Cross Platform Development with Ionic

By Oscar Bout

About Oscar

oscoweb.com

Amsterdam

Web development since 2000

Mobile development since 2008

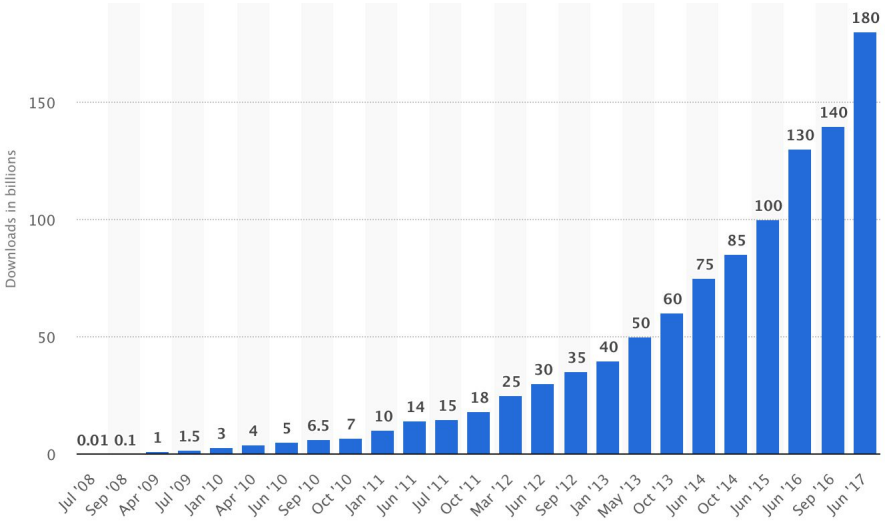
- BlackBerry
- Ios/Android
- Industry (Android barcode scanners)

Table of Contents

Introduction

Hello World

App Development

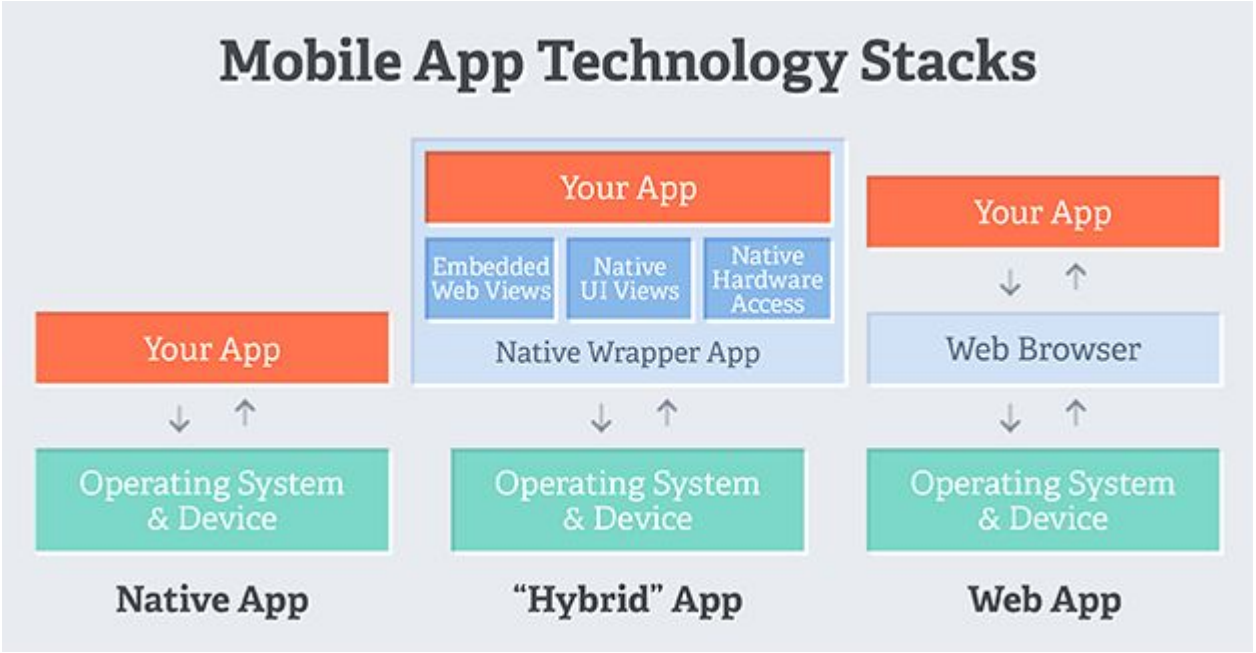


6.000.000 apps today,
180.000.000.000 downloads

3 ways to make one

- Native
- Mobile / Progressive Web Apps
- Hybrid

App Development Technology



Native Development

👍

Fast performance

Smooth and beautiful

Full Device Access

Easy Appstore access

👎

Steep learning curve

Double costs

Slow development

Hard to maintain

Web Development / PWA

👍

No learning curve

Cheap

Fast development

Easy to maintain

Works without App Store

On every device

Always up to date (web app)

👎

Slow

Limited device access

Webserver required

Bad UX/UI

No App Store

Hard to update (PWA)

Hybrid Development

👍

Full Device Access

Cheap

Fast development

Easy to maintain

On every device

App Store Access

Web / PWA app for free as a bonus

👎

Bad UX/UI

Lowest Common Denominator

Hybrid

So not native

Hybrid (!== Native)

Compiles to Native

React Native uses Javascript
(IOS, Android)

NativeScript uses Javascript
(IOS, Android)

Xamarin uses C#
(IOS, Android)

Uses Native Wrapper

Cordova uses Javascript
(IOS, Android)

Phonegap uses Javascript
(IOS, Android)

Electron uses Javascript
(Windows, Linux MacOS)

Web === every device

Cordova Frameworks

Ionic

OnsenUI

Sencha Touch

Trigger.io

Framework 7

Ionic

Ionic

Mobile app development framework since 2013

It is a complete package of components for design, animations and native access.

Major Ionic versions:

1. Based on Angular 1 using Javascript
2. Based on Angular 2 using Typescript
3. Up to Angular 5, RXJS and PWA support
4. (BETA) Free choice of Javascript Framework

