



# Building Hybrid Apps with Ionic 2

@nicobytes

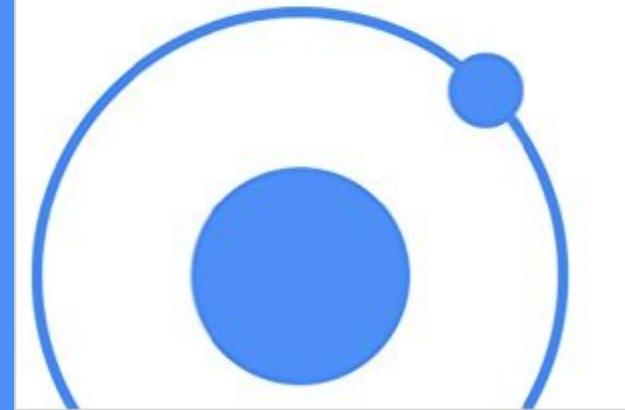
```
{  
  name: 'Nicolas Molina Monroy',  
  twitter: '@nicobytes',  
  website: 'https://nicobytes.com',  
  github: 'https://github.com/nicobytes'  
  jobs: [  
    'Hybrid App Developer',  
    'Front-End Developer'  
,  
  organizer: [  
    'http://www.meetup.com/es/Hybrid-Apps-Colombia',  
    'http://www.meetup.com/es/Django-Bogota',  
    'http://www.meetup.com/es/Meteor-Bogota'  
,  
  ]  
}
```



# ion-book

Prototipa tu Startup  
rápidamente.

CARLOS ROJAS  
NICOLAS MOLINA



[www.ion-book.com](http://www.ion-book.com)

# Todo sobre IONIC en español

Tips, noticias, tutoriales, demos, actualizaciones, buenas prácticas...

[Todo](#) [Demos](#) [Ionic](#) [Ionic2](#) [Noticias](#) [Tips](#)

Share



## Usando la cámara con Ionic 2

July 11, 2016 por [Nicolas Molina](#)

[ionic2](#) [plugins](#)

En los últimos artículos hemos visto una introducción a Ionic 2 y como crear un hola mundo con Ionic 2, ahora vamos hacer uso de Ionic native para usar la cámara del dispositivo y tomar una foto...

[Leer más...](#)



## Observables en Angular 2

July 07, 2016 por [Carlos Rojas](#)

[ionic2](#)

Los Observables son una de las grandes decisiones que creo hace que Angular 2 sea ...

[Leer más...](#)



## ¿Qué es Ionic Native?

July 05, 2016 por [Nicolas Molina](#)

[ionic2](#)

Ionic-native es todo un conjunto de envolturas en ES5/ES6/TypeScript para cualq...

[Leer más...](#)



## Ionic 2 Beta 10



## Ionic Platform

**ion-book**

Protótipos y Startup rápidamente.

[LIBRERÍAS](#) [CÓDIGO FUENTE](#)



Aprende Ionic1 & Ionic2 y construye increíbles aplicaciones mientras aprendes.

[Muy pronto](#)

**ION-BOOK**

Aprende con cursos más específicos con videos online.

[Ver cursos](#)

Siguenos En:





[www.sponzor.me](http://www.sponzor.me)

# Colombia



# Peru



# Bolivia

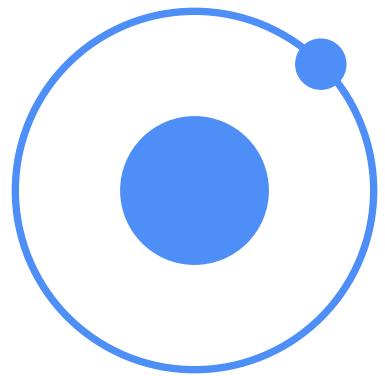


# Chile



*“Travel is the only thing you buy that makes  
you richer!”*

# Review



ionic

# Web Technologies You Already Know & Love

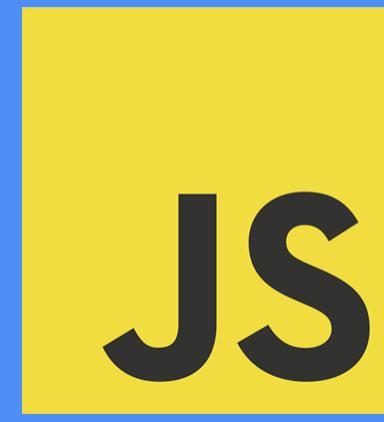
**HTML**



**CSS**



**JS**



(You'll feel right at home)

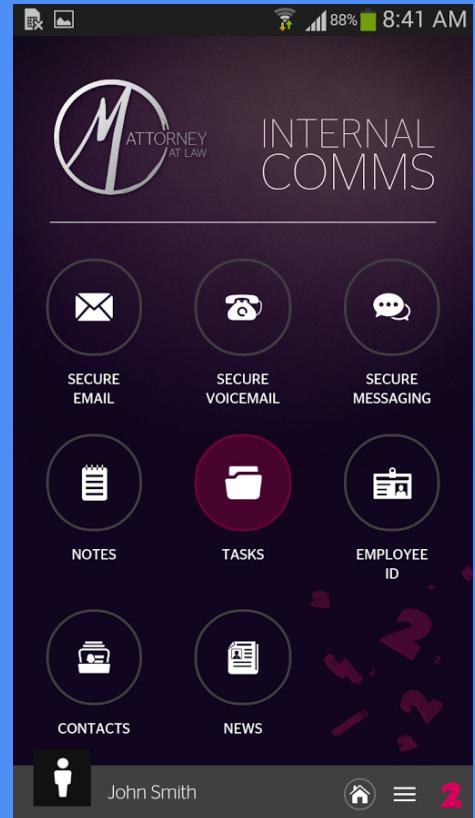
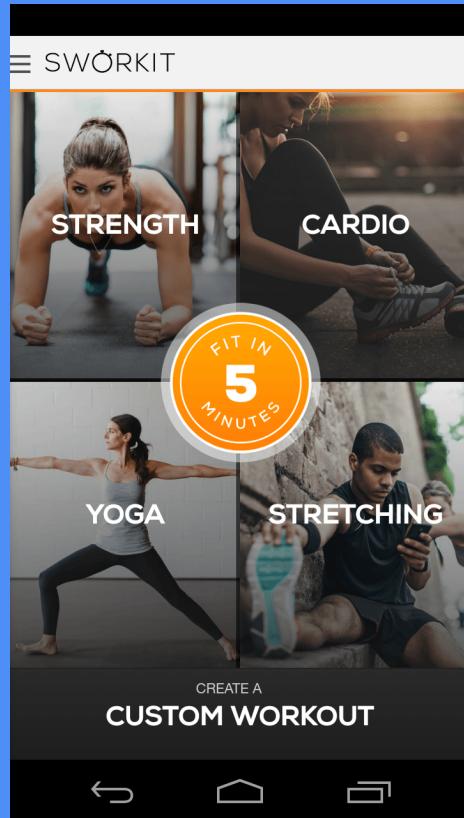
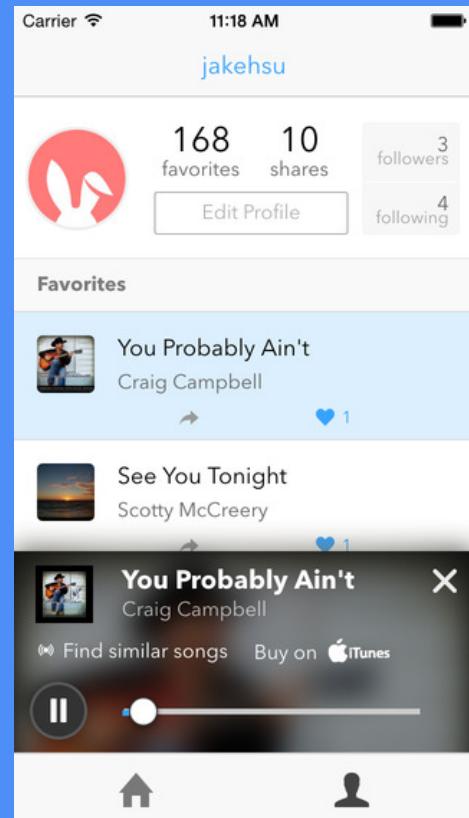
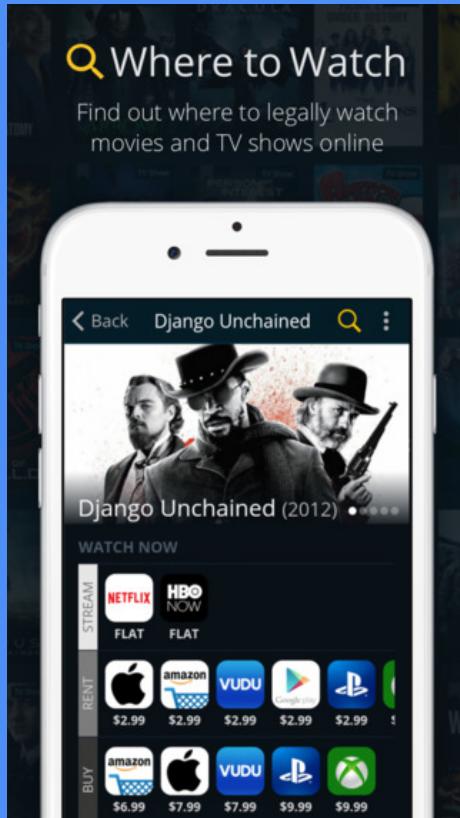
# IBM Mobile

<http://www.ibm.com/mobilefirst/>

The screenshot shows a web browser window with the title bar "IBM MobileFirst - Enterprise Mobility - United States - Chromium". The address bar displays the URL "www.ibm.com/mobilefirst/us/en/". The browser interface includes standard controls like back, forward, and search, along with a toolbar with icons for star, ABP, and other functions.

The main content area features the IBM MobileFirst logo at the top left. A navigation menu with links to "Enterprise mobility", "Downloads", "Case studies", "Solutions", "News & events", and "IBM.com" is positioned above a large headline. The headline reads "Take your business from mobile ready to mobile first" and includes a subtext "Rapidly deliver native and hybrid apps with IBM MobileFirst". To the right of the headline is a photograph of a woman with curly hair looking down at her smartphone. On the left side of the main content area, there is a callout box containing text and a link: "What makes a great app" and "Free Forrester report". A vertical sidebar on the far left contains the text "Contact IBM MobileFirst". At the bottom of the page, there is a blue footer bar with the text "+ Learn more about IBM MobileFirst™" and "Outthink competition – Everything is mobile in the cognitive era – Learn more".

# Apps



# Ionic V1 is great

But a lot's changed since its release

2 years is a long time in web tech!

