

2017.11

使用 Ionic 框架开发 Hybrid App

Hybrid mobile development with Ionic Framework

赵知非

2017-11-03

1 What is Hybrid App?

什么是Hybrid App

2 Comparison of Development Frameworks

主流开发框架的比较

3 Introduction to Ionic

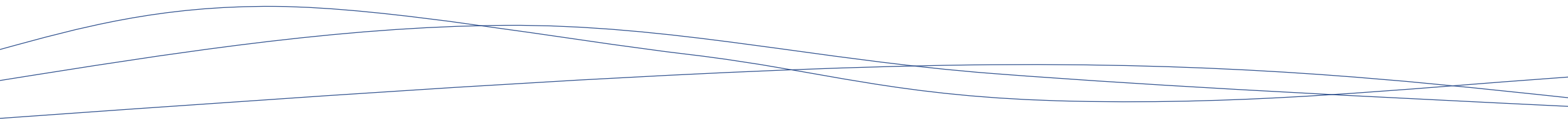
Ionic 简介

4 Usage of Ionic Framework

Ionic 框架的使用

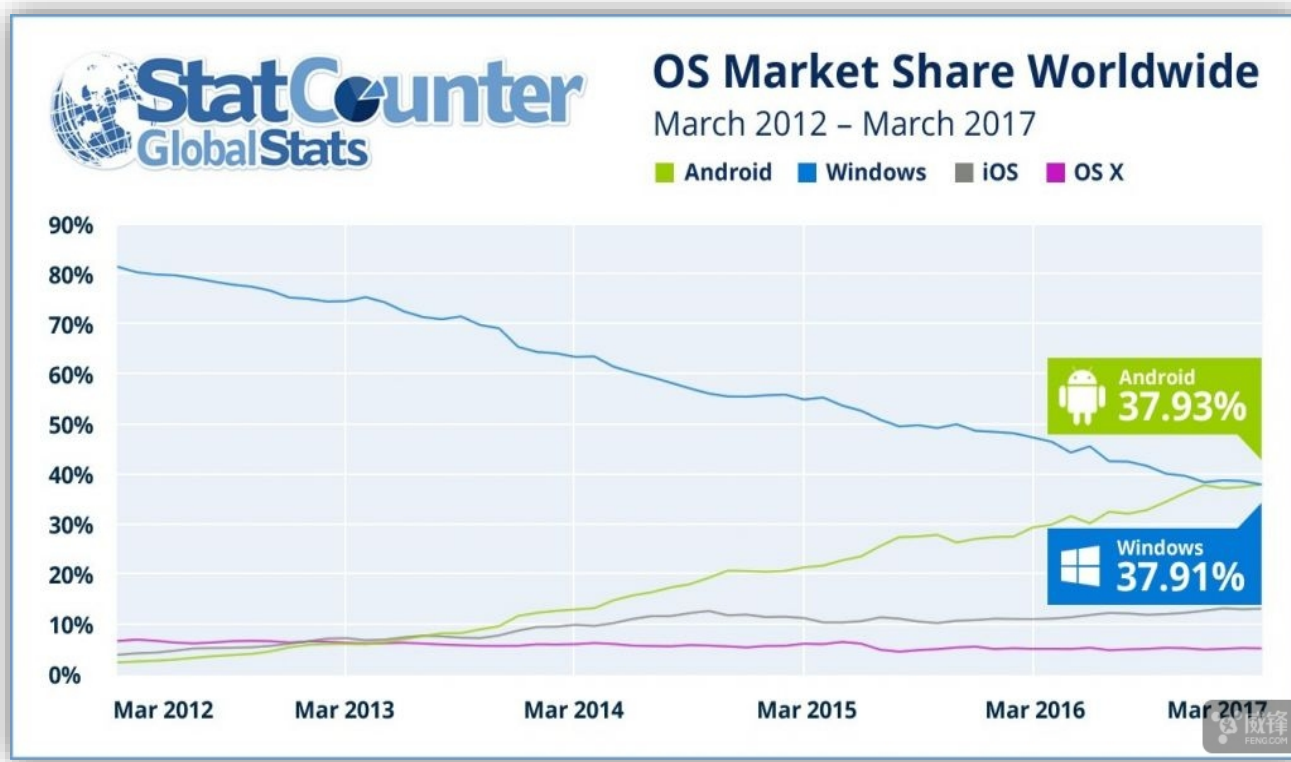
什么是Hybrid App?