Ionic as a framework

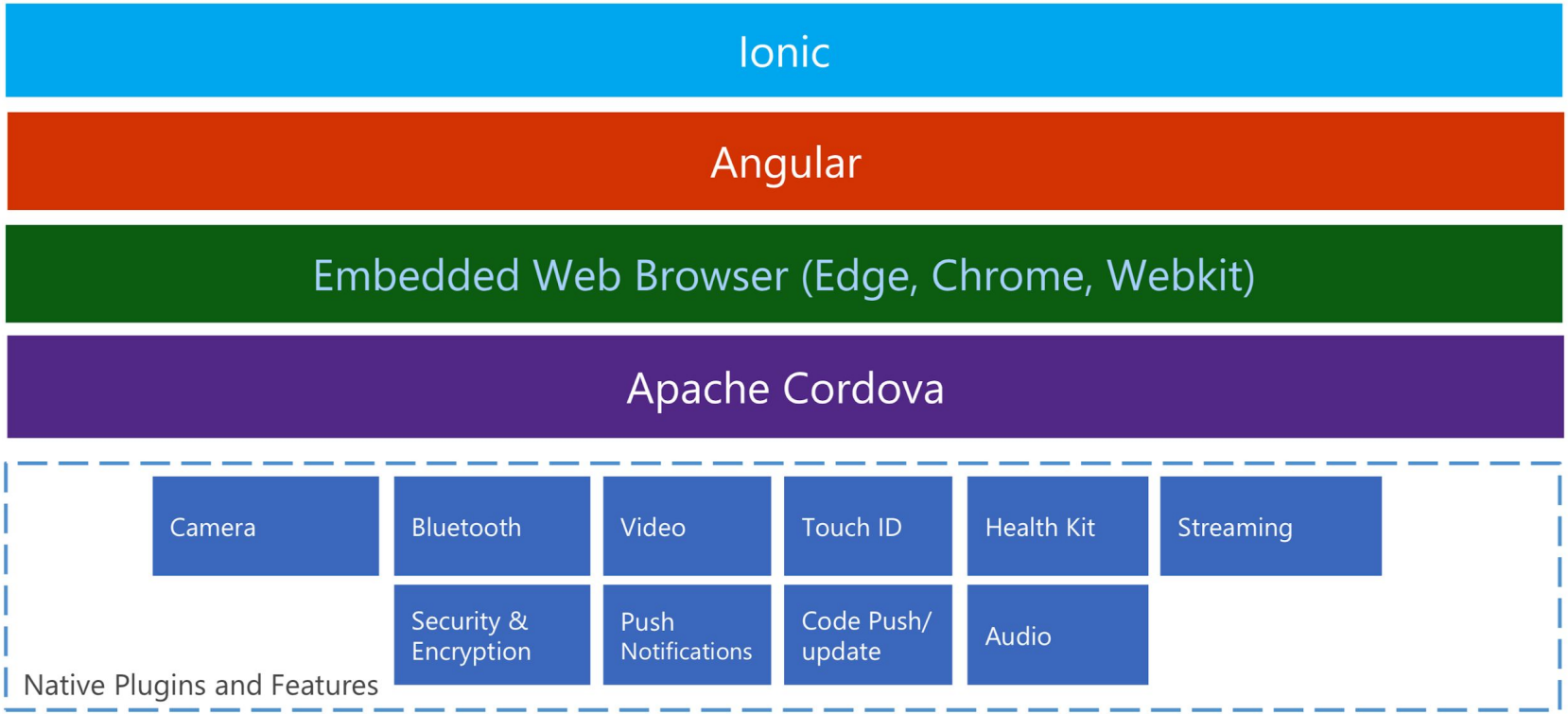
Ionic combines 3 key techniques

Ionic UI framework `<ion-button>Default</ion-button>` Default

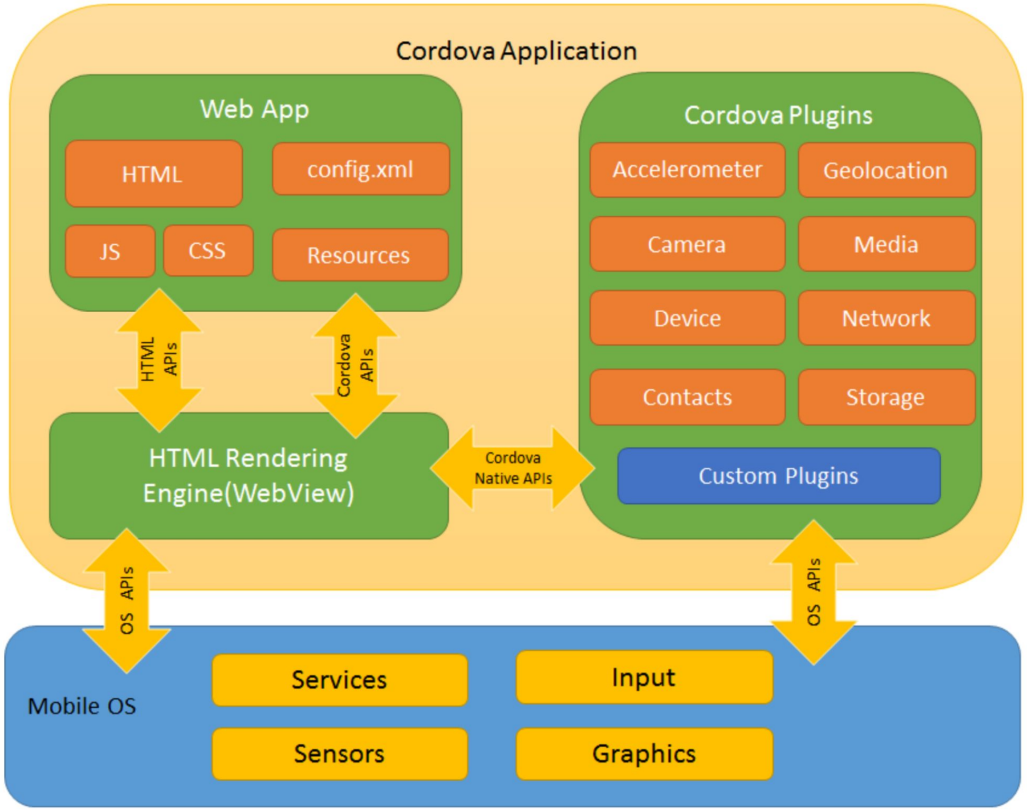
Angular

Apache Cordova

Architecture



Cordova


Cordova Application
Web App
HTML
config.xml
JS
CSS
Resources
Cordova Plugins
Accelerometer
Geolocation
Camera
Media
Device
Network
Contacts
Storage
Custom Plugins
HTML APIs
Cordova APIs
HTML Rendering Engine(WebView)
Cordova Native APIs
OS APIs
OS APIs
Mobile OS
Services
Input
Sensors
Graphics

Plugins

With plugins you can connect your code to your specific device.

Native plugins - https://ionicframework.com/docs/native/

Community - https://www.npmjs.com/

Custom plugins

Development Environment

What we need

- Nodejs - https://nodejs.org/en/ * ALWAYS USE LTS VERSION *
- Git
- Text Editor - **VS Code**, Sublime, Atom
- Decent Browser - **Chrome**, Firefox
- Java JDK and Android SDK tools
- **Android Studio** and/or XCode
- Help! We always need help... https://stackoverflow.com/ is your best friend
- Documentation - https://ionicframework.com/docs/

Assignment 1 - installation

1. Install nodejs from **https://nodejs.org/**
2. Install a text editor of your choice
3. Install a proper browser
4. Open a terminal / command prompt / Powershell
5. Type **`node -v`**
   -> nodejs is installed properly and gives you its version number
6. Type **`npm -v`**
   -> npm is installed properly and gives you its version number
   -> npm === “Node Package Manager”
7. Type **npm install cordova**
8. Type **npm install ionic**
9. Type **ionic start -l**

WDWJD?

What did we just do?

Node.js

Javascript Server-side environment

Includes a package manager NPM (Node Package Manager)

Package.json

Ionic Command Line Interface (CLI)

- Scaffolding
- Building
- Developing
- Debugging
- Previewing

Depends on Node.JS and Cordova

Useful Ionic CLI commands

`ionic -v` shows you all!

```
 _  _ ___ ___  _  ___
| || / _ \ | \| |/ __|
| __ \ (_) | | | | (__
|_||_\___/|_|_|\_|\___|  CLI PRO 4.2.1

Usage:

  $ ionic <command> [<args>] [--help] [--verbose] [--quiet] [--no-interactive] [--no-color] [--confirm] [options]

Global Commands:

  config <subcommand> ..... Manage CLI and project config values
                            (subcommands: get, set, unset)
  docs ..................... Open the Ionic documentation website
  info ..................... Print project, system, and environment
                            information
  login .................... Login to Ionic Pro
  logout ................... Logout of Ionic Pro
  signup ................... Create an account for Ionic Pro
  ssh <subcommand> ......... Commands for configuring SSH keys (subcommands:
                            add, delete, generate, list, setup, use)
  start .................... Create a new project

Project Commands:

  build .................... Build web assets and prepare your app for any
                            platform targets
  capacitor <subcommand> ... (beta) Capacitor functionality (subcommands: add,
                            copy, open, run, sync, update) (alias: cap)
  cordova <subcommand> ..... Cordova functionality (subcommands: build,
                            compile, emulate, platform, plugin, prepare,
                            requirements, resources, run)
  doctor <subcommand> ...... Commands for checking the health of your Ionic
                            project (subcommands: check, list, treat)
  generate ................. Automatically create framework features (alias:
                            g)
  git <subcommand> ......... Commands relating to git (subcommands: remote)
  integrations <subcommand> Manage various integrations in your app
                            (subcommands: disable, enable, list)
  link ..................... Connect local apps to Ionic Pro
  monitoring <subcommand> .. Commands relating to Ionic Pro error monitoring
                            (subcommands: syncmaps)
  repair ................... Remove and recreate dependencies and generated
                            files
  serve .................... Start a local dev server for app dev/testing
                            (alias: s)
  ssl <subcommand> ......... (experimental) Commands for managing SSL keys &
                            certificates (subcommands: generate)
```

Assignment 2 - start

1. Open terminal/cmd in the folder you want your code
2. Type **`ionic start test-installation tabs`**
3. When asked if you want to try Ionic 4 you answer No
4. When asked if you want to target native iOS and Android you answer Yes
5. When asked if you want to connect Ionic Pro you answer No
6. Type **cd test-installation**
7. Type **ionic serve**

Reset

- In terminal go to your code folder (cd..)
- Type ‘ls’
- Type: **git clone** https://github.com/OBout/hello-world-ionic3.git
- Type: cd hello-world-ionic3

Now we are all on the same page!

Assignment 3 - The boilerplate

1. Use keys “CTRL-C” to stop the serving of the app
2. Type `code.` to open VS Code in the project root folder
   In our case please use the hello-world-ionic3 folder
   (or just scroll there using ‘open Folder’)
3. Look at the boilerplate code

Note: Users on macOS must first run a command (Shell Command: Install 'code' command in PATH) to add VS Code executable to the PATH environment variable.

Config.xml

Config.xml is a global configuration file that controls many aspects of a cordova application's behavior.

Comparable to <appname>.plist (IOS) and manifest.xml (Android).

Examples of its function:

- Whitelisting
- Splash screen behaviour
- Status Bar colour
- Plugin constants (API keys Google Maps)

Package.json (1)

Lists the packages that your project depends on.

Allows you to specify the versions of a **package** that your project can **use** using semantic versioning rules.

Remember this:

Edit with care, it **makes** AND **breaks** your entire application!

Package.json (2)

Installs in ‘node_modules’ folder

Uses semver (semantic versioning)

Lists dependencies - what your app needs on compile & runtime

Lists development dependencies - what is needed for development

Semantic Versioning (semver)

major.minor.patch

- Major - API breaking changes
- Minor - New features but always backwards compatible
- Patch - Bug fixes and security patches, no new features and always backwards compatible.

