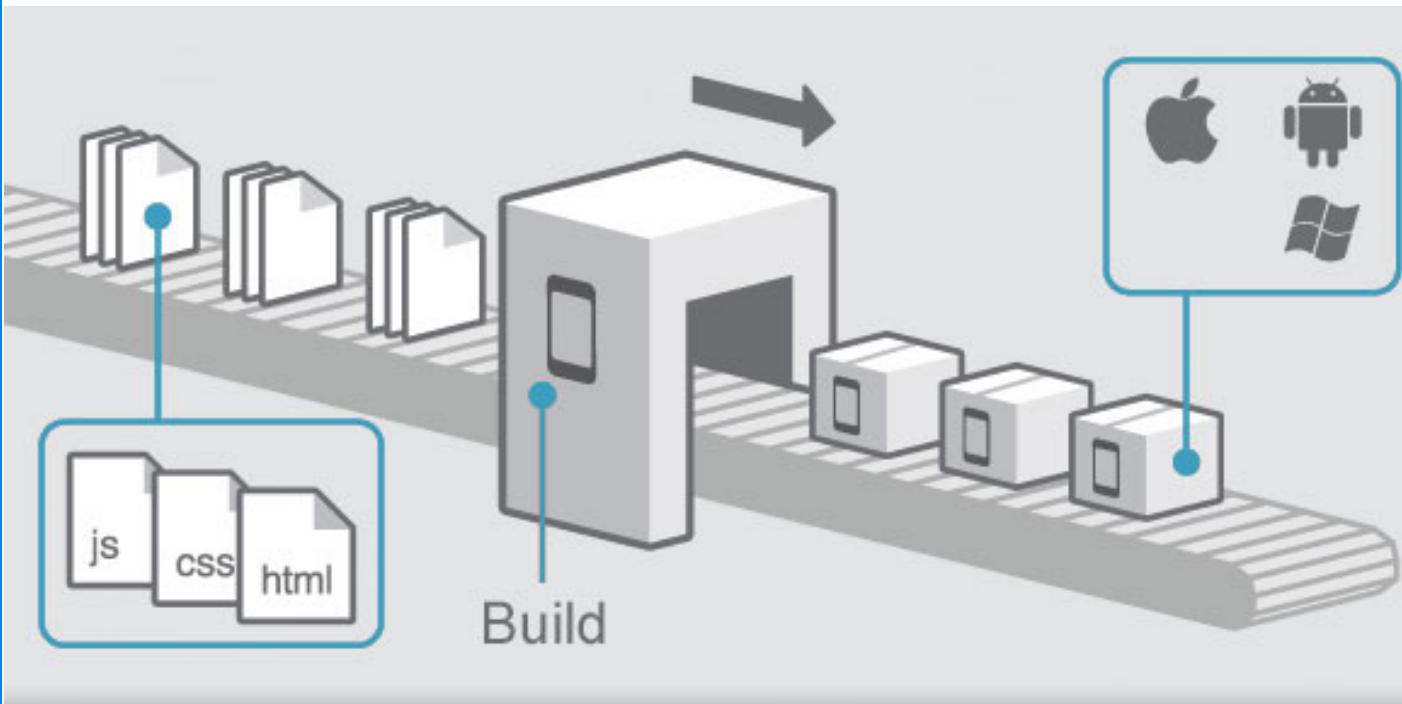
# Comment fonctionne l'hybride ?

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
```

- Créer un projet

```
$ cordova create hello com.example.hello
```

---

- 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
```

- Créer un projet

```
$ ionic start hello
```

---

- Ajouter une nouvelle plateforme

```
$ ionic platform add ios
```

```
$ ionic platform add android
```

```
$ ionic platform add wp8
```

---

- Compiler

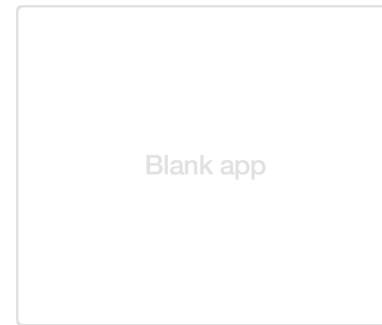
```
$ ionic build android
```

```
$ ionic emulate android
```

```
$ ionic run android
```

