

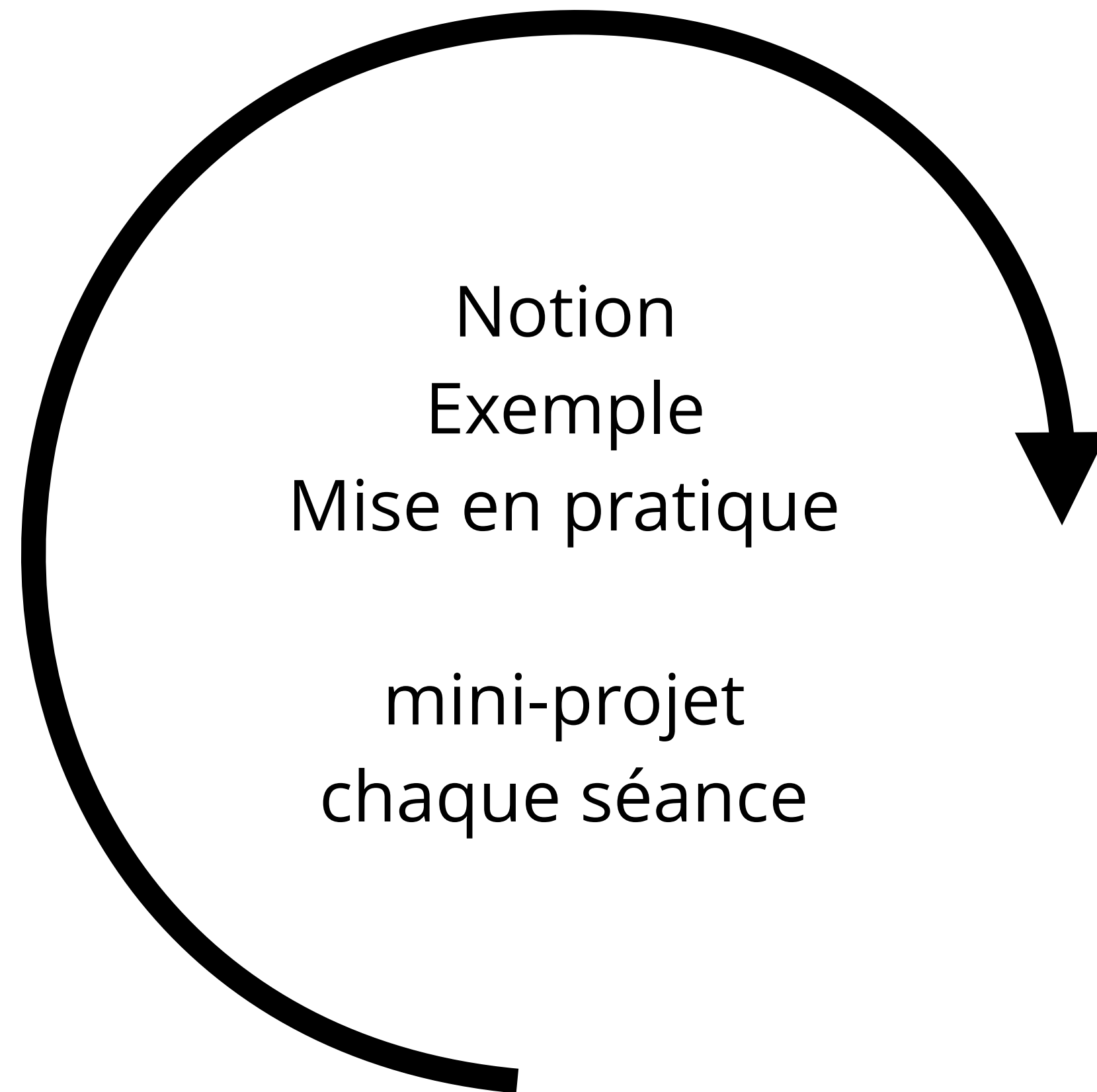
Programmation mobile et web services

Cours 1

Développement mobile hybride avec Ionic



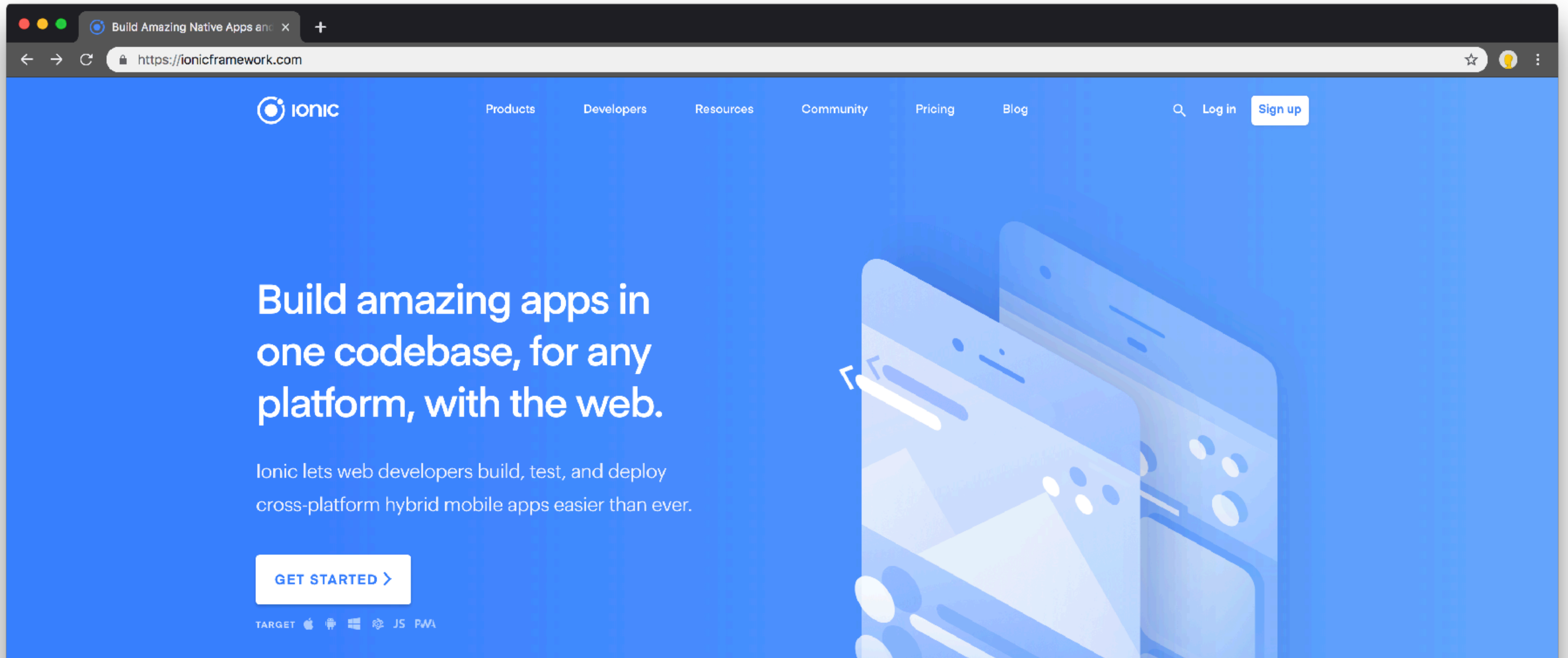
Le cours



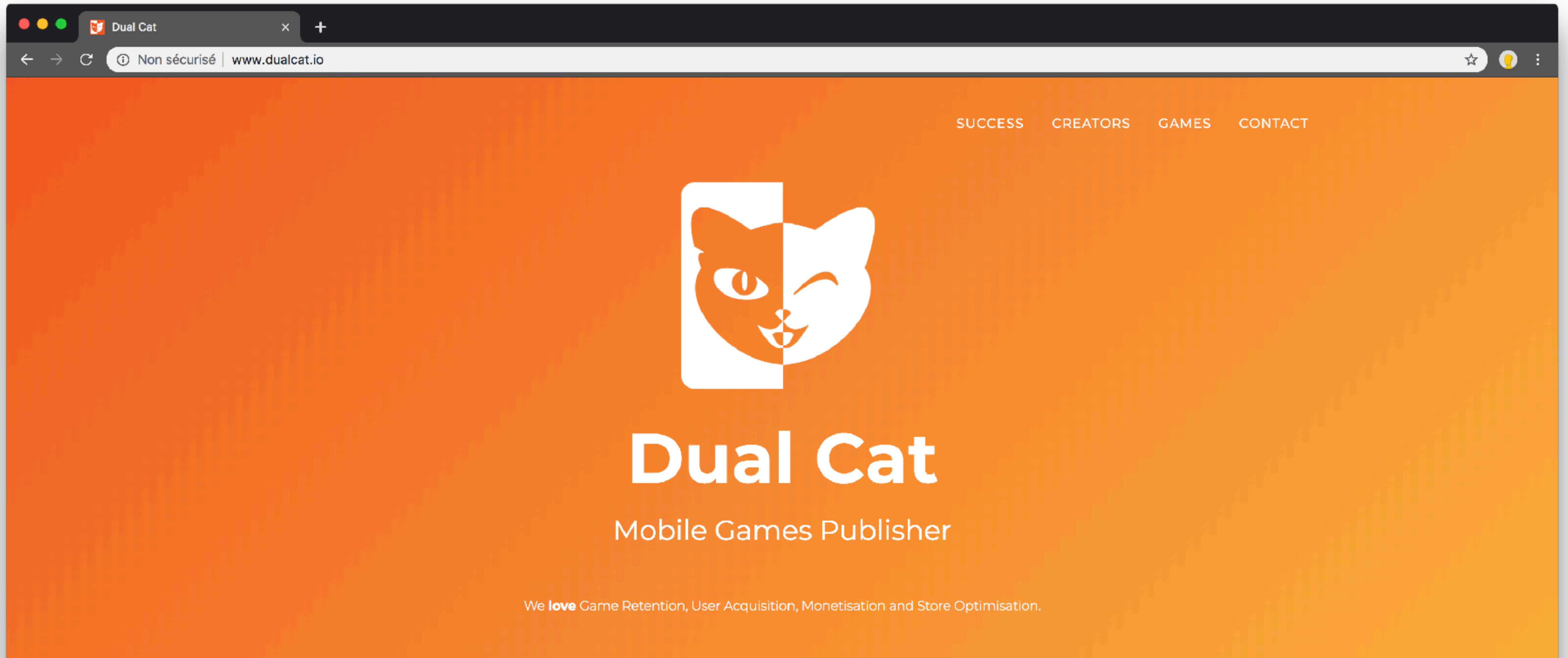
1 projet de 1 semaine
1 examen écrit en dernière séance

Les évaluations

Ionic v3



Préparation en immersion



Applications Natives vs. Web vs. Hybrides

Applications **Natives** vs. Web vs. Hybrides

Applications **Natives** vs. Web vs. Hybrides

Une application native est une application développée pour une plateforme spécifique

Applications **Natives** vs. Web vs. Hybrides



android



Windows



Applications **Natives** vs. Web vs. Hybrides



Les performances et l'expérience utilisateur d'une application sont optimisées, puisqu'elle est implémentée pour une plateforme spécifique.