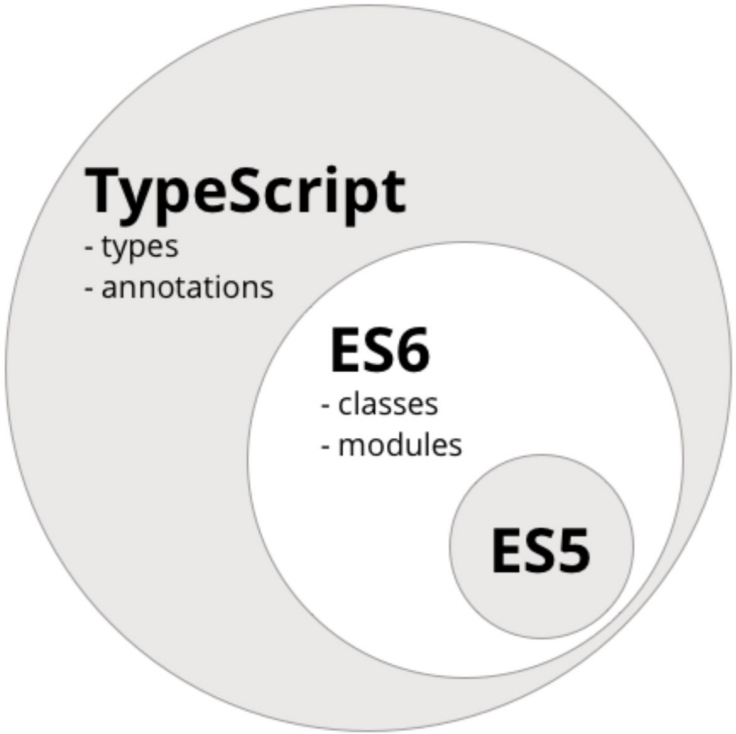
Semantic Versioning (2)

major.minor.patch

- Major - API breaking changes
- Minor - New features but always backwards compatible
- Patch - Bug fixes and security patches.

```
{
  "dependencies" : {
    "dependency_a" : "*", // any version will do
    "dependency_b" : "1.0.0 - 1.1.9", // any version between of equal to bounds
    "dependency_c" : "3.5.*", // any version in 3.5 range
    "dependency_d" : ">=6.1.2", // any version bigger than 6.1.2
    "dependency_e" : "~1.2.1", // any version >= 1.2.1 < 1.3.0 (patch-level)
    "dependency_f" : "^4.0.4", // any version >= 4.0.4 < 5.0.0 (version level)
  }
}
```

TypeScript



Features

Adds strongly typed to JavaScript

Ability to define classes

Classes can implement interfaces and be generic

Arrow functions

Concatenate strings with template strings

Transcompiles to JavaScript

TypeScript - Type Annotations

Javascript

```
var num = 5;
var name = "Oscar";
var anything = 123;
var list = [1, 2, 3];

function square(num) {
  return num * num;
}
```

Typescript

```
const num: number = 5;
const name: string = "Oscar";
let anything: any = 123;
const list: Array<number> = [1, 2, 3];
```

```
public square(num: number): number {
  return num * num;
}
```

TypeScript - Classes

Javascript

```
var Person = (function () {

  function Person(name) {
    this.name = name;
    this.speek = function (word) {
      alert(word);
    }
  }
  return Person;
}());

var aPerson = new Person("Oscar Bout")
aPerson.speek("Typescriptish Ionical")
```

Typescript

```
class Person {

  constructor(public name: string) { }

  speek(word: string) {
    alert(word)
  }

}

const aPerson = new Person("Oscar Bout")
aPerson.speek("Typescriptish Ionical")
```

TypeScript - Interfaces

Provide type safety

```typescript
interface Person {
  id: number;
  name: string;
  email: string;
  age: number;
  phone: string;
  birthday: Date;
}
```

```typescript
const contact: Person = {
  id: 1,
  name: 'Oscar Bout',
  email: 'bb@oscoweb.com',
  age: 44,
  phone: '+31653422514',
  birthday: new Date('1974-05-16')
}
```

Modules (1)

ES6+

Enable modular code

Uses export/import mechanism

Modules (2)

```
// car.ts

export class Car {

  constructor(public color: string) { }

}
export class Convertible extends Car {

}
```

```
// app.ts

import { Car, Convertible } from './car'

const Bmw = new Car ('red')
let Cabrio = new Convertible('white')
```

Angular

- Open Source
- Started 2009
- Google
- Data-driven
- Scalable

Components

- Everything in Angular is a component
- Elements and logic are combined in a component

Angular component

Every component must be part of a NgModule

```
import { Component, } from '@angular/core';

@Component({
  selector: 'page-modalpage',
  templateUrl: 'modalpage.html',
  styleUrls: ['modalpage.scss'],
})
export class ExampleComponent {
  constructor( ) {
  }
}
```

NgModule (Angular Module)

- Consolidates components, directives and pipes into cohesive blocks of functionality
- Focused on a feature area, application business domain, workflow, or common collection of utilities
- Can import other modules

Angular Module

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { MyComponent } from './my.component';

@NgModule({
    imports: [BrowserModule],
    declarations: [MyComponent],
    bootstrap: [MyComponent]
})

export class AppModule { }
```

app.module.ts

The aorta of your Ionic 3 app

Assignment 4

Open app.module.ts in your example project

Interacting with data (DOM ‹-› JS)

{{ }} - displays components value, or function result in HTML
ie: **{{showSomething()}}** or **{{showSomethingAsGet}}**

[ ] - binds to the property of a custom element

( ) - event handling
ie. **(click)=”doSomethingFunction()”**

[( )] - two way binding
ie. **<input type=“number” [(ngModel)]=”jsvariable” />**

***** Manipulate DOM structure
ie. ***ngIf=”result”** or ***ngFor=”let i of items”**

Assignment 5

In your example project:

1. Open pages-home-home.html
2. Make a button (preferably an ionic button --- use docs!)
3. Add the click event with a function “**clickButton()**”
4. Make a “**clickButton()**” function in your home.ts file
5. Make the button **console.log(‘clicked’)**

Result

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 `src/pages/` directory to add or change tabs, update any existing page or create new pages.

CLICK [redacted]

Home About

Elements Console Sources Network Performance Memory Application Security

top Filter Default levels Group similar

```
Angular is running in the development mode. Call enableProdMode() to enable the production mode.
▶ Ionic Native: tried calling StatusBar.styleDefault, but Cordova is not available. Make sure to a
index.html
▶ Ionic Native: tried calling SplashScreen.hide, but Cordova is not available. Make sure to a) run
index.html
clicked [redacted]
```

>

Reset

git checkout assignment5

Assignment 6

In your example project:

1. Open pages-home-home.ts
2. Make a private accessible number variable **“clickValue”**
3. Default the number to 0
4. Make the function “clickButton()” add +1 to **clickValue** at each click
5. Make a get function “showButton()” to return the value of clickValue
6. Make a ion-card (--- use docs!) with title **“Clicked Value”**
7. Show the value of “**showButton**” there in the **ion-card-content**

Result

Home

Welcome to Ionic!

This starter project comes with simple tabs-based layout for apps that

Take a look at the `src/pages/` directory to add or change tabs, updat

CLICK

Clicked Value

showButton: 1

Reset

git checkout assignment6

Assignment 7

In your example project:

1. Open pages-home-home.ts
2. Make a public accessible number variable **“twoWayValue”**
3. Default the number to 0
4. Show the value of **twoWayValue**
5. Make an ion-input number type with two way binded value **twoWayValue**
6. Make the function “clickButton()” add +1 to **twoWayValue** at each click

Result

Welcome to Ionic!

This starter project comes with simple ta

Take a look at the `src/pages/` director

CLICK

Clicked Value

showButton: 1

twoWayValue: 1

input:

1

Reset

git checkout assignment7

WDWJD?

What did we just do?

Services (providers)

- A class with the @Injectable decorator
- The service needs to be added to the provider array of a module
- Can be injected inside another class' constructor

```
import { Injectable } from '@angular/core';

@Injectable()
export class StorageProvider {

  constructor() {
    console.log('Hello StorageProvider Provider');
  }

}
```

Create Services

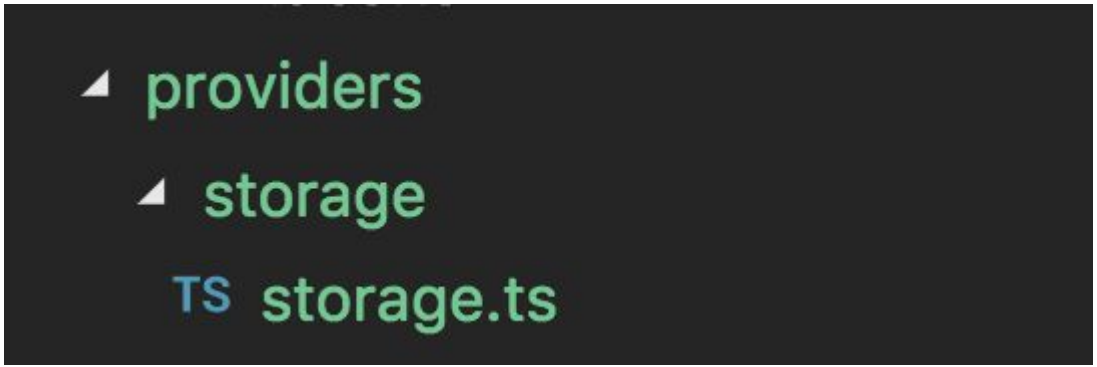
ionic g provider storage

Assignment 8

In your example project:

1. Type ionic g provider storage

Result



WDWJD?

What did we just do?

Reset

git checkout assignment8

Restart development server!

After adjusting app.module **ALWAYS** restart your server!

- In terminal use keys: **CTRL-C**
- (re)type: **ionic serve**
- Close old browertab

Assignment 9

In your example project:

1. Open providers-storage-storage.ts
2. Make a private accessible number variable **“numberFromHome”**
3. Default the number to 99
4. Make a public get and a set function and call them both **“getNumberFromHome()”**

Result