# Let's go ionic !

Créer un projet



\$ ionic start hello blank

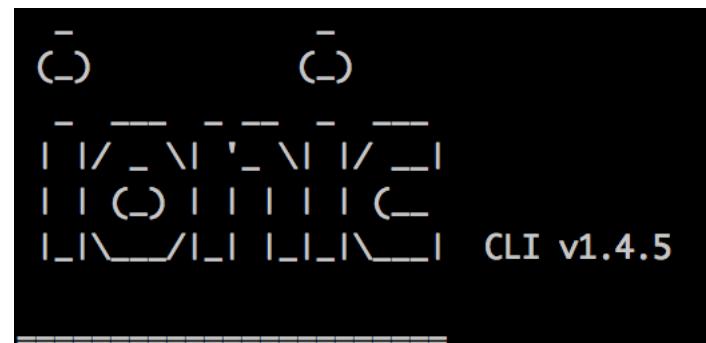
---

Lancer le projet dans le navigateur



\$ ionic serve --livereload

---



\$ ionic help

```
$ 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

# Structure d'un projet

FOLDERS

- ▶ tabs
- ▶ hooks
- ▶ platforms
  - ▶ ios
  - platforms.json
- ▶ plugins
- ▶ scss
  - ionic.app.scss
- ▶ www
  - ▶ css
  - ▶ img
  - ▶ js
  - ▶ lib
    - ▶ ionic
      - ▶ css
      - ▶ fonts
      - ▶ js
      - ▶ scss
      - version.json
  - ▶ templates
    - chat-detail.html
    - tab-account.htm
    - tab-chats.html
    - tab-dash.html
    - tabs.html
- index.html
- .bowerrc
- .gitignore
- bower.json
- config.xml
- gulpfile.js
- ionic.project
- package.json

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="initial-scale=1, maximum-scale=1, user-scalable=no,
6      <title></title>
7
8      <link href="lib/ionic/css/ionic.css" rel="stylesheet">
9      <link href="css/style.css" rel="stylesheet">
10
11     <!-- IF using Sass (run gulp sass first), then uncomment below and remove the CSS
12     <link href="css/ionic.app.css" rel="stylesheet">
13     -->
14
15     <!-- ionic/angularjs js -->
16     <script src="lib/ionic/js/ionic.bundle.js"></script>
17
18     <!-- cordova script (this will be a 404 during development) -->
19     <script src="cordova.js"></script>
20
21     <!-- your app's js -->
22     <script src="js/app.js"></script>
23     <script src="js/controllers.js"></script>
24     <script src="js/services.js"></script>
25   </head>
26   <body ng-app="starter">
27     <!--
28       | The nav bar that will be updated as we navigate between views.
29     -->
30     <ion-nav-bar class="bar-stable">
31       <ion-nav-back-button>
32         </ion-nav-back-button>
33     </ion-nav-bar>
34
35     <!--
36       | The views will be rendered in the <ion-nav-view> directive below
37       | Templates are in the /templates folder (but you could also
38       | have templates inline in this html file if you'd like).
39     -->
40     <ion-nav-view></ion-nav-view>
41   </body>
42 </html>

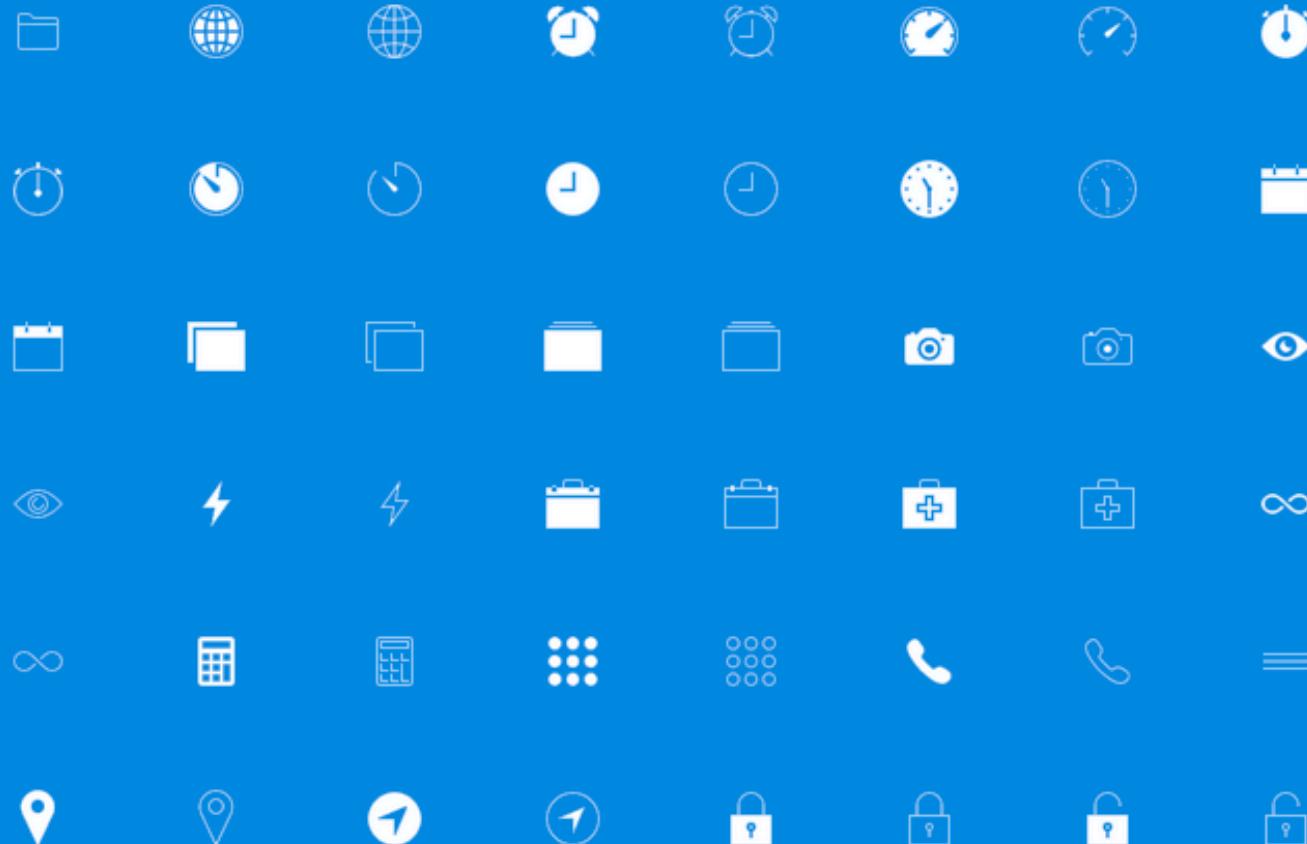
```

Line 39, Column 34

# Les composants Ionic

# *UI Component Overview*

# Ionicons



Over 700 MIT licensed font-icons included  
[ionicons.com](http://ionicons.com)

# CSS Components

Reusable and customizable front-end UI elements

Need icons? Check out Ionicons, our custom-built icon font made specifically for Ionic.

Search

## OVERVIEW

### CSS

Header

Content

Footer

#### Buttons

Block

Full Width

#### Different Sizes

Outlined

Clear

Icons

Headers/Footers

Clear Buttons in  
Headers

Button Bar

List

Cards

Forms

Toggle

Checkbox

Radio Buttons

Range

Select

Tabs

Grid

```
<button class="button button-full button-positive">  
  Full Width Block Button  
</button>
```

## Different Sizes

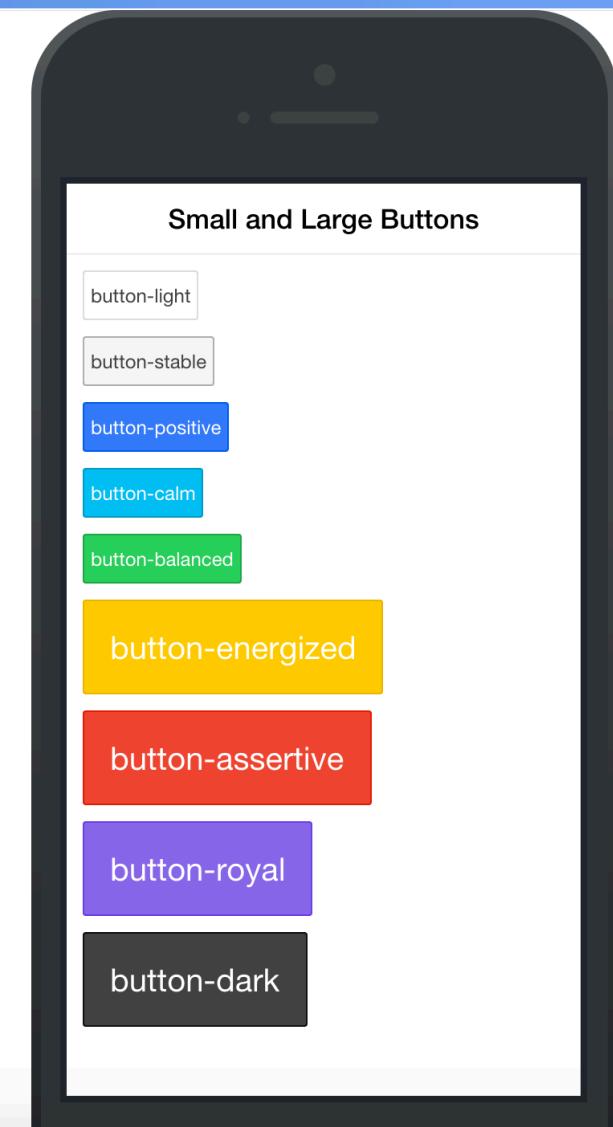
Adding `button-large` to a button makes it larger, adding `button-small` makes it smaller.

```
<button class="button button-small button-assertive">  
  Small Button  
</button>  
<button class="button button-large button-positive">  
  Large Button  
</button>
```

## Outlined Buttons

Use `button-outline` to apply an outline button style, which also has a transparent background.

**Note:** The text and border will take the color of the applied button style, meaning `button-positive` will result in blue text