Applications **Natives** vs. Web vs. Hybrides



android



Windows



Par nature, le développement n'est pas cross-plateformes. Il faut tout ré-implémenter pour changer de plateformes.



Objective-C / Swift
+
Xcode

Applications **Natives** vs. Web vs. Hybrides



Les performances et l'expérience utilisateur d'une application sont optimisées, puisqu'elle est implémentée pour une plateforme spécifique.



Objective-C / Swift
+
Xcode

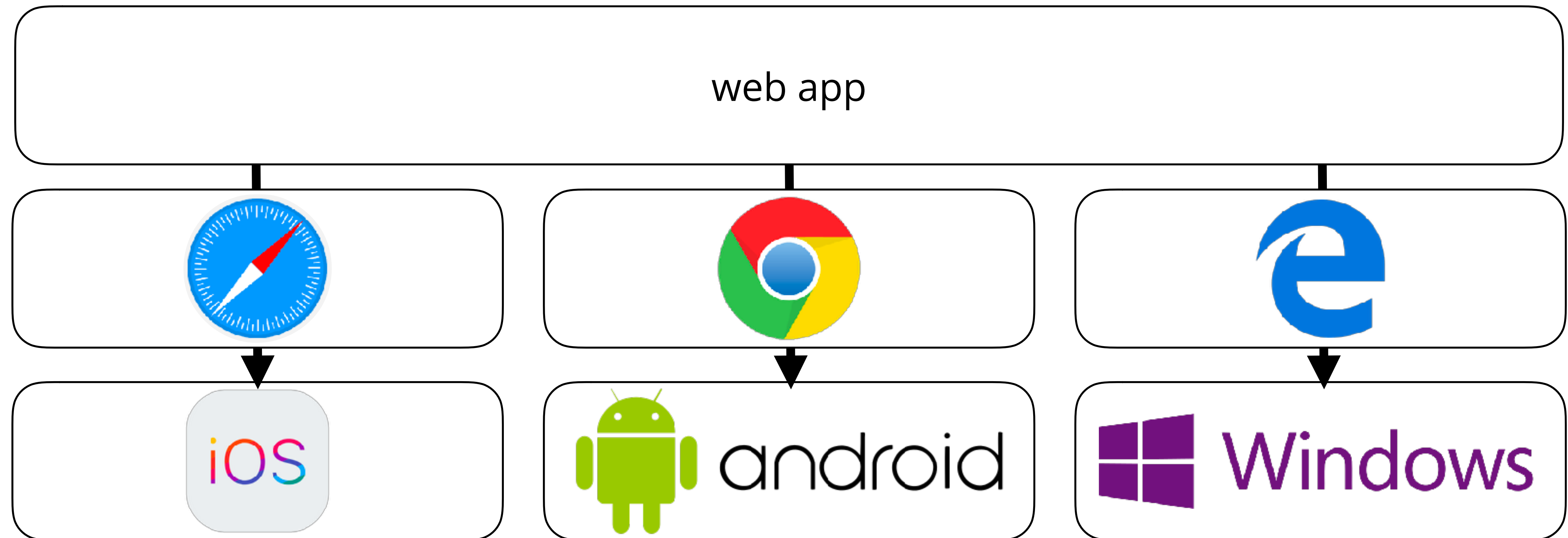
Java
+
Android Studio

C# / C / C++
+
Visual Studio

Applications Natives vs. Web vs. Hybrides

Applications Natives vs. **Web** vs. Hybrides

Une application web est un site web optimisé pour un affichage sur mobile.



Applications Natives vs. Web vs. Hybrides



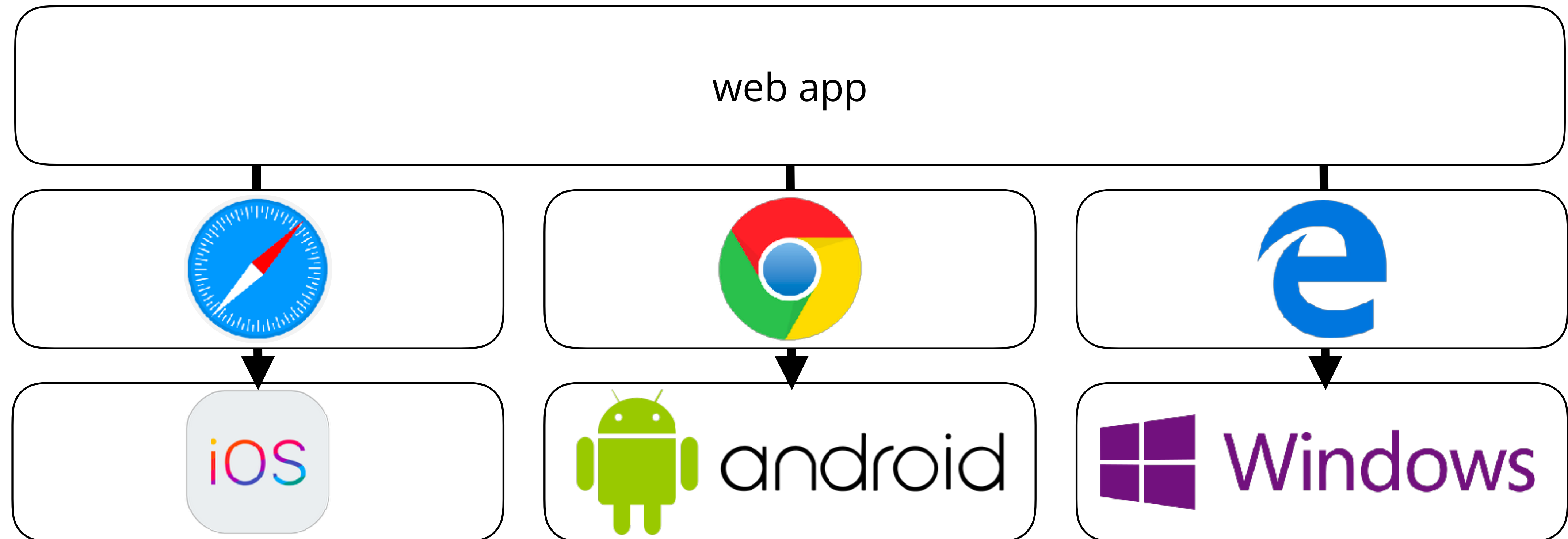
Applications Natives vs. **Web** vs. Hybrides



Se développe
comme un
site web.



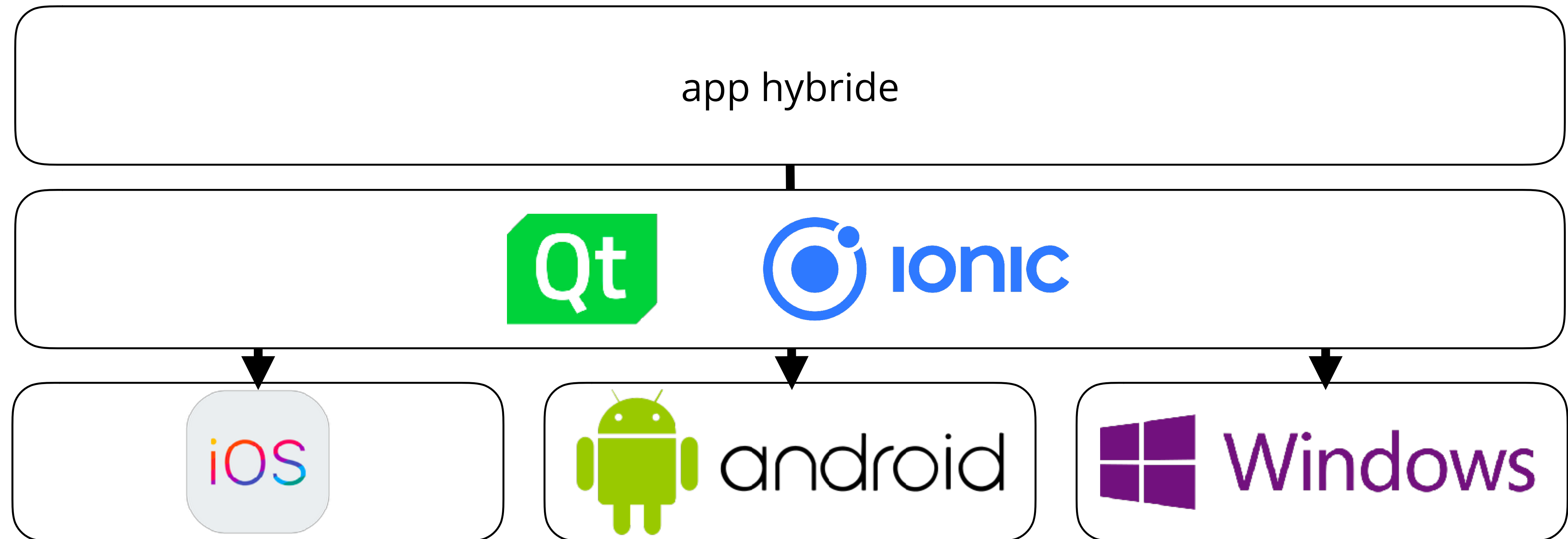
Difficile à tester partout (os, browser, modèle de téléphone, ...).
Ce n'est pas une app et le rendu visuel sera différent des apps natives.
Capteurs difficiles à utiliser.
Non disponible sur les stores.



Applications Natives vs. Web vs. Hybrides

Applications Natives vs. Web vs. Hybrides

Une application hybride est une application implémentée une seule fois et qui est déployée sur toutes les plateformes comme une application native.