```
@Injectable()
export class StorageProvider {

  private numberFromHome: number = 99

  constructor(public http: HttpClient) {
    console.log('Hello StorageProvider Provider')
  }

  public get getNumberFromHome() {
    return this.numberFromHome
  }

  public set getNumberFromHome(arg: number) {
    this.numberFromHome = arg
  }
```

Reset

git checkout assignment9

Assignment 10

In your example project:

1. Open **pages-about.ts**
2. Import the Storage Service you've created before
3. Make it private available in the constructor
4. Make a public function "**fromHome()**" showing the value of **numberFromHome** as stored in the StorageProvider (hint: use the get)
5. Show the value in **about.html**
6. See that it breaks when you navigate to "About"
   ERROR ERROR ERROR

Result

```ts
import { Component } from '@angular/core';
import { NavController } from 'ionic-angular';
import { StorageProvider } from '../../providers/storage/storage'

@Component({
  selector: 'page-about',
  templateUrl: 'about.html'
})
export class AboutPage {

  constructor(
    public navCtrl: NavController,
    private storage: StorageProvider
    ) {

  }

  public get fromHome() {
    return this.storage.getNumberFromHome
  }

}
```

```html
<ion-header>
  <ion-navbar>
    <ion-title>
      About
    </ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>
{{fromHome}}
</ion-content>
```

Result

ERROR Error: Uncaught (in promise): Error:
StaticInjectorError(AppModule)[StorageProvider -> HttpClient]:

StaticInjectorError(Platform: core)[StorageProvider -> HttpClient]:

NullInjectorError: No provider for HttpClient!

FIX

- Goto app.module.ts
- Add:
  import { HttpClientModule } from '@angular/common/http';
- Add **HttpClientModule** to the entryComponents Array
- Save

Result

About

99

Reset

git checkout assignment10

WDWJD?

What did we just do?

Assignment 11

In your example project:

1. Open pages-home.ts
2. Import the Storage Service you've created before
3. Make it private available in the constructor
4. Store the clickValue to the storage provider in the clickButton function

Result

About

1

Reset

git checkout assignment11

WDWJD?

What did we just do?

Building and Deployment

Assignment 12

- To see the installed and available platforms type **“ionic cordova platform ls”**

- To add android type **“ionic cordova platform add android”**

Result

New line in config.xml

New plugins in package.json

New folder: platforms-android

Reset

git checkout assignment12

Running - group assignment

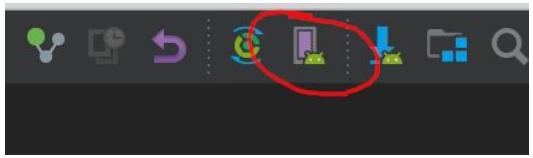
- Type **“ionic cordova run android”**

Common error