# A lot's changed

Better/faster devices!

Fewer slow/bad devices!

Widely available web APIs!

Improved browser engines!



# V2 Goals

1. Simplicity
2. Native UX
3. Navigation
4. Creative Freedom
5. Future Proof

# Talk is cheap

Let's see some code

# 1. Simplicity

Using properties instead of classes.

# 2. Native UX

The different modes for Ionic 2.

# 3. Navigation

More robust and powerful.

# 5. Creative Freedom

Make anything possible.

# 6. Future Proof

Building Ionic with new standards.

# JavaScript has changed

Ionic 1 was built with ES5

ES6 brings many new features

NG2 is pushing for ES6

Different look, but all JS Classes

# TypeScript

ES6 + ES7 Decorators + Types

Ionic 2 and NG2 built using TS

Code completion in your editor

# Ionic 2

*The course*



Class 1  
Basics



Class 2  
Understanding Ionic



Class 3  
Decorators

---



Class 4  
Navigation



Class 5  
UI/UX



Class 6  
Forms

---



Class 7  
Data



Class 8  
Native



Class 9  
Build & Deploy



# Class 1: Basics

# Basics

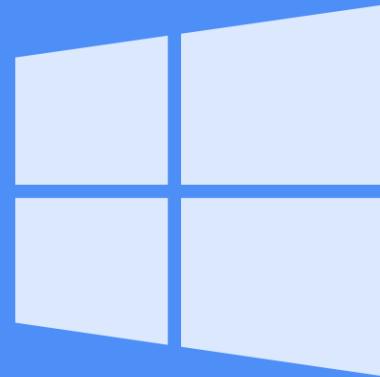
1. ¿What is HybridApp?
2. Showcases
3. New Concepts (ES6, Typescript, Angular)
4. Transpiling
5. Web Components.

¿What is  
HybridApp?

*“I want to build an app!”*

# More Platforms. More Problems.

- Proficiency in each platform required
- Entirely separate code bases
- Timely & expensive development
- Diminishing returns



Why are we still coding for multiple platforms?



# Hybrid Apps!



**HTML5** that acts like native

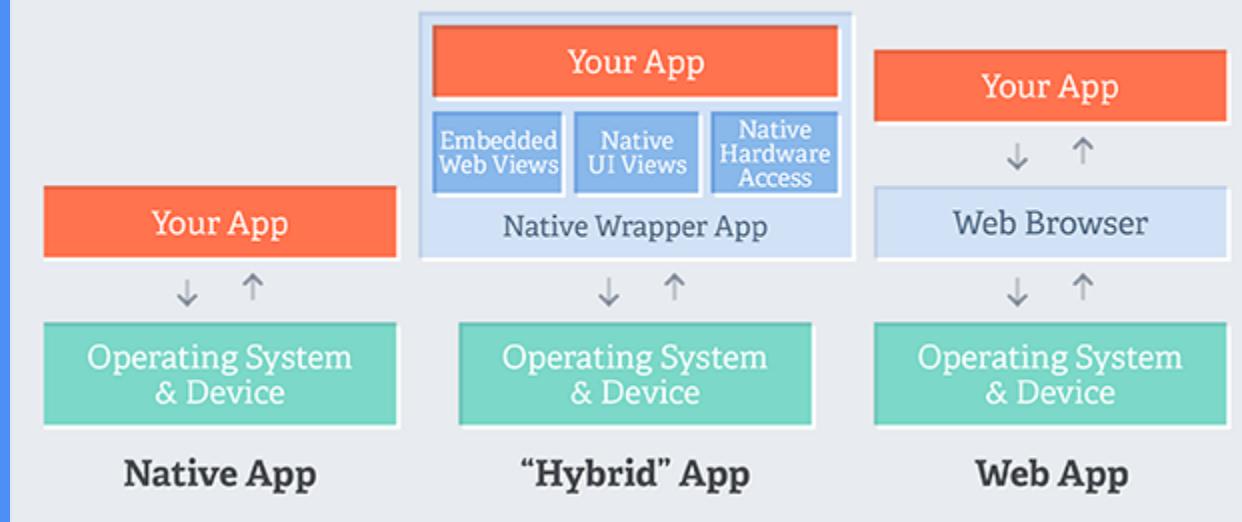
Web wrapped in **native** layer

Direct access to **native APIs**

Familiar **web dev** environment

A **single code base** (web platform!)

## Mobile App Technology Stacks



*“Oh No! The Zuck has spoken!”*

**TC** News TCTV Events

Social michael arrington Facebook Mark Zuckerberg Disrupt

# Mark Zuckerberg: Our Biggest Mistake Was Betting Too Much On HTML5

Posted Sep 11, 2012 by *Drew Olanoff (@drew)*

168 [Share](#) 1.5k [Share](#) 0 [Tweet](#) 2,501 [Next Story](#)

---



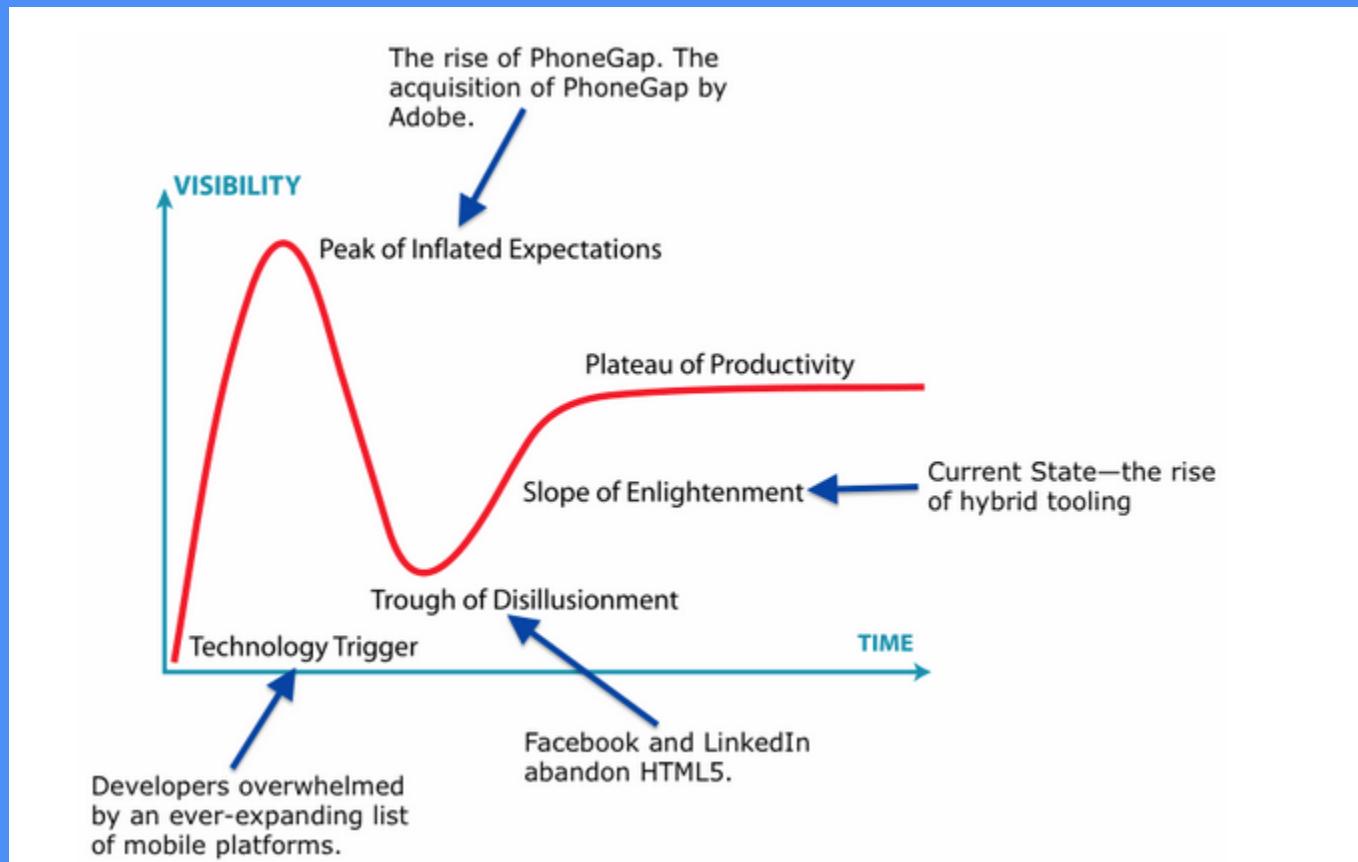
Today, Mark Zuckerberg revealed that Facebook's mobile strategy relied too much on HTML5, rather than native applications. Not only was this a big mistake with mobile, but Zuckerberg says that its biggest mistake period was the focus on HTML5. This is the first time that the Facebook CEO has openly admitted this, but things are looking good for the new iOS native app. According to Zuckerberg, people are consuming twice as many feed stories since the update to the new iOS app, which is great.