Applications Natives vs. Web vs. Hybrides



Applications Natives vs. Web vs. Hybrides



Une seule base de code pour toutes les plateformes
(économise du temps et de l'argent)
Rendu visuel proche des apps natives.



Les performances sont moins optimisées que pour les
applications natives.
Latence entre les releases "natives" et "hybride".

Différence entre library / framework

<library/>

Collection de routines, qui peuvent être déjà compilées et prêtes à être utilisées par des programmes.

<framework/>

Ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel (architecture). Un framework peut à ce titre être constitué de plusieurs bibliothèques. Il impose de suivre certains patrons de conception.

Exemple de framework



C++ (ou Python)

Qt Creator



Typescript

Visual Studio Code, Sublime Text 3, ...

Qt



C++ (qui est un langage natif sur iOS et Android), donc performance.
Déploiement sur Windows, OSX, Linux.



Nécessite du temps d'apprentissage

Ionic



Rapide à développer et donc à sortir sur les stores.
Accès à toutes les library Javascript (langage le plus populaire).
Puissance du HTML5, CSS.
Respecte les thèmes graphiques.
Rapide à maîtriser.



Les performances sont moins optimisées.
Lourdeur des applications (inclus un navigateur web).

Architecture d'un projet ionic



Framework Javascript open-source de développement front-end d'application web. Il introduit une sur-couche du langage HTML en autorisant l'utilisation de balises qui sont ensuite liées à des entrées et/ou sorties d'un modèle décrit en Javascript.



Framework de développement d'application mobile. Apache Cordova permet aux développeurs de créer des applications sur mobile à l'aide de CSS3, HTML5 et Javascript.

(≈ navigateur internet encapsulé)

Liens utiles ionic

De l'installation à la première application:

<https://ionicframework.com/docs/intro/installation/>

Documentation:

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

Autre tutoriel:

<https://soat.developpez.com/tutoriels/mobiles/ionic-3-creer-application-mobile/>

Mon premier projet ionic

Installation à l'aide de npm :



```
>  
> npm install -g ionic cordova  
>
```

-g pour une installation globale

Mon premier projet ionic

Création du projet Ionic :

```
>  
> ionic start unNomBien --type=angular  
>
```


Mon premier projet ionic

```
Bacasable — ionic TERM_PROGRAM=Apple_Terminal SHELL=/bin/bash TERM=xterm-256color — 130x35
[eduroam-108108:Bacasable alixgoguey$ ionic start unNomBien]
[INFO] You are about to create an Ionic 3 app. Would you like to try the release candidate for Ionic 4?

Ionic 4 uses the power of the modern Web and embraces the Angular CLI and Angular Router to bring you the best
version of Ionic ever. See our blog announcement[1] and documentation[2] for more information. We'd love to hear
your feedback on our latest version!

[1]: https://blog.ionicframework.com/ionic-framework-4-0-rc-shipped-paving-way-for-final
[2]: https://beta.ionicframework.com/docs/

[? Try Ionic 4? No]

Let's pick the perfect starter template! 🍌

Starter templates are ready-to-go Ionic apps that come packed with everything you need to build your app. To bypass this
prompt next time, supply template, the second argument to ionic start.

? Starter template: (Use arrow keys)
> tabs      | A starting project with a simple tabbed interface
blank      | A blank starter project
sidemenu   | A starting project with a side menu with navigation in the content area
super      | A starting project complete with pre-built pages, providers and best practices for Ionic development.
tutorial   | A tutorial based project that goes along with the Ionic documentation
aws        | AWS Mobile Hub Starter
```

Mon premier projet ionic

Puis :

```
>  
> cd unNomBien/  
> ionic serve  
>
```

Mon premier projet ionic

```
unNomBien — node • ionic TERM_PROGRAM=Apple_Terminal SHELL=/bin/bash — 130x35
[app-scripts] [08:45:56] ionic-app-scripts 3.2.1
[app-scripts] [08:45:56] watch started ...
[app-scripts] [08:45:56] build dev started ...
[app-scripts] [08:45:56] clean started ...
[app-scripts] [08:45:56] clean finished in 1 ms
[app-scripts] [08:45:56] copy started ...
[app-scripts] [08:45:56] deeplinks started ...
[app-scripts] [08:45:56] deeplinks finished in 15 ms
[app-scripts] [08:45:56] transpile started ...
[app-scripts] [08:45:59] transpile finished in 2.80 s
[app-scripts] [08:45:59] preprocess started ...
[app-scripts] [08:45:59] preprocess finished in less than 1 ms
[app-scripts] [08:45:59] webpack started ...
[app-scripts] [08:45:59] copy finished in 2.94 s
[app-scripts] [08:46:02] webpack finished in 3.56 s
[app-scripts] [08:46:02] sass started ...
[app-scripts] [08:46:03] sass finished in 1.14 s
[app-scripts] [08:46:03] postprocess started ...
[app-scripts] [08:46:03] postprocess finished in 8 ms
[app-scripts] [08:46:03] lint started ...
[app-scripts] [08:46:03] build dev finished in 7.59 s
[app-scripts] [08:46:03] watch ready in 7.65 s

[INFO] Development server running!

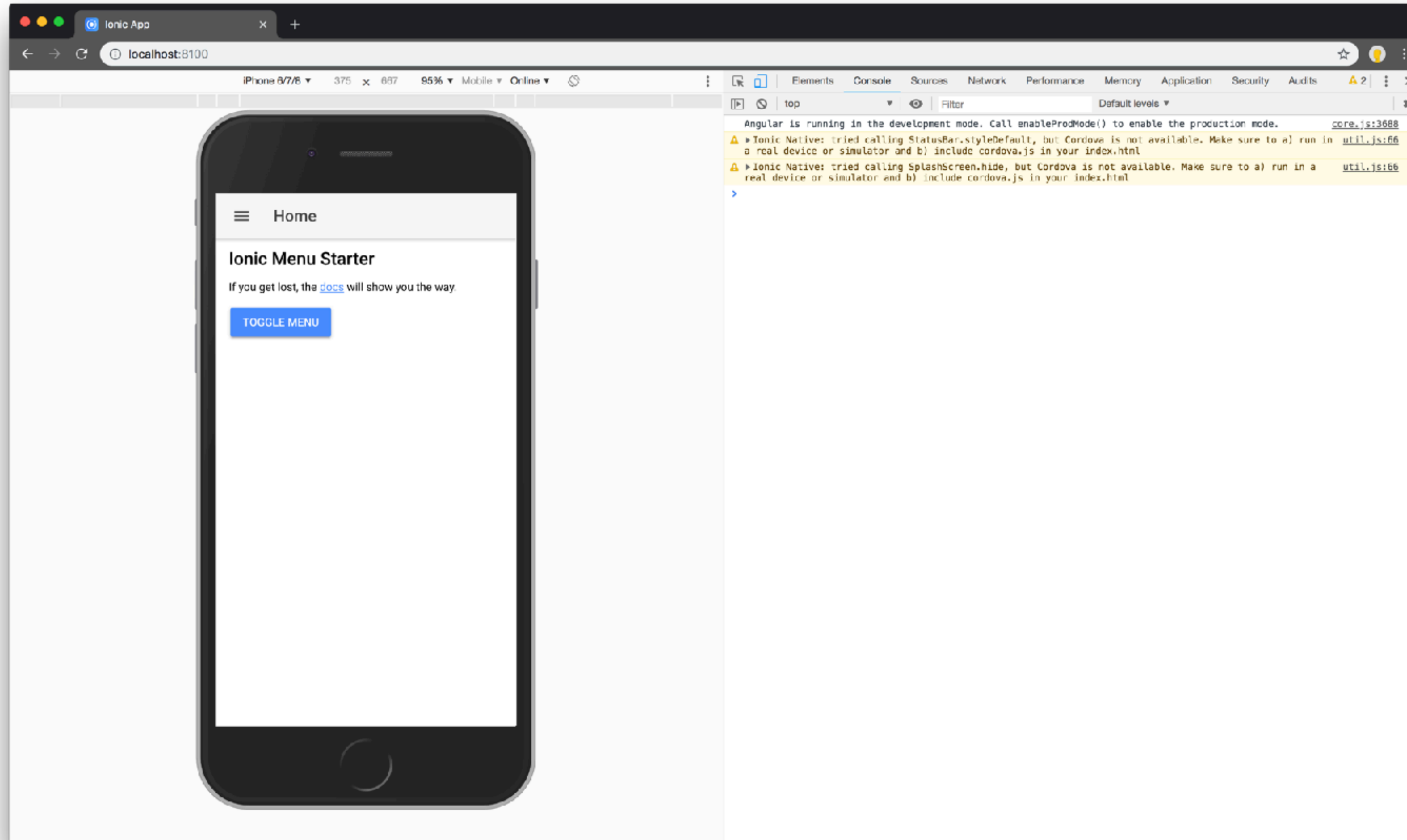
Local: http://localhost:8100
External: http://130.190.108.108:8100
DevApp: unNomBien@8100 on eduroam-108108.grenet.fr

Use Ctrl+C to quit this process

[INFO] Browser window opened to http://localhost:8100!

[app-scripts] [08:46:07] lint finished in 3.47 s
```


Mon premier projet ionic



Mon premier projet ionic

Ajout de plateformes (pour le déploiement) :

```
>  
> ionic cordova platform add android  
> ionic cordova platform add ios  
>
```


Architecture d'un projet ionic

Ionic se base sur les *components*, ce qui veut dire que chaque page est vue comme un composant individuel avec son propre habillage et comportement. Cela facilite la maintenabilité du projet vu que tout le code se trouve au niveau du composant.

Architecture d'un projet ionic

Structure d'un projet :

/

/config.xml

/package.json

/src/

 /src/app/

 /src/app/XXX

 /src/assets/

 /src/provider/

 /src/theme/

/www/

/resources/

Architecture d'un projet

Structure d'un projet :

/

/config.xml

/package.json

/src/

/src/app/

/src/app/XXX

/src/assets/

/src/provider/

/src/theme/

/www/

/resources/

Fichiers de configuration dont les plus importants sont :

- config.xml (après ajout des plateformes) : utilisé par Cordova. Contient les configurations pour les builds iOS et Android.

- package.json : fichier de config de npm, contenant des informations sur le projet (nom, version, auteur, licence...), dépendances en termes de packages npm, et les scripts perso.