```
error: device still connecting
Your emulator is out of date, please update by launching Android Studio:
 - Start Android Studio
 - Select menu "Tools > Android > SDK Manager"
 - Click "SDK Tools" tab
 - Check "Android Emulator" checkbox
 - Click "OK"
```

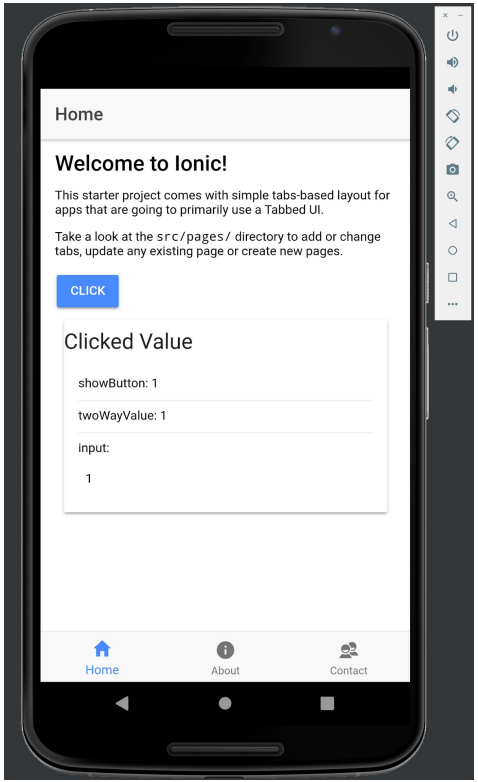
FIX

- Open a random android studio project
- Open AVD (Android Virtual Device)



- Install new emulator (ie. Nexus 6) with at least API 28.
- Start it up
- Re-run **“ionic cordova run android”**

Result

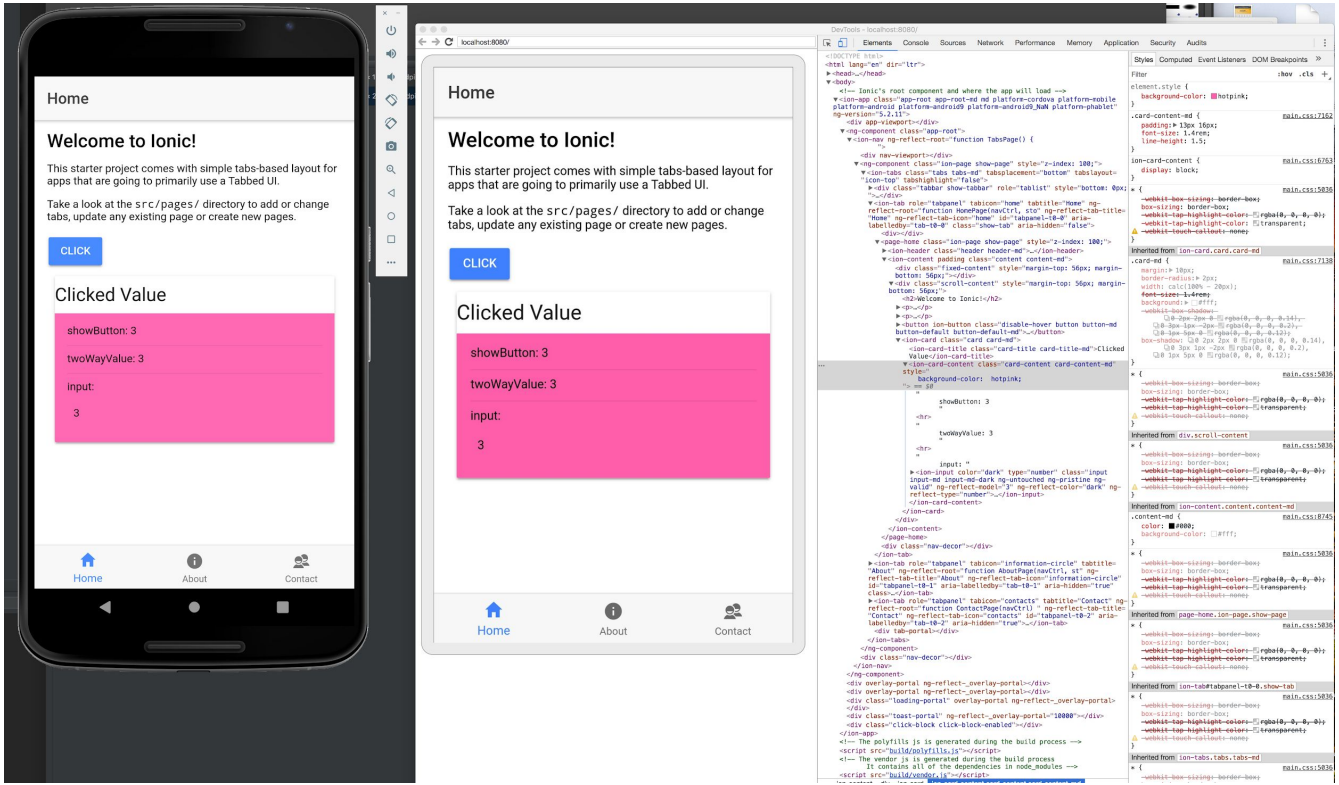

Home
Welcome to Ionic!
This starter project comes with simple tabs-based layout for apps that are going to primarily use a Tabbed UI.
Take a look at the src/pages/ directory to add or change tabs, update any existing page or create new pages.
CLICK
Clicked Value
showButton: 1
twoWayValue: 1
input:
1
Home
About
Contact

Assignment 13

- Open Google Chrome
- Open url: **chrome://inspect/#devices**
- Klik on your running emulator
- Select the ion-card section in the inspector using the selector on the top left of the debugger 

- Make the background hot-pink **(background-color: hotpink;)**

Result


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 src/pages/ directory to add or change tabs, update any existing page or create new pages.
CLICK
Clicked Value
showButton: 3
twoWayValue: 3
input:
3
Home
About
Contact

WDWJD?

What did we just do?

ionic cordova run android

1. Ionic is build to deployable javascript
2. Cordova made a wrapper around (1.)
3. Android build it all (1. and 2.) to a deployable .apk file
4. Android-build is deployed to an active device

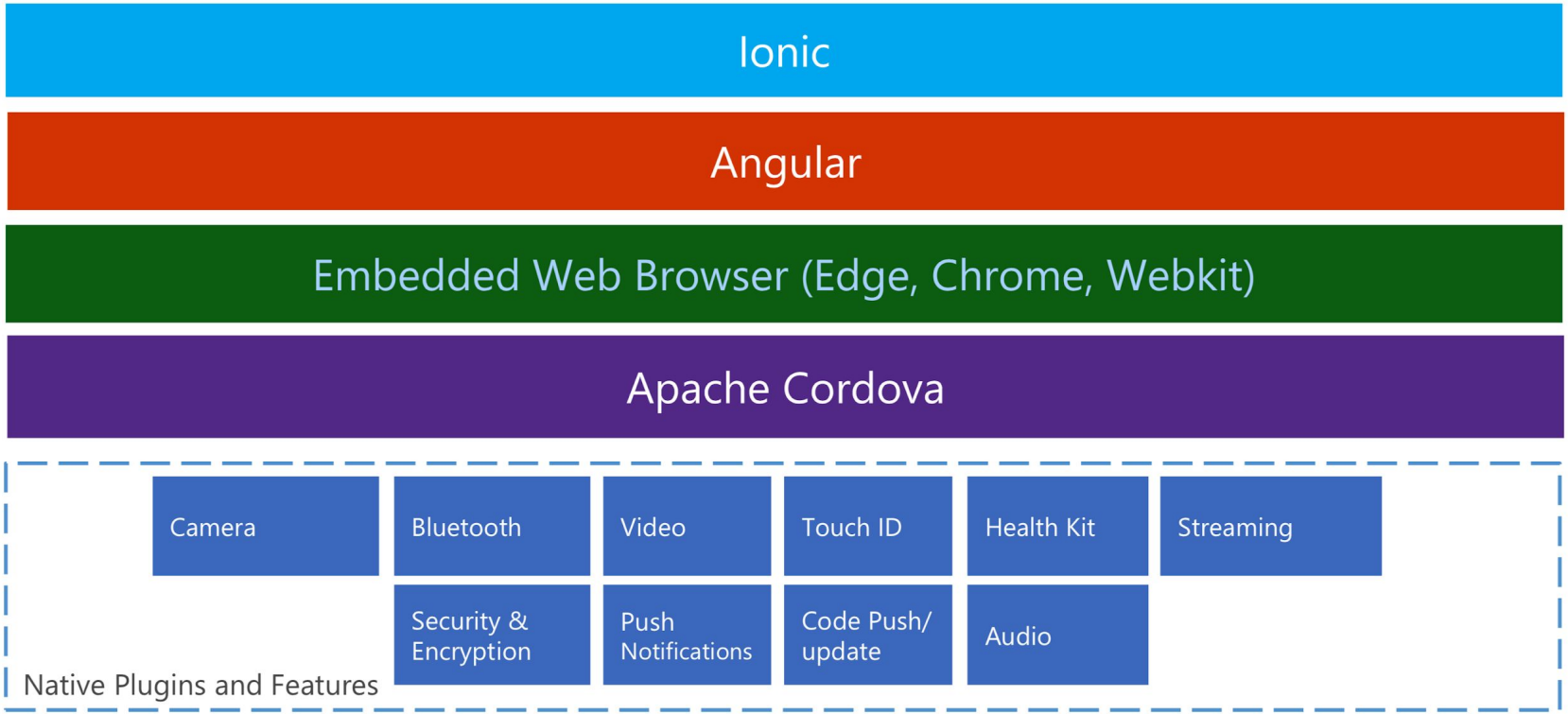
ionic cordova build android

1. Ionic builds
2. Cordova builds
3. Android builds

With options!

- **ionic cordova build android --prod --release** for production builds
- **ionic cordova build android** for development builds (usually slower in performance)

Remember this?



Code Signing

Code signing is an instrument used to be trusted by app stores

Code signing prevents 3th party access

Code signing is a way to make your APK unique

Androids code signing

Jks file (java key store) - you create your own or use your company’s

Apple code signing

- Get a license from Apple
- Developer Account - free only usable for testing and local
- IOS developer license (89,00/year) - App Store distribution
- Enterprise License (299,00/year) limited distribution for employees no appstore needed

Code signing with Ionic

- Ionic CLI
  https://ionicframework.com/docs/intro/deploying/
- Android Studio
- XCode

Deploying to production

- For browser builds, you copy the build folder and paste it in any apache/iis served folder

- Android needs the signed APK to be uploaded in their “Play Store Console”
  https://play.google.com

- Apple uses XCode and iTunes connect
  https://appstoreconnect.apple.com

Ionic In Depth

Infinite Loop

Project Structure

src	This directory will contain the actual application code.
hooks	This directory contains scripts that are used by Cordova during the build process.
node_modules	The supporting third-party libraries can be found here.
resources	The default icons and splash screens for both iOS and Android are included.
platforms	This directory contains the specific build elements for each installed build platform.
plugins	This directory contains Cordova plugins.
www	This directory contains the index.html that will bootstrap the Ionic application with the transpiled output from the app directory.
.gitignore	A default gitignore file is generated.
config.xml	Used by Cordova to define various app-specific elements.
ionic.config.json	Used by the Ionic CLI to define various settings when executing commands.
package.json	A list of all the npm packages that have been installed for the project.
tsconfig.json	The tsconfig.json file specifies the root files and the compiler options required to compile the project.
tslint.json	TypeScript linter rules.

Previewing the app

ionic serve
Opens a webbrowser on port 8100 with live reload

ionic serve --lab
Opens a webbrowser on port 8100 with live reload
Runs AS IF it was run on a device

ionic cordova run android -l (live reload)
Runs the app on an Android device/emulator with live reload

App Module

```
import { NgModule, ErrorHandler } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { IonicApp, IonicModule, IonicErrorHandler } from 'ionic-angular';
import { MyApp } from './app.component';
import { StatusBar } from '@ionic-native/status-bar';
import { SplashScreen } from '@ionic-native/splash-screen';
import { HttpClientModule } from '@angular/common/http';
import { AboutPage } from '../pages/about/about';

@NgModule({
  declarations: [
    MyApp,
    AboutPage,
  ],
  imports: [
    BrowserModule,
    IonicModule.forRoot(MyApp),
    HttpClientModule
  ],
  bootstrap: [IonicApp],
  entryComponents: [
    MyApp,
    AboutPage,
  ],
  providers: [
    StatusBar,
    SplashScreen,
    {provide: ErrorHandler, useClass: IonicErrorHandler}
  ]
})
export class AppModule {}
```

App Module

- IonicModule - Bootstrap the App & Config object
- IonicApp - The main app defined in index.html
- EntryComponents - Contains components the Angular compiler needs
- IonicErrorHandler - display errors as overlay

App template

- app.html
  <ion-nav> represents the NavController, helps you nagivate
- [root] represents the start page
- app.component.ts

```
@Component({
  templateUrl: 'app.html'
})
export class MyApp {
  rootPage:any = TabsPage;
```

App Component

```
import { Component } from '@angular/core';
import { Platform } from 'ionic-angular';
import { StatusBar } from '@ionic-native/status-bar';
import { SplashScreen } from '@ionic-native/splash-screen';

import { TabsPage } from '../pages/tabs/tabs';

@Component({
  templateUrl: 'app.html'
})
export class MyApp {
  rootPage:any = TabsPage;

  constructor(platform: Platform, statusBar: StatusBar, splashScreen: SplashScreen) {
    platform.ready().then(() => {
      // Okay, so the platform is ready and our plugins are available.
      // Here you can do any higher level native things you might need.
      statusBar.styleDefault();
      splashScreen.hide();
    });
  }

