

Intro to Ionic

TAs Lucas and JongHo

Today's goals

- Overview of Ionic
- Ionic Components
 - Structural, Item, Inputs, Lists
- Ionic Services
- Ionic Routing
- Create a simple project
 - Components
 - Navigation between components

Setting Up Ionic

- Install Ionic

- `npm info ionic` // Check
current versions of ionic
- `npm install -g ionic@latest-4` // We'll be using 4 in class
- `ionic -v` // Check
ionic version

- Create Ionic project

- `ionic start [project_name]`

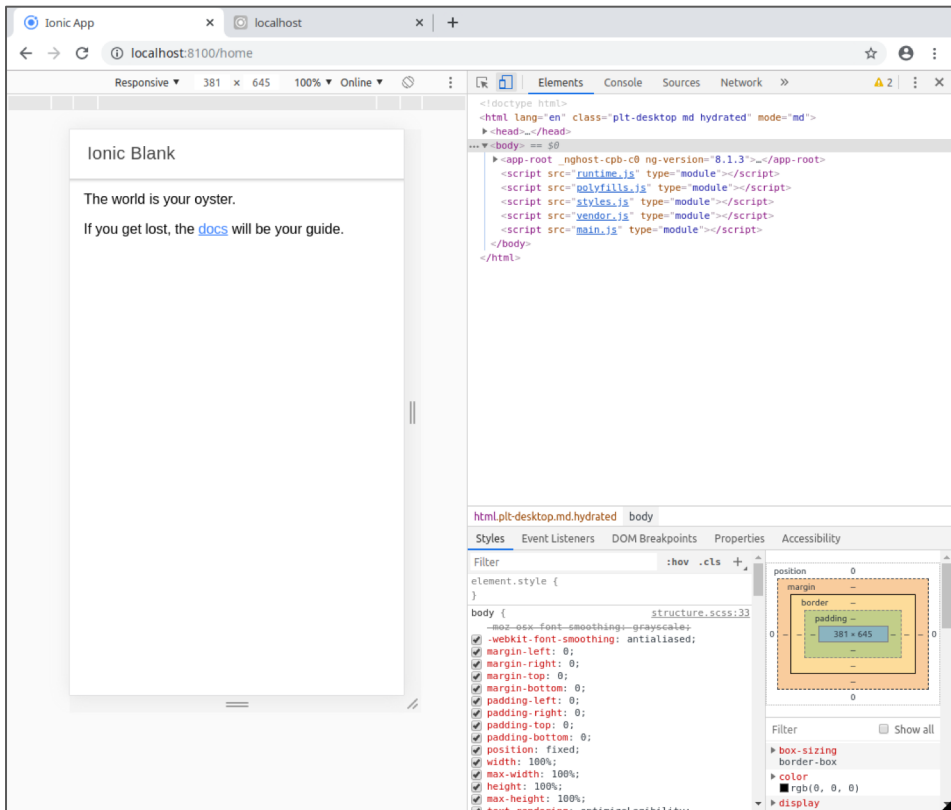
- Serve Ionic project

- `cd [project_name]` // Change directory to project
- `ionic serve` // Default: localhost:8100

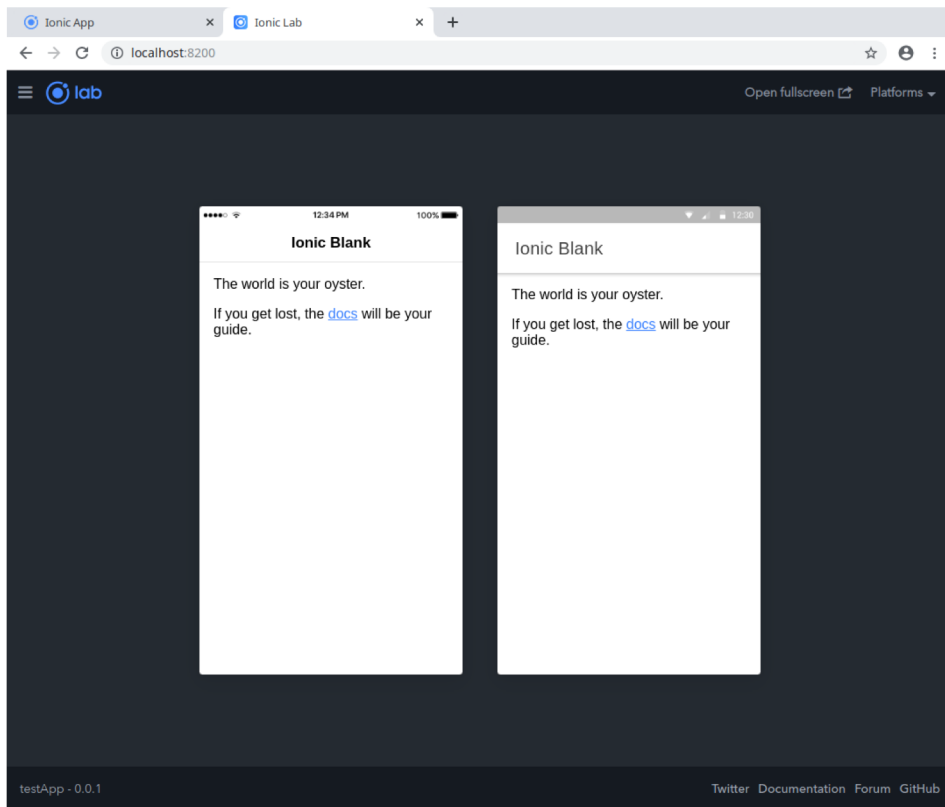
- Serve Ionic project in a lab

- `cd [project_name]` // Change directory to project
- `npm install @ionic/lab` // Install lab in the project (NO -g
flag)