Architecture d'un projet ionic

Exemple de script perso :

```
"scripts": {  
  "start": "ionic-app-scripts serve",  
  "clean": "ionic-app-scripts clean",  
  "build": "ionic-app-scripts build",  
  "lint": "ionic-app-scripts lint",  
  "toto": "ionic serve --port=8101"  
},
```

```
>  
> npm run toto  
>
```

Architecture d'un projet ionic

Structure d'un projet :

/

/config.xml

/package.json

/src/

/src/app/

/src/app/XXX

/src/assets/

/src/provider/

/src/theme/

/www/

/resources/

Code source du projet

Architecture d'un projet ionic

Structure d'un projet :

/

/config.xml

/package.json

/src/

/src/app/

/src/app/XXX

/src/assets/

/src/provider/

/src/theme/

/www/

/resources/

Contient le composant racine de l'application, où l'on définit la page de démarrage, déclare nos modules, etc

Architecture d'un projet ionic

Structure d'un projet :

/

/config.xml

/package.json

/src/

/src/app/

/src/app/XXX

/src/assets/

/src/provider/

/src/theme/

/www/

/resources/

Contient toutes les pages de l'application.

Contient toutes les providers de l'application.

```
>  
> ionic generate  
> ionic generate <type> <name> [options]  
>
```

Architecture d'un projet ionic

Structure d'un projet :

/

/config.xml

/package.json

/src/

/src/app/

/src/app/XXX

/src/assets/

/src/provider/

/src/theme/

/www/

/resources/

Contient toutes les ressources statiques (images, JSON, etc).

Architecture d'un projet ionic

Structure d'un projet :

/

/config.xml

/package.json

/src/

/src/app/

/src/app/XXX

/src/assets/

/src/provider/

/src/theme/

/www/

/resources/

Contient variables SCSS pour les couleurs de l'application.

Architecture d'un projet ionic

Structure d'un projet :

/

/config.xml

/package.json

/src/

/src/app/

/src/app/XXX

/src/assets/

/src/provider/

/src/theme/

/www/

/resources/

Contient le code compilé de l'application.

Architecture d'un projet

Structure d'un projet :

/

/config.xml

/package.json

/src/

/src/app/

/src/app/XXX

/src/assets/

/src/provider/

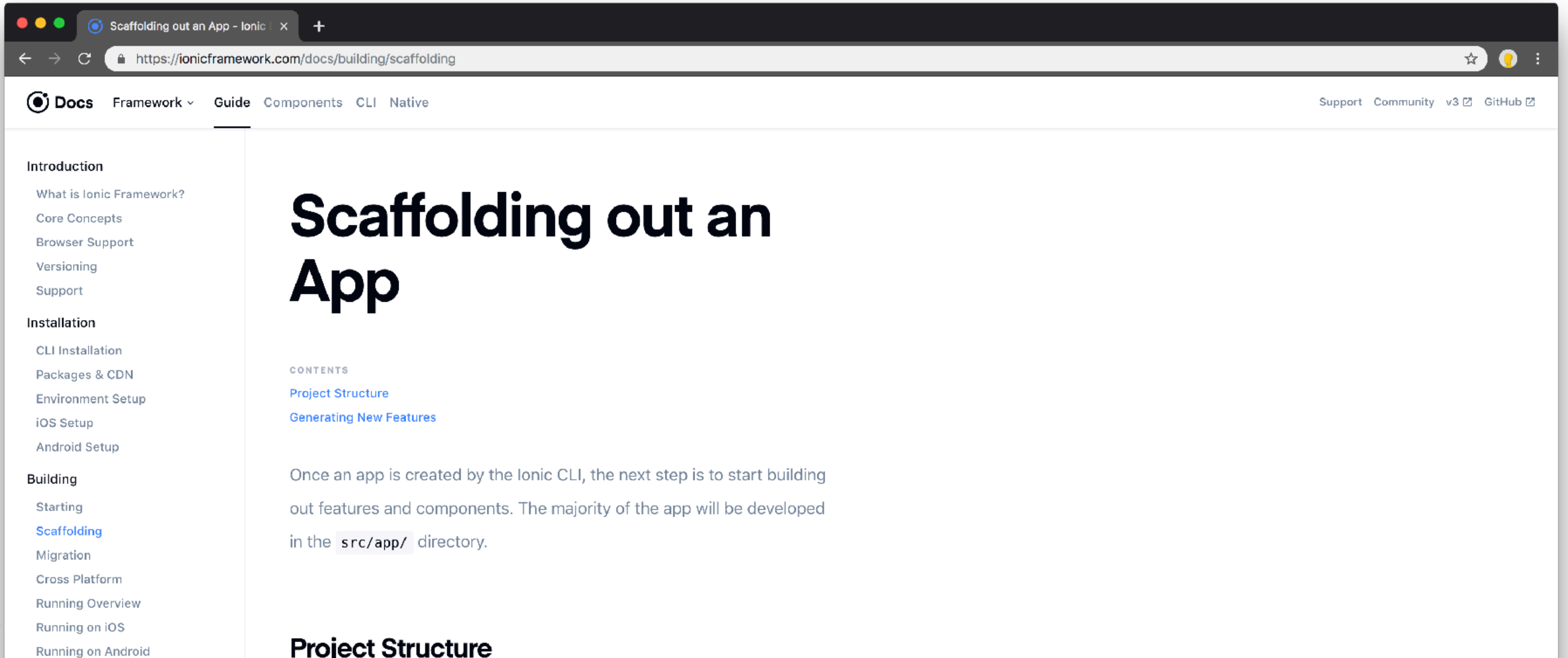
/src/theme/

/www/

/resources/

Contient les ressources pour les plateformes Android et iOS, notamment les images utilisées pour le splash screen (écran de démarrage) et les icônes.

Architecture d'un projet ionic



The screenshot shows a web browser window with the URL <https://ionicframework.com/docs/building/scaffolding>. The page title is "Scaffolding out an App - Ionic". The navigation bar includes "Docs", "Framework", "Guide", "Components", "CLI", and "Native". The main content area has a large heading "Scaffolding out an App" and a "CONTENTS" section with links to "Project Structure" and "Generating New Features". The left sidebar lists various topics under "Introduction", "Installation", and "Building".

Introduction

- What is Ionic Framework?
- Core Concepts
- Browser Support
- Versioning
- Support

Installation

- CLI Installation
- Packages & CDN
- Environment Setup
- iOS Setup
- Android Setup

Building

- Starting
- Scaffolding**
- Migration
- Cross Platform
- Running Overview
- Running on iOS
- Running on Android

Scaffolding out an App

CONTENTS

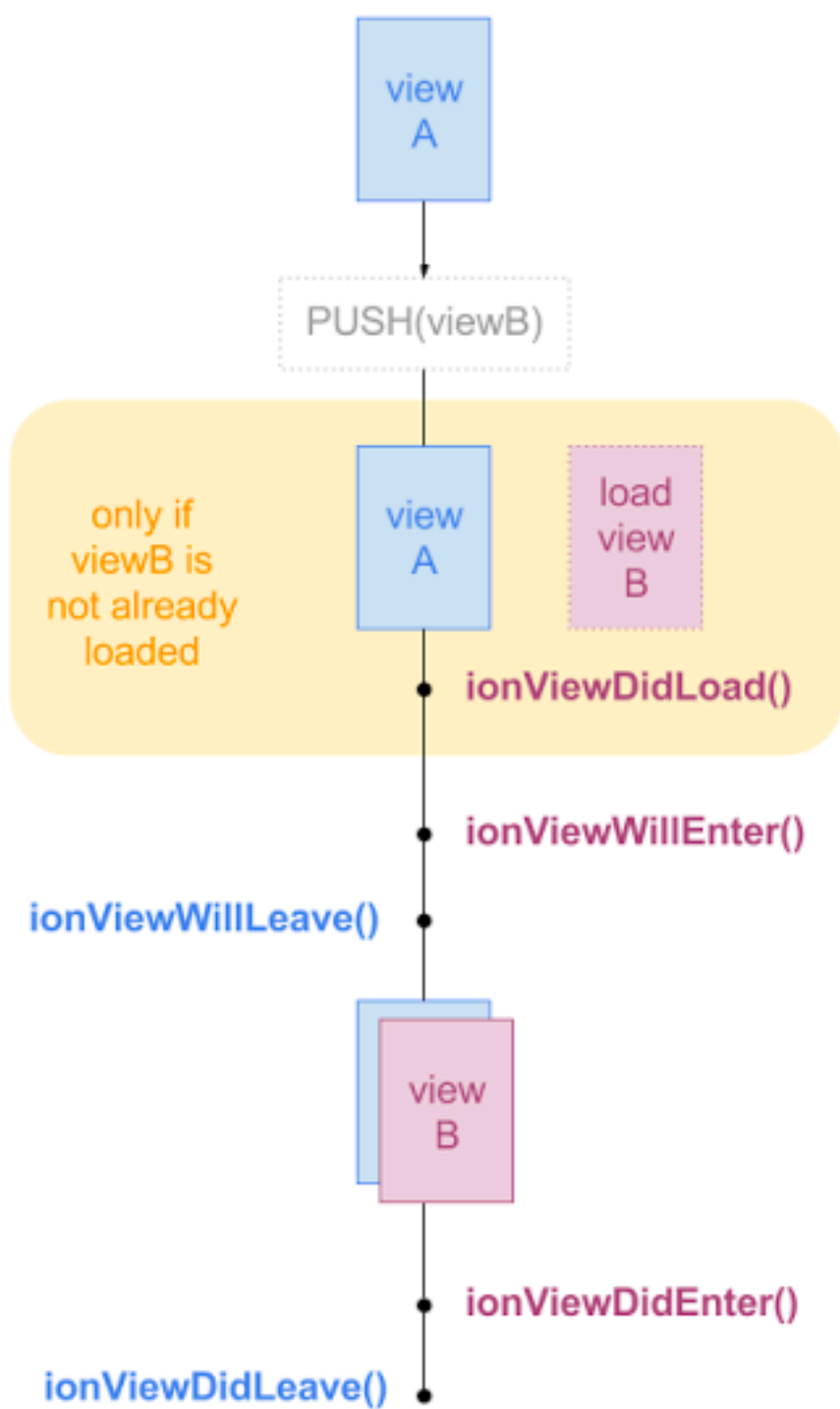
- [Project Structure](#)
- [Generating New Features](#)

Once an app is created by the Ionic CLI, the next step is to start building out features and components. The majority of the app will be developed in the `src/app/` directory.