*“Hybrid apps are slow!”*

*“The Times They Are a-Changin’”*

# STATE



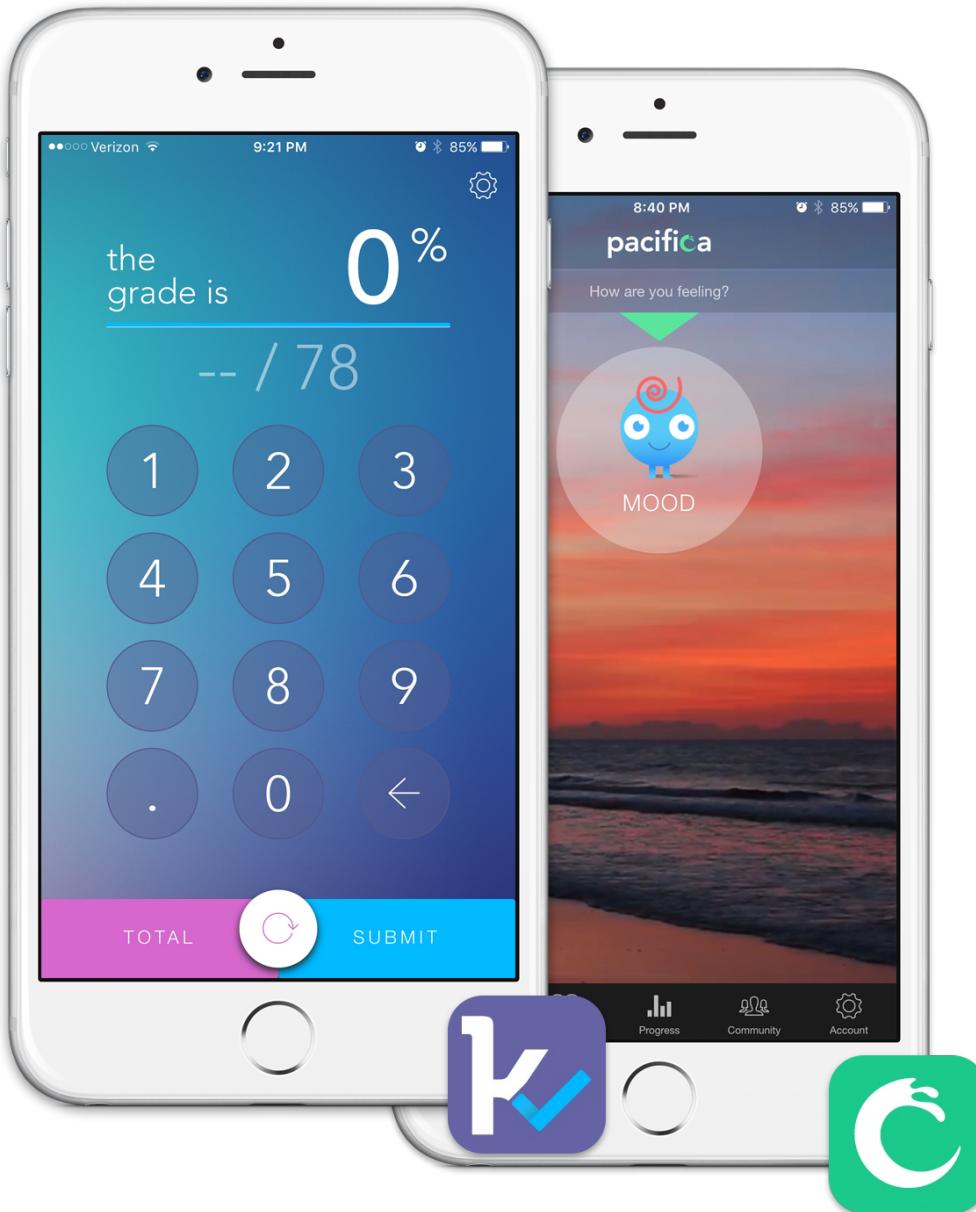


**“It's not 2007  
anymore”**

# **THE TIMES THEY ARE A-CHANGIN'**

Year	Device	Processor	RAM
2007	iPhone	400 MHz	128 MB
2010	iPhone 4	1 GHz	512 MB
2015	iPhone 6	1.4 GHz dual-core	1 GB

# Showcases



<http://showcase.ionicframework.com/>

<http://blog.ionic.io/tag/built-with/>

# New Concepts

# ECMAScript 6 (ES6)

# Classes

```
class Person {  
    name;  
    age;  
  
    constructor (name, age) {  
        this.name = name;  
        this.age = age;  
    }  
  
    getName () {  
        return this.name;  
    }  
}
```

# Modules

```
export function multiply (x, y) { return x * y }
export var url = 'http://api.domain.com';

import { multiply, url } from "./math"
console.log(multiply(2, 2))
```

# Promises

```
doLogin().then((rta) => {  
    console.log(rta);  
} );
```

```
a(function (resultsFromA) {
  b(resultsFromA, function (resultsFromB) {
    c(resultsFromB, function (resultsFromC) {
      d(resultsFromC, function (resultsFromD) {
        e(resultsFromD, function (resultsFromE) {
          f(resultsFromE, function (resultsFromF) {
            console.log(resultsFromF);
          })
        })
      })
    })
  });
});
```

```

function register()
{
    if (!empty($_POST)) {
        $msg = '';
        if ($_POST['user_name']) {
            if ($_POST['user_password_new']) {
                if ($_POST['user_password_repeat']) {
                    if (strlen($_POST['user_password_new']) > 5) {
                        if (strlen($_POST['user_name']) < 65 || strlen($_POST['user_name']) > 15) {
                            if (preg_match('/[a-zA-Z][a-zA-Z\d]{5,14}[a-zA-Z]$/i', $_POST['user_name'])) {
                                $user = read_user($_POST['user_name']);
                                if (!isset($user['user_name'])) {
                                    if (!$_POST['user_email']) {
                                        if (strlen($_POST['user_email']) < 65) {
                                            if (filter_var($_POST['user_email'], FILTER_VALIDATE_EMAIL)) {
                                                create_user();
                                                $_SESSION['msg'] = 'You are now registered so please login';
                                                header('Location: ' . $_SERVER['PHP_SELF']);
                                                exit();
                                            } else $msg = 'You must provide a valid email address';
                                        } else $msg = 'Email must be less than 65 characters';
                                    } else $msg = 'Email cannot be empty';
                                } else $msg = 'Username already exists';
                            } else $msg = 'Username must be only a-z, A-Z, 0-9';
                        } else $msg = 'Username must be between 2 and 64 characters';
                    } else $msg = 'Password must be at least 6 characters';
                } else $msg = 'Passwords do not match';
            } else $msg = 'Empty Password';
        } else $msg = 'Empty Username';
        $_SESSION['msg'] = $msg;
    }
    return register_form();
}

```



[icompile.eladkarako.com](http://icompile.eladkarako.com)

# Promises

```
doLogin()
  .then((rta) => {
    console.log(rta);
  })
  .then((rta) => {
    console.log(rta);
  })
  .then((rta) => {
    console.log(rta);
  })
  .catch((error) => {
    console.log(error);
  })
}
```

# Block Scoping

```
for (let i = 0; i < a.length; i++) {  
    let x = a[i];  
}
```

# Block Scoping

```
(function() {  
    'use strict';  
  
    //Code  
} ) () ;
```

# Fat Arrow Functions

```
doLogin((response) => {
  console.log(response);
}) ;
// 
doLogin(function(response) {
  console.log(response);
}) ;
```



# TypeScript

```
class Person {  
    private name: string;  
    private age: number;  
    private single: boolean;  
  
    constructor (name, age) {  
        this.name = name;  
        this.age = age;  
        this.single = false;  
    }  
  
    getName():string{  
        return this.name;  
    }  
}
```

# Transpiling

# Web Components

# Decorators

# Import & Export

# Dependency Injection



## Class 2: Understanding Ionic

# Understanding Ionic

1. Generate your first app
2. Syntax (Ionic 2 & Angular 2)
3. Anatomy
4. Ionic CLI

Generate your  
first app

```
npm install -g ionic@beta cordova
ionic info
ionic start appBlank blank --v2 --ts
ionic start appSide sidemenu --v2 --ts
ionic start appTabs tabs --v2 --ts
```

```
ionic platform add android  
ionic platform add ios
```

```
ionic serve --lab
```

# Syntax

# Data Binding

```
<input [value]="twitter">
<button (click)="doChange()">
<p> Hola {{twitter}} </p>
<input [value]="twitter" (input)="twitter = $event.target.value">
<input [(ngModel)]="twitter">
```

# Local Variable

```
<p #myVar></p>
<button (click)="myVar.innerHTML = 'test.'">
```

# Directives

- \*ngIf
- \*ngFor

# Anatomy

# Ionic Cli

# Basic Usage

```
ionic start myAwesomeApp --v2 --ts  
cd myAwesomeApp  
ionic info  
ionic serve
```

# Build

```
ionic build android  
ionic build ios  
ionic build android --release  
ionic build ios --release
```

# Emulate

```
ionic emulate android  
ionic emulate ios
```

# Platform

```
ionic platform add android  
ionic platform add ios  
ionic platform add wp
```

# Run

```
ionic run android  
ionic run ios
```

# State

```
ionic state save  
ionic state restore
```

# Resources

ionic resources

# Plugins (Native)

```
ionic plugin add cordova-plugin-XXXXXX  
//Examples  
ionic plugin add cordova-plugin-camera  
ionic plugin add cordova-plugin-googlemaps  
ionic plugin add phonegap-plugin-push
```

# Libs (npm)

```
npm install moment --save
```

# Typings

```
npm install -g typings
//Now
npm install underscore --save
typings install underscore --ambient --save
```

Ionic2 External Libraries/



## Class 3: Decorators

# Decorators

1. Decorators (@Component, @Directive, @Pipe and @Injectable)
2. Ionic Generator
3. Pages
4. Components
5. Directives
6. Pipes
7. Injectables

```
@Decorator ( {
    //meta data
} )
export class MyClass {
    //Your class
}
```

- @Component
- @Injectable
- @Pipe
- @Directive

# Generator

```
ionic g --list
```

# Page

ionic g page heroes

# It is a component

```
import { Component } from '@angular/core';
import { NavController } from 'ionic-angular';

@Component({
  templateUrl: 'build/pages/heroes/heroes.html',
})
export class HeroesPage {

  constructor(private nav: NavController) {

  }

}
```

# Hero Component

```
ionic g component my-hero
```

# Hero Component

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

@Component({
  selector: 'my-hero',
  templateUrl: 'build/components/my-hero/my-hero.html'
})
export class MyHero {

  text: string;
  @Input() hero: any;

  constructor() {
    this.text = 'Hello World';
  }
}
```

# Include

```
import { Component } from '@angular/core';
import { NavController } from 'ionic-angular';
import {MyHero} from '../../components/my-hero/my-hero';

@Component({
  templateUrl: 'build/pages/heroes/heroes.html',
  directives: [MyHero]
})
export class HeroesPage {
```

# Template

```
<my-hero *ngFor="let hero of heroes" [hero]="hero"></my-hero>
```

# ViewEncapsulation

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

@Component({
  selector: 'my-hero',
  templateUrl: 'build/components/my-hero/my-hero.html',
  encapsulation: ViewEncapsulation.Native
})
```

# Directives

```
ionic g directive my-highlight
```

```
import { Directive, ElementRef } from '@angular/core';
@Directive({
  selector: '[my-highlight]' // Attribute selector
})
export class MyHighlight {
  constructor(el: ElementRef) {
    el.nativeElement.style.backgroundColor = 'yellow';
  }
}
```

# Template

```
<h1 my-highlight>Hola</h1>
```

```
import { Component } from '@angular/core';
import { NavController } from 'ionic-angular';
import {MyHero} from '../../components/my-hero/my-hero';
import {MyHighlight} from '../../components/my-highlight/my-highlight';

@Component({
  templateUrl: 'build/pages/heroes/heroes.html',
  directives: [MyHero, MyHighlight]
})
export class HeroesPage {
```

# Pipe

```
ionic g pipe reserve
```

```
import { Injectable, Pipe } from '@angular/core';
@Pipe({
  name: 'reserve'
})
@Injectable()
export class Reserve {
  transform(value: string, args: any[]) {
    return value.split(' ').reverse().join(' ');
  }
}
```