What is Hybrid App?



Background

背景



我想开发一个移动App,
但是我...

- 只会前端语言 (HTML/CSS/JS)
- 不会Java, Swift/Object-C, C#...
- 没有一点Android/iOS的设计开发经验
- 没有配备苹果电脑
- 想要使代码具备跨平台能力, 做到“一次编写, 到处运行”
- ...

Background

背景

Web App (H5应用/网页应用)



- 不需要下载安装
- 浏览器渲染
- 只能调用浏览器提供的API
- 不同平台, 同一语言版本
(HTML + CSS + JavaScript)
- 运行速度慢
- 消耗资源多



Native App (原生应用)

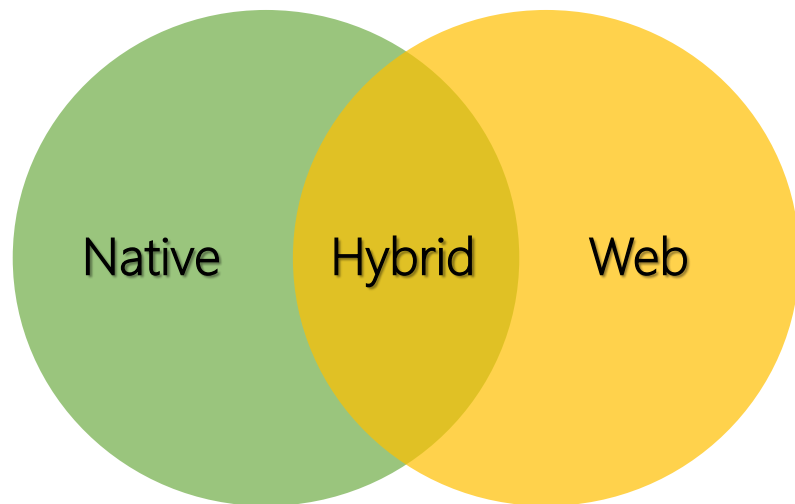


- 需要下载安装
- 系统渲染
- 可调用系统提供的API
- 不同平台, 不同语言版本
(Java/Swift/C# ...)
- 运行速度快
- 消耗资源少

Let's try Hybrid

Hybrid App (混合应用)

- Hybrid App 是 Native + Web 的混合类App
- 兼具“Native App 良好用户交互体验”和“Web App 跨平台开发”的优势。
- 看上去类似Native App，需要下载安装。
- 接近原生的使用体验，但访问主体内容是 Web 页面。



Hybrid vs Native

前端语言写的
Tab按钮

嵌套的
Webview

原生的
按钮控件

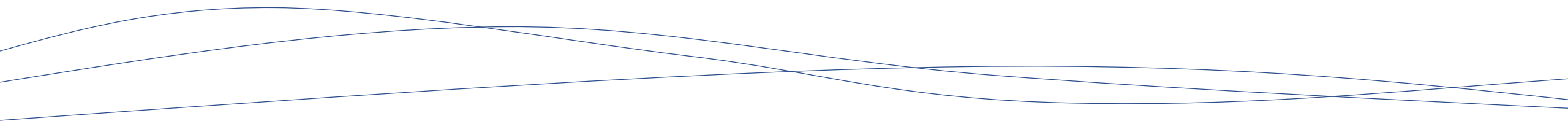


App: Hybrid vs Native vs Web

	Web App（网页应用）	Hybrid App（混合应用）	Native App（原生应用）
开发成本	低	中	高
维护更新	简单	简单	复杂
体验	差	优	优
Store或market认可	不认可	认可	认可
安装	不需要	需要	需要
跨平台	优	优	差