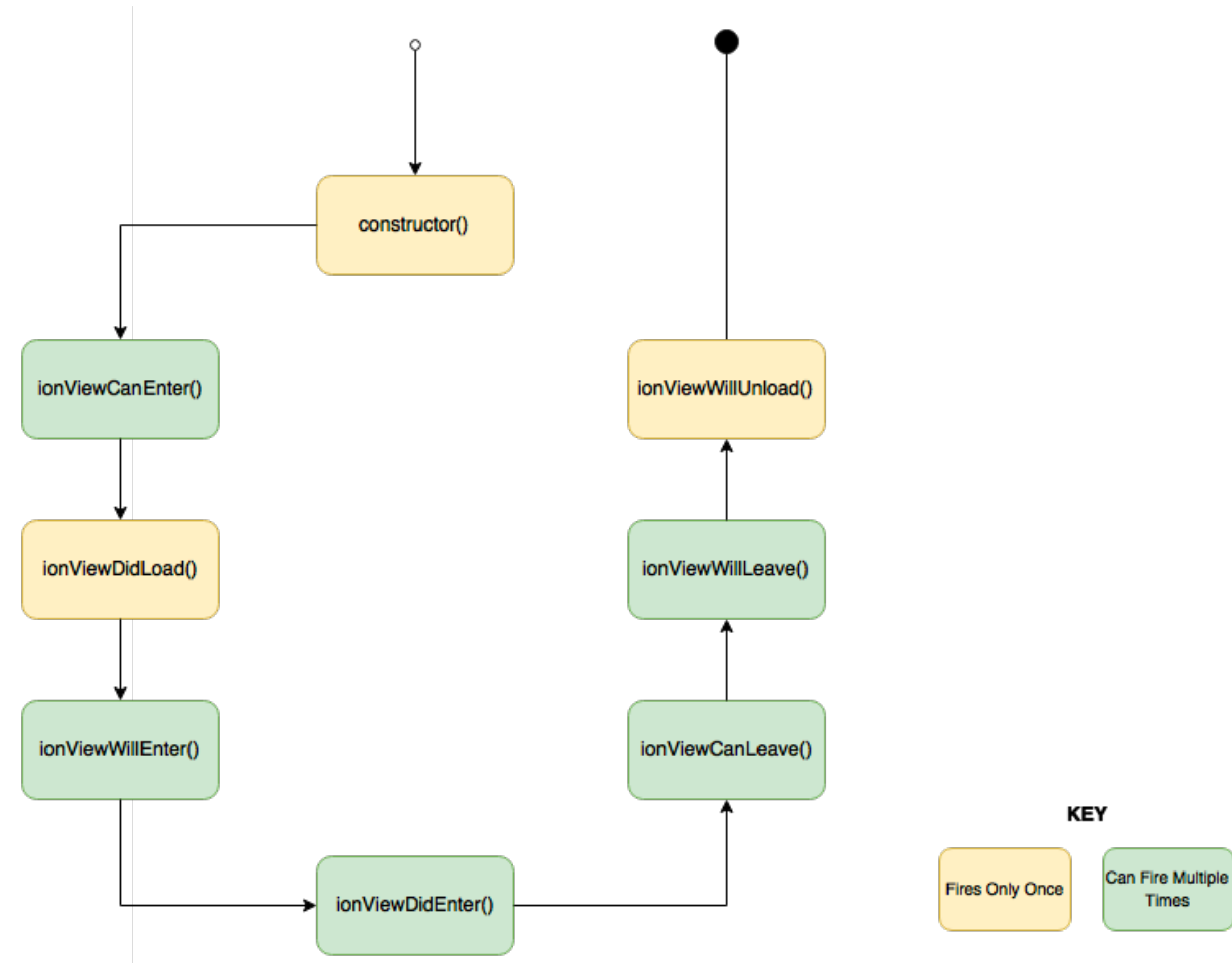
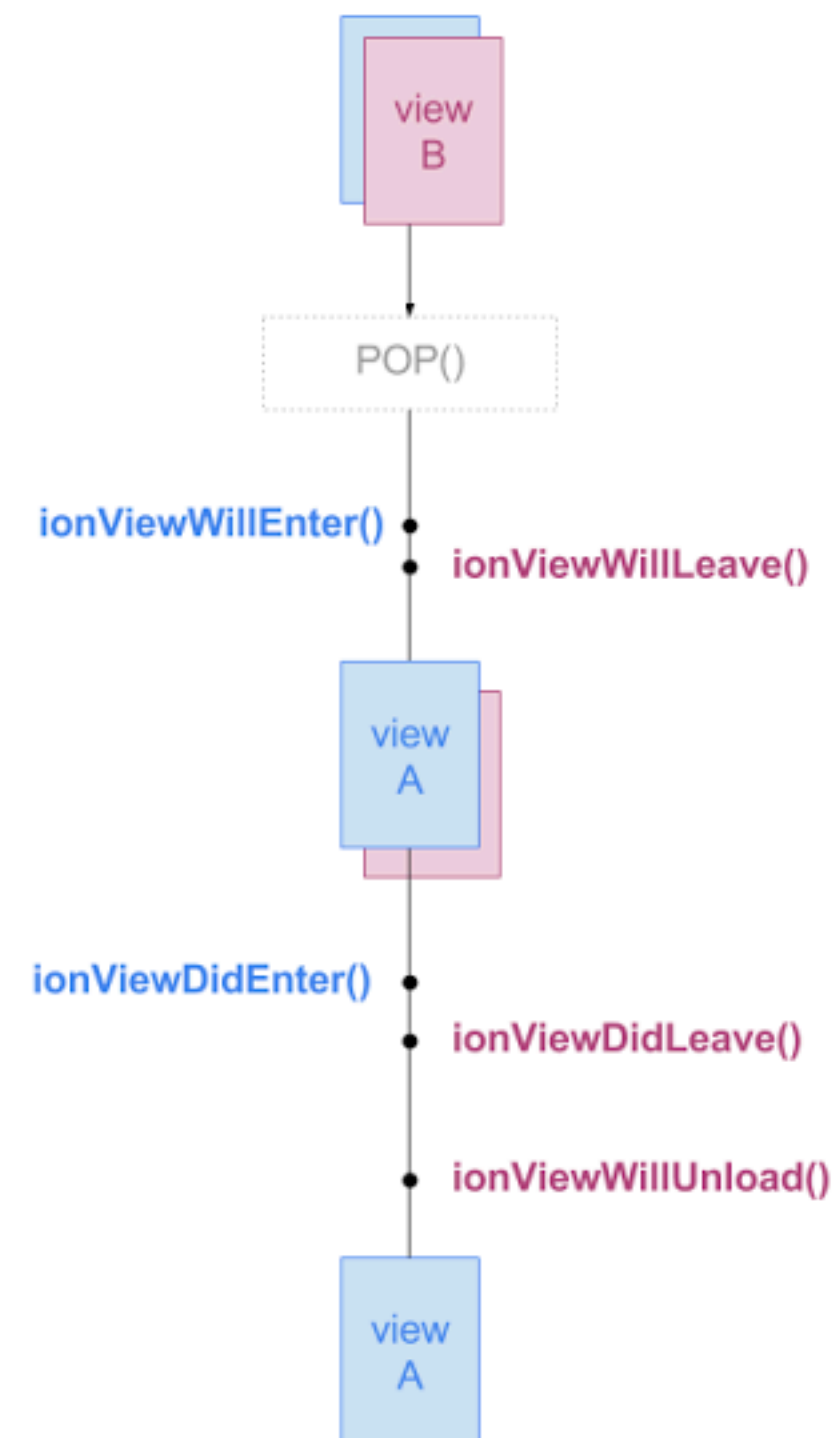
Project Structure

Principe de gestion des pages

View lifecycle events on pushing a new view



View lifecycle events on pop current view



Principe de gestion des pages ionic

Documentation

<https://blog.ionicframework.com/navigating-lifecycle-events/>
<https://saniyusuf.com/ionic-by-component-page-lifecycle/>

Déploiement et tests ionic

Tests sur mobile (nécessite l'installation des SDK) :

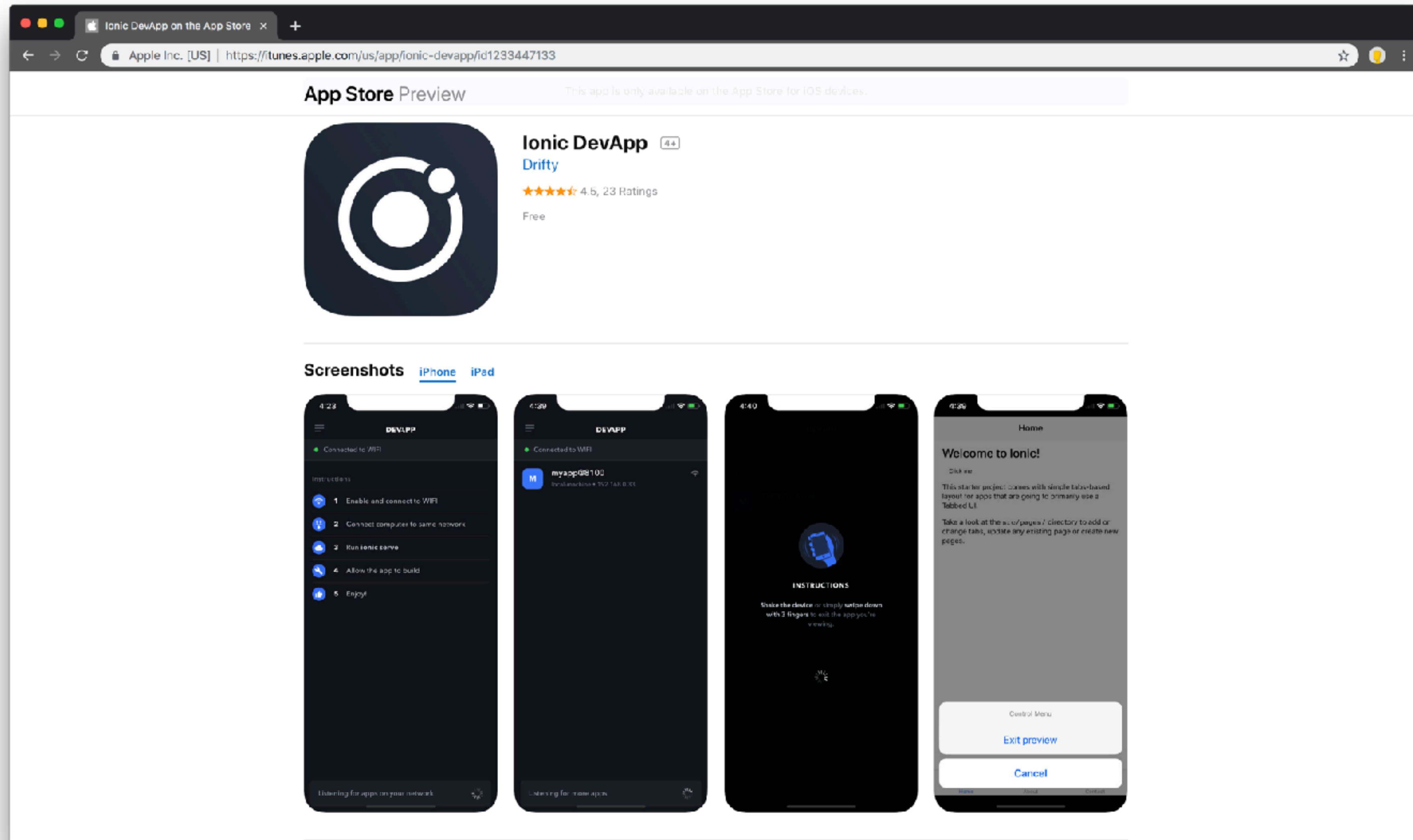
```
>  
> ionic cordova platform add android  
> ionic run android  
>
```