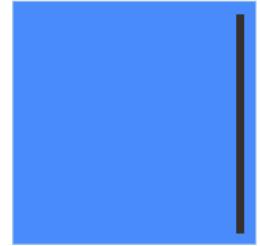


# Ionic UI scaffolding

<ion-pane>



<ion-content>



<ion-header-bar>



<ion-footer-bar>



<ion-nav-bar>



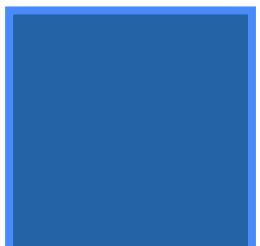
<ion-nav-buttons>



<ion-nav-view>



<ion-view>

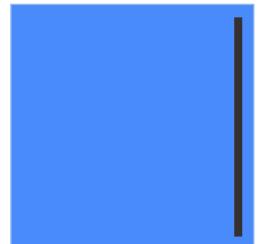


play.ionic.io

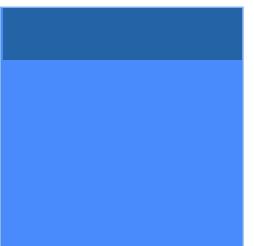
<ion-pane>



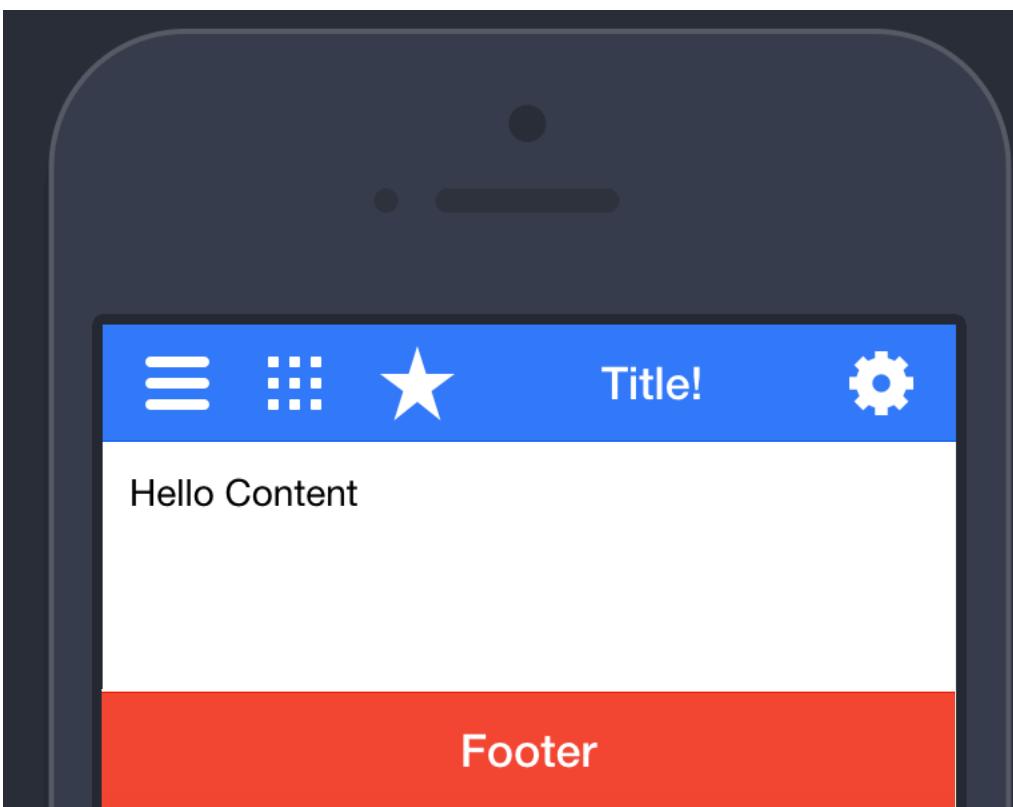
<ion-content>

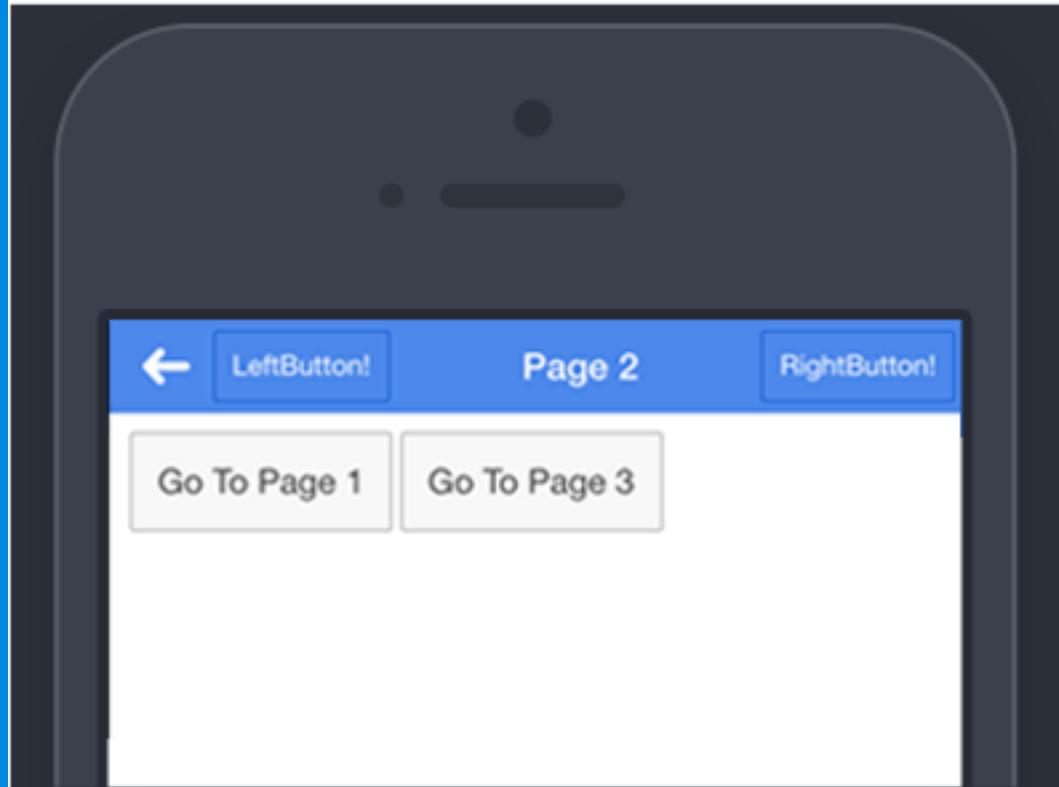


<ion-header-bar>



<ion-footer-bar>





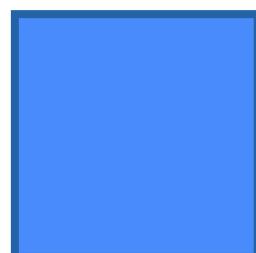
<ion-nav-bar>



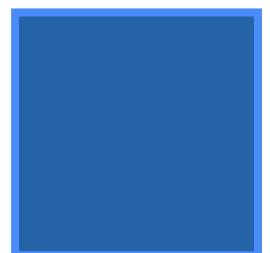
<ion-nav-buttons>



<ion-nav-view>



<ion-view>



# Gestures

\$ionicGesture