}
```

Notable Directories

- Assets - various uncompilable assets such as Images, Fonts
- Pages - grouping of all pages in the app
- Theme - Sass files for custom theming (overriding defaults)

Project Structure

https://github.com/Robinyo/ionic-angular-schematics

Ionic Component HTML structure

Ionic power

ion-header

- Parent component for:
  Navbar,
  Toolbar

- Is platform aware *

- Fixed above content during page transitions

ion-navbar

- Navigational Toolbar
- Holds Backbutton, Menu toggle, Title
- Behaviour depends on platform (but not always!)

ion-title

- Page Title

Assignment 14 - change title

1. Open about.html
2. Make the title show the “fromHome” value

Result

Number from home 1

fromHome: 1

Reset

git checkout assignment14

Buttons

- <button ion-button color=”secondary”>hello</button>
- Buttons can be enhanced with many attributes:
  color
  outline
  clear
  round
  full
  icon-only
  large

Icons

- Ionic ships with a nice collection of icons
- Icons have Material Design and iOS versions
- Ionic will automatically use the correct version based on the platform
- Platform specific by using attributes
- <ion-icon name="add"></ion-icon>

Icon Button

- Buttons can have icons
- Left of text
- Right of text
- Icon only
- <button ion-button icon-only><ion-icon name="home"></ion-icon></button>

Navbar Button

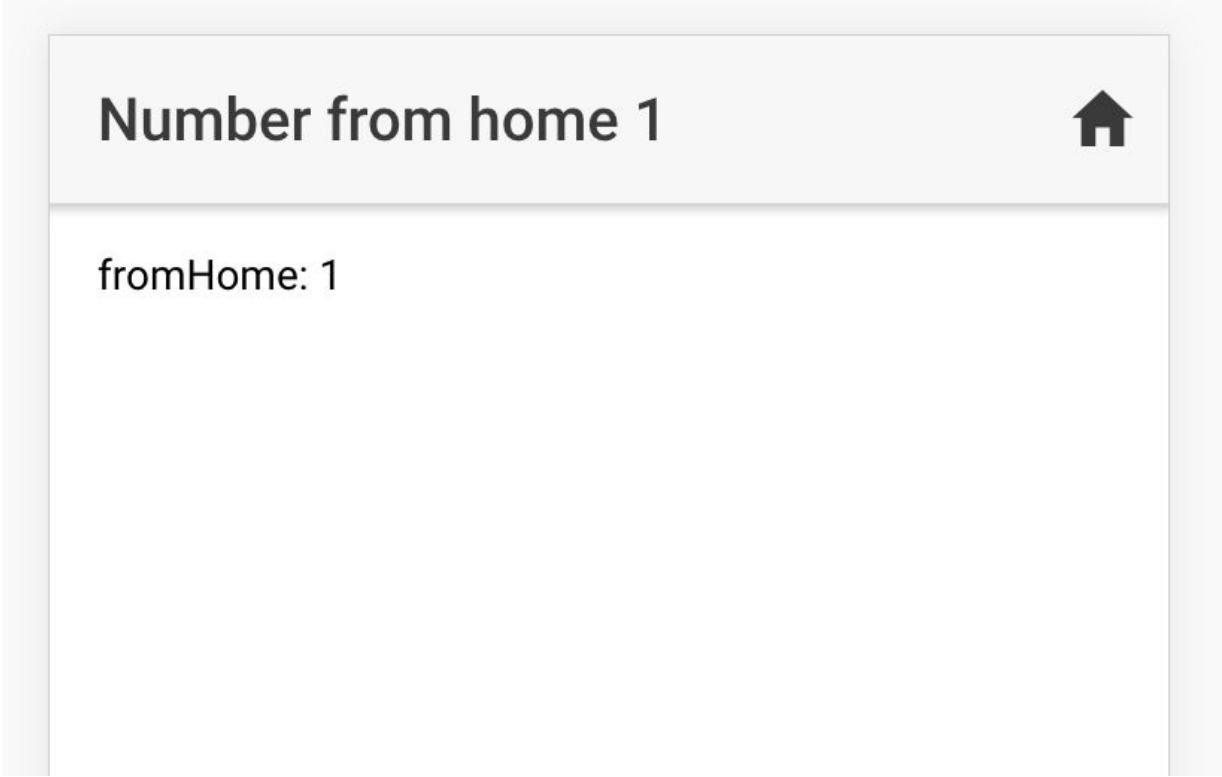
- Buttons can be added to a navbar by using ion-buttons

```
<ion-header>
  <ion-navbar>
    <ion-buttons end>
      <button ion-button menuToggle>
        <ion-icon name="menu"></ion-icon>
      </button>
    </ion-buttons>
    <ion-title>Home</ion-title>
  </ion-navbar>
</ion-header>
```

Assignment 15 - make home button

1. Open about.html
2. In ion-navbar create an ion-buttons component at the end
3. Create a icon-only ion-button with ion-icon “home” in it
4. Make the button clickable and attach function “toHome” on it

Result



Reset

git checkout assignment15

STOP SERVER

CTRL-C

Sidemenu Navigation

Most used navigation

Assignment 16 - start menu project

1. In terminal go 1 folder up
2. Type **`ionic start features-ionic3 sidemenu`**
3. When asked if you want to try Ionic 4 you answer No
4. When asked if you want to target native iOS and Android you answer Yes
5. When asked if you want to connect Ionic Pro you answer No
6. Type **cd features-ionic3**
7. Type **ionic serve**

Reset

- **ONLY NEEDED IF INSTALLATION FAILED!**
- In terminal go to your code folder (cd..)
- Type ‘ls’
- Type: **git clone** **https://github.com/OBout/features-ionic3.git**
- Type: cd features-ionic3

Now we are all on the same page!

Content

- Ion-content - main container for your app
- Padding - Adds padding to the content (for example edged phones)

Lists

- Lists display rows of information
- By default visually separated
- Rows in the list are usually <ion-item>
- Build for used ny *ngFor to display its data

```
<ion-list>
  <button ion-item *ngFor="let item of items" (click)="itemTapped($event, item)">
    <ion-icon [name]="item.icon" item-start></ion-icon>
    {{item.title}}
    <div class="item-note" item-end>
      {{item.note}}
    </div>
  </button>
</ion-list>
```

Premade list types

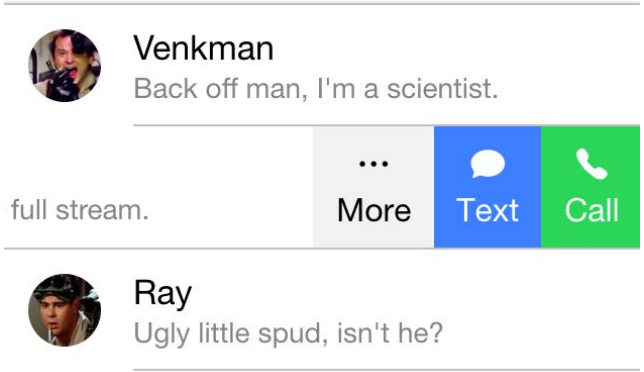
- Basic Lists
- Inset List
- List Dividers
- List Headers
- Icon List
- Avatar List
- Multi-line List
- Sliding List
- Thumbnail List

When lists are not used

- Multi line rows
- Dynamic data in rows

Sliding items

- A common interface pattern is to swipe from right to left on a row to reveal a set of buttons
- Use ion-item-sliding as a parent to ion-item
- Provide ion-item-options for the slide actions




```
<ion-list>
  <ion-item-sliding>
    <ion-item>
      <ion-avatar item-start>
        <img src="img/slimer.png">
      </ion-avatar>
      <h2>Slimer</h2>
    </ion-item>
    <ion-item-options side="right">
      <button ion-button color="primary">
        <ion-icon name="mail"></ion-icon>
        Email
      </button> </ion-item-options>
  </ion-item-sliding>
</ion-list>
```

NavController

Your app's Uber

Navigation between pages

- Navigation is handled through the ion-nav component

- Stack of pages, pushing on and popping off (moving forward or backwards)

- NavController is the main class used for navigation
  Can be injected in a page
  Used by side-menu and tab component

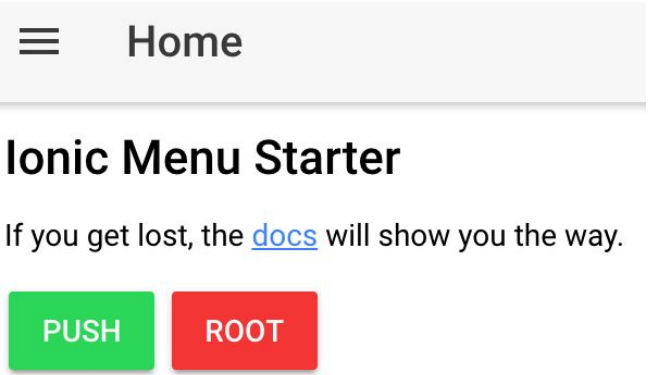
Assignment 16

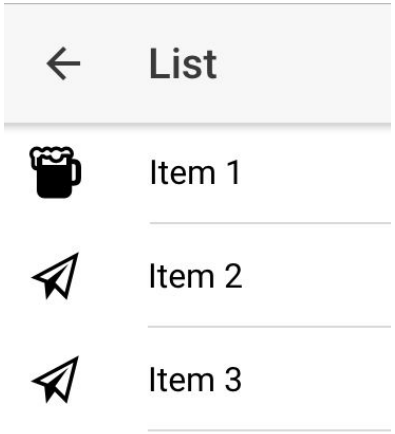
use docs https://ionicframework.com/docs/components/

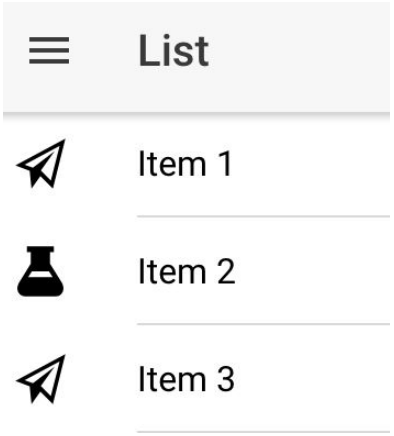
In your features project:

1. Open pages-home.ts
2. Change the “Toggle Menu” bottom IN THE CONTENT
3. Make it call a function in (click) called “navigatePushToList”
4. Let this function push a navigation event to the stack to the “ListPage” component (home.ts file)
5. Rename it “Push” and fix its color
6. Copy the button and rename it’s called function to “navigateRootToList”
7. Call it “Root” and color it “danger”
8. Let this function set the Root of the NavController to “ListPage”

Results


Home
Ionic Menu Starter
If you get lost, the docs will show you the way.
PUSH
ROOT


List
Item 1
Item 2
Item 3


List
Item 1
Item 2
Item 3

Reset

git checkout assignment16

WDWJD?

What did we just do?

Pages

- **ionic g page namepage**
- Page is an angular component
- Page has a selector

  <page-name></page-name>

  Page-name {

  background-color: pink;

  }

Pages

- Lifecycle - ionViewDidLoad & ionViewWillLeave
- Page is a component so it needs to be included in a module
- Enables lazy loading

Popups

- Javascript popups -
  alert(‘never do this’);
  confirm(‘promise me you never will’);

- Toast - Simple notification

- AlertController - Programmable buttons, Native UI

- Modal - Pops up a page!

Assignment 17a - easy start

In your features project:

1. Open pages-home.ts and pages-home.html
2. Create 3 new buttons
3. Make 1 button display a Toast “hello world”
4. Color this button “light” and call it Toast

Assignment 17b - medium

In your features project:

1. Open pages-home.ts and pages-home.html
2. Make 1 button use alertController to display “hello world”
3. Color it “dark” and call it “Alert”

Assignment 17c - hard section

In your features project:

1. Generate a page “modalpage”
2. Don’t forget to add it to “app.module.ts” as a declaration and as an entry component
3. Open pages-home.ts and pages-home.html
4. Make 1 button show a modal using ModalController showing “modalpage”.
5. Color it “default” and name it “Modal”

Assignment 17d - epic section

In your features project:

1. Open modalpage.ts and modalpage.html
2. Make 1 button calling a function to close the current modal.
3. Color it “danger” and name it “Close”

Restart development server!

After adjusting app.module **ALWAYS** restart your server!

- In terminal use keys: **CTRL-C**
- (re)type: **ionic serve**
- Close old browertab

Results