ou

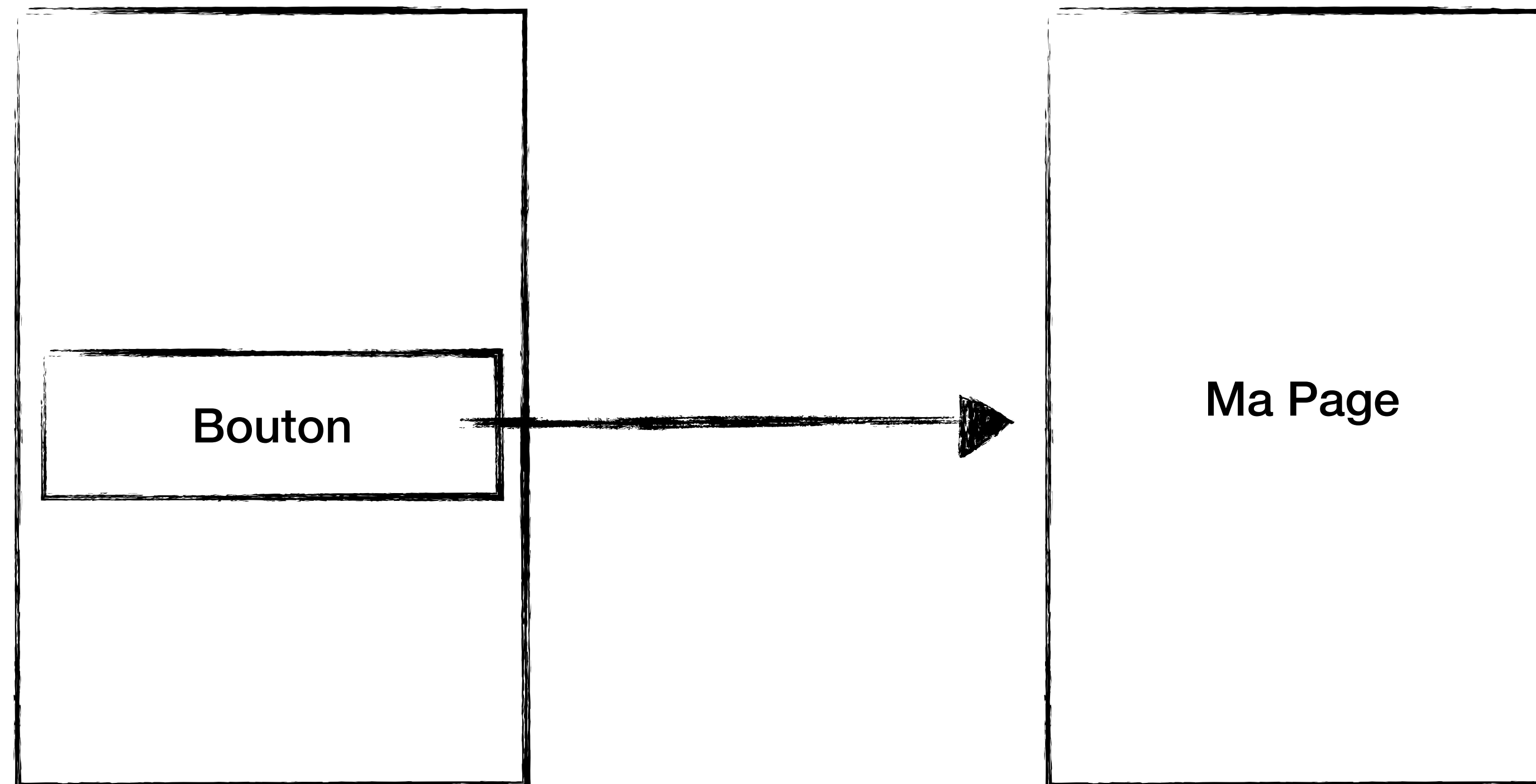
```
>  
> ionic cordova platform add ios  
> ionic run ios  
>
```

Déploiement et tests ionic

ou



Les créations d'entités ionic



Les créations d'entités ionic

Générons une page...

```
>  
> ionic generate  
>
```

Les créations d'entités

... et routons là !

app-routing.module.ts

```
1  import { NgModule } from '@angular/core';
2  import { Routes, RouterModule } from '@angular/router';
3
4  const routes: Routes = [
5    {
6      path: '',
7      redirectTo: 'home',
8      pathMatch: 'full'
9    },
10   {
11     path: 'home',
12     loadChildren: './home/home.module#HomePageModule'
13   },
14   {
15     path: 'list',
16     loadChildren: './list/list.module#ListPageModule'
17   },
18   {
19     path: 'ma-page',
20     loadChildren: './ma-page/ma-page.module#MaPagePageModule'
21   }
22 ];
23
24 @NgModule({
25   imports: [RouterModule.forRoot(routes)],
26   exports: [RouterModule]
27 })
28 export class AppRoutingModule {}
29
```