Tap



Double tap



hold



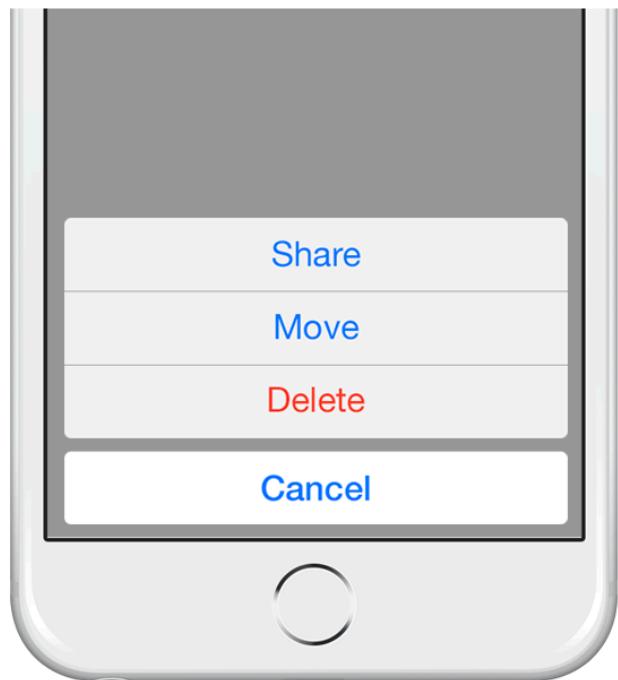
Drag



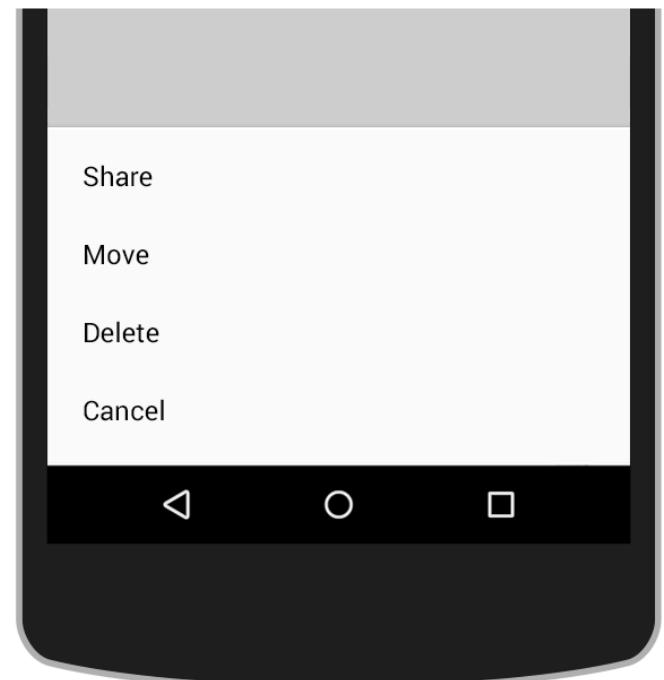
Swipe

# Platform continuity

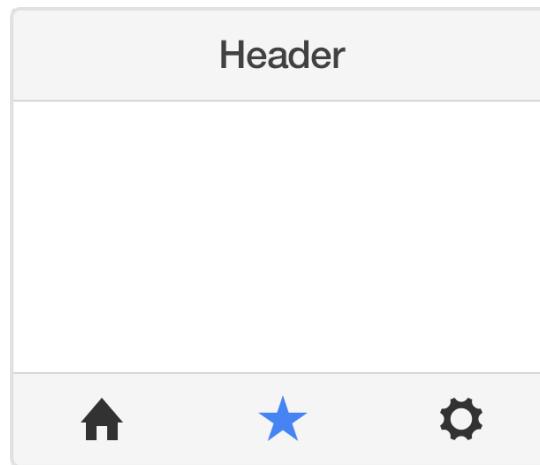
iOS



Android

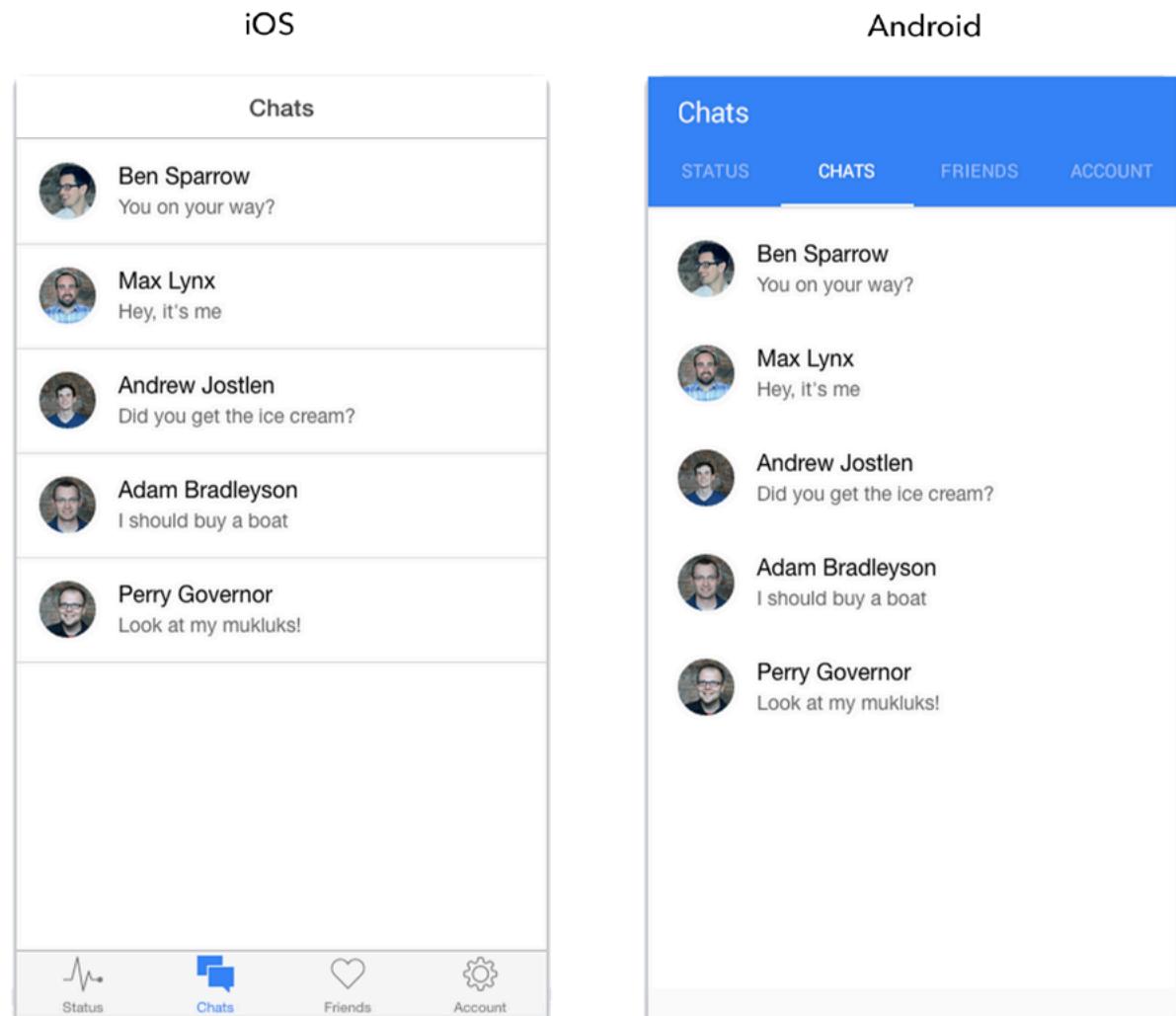


Header



\$ ionic start test tabs

# Platform continuity



\$ ionic serve --lab --livereload

# Testons sur nos téléphones



Ionic View App

The screenshot shows the Ionic View app interface on a mobile device. At the top, it displays "Carrier" and "3:14 PM". Below this is a header with "MY APPS" and a gear icon. The main content area lists five 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