# Inlcude

```
import { Component } from '@angular/core';
import { NavController } from 'ionic-angular';
import {MyHero} from '../../components/my-hero/my-hero';
import {MyHighlight} from '../../components/my-highlight/my-highlight';
import {Reserve} from '../../pipes/reserve';

@Component({
  templateUrl: 'build/pages/heroes/heroes.html',
  directives: [MyHero, MyHighlight],
  pipes: [Reserve]
})
export class HeroesPage {
```

# Template

```
<h1>{ { 'Hola' | reserve } }</h1>
```

# Provider

```
ionic g provider heroes
```

```
import { Injectable } from '@angular/core';
import { Http } from '@angular/http';
import 'rxjs/add/operator/map';
@Injectable()
export class Heroes {
  data: any;

  constructor(private http: Http) {
    this.data = null;
  }

  load() {
    //Code
  }
}
```

# Include

```
import { Component } from '@angular/core';
import { NavController } from 'ionic-angular';
import {MyHero} from '../../components/my-hero/my-hero';
import {MyHighlight} from '../../components/my-highlight/my-highlight';
import {Reserve} from '../../pipes/reserve';
import {HeroesService} from '../../providers/heroes/heroes';

@Component({
  templateUrl: 'build/pages/heroes/heroes.html',
  directives: [MyHero, MyHighlight],
  pipes: [Reserve],
  providers: [ HeroesService ]
})
export class HeroesPage {
```

# Inject

```
export class HeroesPage {  
  
    heroes: any[];  
  
    constructor(  
        private nav: NavController,  
        private heroesService: HeroesService  
    ) {  
        this.heroesService.load();  
        this.heroes = [  
            {  
                name: 'as'  
            },  
            {  
                name: 'as'  
            }  
        ];  
    }  
},
```

# Ionic Generator

# Global

```
import {Component} from '@angular/core';
import {Platform, ionicBootstrap} from 'ionic-angular';
import {StatusBar} from 'ionic-native';
import {TabsPage} from './pages/tabs/tabs';
import {HeroesService} from './providers/heroes/heroes';

@Component({
  template: '<ion-nav [root]="'rootPage'"></ion-nav>'
})
export class MyApp {

  private rootPage: any;

  constructor(
    private platform: Platform,
    private heroesService: HeroesService
  ) {
    this.heroesService.load();
  }
}
```

*Config Global*



# Class 4: Navigation

# Navigation

1. Push, Pop y setRoot.
2. Url parameters.
3. Tabs, Menus, Modal, Alerts.

# Push

```
// 1.  
import { NavController } from 'ionic-angular';  
// 2.  
constructor(  
    private nav: NavController,  
    private heroesService: HeroesService  
) {  
// 3.  
this.nav.push( HeroPage );
```

# Pop

```
// 1.  
import { NavController } from 'ionic-angular';  
// 2.  
constructor(  
    private nav: NavController,  
    private heroesService: HeroesService  
) {  
// 3.  
this.nav.pop();
```

# setRoot

```
this.nav.setRoot( page );
```

# Passing Data

```
this.nav.push(somethingPage, {  
  example1: data1,  
  example2: data2  
} );
```

# Passing Data

```
import {Component} from '@angular/core';
import {NavController, NavParams} from 'ionic-angular';
@Component({
  templateUrl: 'build/pages/second/second.html'
})
export class somethingPage {
  constructor(nav: NavController, navParams: NavParams) {
    this.navParams.get('example1');
  }
}
```

# Modals

Docs

# Tabs

## Docs

# Menus

## Menu

```
<ion-menu [content]="content">
  <ion-content>
    <ion-list>
      <button menuClose ion-item (click)="openTabsPage () ">Tabs</button>
      <button menuClose ion-item (click)="openHeroesPage () ">Heroes</button>
      <button menuClose ion-item (click)="close () " >Cerrar </button >
    </ion-list>
  </ion-content>
</ion-menu>

<ion-nav id="nav" #content [root]="rootPage"></ion-nav>
```

```
export class MenuPage {
    private rootPage: any;

    constructor(private nav: NavController) {
        this.rootPage = TabsPage;
    }

    openHeroesPage() {
        this.rootPage = HeroesPage;
    }

    close() {
        this.nav.setRoot( LoginPage );
    }
}
```

```
<button menuToggle>
  <ion-icon name="menu"></ion-icon>
</button>
```



## Class 5: UI/UX

# UI/UX

1. Ionic & Sass.
2. Material Desing.
3. IOS Stlye.
4. Window Phone Desing.
5. UI Components.

# Sass

## Documentation

# Files

- app.variables.scss
- app.core.scss
- app.md.scss
- app.ios.scss
- app.wp.scss

# Variables

app/theme/app.variables.scss

```
$colors: (  
  primary:      #387ef5,  
  secondary:    #32db64,  
  danger:       #f53d3d,  
  light:        #f4f4f4,  
  dark:         #222,  
  favorite:     #69BB7B  
) ;
```

# Custom Colors

```
$colors: (  
  // ...  
  twitter: #55acee  
)
```

# Apply

```
<button twitter>Twitter</button>
```

# Custom Colors

```
$colors: (  
  // ...  
  twitter: (  
    base: #55acee,  
    contrast: #ffffff  
  )  
)
```

# In scss

```
.home {  
  p{  
    background: color($colors, twitter, base);  
  }  
}
```

# Sass Variables

```
$my-padding: 10px;
```

# Apply

```
.home {  
  p{  
    background: color($colors, twitter, base);  
    padding: $my-padding;  
  }  
}
```

# Utility Attributes

Utilities

# Platform Specific Styles

## Docs

# Apply

```
.md button {  
    background: red;  
}  
.ios button {  
    background: blue;  
}  
.wp button {  
    background: green;  
}
```

# In html

```
<button [class.myclass]="true">Twitter</button>
<button [attr.no-lines]="true">Twitter</button>
```

# Overriding Ionic Sass Variables

Docs

```
// App iOS Variables
// -----
// iOS only Sass variables can go here
$button-ios-border-radius: 20px;
```

# Include yours scss

## app.core.scss

```
@import "../pages/home/home";  
@import "../component/my-avatar/my-avatar";
```

# Include yours scss for OS

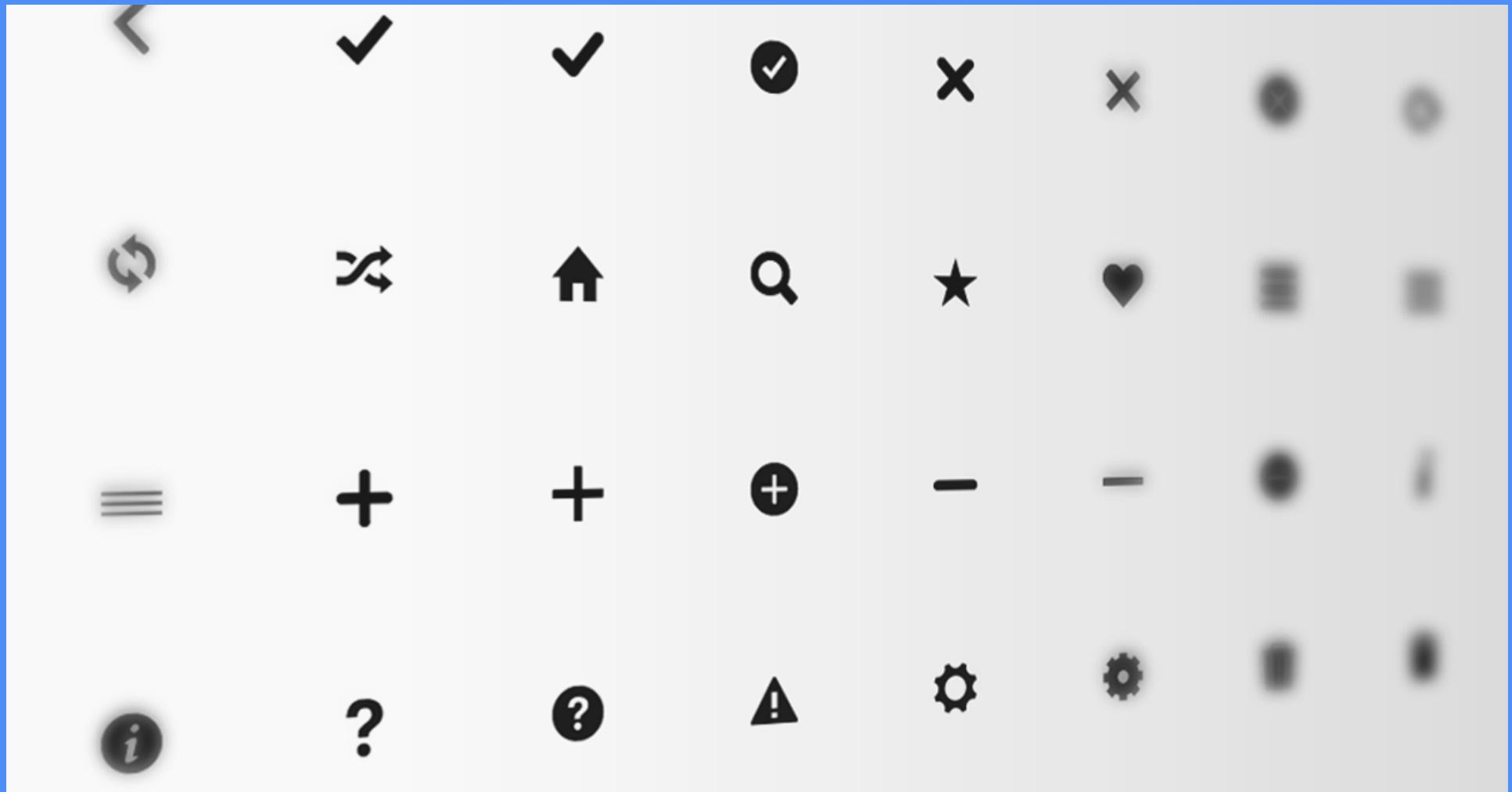
## app.----.scss

```
@import "../pages/home/ios/home";
```

# Overwrite

```
button[primary] {  
    background: black;  
}
```

# Icons



Icons v1

Icons v2

# Custom icons

# Inlcude

```
<link href="build/font-awesome/css/font-awesome.min.css"
```

Icomoon

# UI Components (without JS)

# Badges

# Buttons

# Cards

# Checkbox

# Datetime

# Gestures

# Grid

# List

# Slides

# UI Components (with JS)

# Alerts

Loading



# Class 6: Forms

# Forms

1. Binding with ngModel.
2. FormBuilder.
3. Validations.
4. Custom validations.

# ngModel

```
<form (ngSubmit)="saveData()">
  <ion-input type="text" required name="username" [(ngModel)]="model.username"></ion-inp
  <ion-input type="text" required name="name" [(ngModel)]="model.name"></ion-inp
  <button primary block type="submit">Save</button>
</form>
```