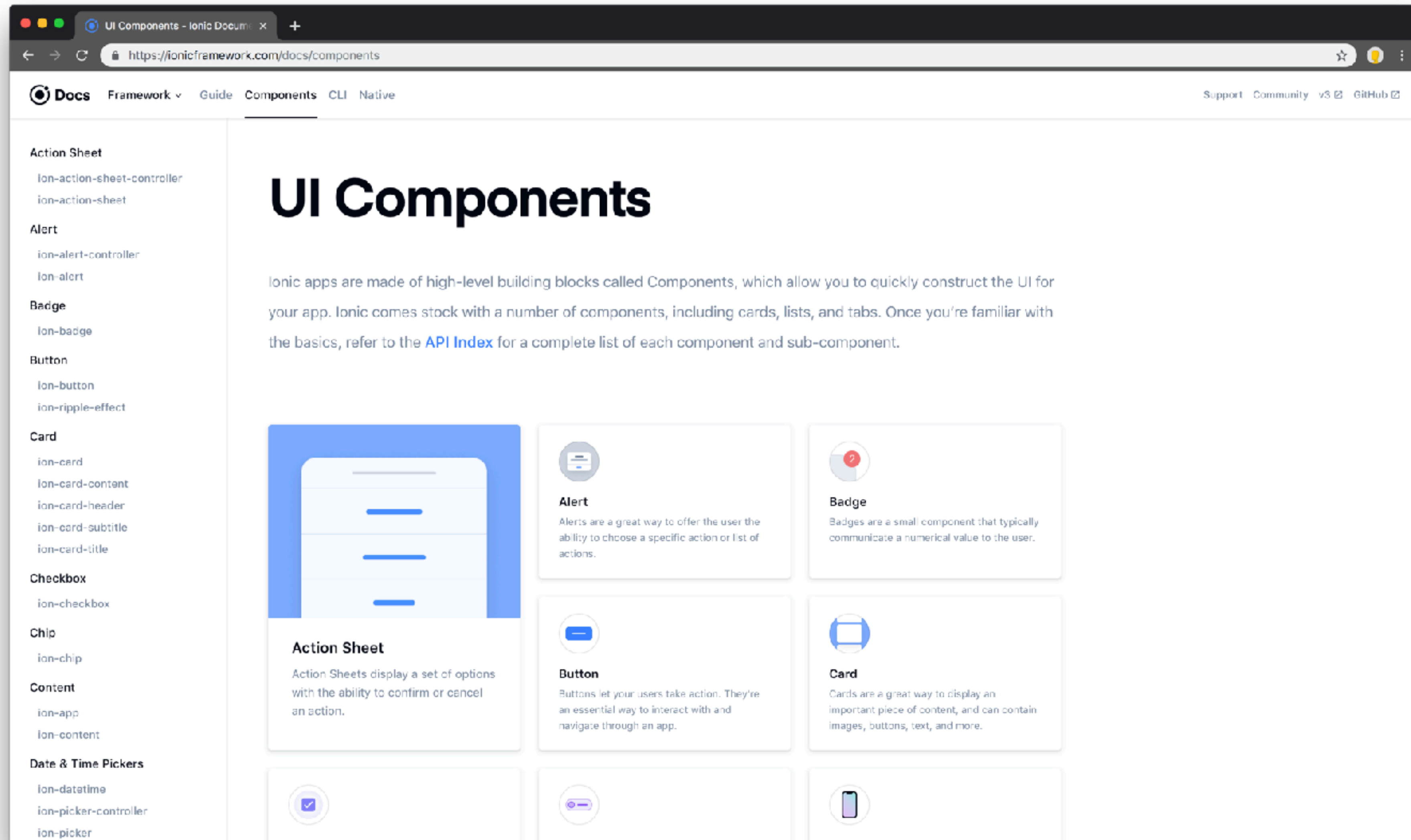
2 references

Les créations d'entités

...ajout au menu

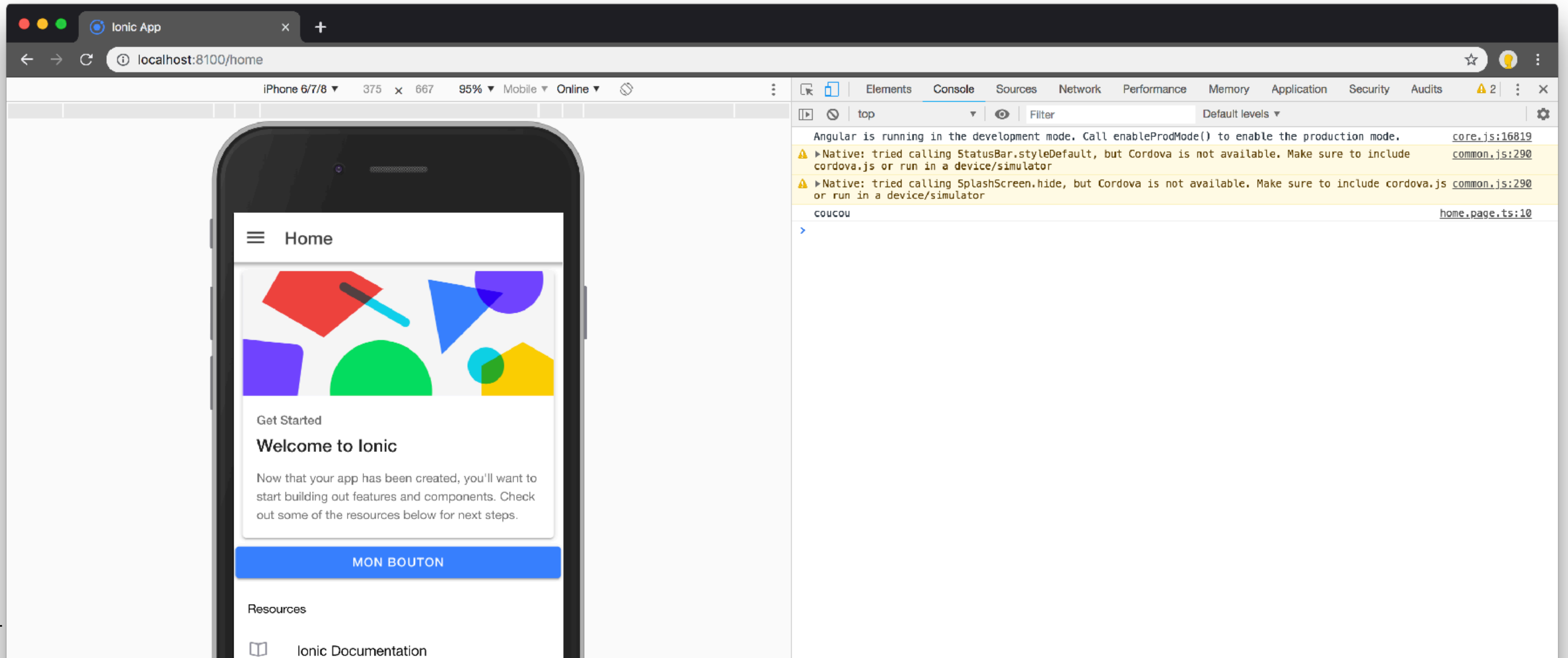
```
app.component.ts x
1  import { Component } from '@angular/core';
2
3  import { Platform } from '@ionic/angular';
4  import { SplashScreen } from '@ionic-native/splash-screen/ngx';
5  import { StatusBar } from '@ionic-native/status-bar/ngx';
6
7  @Component({
8    selector: 'app-root',
9    templateUrl: 'app.component.html'
10  })
11  export class AppComponent {
12    public appPages = [
13      {
14        title: 'Home',
15        url: '/home',
16        icon: 'home'
17      },
18      {
19        title: 'List',
20        url: '/list',
21        icon: 'list'
22      },
23      {
24        title: 'Ma Page',
25        url: '/ma-page',
26        icon: 'list'
27      }
28    ];
29
30    constructor(
31      private platform: Platform,
32      private splashScreen: SplashScreen,
```

Les components



Les components

Exemple : les ion-buttons



Les composants



naviguons

home.page.ts x

```
1 import { Component } from '@angular/core';
2 // import { NavController } from '@ionic/angular';
3 import { Router } from '@angular/router';
4
5 @Component({
6   selector: 'app-home',
7   templateUrl: 'home.page.html',
8   styleUrls: ['home.page.scss']
9 })
10 export class HomePage {
11
12   // constructor(private navCtrl: NavController) { }
13   constructor(private router: Router) { }
14
15   public notrePage() {
16     // this.router.navigate(['/ma-page']);
17     this.router.navigate(['/ma-page', { test: 3456463, back: '/home' }]);
18   }
19
20 }
21
```

home.page.html •

```
1 <ion-header>
2   <ion-toolbar>
3     <ion-buttons slot="start">
4       <ion-menu-button></ion-menu-button>
5     </ion-buttons>
6     <ion-title>
7       Home
8     </ion-title>
9   </ion-toolbar>
10 </ion-header>
11
12 <ion-content>
13   <ion-card class="welcome-card">
14     <ion-img src="/assets/shapes.svg"></ion-img>
15     <ion-card-header>
16       <ion-card-subtitle>Get Started</ion-card-subtitle>
17       <ion-card-title>Welcome to Ionic</ion-card-title>
18     </ion-card-header>
19     <ion-card-content>
20       <p>Now that your app has been created, you'll want to start building out features and c
21     </ion-card-content>
22   </ion-card>
23
24   <ion-button expand="block" (click)="notrePage()">Vers notre page!</ion-button>
25 </ion-content>
26
```

Les composants

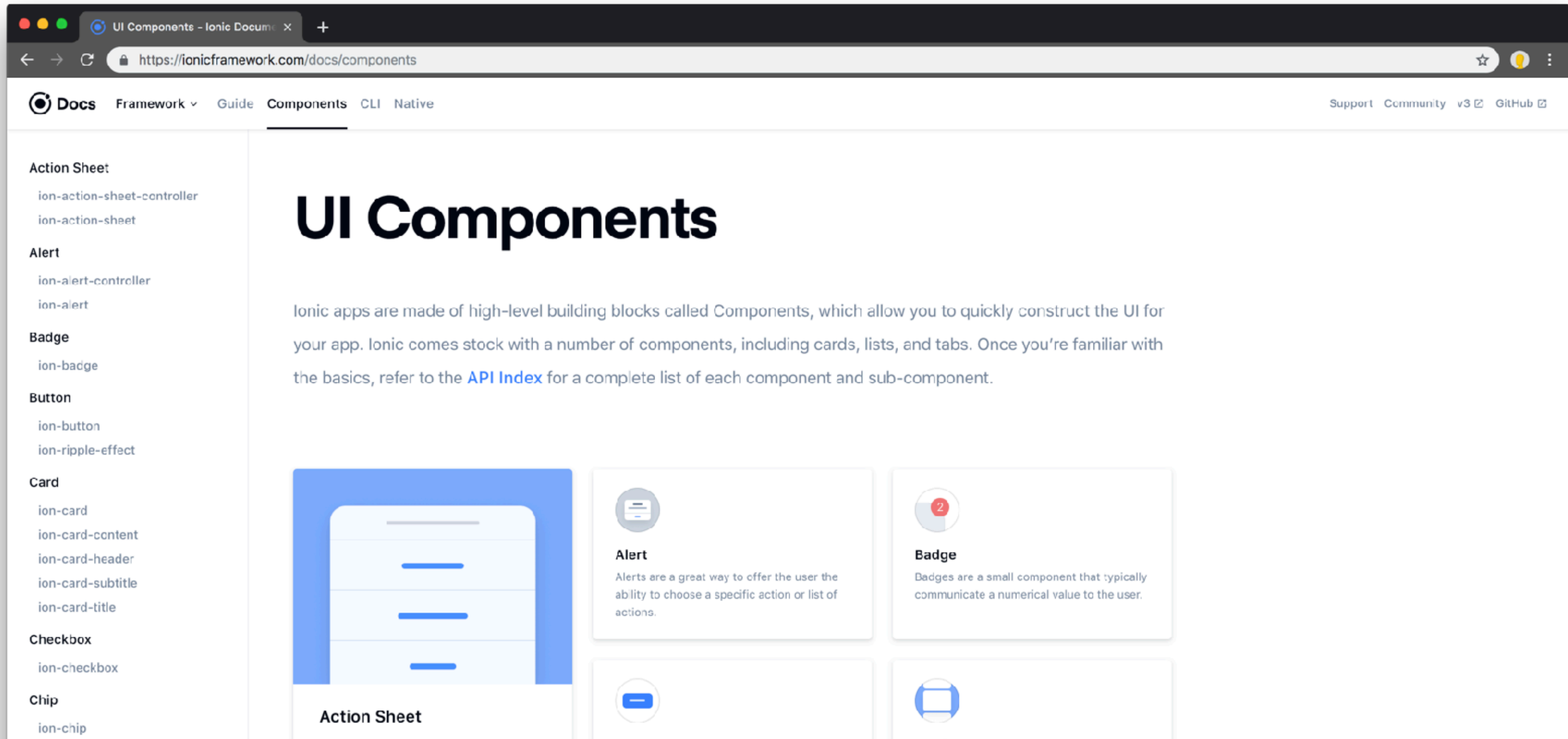


récupération des paramètres

```
ma-page.page.ts
1  import { Component, OnInit } from '@angular/core';
2  import { Router, ActivatedRoute } from '@angular/router';
3
4  @Component({
5    selector: 'app-ma-page',
6    templateUrl: './ma-page.page.html',
7    styleUrls: ['./ma-page.page.scss'],
8  })
9  export class MaPagePage implements OnInit {
10
11    public prev_page: String = '/list';
12
13    constructor(private router: Router, private route: ActivatedRoute) {
14      this.route.params.subscribe(params => {
15        console.log(params);
16        if (params.back) {
17          this.prev_page = params.back;
18        }
19      });
20    }
21
22    ngOnInit() {
23    }
24
25    public maMethode(event) {
26      this.router.navigate([this.prev_page]);
27    }
28  }
29
```


Les composants

Volontaires ! Création de page, routage, navigation, affichage de composant



The screenshot shows the Ionic Framework documentation website. The browser's address bar displays the URL `https://ionicframework.com/docs/components`. The page features a navigation menu on the left with categories like Action Sheet, Alert, Badge, Button, Card, Checkbox, and Chip. The main content area is titled "UI Components" and includes a paragraph explaining that Ionic apps are built from high-level building blocks called Components. Below this text, there are four component preview cards: "Action Sheet" (showing a list of items), "Alert" (showing a dialog box), "Badge" (showing a circular icon with a red dot), and "Button" (showing a blue button).