# Système de navigation Angular

# UI Router

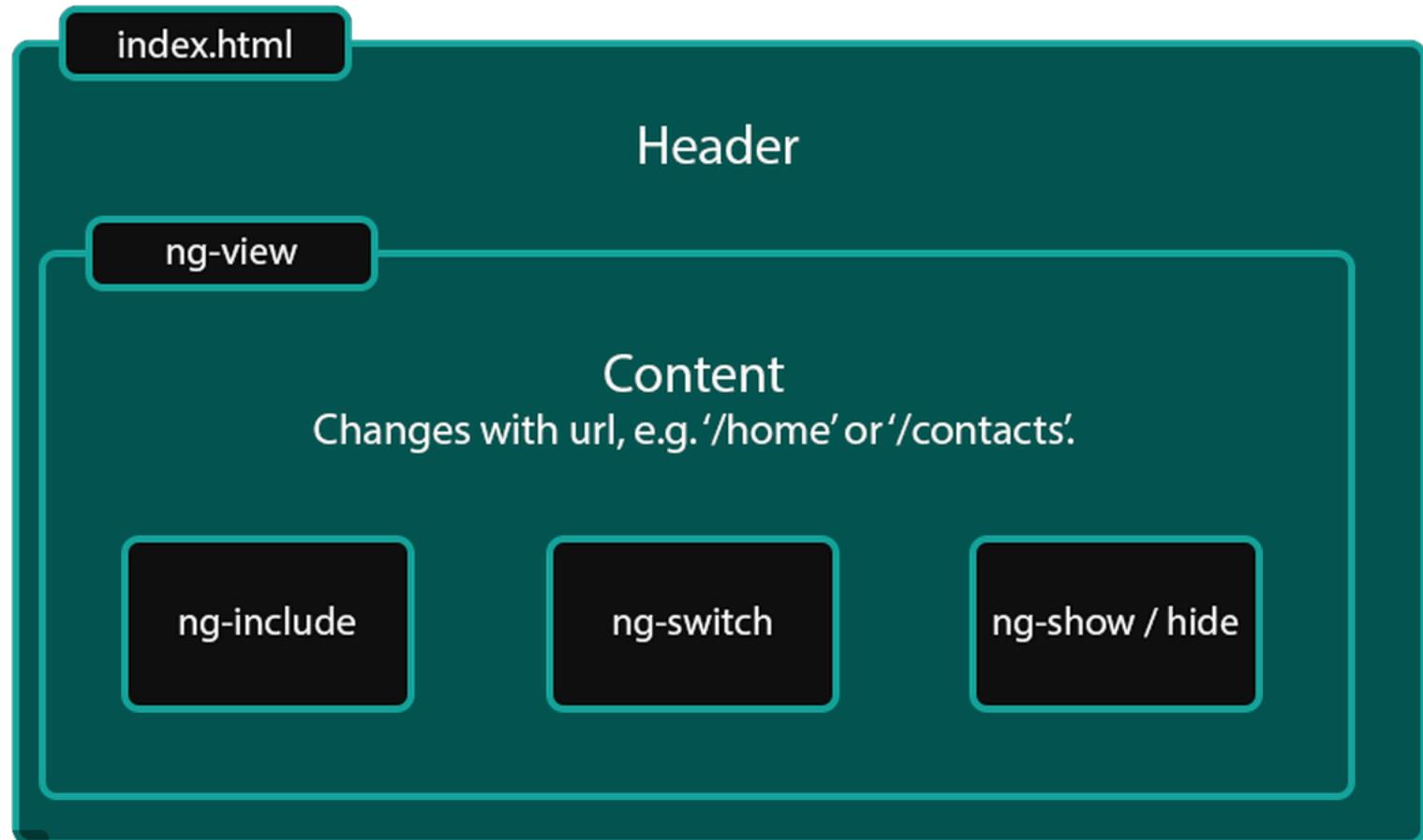


**ui-router** > **ngRoute**

Un module robuste et évolutif

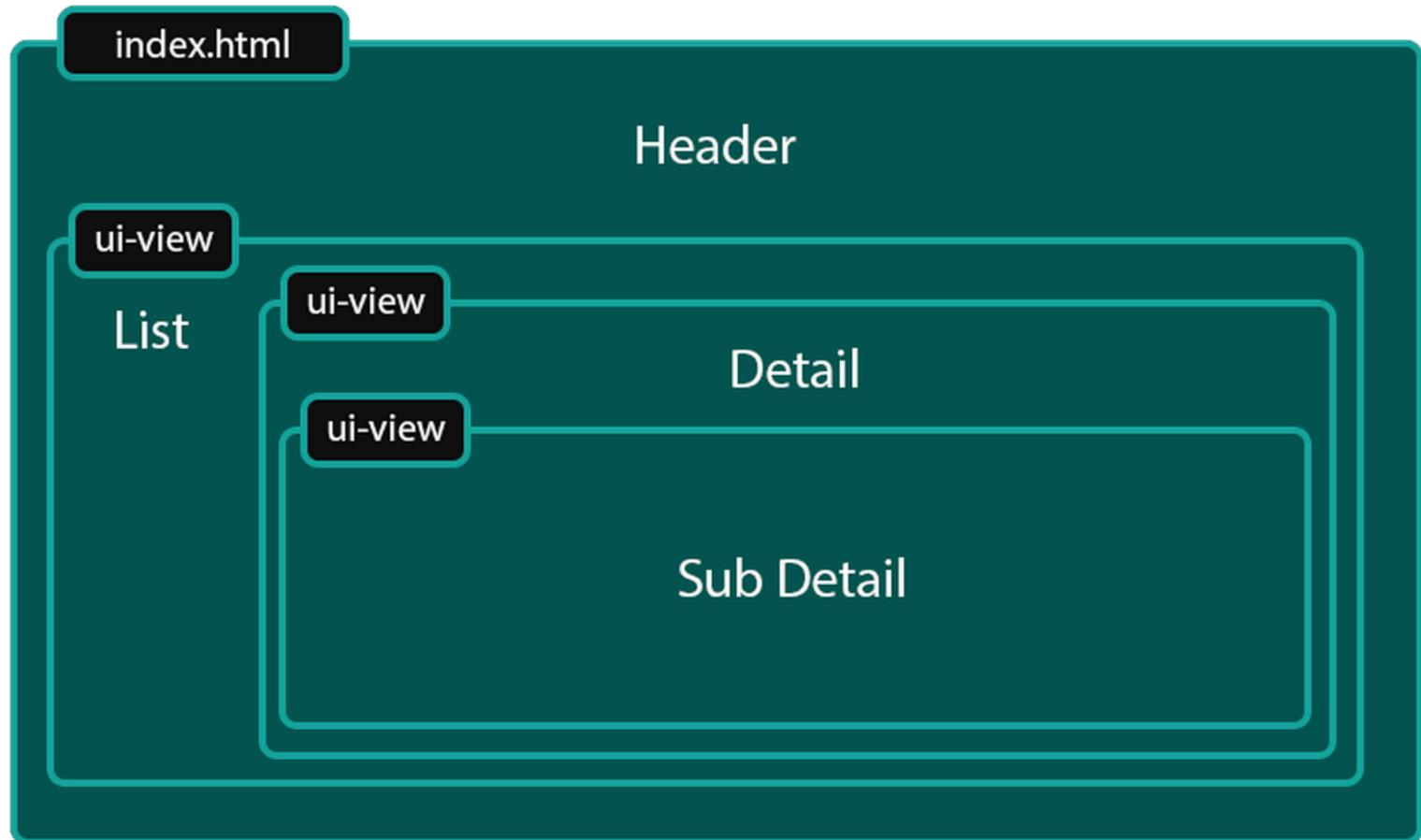
- Changement d'état, pas seulement d'url
  - Side Panel, Modal etc.
- Vues imbriqués: Des vues, dans des vues...





Une seule vue de premier niveau

## \$state/ \$stateProvider



Plusieurs vues imbriquées

## \$stateProvider

Une syntaxe similaire à ngRoute: un Nom + une URL

```
$routeProvider.when('/contacts/:id', {
  template: '<h1>Hello</h1>',
  templateUrl: 'contacts.html',
  controller: function($scope){ ... },
  resolve: { ... }
})
```

```
$stateProvider.state('contact.detail', {
  url: '/contacts/:id',
  template: '<h1>Hello</h1>',
  templateUrl: 'contacts.html',
  controller: function($scope){ ... },
  resolve: { ... }
})
```

Possibilité d'effectuer une redirection:

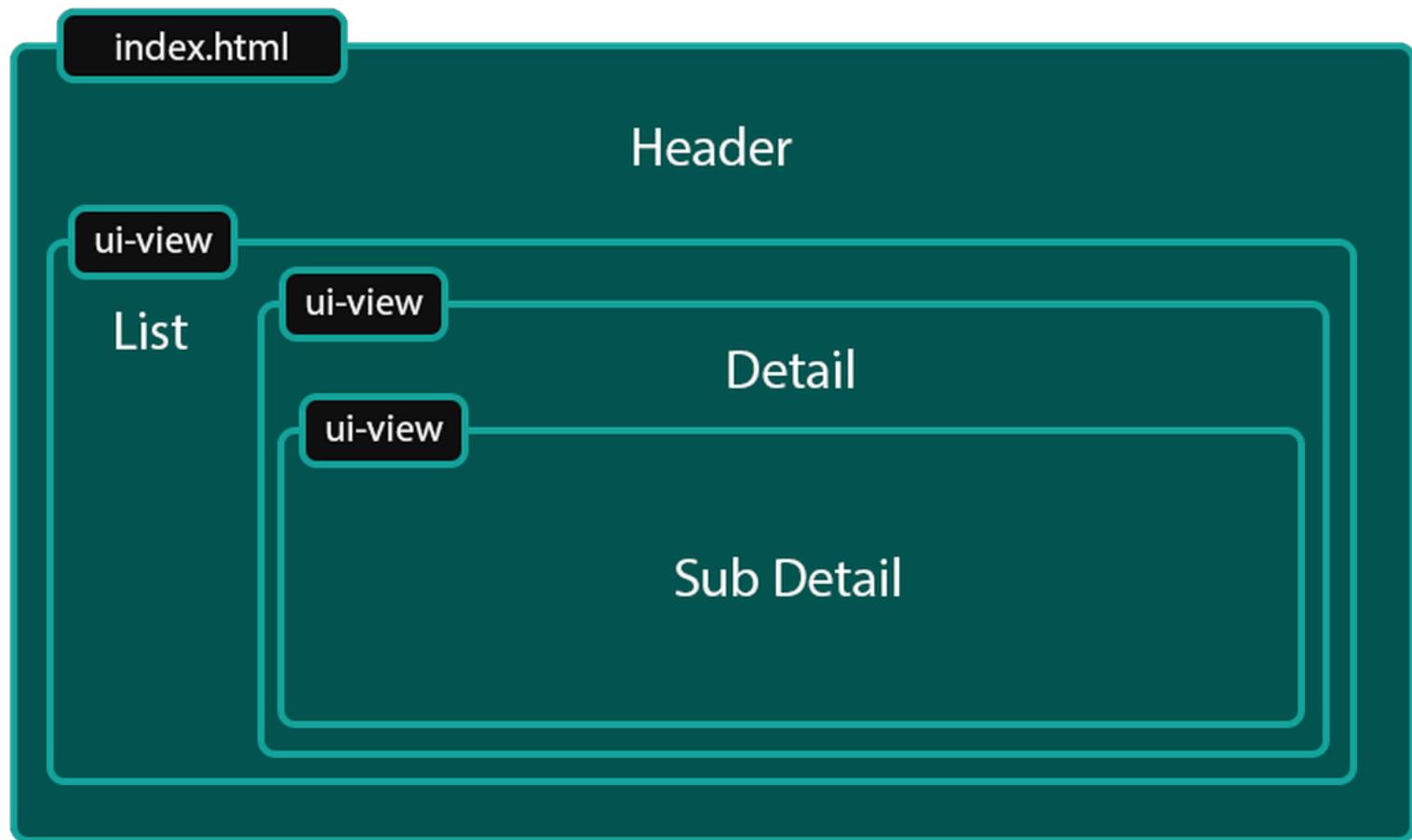
```
$urlRouterProvider
  .when('/user/:id', '/contacts/:id')
  .otherwise('/');
```

## Ajout du script

```
<script src="angular-ui-router.js"></script>
```

Ajout de la dépendance vers le module ui.router

```
angular.module("myApp", ["ui.router"])
```



# Template

## Utilisation de template/templateUrl dans d'un état

```
index.html
<body>
    <ui-view/>
</body>

app.js
.state('top', {
    template:<ui-view/>
})
.state('top.middle', {
    template:<ui-view/>
})
.state('top.middle.bottom', {
    template:<ui-view/>
})
```

Chaque template s'intègre au sein d'une ui-view

```
index.html
<body>
    <ui-view/>
</body>

app.js
.state('top', {
    template:<ui-view/>
})
.state('top.middle', {
    template:<ui-view/>
})
.state('top.middle.bottom', {
    template:<ui-view/>
})
```

# Configuration du parent

1. Via le séparateur point: (.)

## 1. \$stateProvider

```
.state('contacts', {})
.state('contacts.list', {});
```

2. Via la propriété parent, dans le config

## 2. \$stateProvider

```
.state('contacts', {})
.state('list', {
  parent: 'contacts'
});
```

3. Via un objet

```
var contacts = {
  name: 'contacts', //mandatory
  templateUrl: 'contacts.html'
}
var contactsList = {
  name: 'list', //mandatory
  parent: 'contacts', //mandatory
  templateUrl: 'contacts.list.html'
}

$stateProvider
  .state(contacts)
  .state(contactsList)
```

# Héritages:

Propriétés et méthodes du scope Angular

## Les états enfants hérités des parents

- Resolve

```
.state('parent', {
  resolve:{
    resA: function(){
      return {'value': 'A'};
    }
  },
  controller: function($scope, resA){
    $scope.resA = resA.value;
  }
})
.state('parent.child', {
  resolve:{
    resB: function(resA){
      return {'value': resA.value + 'B'};
    }
  },
  controller: function($scope, resA, resB){
    $scope.resA2 = resA.value;
    $scope.resB = resB.value;
  }
})
```

- Custom data

```
$stateProvider
  .state('parent', {
    data:{
      customData1: "Hello",
      customData2: "World!"
    }
  })
  .state('parent.child', {
    data:{
      // customData1 inherited from 'parent'
      // but we'll overwrite customData2
      customData2: "UI-Router!"
    }
  });

$rootScope.$on('$stateChangeStart', function(event, toState){
  var greeting = toState.data.customData1 + " " + toState.data.customData2;
  console.log(greeting);
})
```

# Ouverture et fermeture d'un état

- Callbacks

```
$stateProvider.state("contacts", {  
    template: '<h1>{{title}}</h1>',  
    resolve: { title: 'My Contacts' },  
    controller: function($scope, title){  
        $scope.title = 'My Contacts';  
    },  
    onEnter: function(title){  
        if(title){ ... do something ... }  
    },  
    onExit: function(title){  
        if(title){ ... do something ... }  
    }  
})
```

- Evènements

**\$stateChangeStart** (event, toState, toParams, fromState, fromParams)

*can e.preventDefault()*

**\$stateNotFound** (event, unfoundState, fromState, fromParams)

*good for lazy state definitions*

**\$stateChangeSuccess** (event, toState, toParams, fromState, fromParams)

**\$stateChangeError** (event, toState, toParams, fromState, fromParams, error)

## Changer d'état

- \$state.go()

```
myApp.controller('contactCtrl', ['$scope', '$state',
  function($scope, $state){
    $scope.goToDetails = function(){
      $state.go('contact.details', {id: selectedId});
    }
  }
])
```

- ui-sref directive

```
<a ui-sref="home">Home</a>
```

- ui-sref params

```
<li ng-repeat="contact in contacts">
  <a ui-sref="contacts.detail({ id: contact.id })"></a>
</li>
```

## Vérifier l'activation d'un état

- \$state.is();