主流开发框架比较

Comparison of Development Frameworks



Mainstream Hybrid Framework

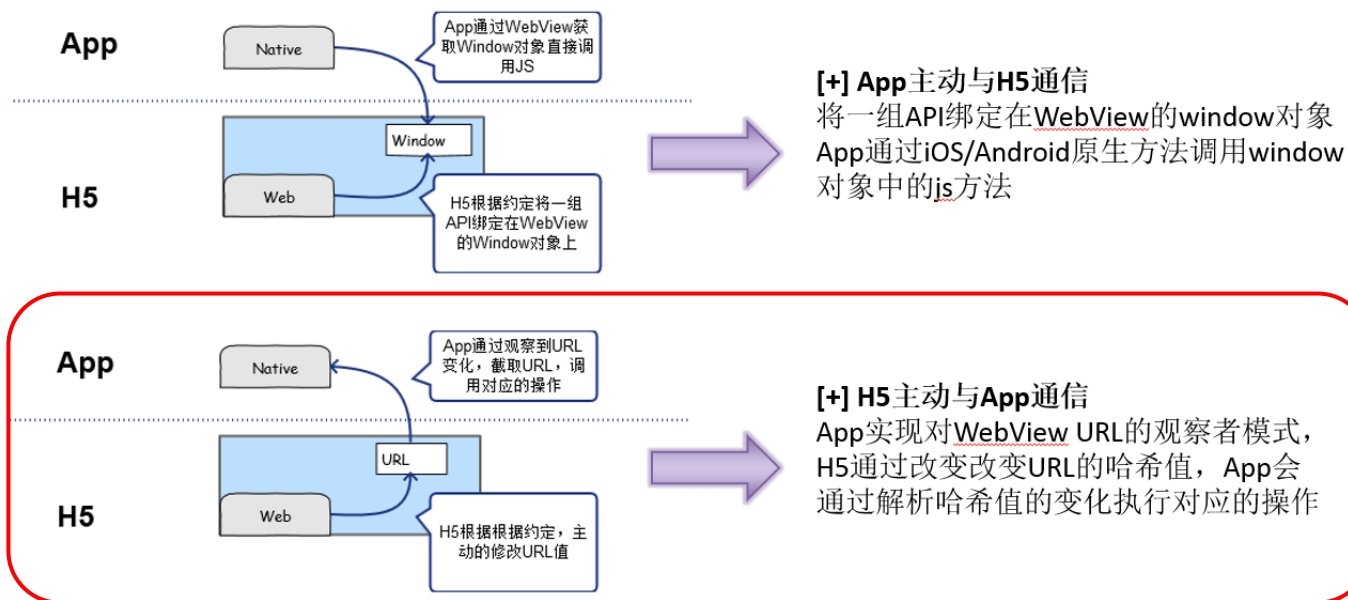
主流的 Hybrid App 开发框架

国内:

- AppCan (2012)
- DCloud (2013)
- ApiCloud (2014)
-

国外:

- Ionic
- React Native
-



>> 开发者工作

20%

>> AppCan平台工作

80%

UI

第三方JS

插件

JS

SDK

数据存储

访问统计

数据统计

数据分析

用户管理

Oauth组件

数据管理

文件存储

即时通讯

推送服务

移动用户认证

版本

文档中心

应用门户

行为

云补丁

编译报告

SVN

Comparison of Hybrid Frameworks

主流 Hybrid App 框架之间的对比

	AppCan	DCloud	ApiCloud	Ionic	React-Native
支持平台	Android/iOS	Android/iOS	Android/iOS	Android/iOS/WP/ Ubuntu	Android/iOS/WP
开发环境	Eclipse插件	Hbuilder（自主产 品）	Sublime/WebStor m/Eclipse插件	不限	不限
打包方式	在线	在线/离线	在线	离线	离线
UI 框架	无	MUI（自主产品）	少量自带UI模块	Ionic/Angular UI	原生UI
运行速度	慢	一般	慢	一般	快