# Control

```
this.username = new Control(  
    'Default value',  
    Validators.required  
    /*Async*/  
) ;
```

# Control

```
<input type="text"  
ngControl="username"  
[ (ngModel) ]="model.username"  
name="username" />
```

# FormBuilder & ControlGroup

```
myForm: ControlGroup;  
//  
constructor(  
    private formBuilder: FormBuilder  
) {
```

# Form

```
this.username = new Control(' ', Validators.required);
this.name = new Control(' ', Validators.required);
this.myForm = this.formBuilder.group({
  username: this.username,
  name: this.name,
})
```

# Form

```
<form (ngSubmit)="saveData()" [ngFormModel]="myForm" novalidate>
  <ion-input type="text"
    name="username"
    [(ngModel)]="model.username"
    ngControl="username"></ion-input>
  <ion-input type="text"
    name="name"
    [(ngModel)]="model.name" ngControl="name"></ion-input>
  <button primary block type="submit">Save</button>
</form>
```

# Show errors

```
<div *ngIf="username.dirty && !username.valid">
  <p *ngIf="username.errors.required">
    Este campo es requerido
  </p>
</div>
```

# Validations

- Validators.required
- Validators.minLength(4)
- Validators.maxLength(8)
- Validators.pattern('[a-zA-Z ]\*)')

# Validations compose

```
Validators.compose([  
    Validators.required,  
    Validators.minLength(4),  
    Validators.maxLength(8),  
    Validators.pattern(' [a-zA-Z ]* ')  
])
```

# Control

```
this.username = new Control(  
    '',  
    Validators.compose([  
        Validators.required,  
        Validators.minLength(4),  
        Validators.maxLength(8),  
        Validators.pattern(' [a-zA-Z ]* ')  
    ])  
) ;
```

# Form

```
this.myForm = this.formBuilder.group({  
  username: ['', Validators.compose([....])],  
  name: ['', Validators.compose([....])],  
})
```

# Show errors

```
<div *ngIf="myForm.controls.username.dirty && !myForm.controls.username.valid">
  <p *ngIf="myForm.controls.username.errors.required">
    Este campo es requerido
  </p>
</div>
```

# saveDate

```
<button primary block type="submit" [disabled]="!myForm.valid" >Save</button>
//Or
(ngSubmit)="myForm.valid && saveData()"
```

# Customs Validations

```
import {Control} from '@angular/common';

export class AgeValidator{
  static isOld(control: Control) {
    let value = control.value;
    if(value > 18) {
      return {
        'isOld': true
      }
    }
    return null;
  }
}
```

# Include

```
import { AgeValidator } from '../../validators/age';
//Control
age: ['', AgeValidator.isReal]
```

# Include

```
import { AgeValidator } from '../../validators/age';
//Control
age: ['', AgeValidator.isReal]
```



## Class 7: Data

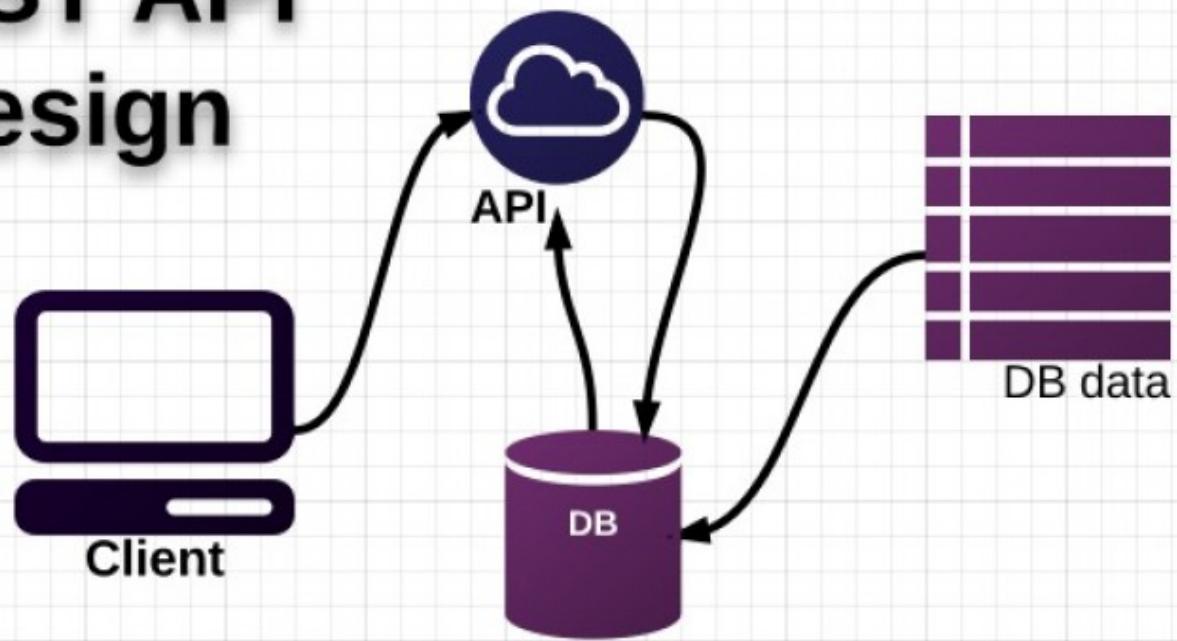
# Data

1. REST API.
2. LocaStorage.
3. SQLite.
4. Firebase.

# REST API

# REST API Design

<b>GET</b>	/tasks - display all tasks
<b>POST</b>	/tasks - create a new task
<b>GET</b>	/tasks/{id} - display a task by ID
<b>PUT</b>	/tasks/{id} - update a task by ID
<b>DELETE</b>	/tasks/{id} - delete a task by ID



GET

<https://randomuser.me/>

```
return new Promise(resolve => {
  this.http.get('https://randomuser.me/api/?results=25')
    .map(res => res.json())
    .subscribe(data => {
      this.data = data.results;
      resolve(this.data);
    });
});
```

# JSONPlaceholder

<https://jsonplaceholder.typicode.com/>

- /todos (GET)
- /todos (POST)
- /todos/1 (GET)
- /todos/1 (PUT)
- /todos/1 (DELETE)

# getAllTodos

```
getAllTodos ()  {
  return new Promise((resolve, reject) => {
    this.http.get(` ${this.path}/todos?_expand=user`)
      .map(res => res.json())
      .subscribe(data => {
        resolve(data);
      }, error =>{
        reject(error);
      })
  ) ;
}
```

# getTodo

```
getTodo(id: number) {
  return new Promise((resolve, reject) => {
    this.http.get(`.${this.path}/todos/${id}`)
      .map(res => res.json())
      .subscribe(data => {
        resolve(data);
      }, error =>{
        reject(error);
      })
  ) ;
}
```

# createTodo

```
createTodo (data: any) {
  return new Promise((resolve, reject) => {
    this.http.post(` ${this.path}/todos`, data)
      .map(res => res.json())
      .subscribe(data => {
        resolve(data);
      }, error =>{
        reject(error);
      })
  ) ;
}
```

# editTodo

```
editTodo(data: any) {  
    return new Promise((resolve, reject) => {  
        this.http.post(` ${this.path}/todos`, data)  
            .map(res => res.json())  
            .subscribe(data => {  
                resolve(data);  
            }, error => {  
                reject(error);  
            } )  
    } );  
}
```

# deleteTodo

```
deleteTodo (data: any) {
  return new Promise((resolve, reject) => {
    this.http.post(` ${this.path}/todos`, data)
      .map(res => res.json())
      .subscribe(data => {
        resolve(data);
      }, error =>{
        reject(error);
      })
  ) ;
}
```

# Headers

```
var headers = new Headers();
headers.append('Authorization', 'Basic -----');
this.http.post('http://api.domain.com/users', data, {
  headers: headers
})
```

# Proxies

```
"proxies": [
    {
        "path": "/v1",
        "proxyUrl": "https://api.instagram.com/v1"
    }
]
```

# LocalStorage

```
import { Storage, LocalStorage } from 'ionic-angular';
@Injectable()
export class TodosService {

  todos: Storage;

  constructor() {
    this.todos = new Storage(LocalStorage);
  }

  saveTodos( todos ) {
    this.todos.set('todos', JSON.stringify(todos));
  }

  getAllTodos() {
    return this.todos.get('todos');
  }
}
```

Limit

5MB

# SQLite

# WEB SQLite / SQLite Native

```
import { Storage, SqlStorage } from 'ionic-angular';

@Injectable()
export class TodosService {

  todos: Storage;

  constructor() {
    this.todos = new Storage( SqlStorage, {name: 'dbname'} );
  }
}
```

```
createTable() {
  let sql = "CREATE TABLE IF NOT EXISTS todos(id INTEGER PRIMARY KEY AUTOINCREMENT,
  title TEXT, description TEXT, created_at DATETIME DEFAULT CURRENT_TIMESTAMP);
  this.todos.query( sql )
  .then(data => {
    console.log( data );
  })
  .catch(error => {
    console.log( error );
  })
}
```

```
createTodo(data: any) {
  let sql = `INSERT INTO todos(title, done) values("${data.title}", "${data.done}")
this.todos.query( sql )
.then(data => {
  console.log( data );
} )
.catch(error => {
  console.log( error );
} )
}
```

```
getAllTodos():Promise<any>{
  let sql = `SELECT * from todos`;
  return this.todos.query( sql )
    .then(response => {
      let data = [];
      for (let index = 0; index < response.res.rows.length; index++) {
        data.push({
          id: response.res.rows[index].id,
          title: response.res.rows[index].title,
          completed: response.res.rows[index].done == "true" ? true : false
        });
      }
      return Promise.resolve( data );
    })
    .catch(error => {
      return Promise.reject(error)
    })
}
```

# limits

<https://www.sqlite.org/limits.html>

# Firebase

# Create Project

# Install angularfire2

```
npm install angularfire2 --save
npm install firebase --save
```

# Install typings

```
npm install typings -g  
typings install --save firebase
```

# Add typing typings/index.d.ts

```
//> <reference path="../../node_modules/angularfire2/firebase3.d.ts" /
```

# Config App

```
import {FIREBASE_PROVIDERS, defaultFirebase} from 'angularfire2';

ionicBootstrap(MyApp, [
  FIREBASE_PROVIDERS,
  defaultFirebase({
    apiKey: "-----",
    authDomain: "-----",
    databaseURL: "-----",
    storageBucket: "-----",
  }),
]) ;
```

# Rules

```
{  
  "rules": {  
    ".read": true,  
    ".write": true  
  }  
}
```

# Service

```
import { Injectable } from '@angular/core';
import { AngularFireListObservable, AngularFire } from 'angularfire2'

@Injectable()
export class TodosService {

  todos: AngularFireListObservable<any>

  constructor(
    private af: AngularFire
  ) {
    this.todos = this.af.database.list('/todos');
  }
}
```