```
ionViewDidLoad ModalpagePage
undefined
Dismissed toast
ionViewDidLoad ModalpagePage
undefined
>
```

Home

Ionic Menu Starter

If you get lost, the docs will show you the way.

PUSH ROOT TOAST ALERT MODAL

Hello World

Hello

World

AWESOME

modalpage

CLOSE

Reset

git checkout assignment17

WDWJD?

What did we just do?

Forms and inputs

User interactions

Forms

- Forms are used to gather user input
- Input elements are provided by Ionic for Native Look ‘n Feel
- Input fields do not NEED to be in a form, but a form provides benefits.

```
import { FormsModule } from '@angular/forms';
@NgModule({
  imports: [FormsModule]
})
export class MyModule { }
```

Forms

- ngForm needs to be added to the form element
  This provides information about the state of the form
  Lets you access the form values
  Allows you to submit events

```
<form #myForm="ngForm">
  // inputs
</form>
```

Inputs

- <ion-input> is Ionic’s preferred text-type input element
- It prefers to be inside an <ion-item>
- text, password, email, number, search, tel, url
- <ion-textarea> works the same as ion-input but is text input and has a little different UI presentation

```
<ion-item>
  <ion-input type="text" placeholder="My Ionic Input" ([model])="model"></ion-input>
</ion-item>
```

Checkbox

- Checkbox is an input element that holds a boolean value (checked)
- Multiple checkboxes can be selected

```
<ion-item>
  <ion-label>Cool Story Bro</ion-label>
  <ion-checkbox></ion-checkbox>
</ion-item>
```

Radio

- Radio is an input element that holds a boolean value (checked)
- Only 1 radio inside a group can be selected

```
<ion-list radio-group [(ngModel)]="modal">
  <ion-item>
    <ion-label>Value 1</ion-label>
    <ion-radio value="value1"></ion-radio>
  </ion-item>
  <ion-item>
    <ion-label>Value 2</ion-label>
    <ion-radio value="value2"></ion-radio>
  </ion-item>
</ion-list>
```

Select

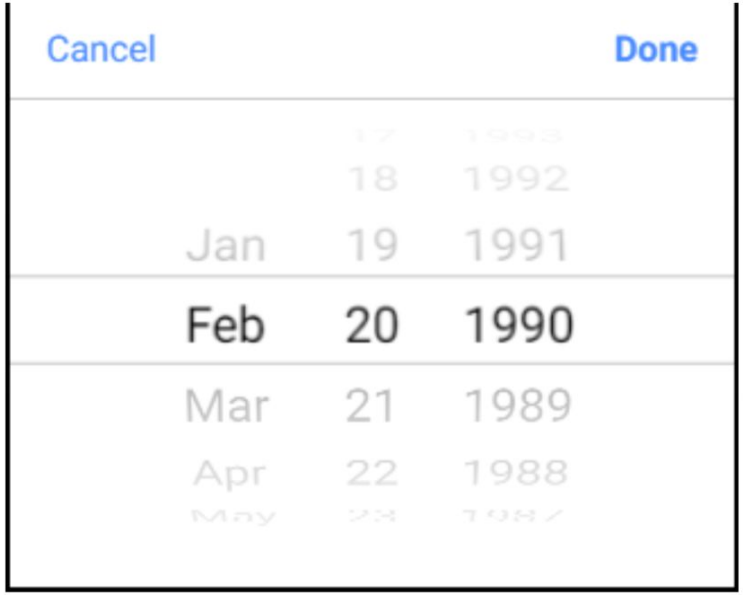
- A select is a list of items the user can choose from (sometimes called dropdown box)
- A select can be either multi-value or single value

```
<ion-item>
  <ion-label>Your Answers</ion-label>
  <ion-select [(ngModel)]="modal" multiple="false">
    <ion-option *ngFor="let answer of answers; let l = index">{{answer.DisplayValue}}</ion-option>
  </ion-select>
</ion-item>
```

DateTime

- Presents an interface which makes it easy for users to to to select dates and time

```
<ion-item>
  <ion-label>Date</ion-label>
  <ion-datetime displayFormat="MM/YYYY" pickerFormat="MMMM YYYY"></ion-datetime>
</ion-item>
```



Toggle

- Toggle is an “on-off” switch
- It is an alternative for checkbox


```
<ion-item>
  <ion-label>Toggle</ion-label>
  <ion-toggle></ion-toggle>
</ion-item>
```

Retrieve values

- Filled value can be retrieved on form submission

```
<form #myForm="ngForm" (ngSubmit)="add(myForm.value)">
  <label>Title:</label>
  <input type="text" name="title" ngModel>
</form>
```

- Or directly via two way binding

```
<label>Title:</label>
<input type="text" name="title" ([model])="myForm.value">
```

Search bar

- Ionic has a special search component ion-searchbar

```
<ion-searchbar
  [(ngModel)]="myInput"
  [showCancelButton]="shouldShowCancel"
  (ionInput)="onInput($event)"
  (ionCancel)="onCancel($event)">
</ion-searchbar>
```

- Works best in <ion-header>

Searchbar

Open this:

https://ionicframework.com/docs/api/components/searchbar/Searchbar/

(Or just Google “ionic search bar” and click the first hit)

Assignment 18a

- In modalpage.ts
- Add this line on top: **import { OnInit } from '@angular/core';**
- Implement OnInit on your class
- Create a public empty string array called **“countries”**
- Create a setCountries function setting the countries array with: **[“Belgium”, “Holland”, “Germany’, “France”]**
- Create an ngOnInit function calling the setCountries function
- Copy the “**filterItems(ev: any)**” function from the ionic docs example and fix it to work with your code

Assignment 18b - a little help

```
filterItems(ev: any) {
  this.setCountries();
  let val = ev.target.value;