UI Components

Ionic apps are made of high-level building blocks called Components, which allow you to quickly construct the UI for your app. Ionic comes stock with a number of components, including cards, lists, and tabs. Once you're familiar with the basics, refer to the [API Index](#) for a complete list of each component and sub-component.

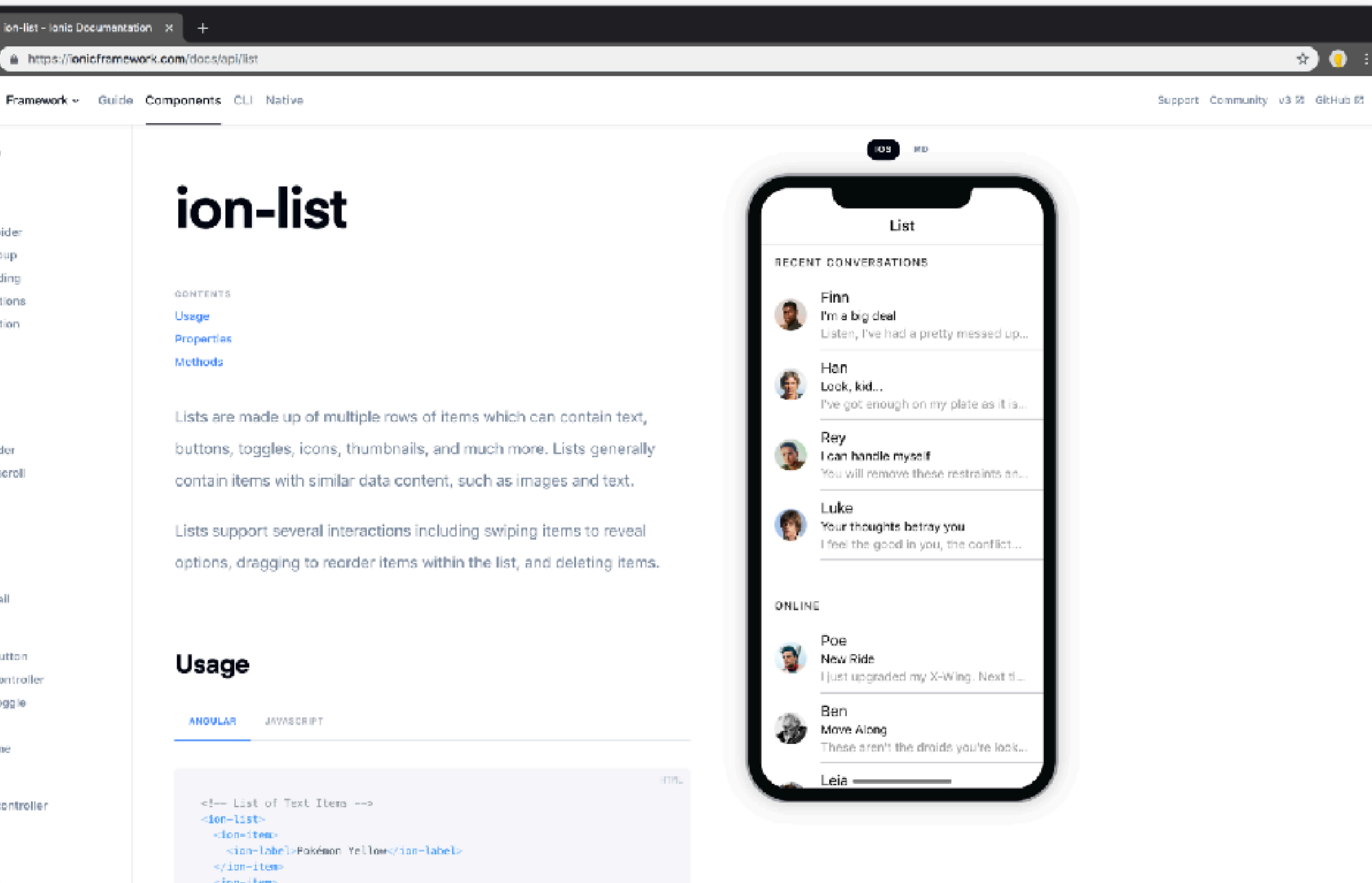
Action Sheet

Alert

Badge

Les composants ionic

Volontaires ! Création d'un page, routage, navigation, affichage de composant



The screenshot shows the Ionic Framework documentation page for the `ion-list` component. The page title is "ion-list". The left sidebar contains a table of contents with links to "Usage", "Properties", and "Methods". The main content area describes the component and includes a "Usage" section with a code snippet for an Angular template. A mobile device mockup on the right displays a list of recent conversations and online users.

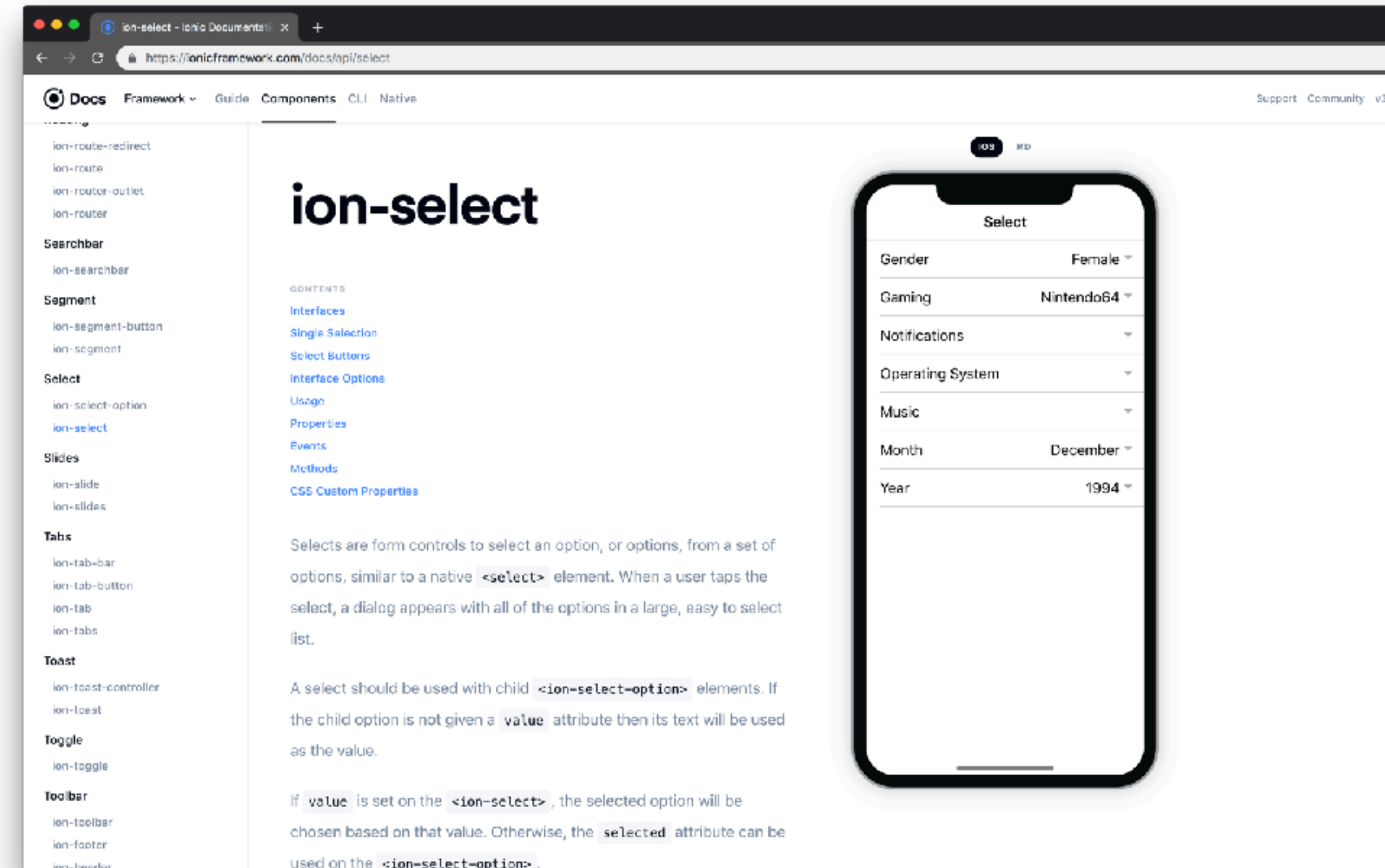
ion-list

Lists are made up of multiple rows of items which can contain text, buttons, toggles, icons, thumbnails, and much more. Lists generally contain items with similar data content, such as images and text.

Lists support several interactions including swiping items to reveal options, dragging to reorder items within the list, and deleting items.

Usage

```
<!-- List of Text Items -->
<ion-list>
  <ion-item>
    <ion-label> Pokémon Yellow</ion-label>
  </ion-item>
</ion-list>
```



The screenshot shows the Ionic Framework documentation page for the `ion-select` component. The page title is "ion-select". The left sidebar contains a table of contents with links to "Interfaces", "Single Selection", "Select Buttons", "Interface Options", "Usage", "Properties", "Events", "Methods", and "CSS Custom Properties". The main content area describes the component and includes a "Usage" section with a code snippet for an Angular template. A mobile device mockup on the right displays a select form with various options.

ion-select

Selects are form controls to select an option, or options, from a set of options, similar to a native `<select>` element. When a user taps the select, a dialog appears with all of the options in a large, easy to select list.

A select should be used with child `<ion-select-option>` elements. If the child option is not given a `value` attribute then its text will be used as the value.

If `value` is set on the `<ion-select>`, the selected option will be chosen based on that value. Otherwise, the `selected` attribute can be used on the `<ion-select-option>`.

Exercice ionic