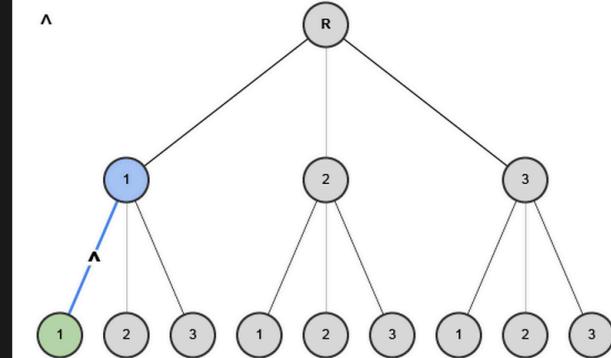
```
$state.is("home");
```

# Navigation relative

(^) relatif au parent

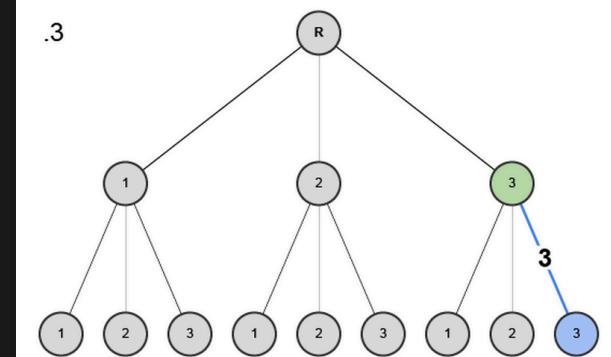
Start      Middle      End

*Go to parent - \$state.go('^')*



(.) relatif à l'enfant

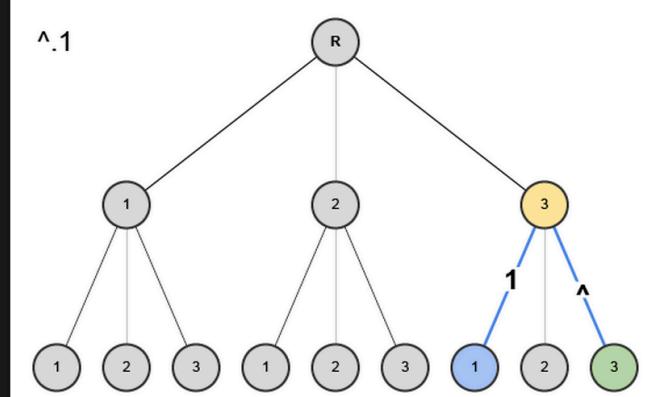
*Go to child - \$state.go('.3')*



# Navigation relative

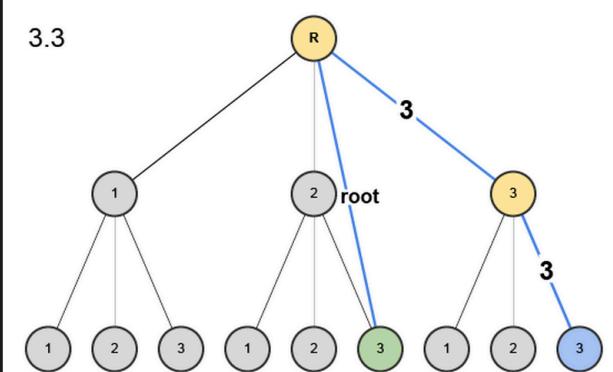
$\wedge.1$

*Go to sibling - \$state.go('^.1')*



.3

*Absolute Path - \$state.go('3.3')*



# Gestion d'url

## URL imbriqués

```
$stateProvider
  .state('contacts', {
    url: '/contacts',
  })
  .state('contacts.list', {
    url: '/list',
  });
});
```

/contacts/list

## URL absolu

```
$stateProvider
  .state('contacts', {
    url: '/contacts',
  })
  .state('contacts.list', {
    url: '^/list',
  });
});
```

/list

# Gestion d'url

## Paramètres

### Basic

```
url: '/contacts/:contactId'  
url: '/contacts/{contactId}'
```

### Regex

```
url: '/contacts/{contactId:[0-9a-fA-F]{1,8}}' //Hexadecimals
```

### Query

```
url: '/contacts?contactId&contactRegion' //Separate with '&'
```

## Exemple

```
//State URL:  
url: '/users/:id/details/{type}/{repeat:[0-9]+}?from&to'  
  
//Navigate to:  
'/users/123/details//0'  
  
//$$stateParams will be  
{ id:'123', type:'', repeat:'0' }  
  
//Navigated to:  
'/users/123/details/default/0?from=there&to=here'  
  
//$$stateParams will be  
{ id:'123', type:'default', repeat:'0',  
  from:'there', to:'here' }
```

# Gestion d'url

IMPORTANT: pas d'héritage pour les paramètres d'url

```
$stateProvider.state('contacts.detail', {  
    url: '/contacts/:contactId',  
    resolve: { depA: function(){  
        return $state.current.params.contactId + "!" ;  
    },  
    controller: function($stateParams){  
        $stateParams.contactId // Exists!  
    }  
}).state('contacts.detail.subitem', {  
    url: '/item/:itemId',  
    controller: function($stateParams){  
        $stateParams.contactId // Doesn't exist  
        $stateParams.itemId // Exists!  
    }  
})
```

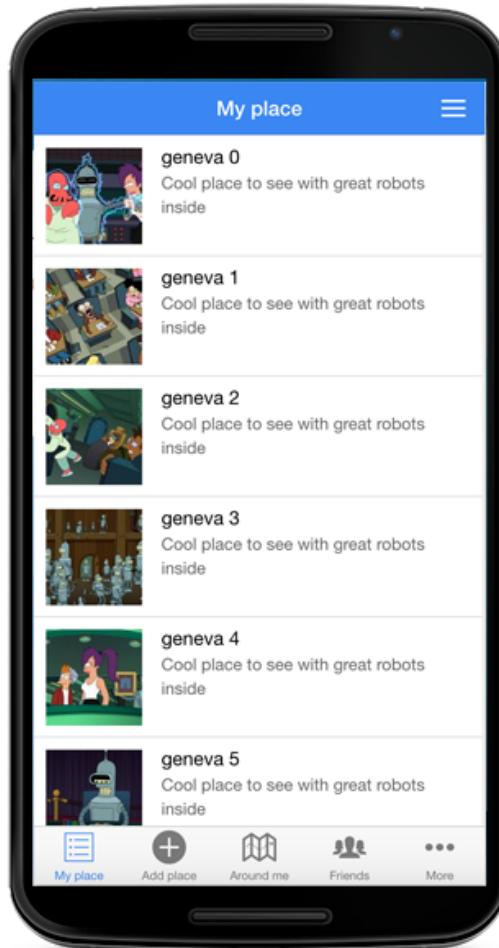
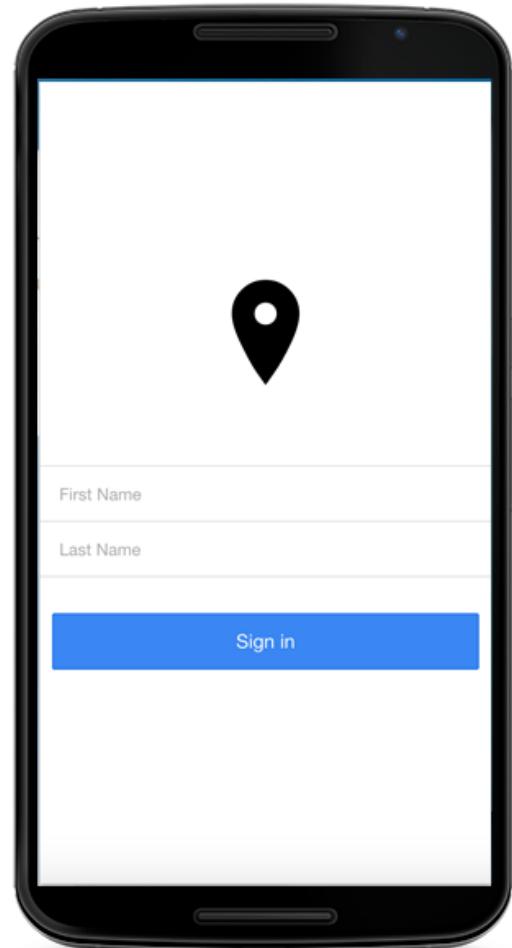
```
{  
  name: '',  
  url: '^',  
  'abstract': true  
}
```

## Utilisation d'état abstrait

- Pour préfixer l'URL des états enfants
- Pour initialiser un template et son contrôleur si nécessaire
- Pour centraliser les Resolve, CustomData et évènements d'un état