# GetAllTasks

```
getAllTasks () {  
    return this.todos;  
}
```

# GetAllTodos

```
getAllTodos () {  
    return this.todos;  
}
```

# createTodo

```
createTodo(todo: any) {  
    return this.todos.push( todo );  
}
```

# updateTodo

```
updateTodo(task: any) {
  return this.todos.update( task.$key, {
    title: task.title,
    completed: task.completed
  ) ;
}
```

# updateTodo

```
updateTodo(task: any) {
  return this.todos.update( task.$key, {
    title: task.title,
    completed: task.completed
  ) ;
}
```

# removeTodo

```
removeTodo(task: any) {  
    return this.todos.remove( task.$key ) ;  
}
```



## Class 8: Native

# Native

1. Cordova / Phonegap.
2. Emulations.
3. Ionic Native.
4. Plugins (Camera, Geolocation, Vibration etc).

# Cordova / Phonegap

Genymotion == Android

Xcode == IOS

VS == WP

# GenyMotion

Check your sdk

android

ngCordova

Ionic native

# Plugins

# Install

```
ionic plugin add xxxx-xxx--xxxx --save
```

# Adb (Android)

```
adb devices  
adb start-server  
adb kill-server
```

# Run

```
ionic run android  
ionic run ios
```



## Class 9: Build

# Build

1. Preparate.
2. Signing
3. Ionic Platform.
4. Phonegap Build y ionic Package.
5. PlayStore y AppStore.

# Enable Production Mode

```
ionicBootstrap(MyApp, [ Providers ], {  
  prodMode: true  
} );
```

# Generate Icons and Splash Screens

ionic resources

# Set the Bundle ID and App Name

```
<?xml version='1.0' encoding='utf-8'?>
<widget id="io.ionic.starter" version="0.0.1" xmlns="http://www.w3.org/ns/widgets" xmlns:cdv="http://cordova.apache.org/ns/1.0">
  <name>V2 Test</name>
  <description>An Ionic Framework and Cordova project.</description>
  <author email="hi@ionicframework" href="http://ionicframework.com/">Ionic Framework Team</author>
```

# Global Preferences

# Minify Assets

<https://tinyjpg.com/>

# Signing Android Applications

# Generate key

```
keytool -genkey -v -keystore my-release-key.
```

# Release

```
Ionic build android --release
```

# To sign the unsigned APK

```
jarsigner -verbose -sigalg SHA1withRSA -dige
```

Run the zip align tool to  
optimize the APK.

```
zipalign -v 4 platforms/android/build/output
```

Google Play

# Signing iOS Applications

Apple Developer Account  
Required

# Generate .p12

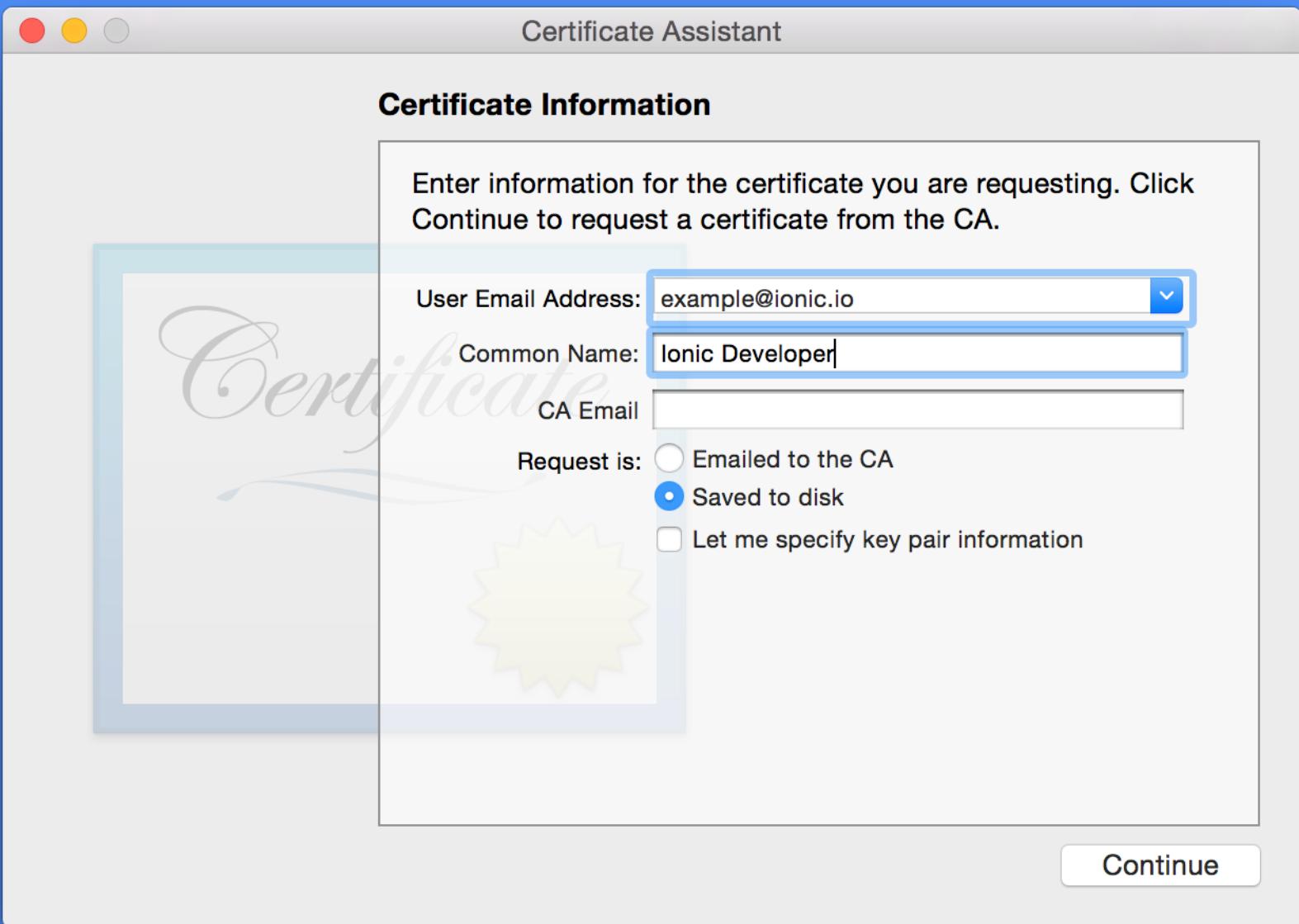
::before Generate .certSigningRequest

::before Generate mykey.key

# Signing iOS Applications with MAC

Tutorial

Keychain Access > Certificate Assistant > Request a  
Certificate From a Certificate Authority



# Signing iOS Applications without Mac

openssl

# Signing iOS Applications without Mac

```
openssl genrsa -out mykey.key 2048
openssl req -new -key mykey.key -out myCSR.csr
```

Done! .certSigningRequest

Upload => .certSigningRequest  
Download <= .cer

developer

# Generate .p12 with MAC

```
Open Keychain Access
```

```
Open .cer > Export .p12
```

# Generate .p12 without MAC

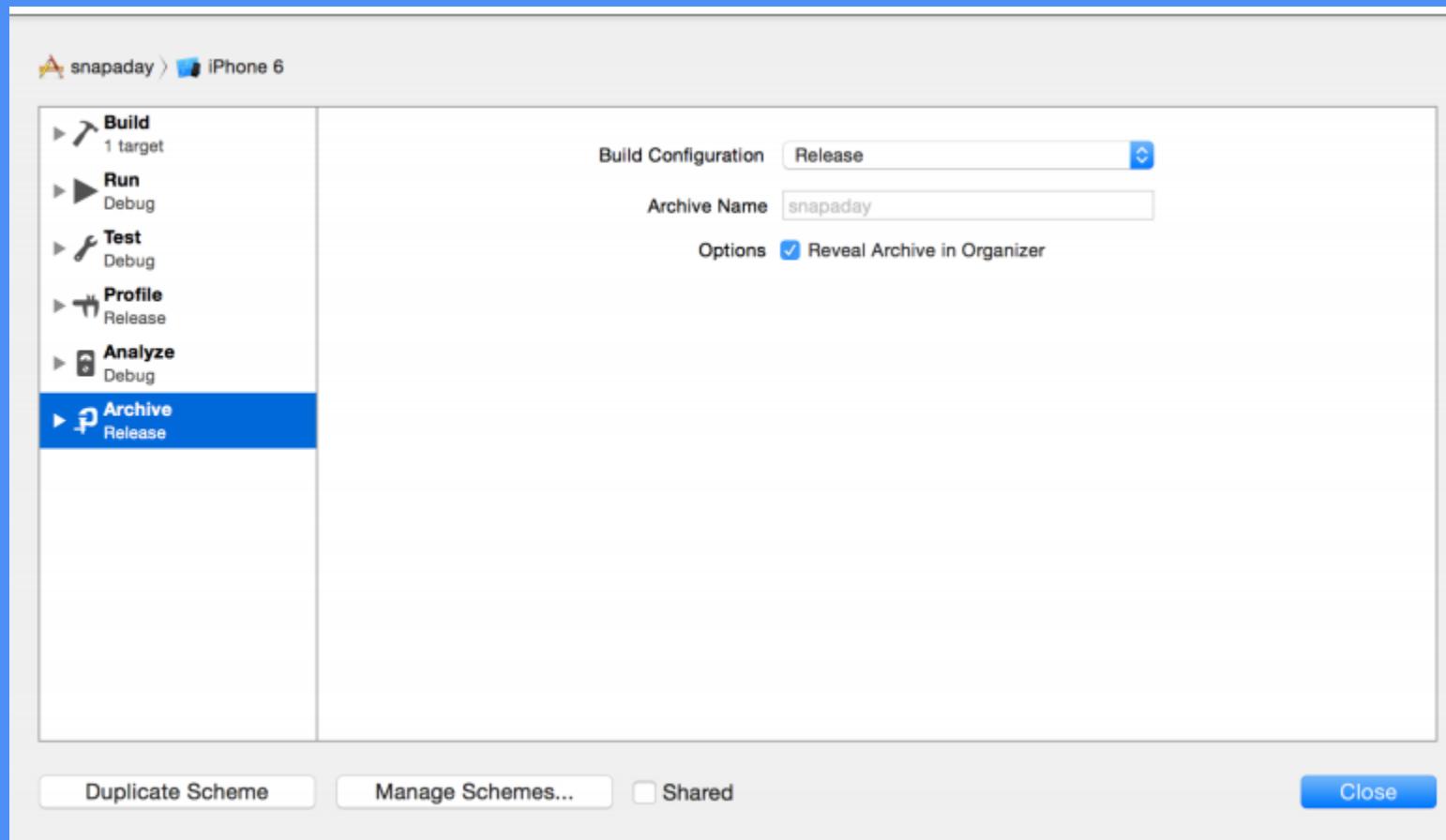
```
openssl x509 -in ios_development.cer -inform der -out ios_development.pem  
openssl pkcs12 -export -inkey mykey.key -in ios_development.pem -out ios_development.p12
```