  if (val && val.trim() !== '') {
    this.countries = this.countries.filter(function(item) {
      return item.toLowerCase().includes(val.toLowerCase());
    });
  }
}
```

Assignment 18c

- In modalpage.ts
- Create a function called “select”.
- Have it accept a value “country” with type “string”.
- Have the function call the view-controller to dismiss, but pass through the value “country”.

Assignment 18d

- In modalpage.html
- Create a <ion-toolbar> containing a <ion-searchbar> in the ion-header
- Add this function to it **(ionInput)="filterItems($event)"**
- Create a **<ion-list>**
- Make a hyperlinked ion-item (**<a ion-item>)** for every "**country**" in the "**countries**" array

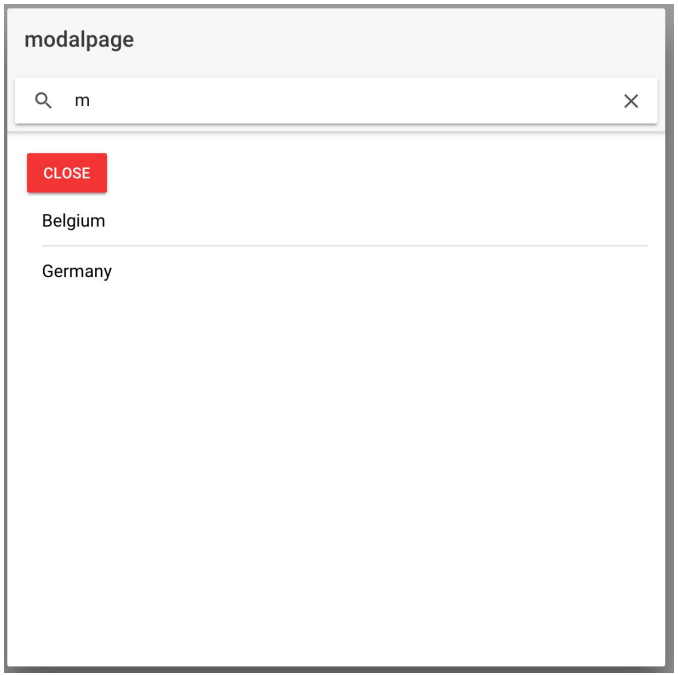
  ```
  <a ion-item>bla bla</a>
  ```

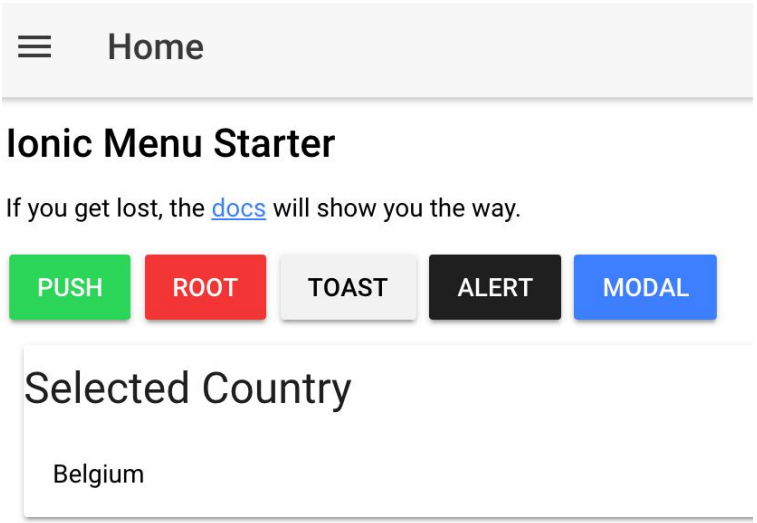
  HINT: USE *ngFor
- Make in every country a (click) event passing its value to the "select" function

Assignment 18e - now.. show me the money

- In home.html
- Make a public string “country”
- Make a **<ion-card>** only visible if country exists
  HINT: use *ngIf
- Have the card have a title “Selected Country”
- Have the content display the value of the public string “country”
- Adjust the modal opener function you’ve created earlier in **Assignment 17d**
- Have this function catch the **.onDismiss** function of the modal
- Capture the data from the modal
- Set the public string “**country**” to the data given in the modal

Results


modalpage
m
CLOSE
Belgium
Germany


Home
Ionic Menu Starter
If you get lost, the docs will show you the way.
PUSH
ROOT
TOAST
ALERT
MODAL
Selected Country
Belgium

Reset

git checkout assignment18

WDWJD?

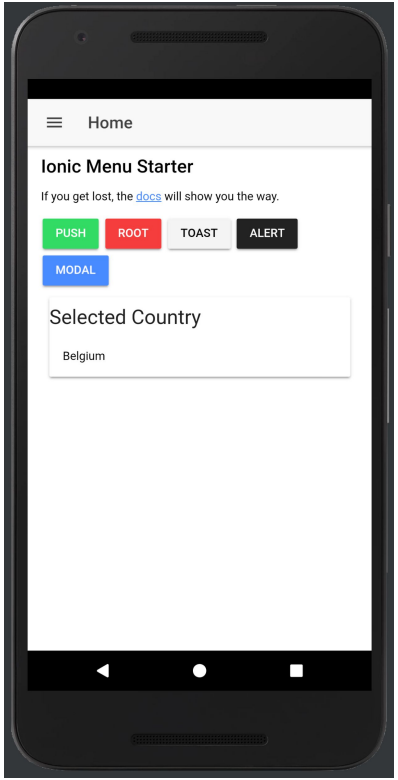
What did we just do?

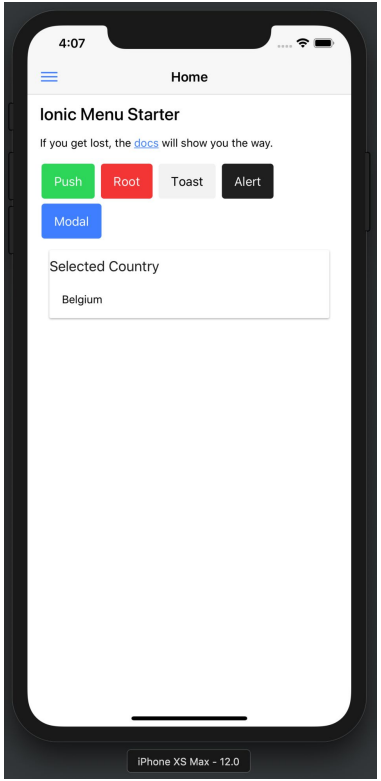
Assignment 19

- Run it on Android
  ionic cordova run android
- Run in on IOS (only if you are on a mac!)
  XCode 10 (latest) users use this:
  ionic cordova run ios -- --buildFlag="-UseModernBuildSystem=0"

  IOS see this:
  https://github.com/apache/cordova-ios/issues/407

Result


Home
Ionic Menu Starter
If you get lost, the docs will show you the way.
PUSH
ROOT
TOAST
ALERT
MODAL
Selected Country
Belgium


4:07
Home
Ionic Menu Starter
If you get lost, the docs will show you the way.
Push
Root
Toast
Alert
Modal
Selected Country
Belgium
iPhone XS Max - 12.0

Reset

git checkout assignment19

Ionic Storage

Remembers stuff

Ionic Storage

- Ionic Storage allows you to save key/value pairs and JSON data with little effort
- It uses a variety of different storage solutions, automatically selecting the proper one for the platform the app is running on

- When running in the web or as a Progressive Web App, Storage will attempt to use IndexedDB, WebSQL, and localstorage, in that order.
- For app it uses SQLite (needs a plugin)

- **https://ionicframework.com/docs/storage/**

Working with data

- Inject the Storage service and use the set / get methods
- Each method returns a Promise, use that to know when it has completed

Data example

```
import { Storage } from '@ionic/storage';

export class MyClass {

    constructor(private storage: Storage) {

    }

    getIt() {
        this.storage.get('code').then((val) => {
            console.log('The code is', val);
        });

    }

    setIt(val: any) {
        this.storage.set('code', val);
    }
}
```

Resources

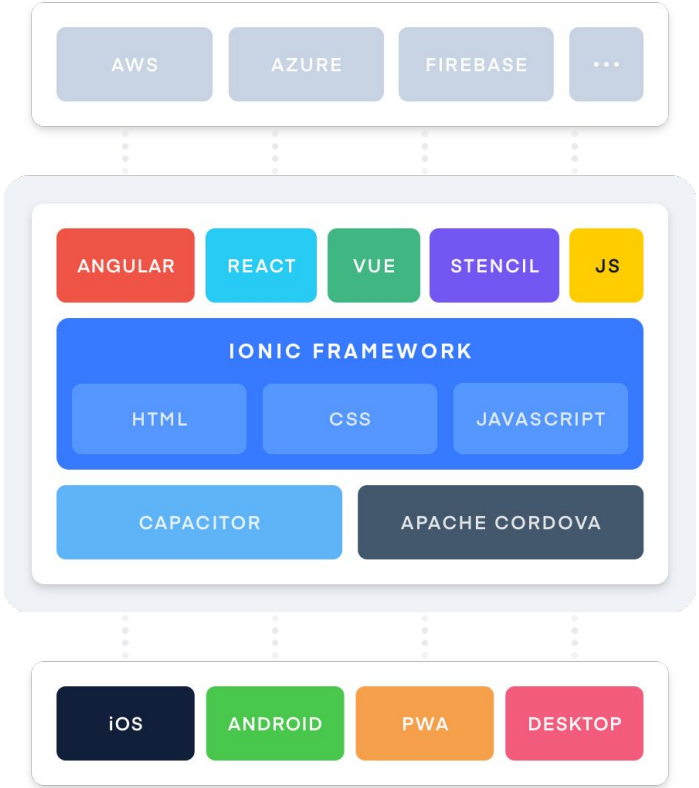
For my app

App Icons and Splash Screens

- **resources** folder in root

- Each platform needs it own icons and splash screens
- The Ionic CLI can generate all the necessary assets from images using:
  > **ionic resources**
- Icons will be generated from a file **icon.png**
  > needs to be **1024 x 1024 px** with no transparency
- Splash screen will be generated from a file **splash.png**
  > needs to be **2732 x 2732 px** with no transparency

Future -> Ionic 4+



Ionic 4+

- Angular navigation using routing file
  Deprecation of the NavController :(:(

- <ion-button>
- Lazy loading
- Webpack
- Angular-CLI
  (ng generate module)

- Tappable items

Questions

To ask

Thanks - merci - dank u - danke - 谢谢

https://ionicframework.com/what-is-ionic

Twitter: oscar.bout

https://oscoweb.com