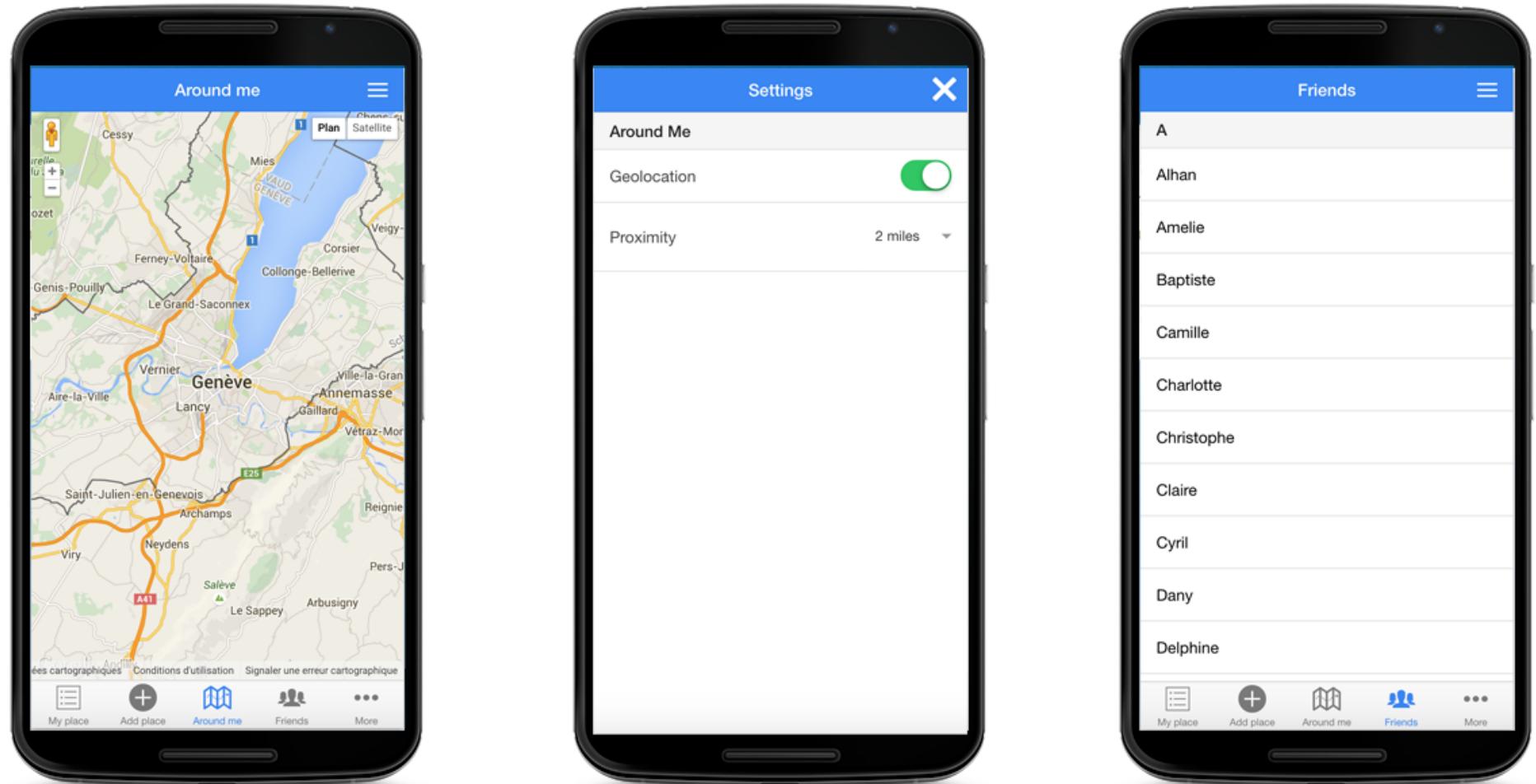
# Démarrons un nouveau projet

# Objectif:



Prototype « Ionic Places »

# Objectif:



Prototype « Ionic Places »

# Installation

Récupérer le projet

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

Un compte github ? « Fork me. »



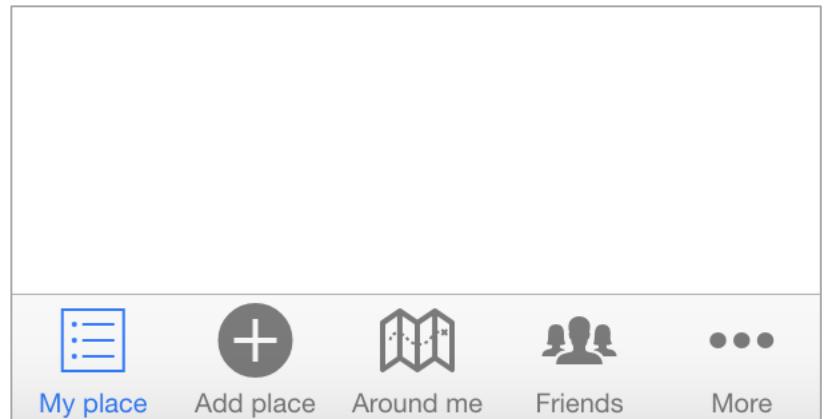
---

```
$ ionic serve --livereload
```

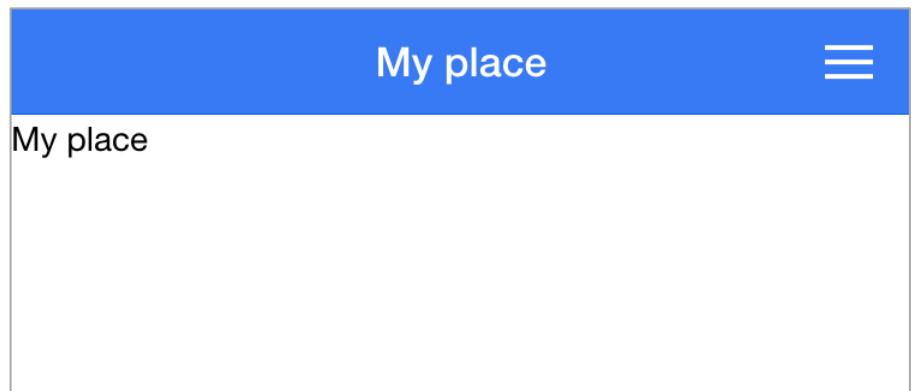
First Name
Last Name
<b>Sign in</b>

# Pratique

1) Mettre en place le système de navigation :  
<ion-tabs>

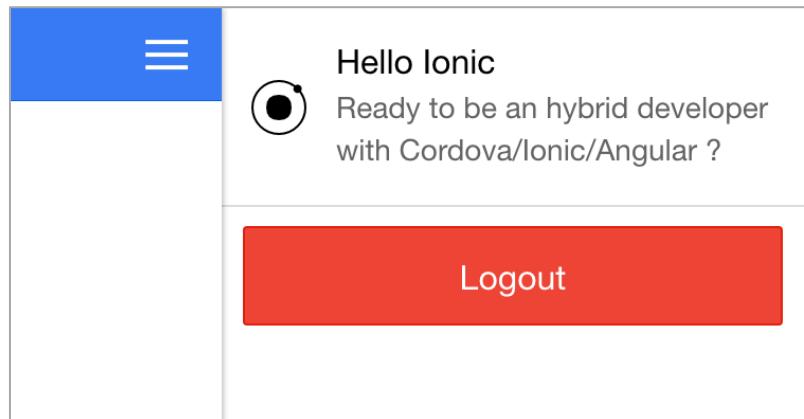


2) Mettre en place le header



# Pratique

3) Mettre en place le système de navigation :  
<ion-side-menus>



4) Mettre en place une liste dynamique  
collection-repeat