# Submitting an app using XCode

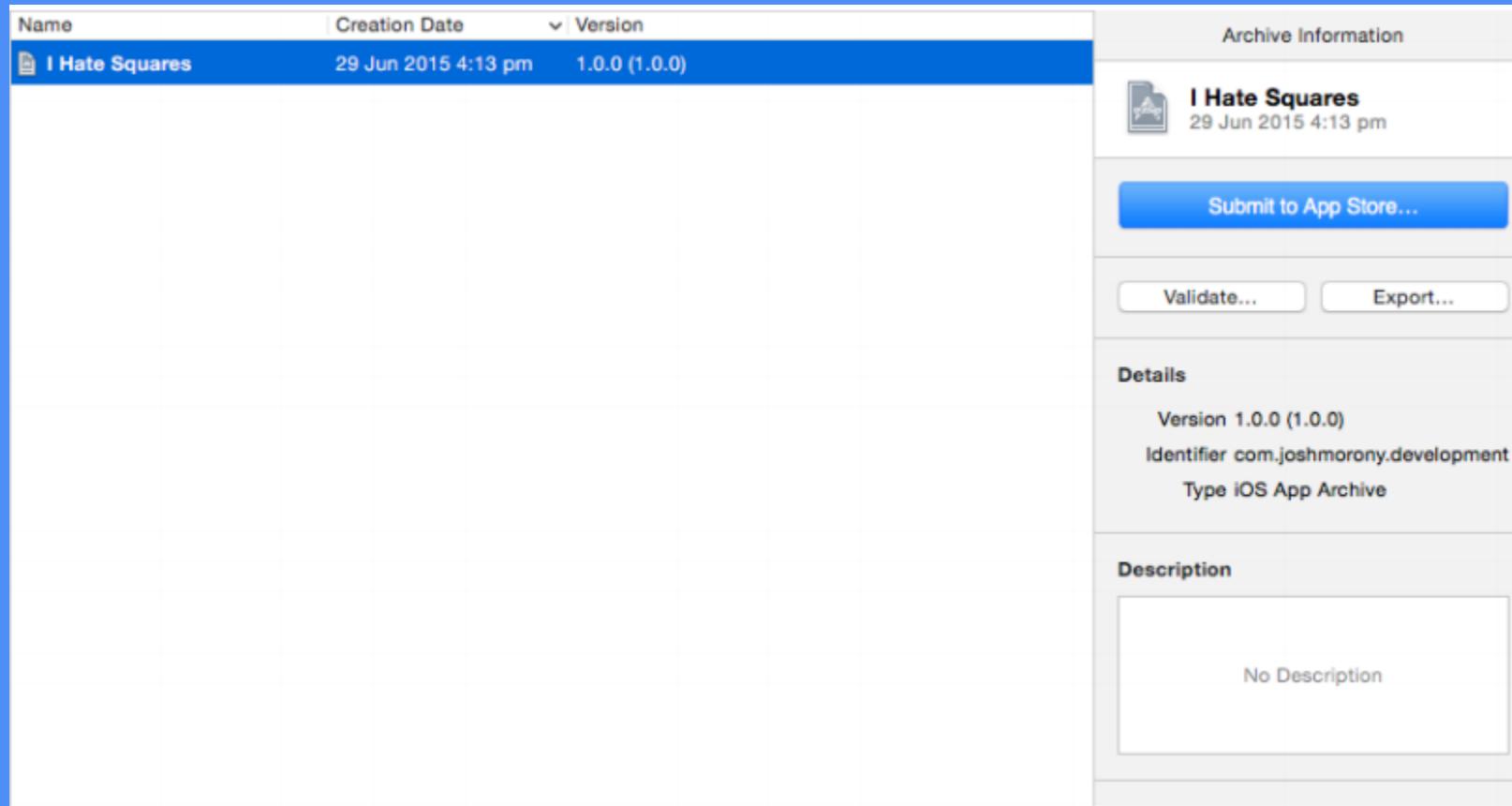
```
ionic build ios
```

# Open .xcodeproj

## platforms/ios/snapaday.xcodeproj



# Open .xcarchive



# Validate

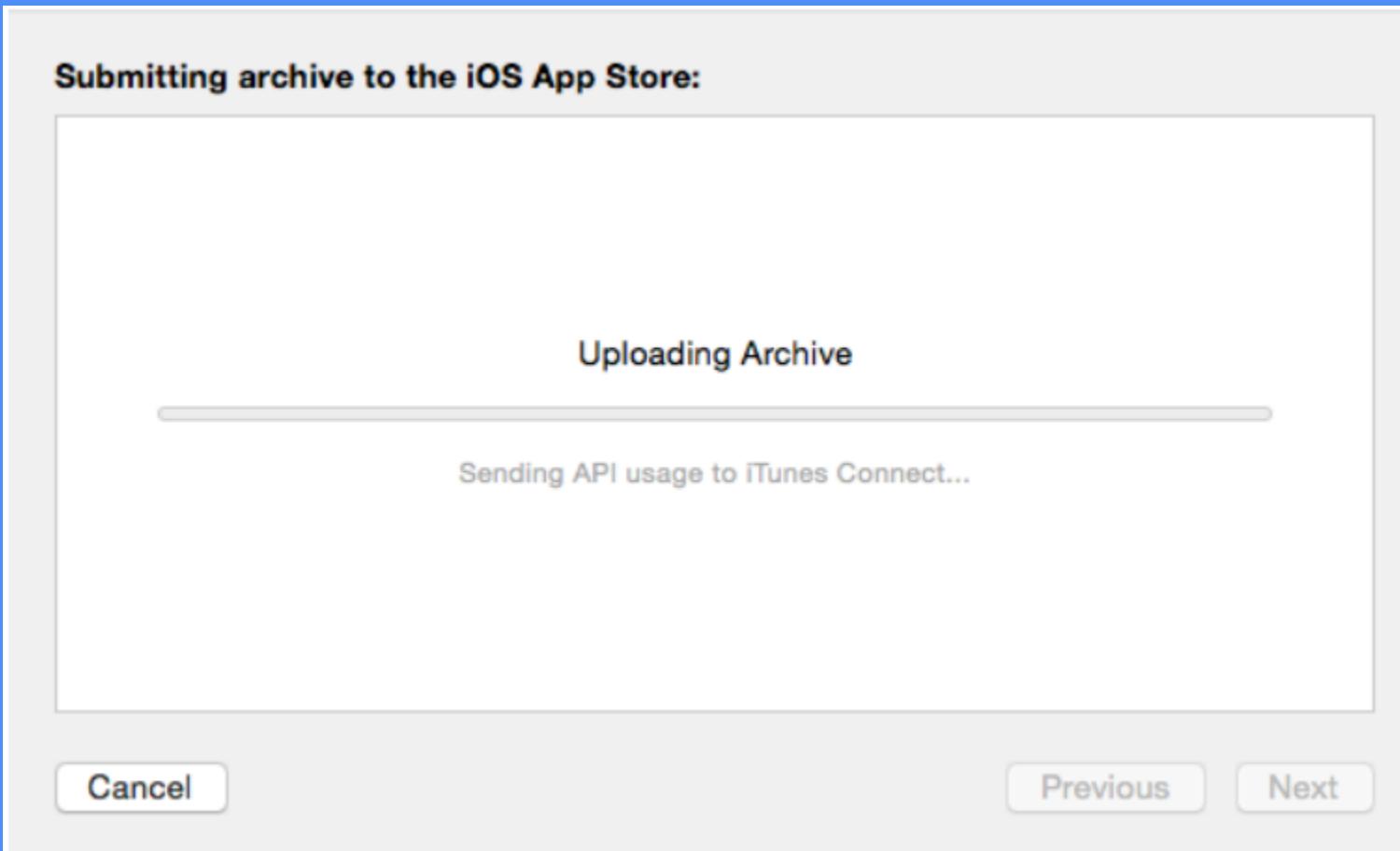
Archive validation process complete:

**Validation Successful**

Your app successfully passed all validation checks.

[Cancel](#) [Previous](#) [Done](#)