ionic 简介

Introduction to Ionic

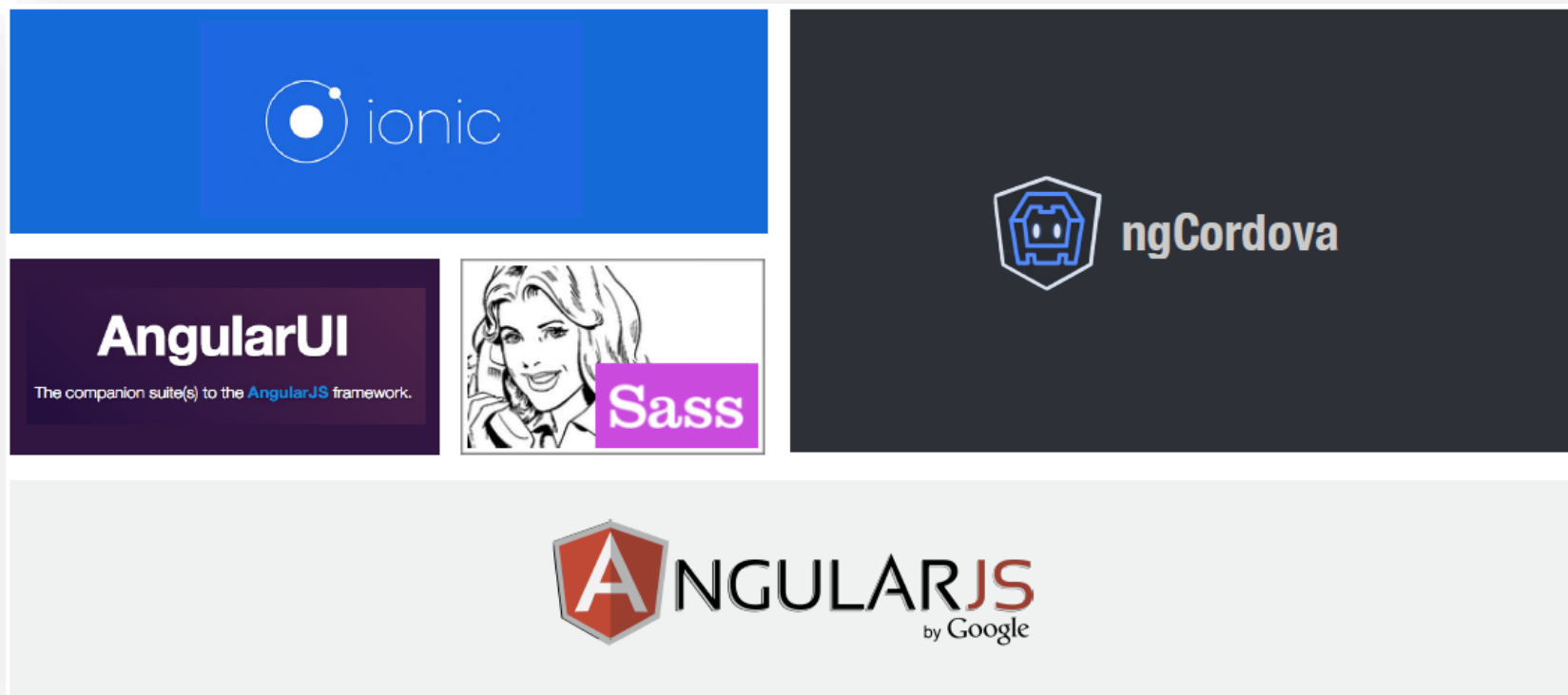
Ionic Technology Stack

Ionic 技术栈

- 基于 Angular 语法，简单易学；支持 Angular 特性，MVC 结构的代码易于维护
- 提供美观的 UI 设计，并通过 SASS 构建应用程序
- 专注原生，性能优越，运行速度快
- 提供了强大的命令行工具
- 完全开源



=



Based on Angular.js

基于Angular.js

Providers

angular.service()
angular.factory()
angular.provider()

Directives

angular.directive()

Controllers

angular.controller()