Ionic Project in Browser



Ionic Lab in Browser



Ionic Introduction

- Cross-platform

- Shares **same code** across multiple platforms (iOS, Android, desktop, web etc)
- Means that we can deploy apps that are **native** to its platform!
 - E.g., generate APK file for Android

- Web Standards-based

- Ionic Framework is **built on top of** HTML, CSS, Javascript using modern Web APIs
- The framework becomes independent from a specific platform

Ionic Structural Components

```
<ion-header>
  <ion-toolbar>
    <ion-title>
      Home Page
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  content
</ion-content>

<ion-footer>
  <ion-toolbar>
    <ion-title>Ionic Footer</ion-title>
  </ion-toolbar>
</ion-footer>
```



Ionic Items

- `<ion-item>` is a basic component
- Essentially like `<div>`
- Creates a block, so its content become a “row”
- Contain other components (e.g.: a button):

```
<ion-content>
  <ion-item>
    <ion-button (click)="goToPage2()">
      Go to Page2
    </ion-button>
  </ion-item>
</ion-content>
```


Ionic Inputs

- Ionic supports various inputs
 - Today we'll only focus on text input
- Use [(ngModel)]= to bind to the variable

```
<ion-input type="text" placeholder="Item" [(ngModel)]="item"></ion-input>  
<ion-button (click)="addItem()"> Add </ion-button>
```



Ionic Lists

- Display rows of information
- Each items are very customizable

```
<ion-list *ngFor="let item of list">
  <ion-item>
    <ion-label>{{item}}</ion-label>
  </ion-item>
</ion-list>
```


```
<ion-list>
  <ion-card>
    <ion-card-header>
      <ion-card-title>My Groceries</ion-card-title>
    </ion-card-header>
  </ion-card>

  <ion-card *ngFor="let item of list">
    <ion-item>
      <ion-label>{{item}}</ion-label>
    </ion-item>
  </ion-card>
</ion-list>
```

Ionic Services

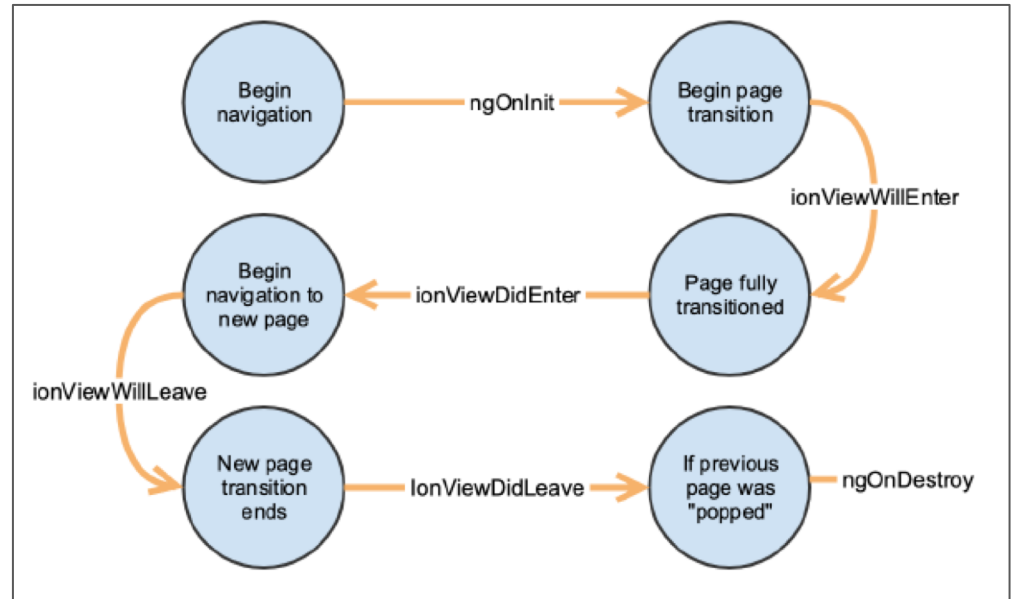
- Works very similarly with Angular Services
- Generate service
 - `ionic generate service [service_name]`
- Must include in the “providers” in the `app.module.ts`
- Injecting services work the same way as Angular

Ionic Routing

- Ionic has largely two types of views: component and pages
 - Page is a component that acts as **an entire view** (with some nested components)
 - Components are “smaller”
- To navigate around multiple pages, Ionic uses the `app-routing.module.ts`
 - Using Ionic CLI to generate pages fills out the `app-routing.module.ts` for you
 - Generate page: `ionic generate page [page_name]`
- Two different ways to navigate
 - href style  `<ion-button href="page2"> Go to page 2 </ion-button>`
 - NavController style
 - More intuitive with an example, so will be explained in the live demo

Ionic Page Lifecycle

- Various events are fired between when a page is generated to its closure
- Mostly used are
 - `ngOnInit`
 - Fired once when a page is loaded
 - `ngOnDestroy`
 - Fired right before the view is destroyed.



Ionic Live Demo