# Upload

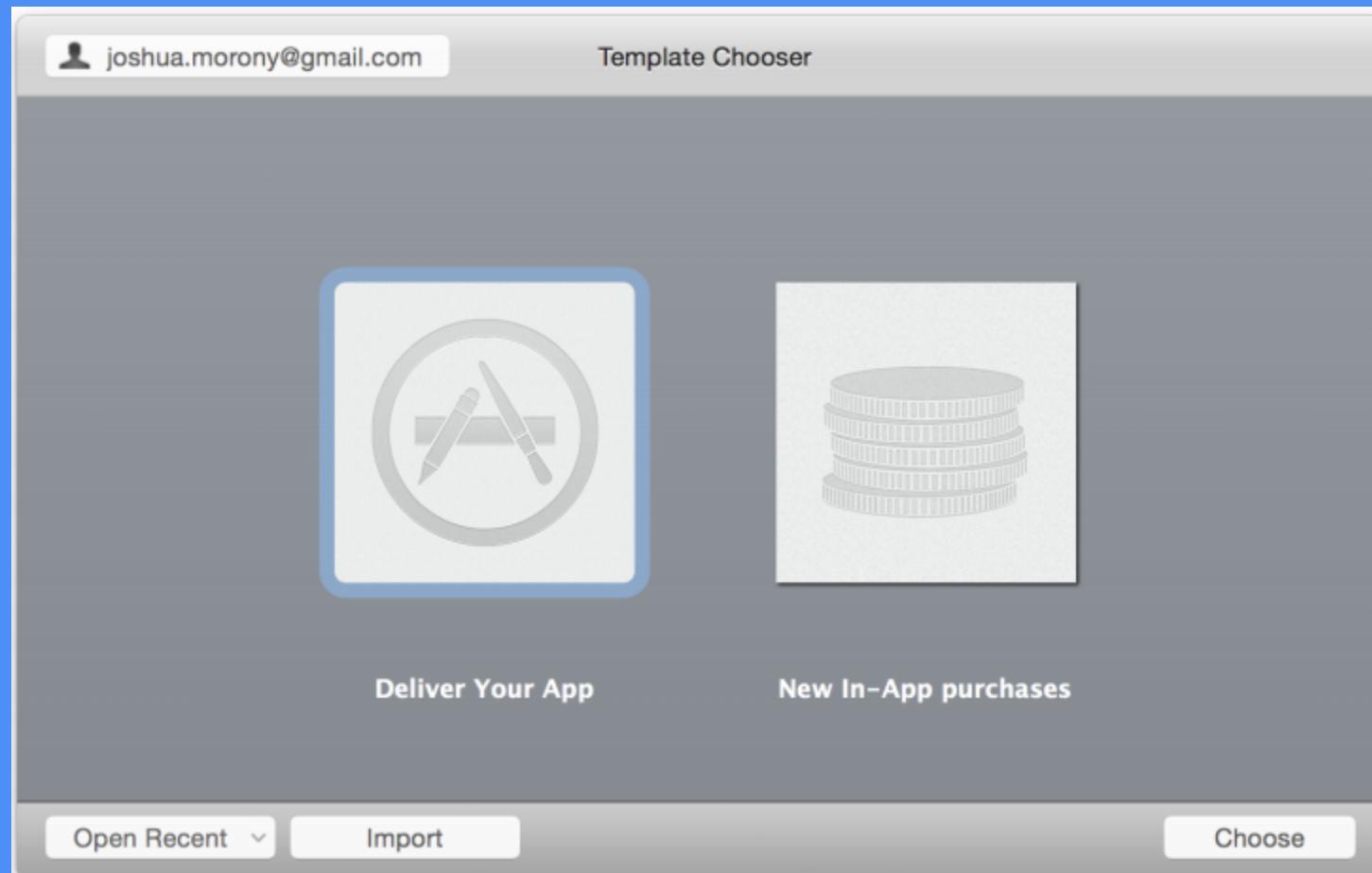


Submitting an app without  
Mac

# Ionic Package

# Phonegap Build

# Application Loader



itunesconnect

# Ionic Platform



## Push Notifications

Start sending push notifications.



## User Auth

Custom user authentication.



## Live Updates

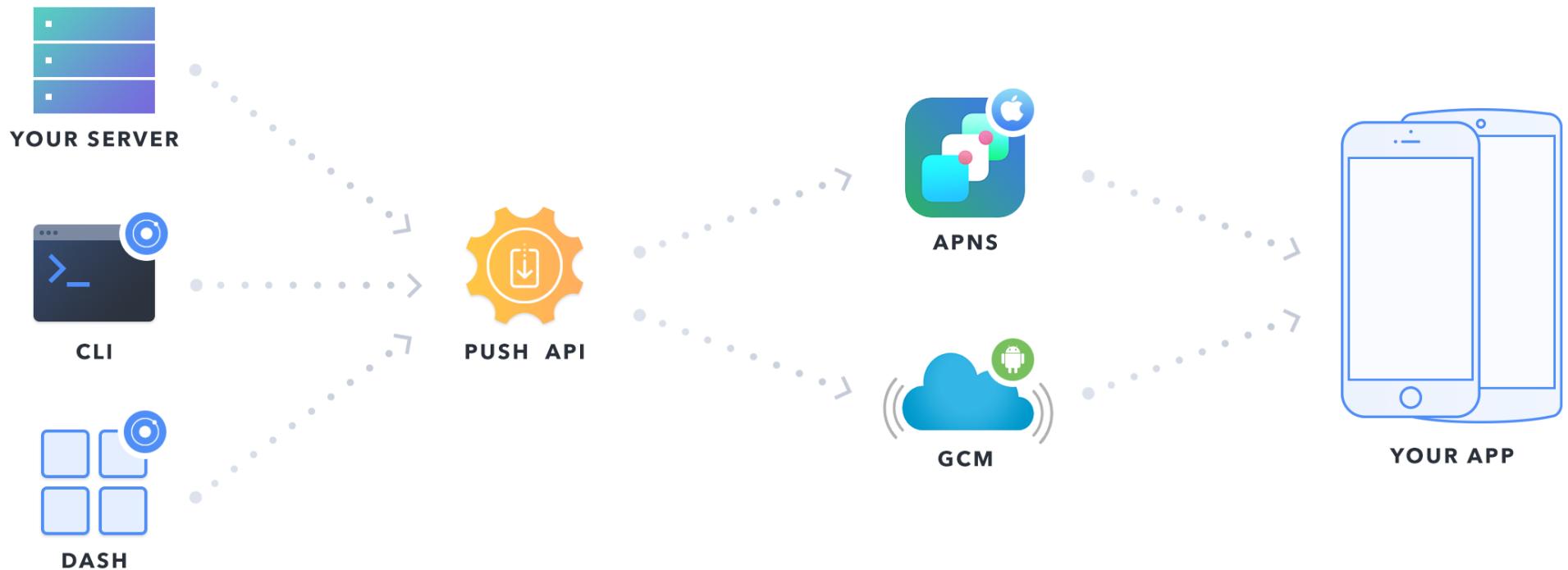
Live remote app updating.



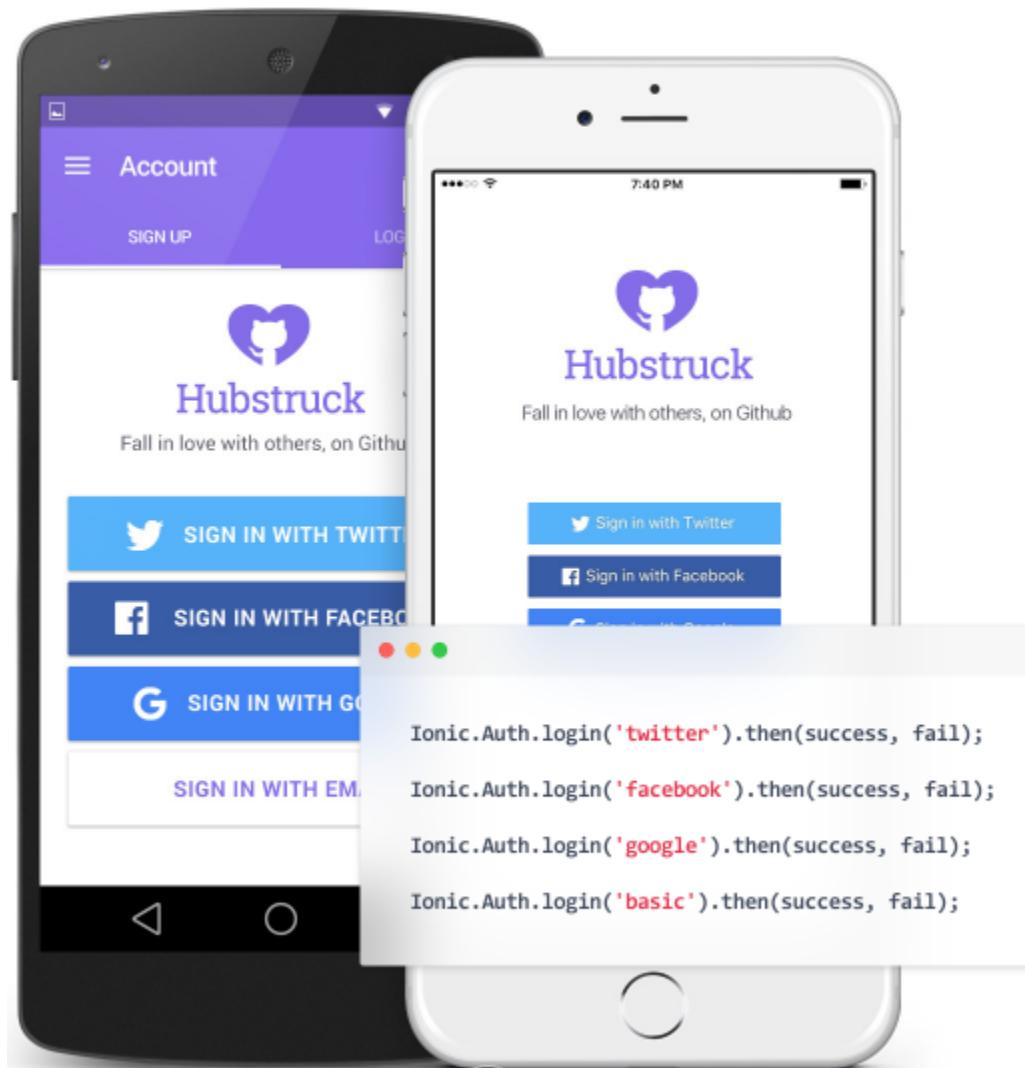
## Native Package

Build for native distribution.

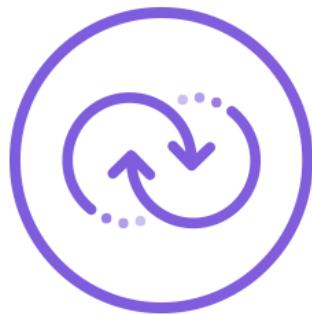
# Push



# Users



# Deploy



ionicdeploy



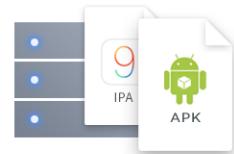
# Package



IONIC CLI



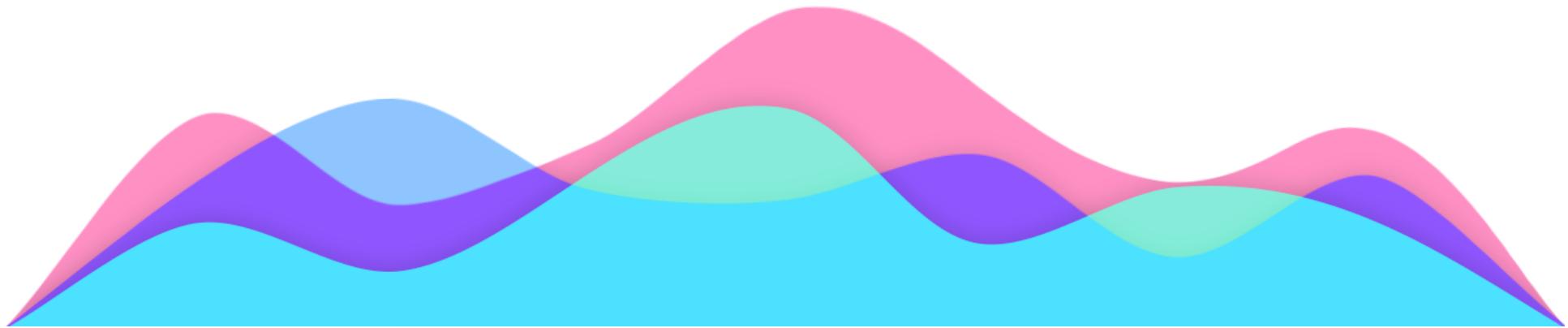
PACKAGE API



BUILD SERVER



ionicanalytics



# News

<http://www.ion-book.com/>

<http://blog.ionic.io/>

# Thanks