Templates

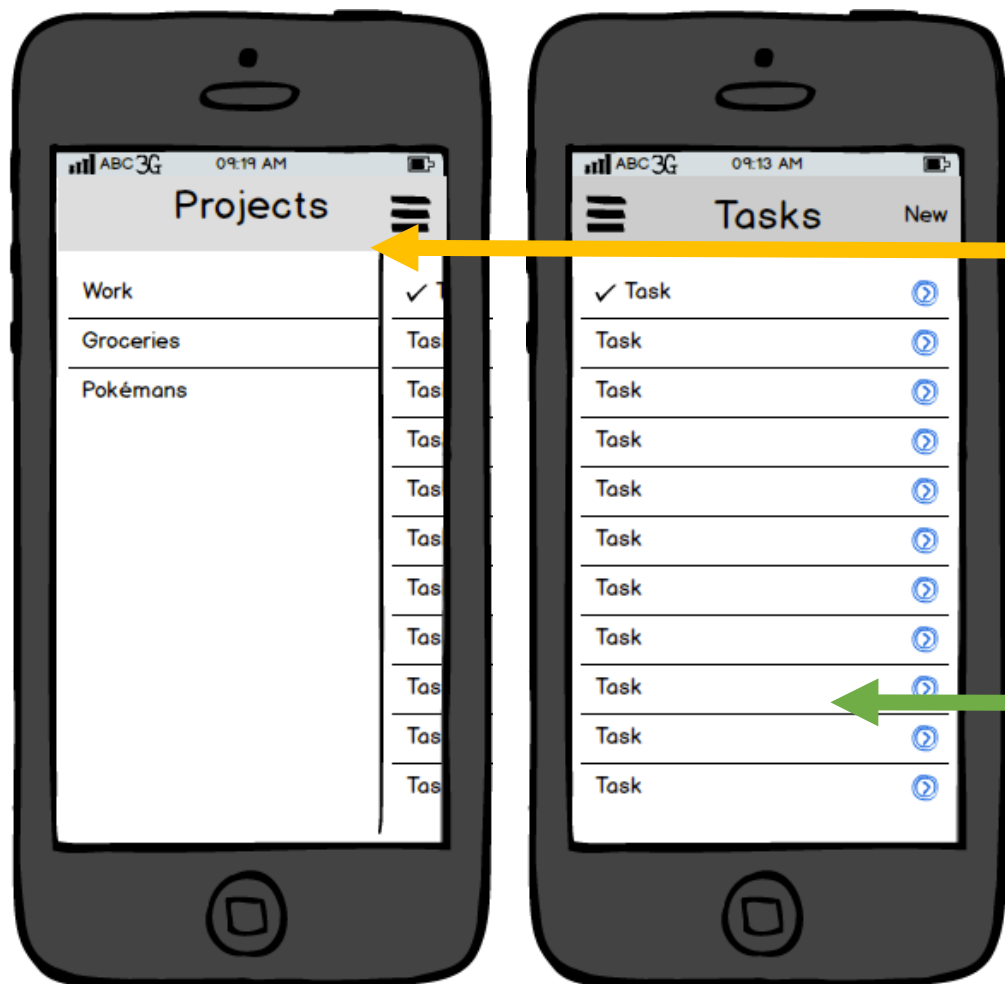
<html> </html>

Angular 2

- 模块 (Modules)
- 组件 (Components)
- 模板 (Templates)
- 元数据 (Metadata)
- 数据绑定 (Data Binding)
- 指令 (Directives)
- 服务 (Services)
- 依赖注入 (Dependency Injection)

States Managed By uiRouter

通过 uiRouter 管理状态切换



```
angular.module('ionicApp', ['ionic'])

.config(function ($stateProvider, $urlRouterProvider) {

  $stateProvider
    .state('menu', {
      abstract: 'true',
      templateUrl: 'templates/menu.html',
      controller: 'MenuCtrl'
    })

    /* ... */

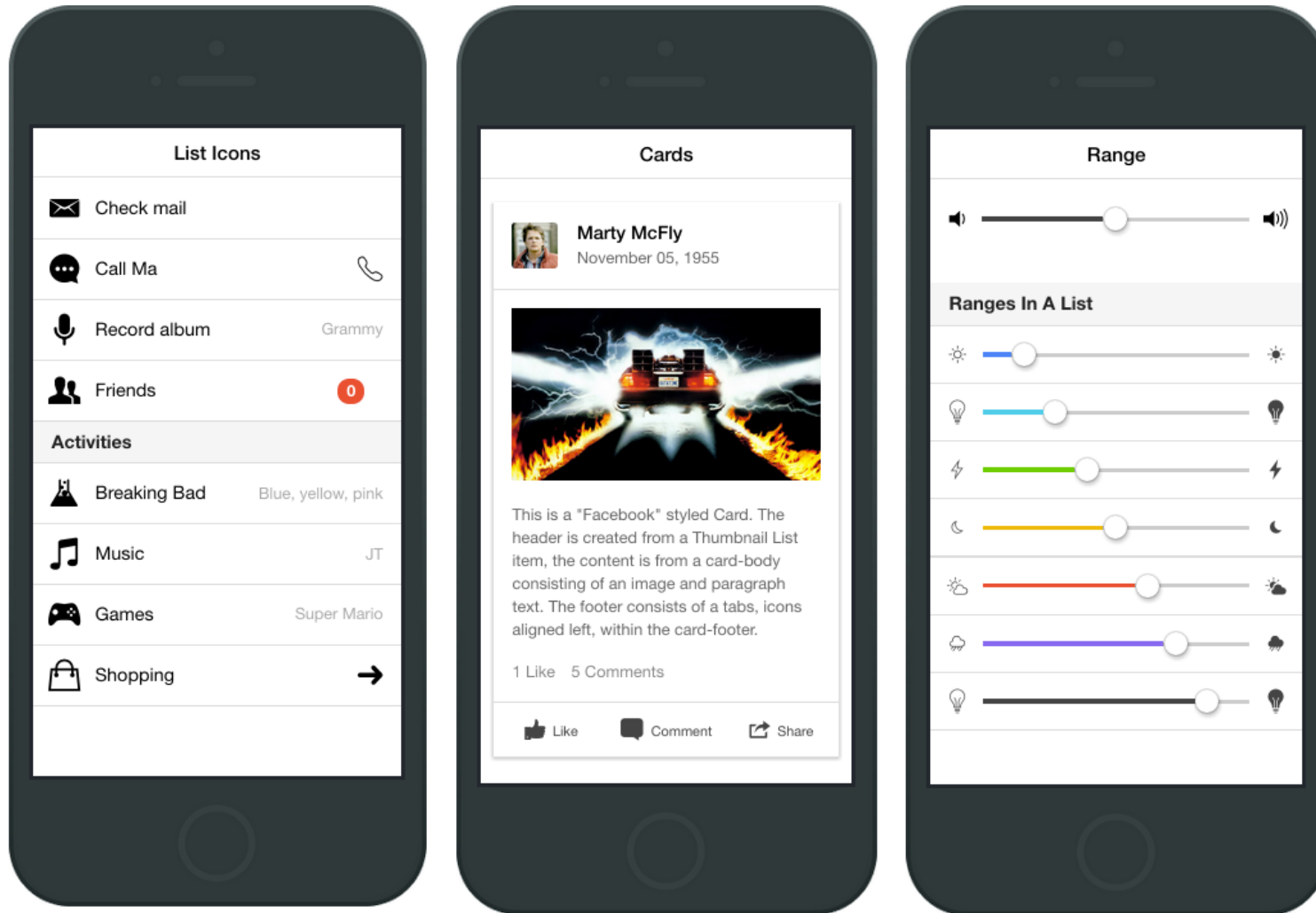
    .state('menu.work', {
      url: '/work',
      views: {
        menuContent: {
          templateUrl: 'templates/work.html',
          controller: 'WorkCtrl'
        }
      }
    });

  $urlRouterProvider.otherwise('/work');

});
```


UI Components in Ionic

Ionic 的 UI 组件



UI Components

Action Sheets
Alerts
Buttons
Cards
Carousels
Checkboxes
Footers
Forms
Headers
Icons
Lists
Modals
Navigations
Radio Buttons
Ranges
Selects
Tabs
Toggles

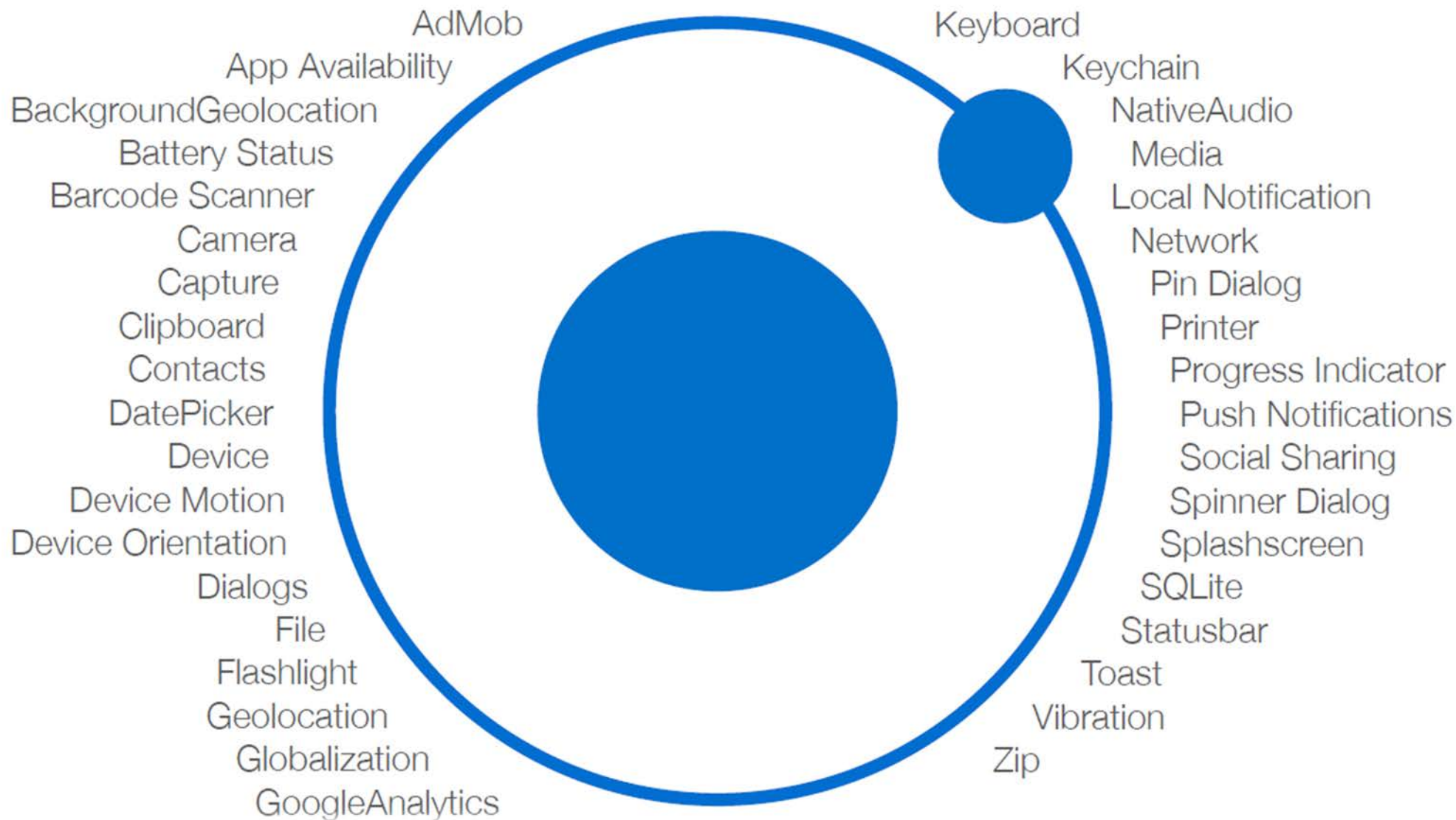
...

Events

on-hold
on-tap
on-touch
on-release
on-drag
on-drag-up
on-drag-right
on-drag-down
on-drag-left
on-swipe
on-swipe-up
on-swipe-right
on-swipe-down
on-swipe-left
...

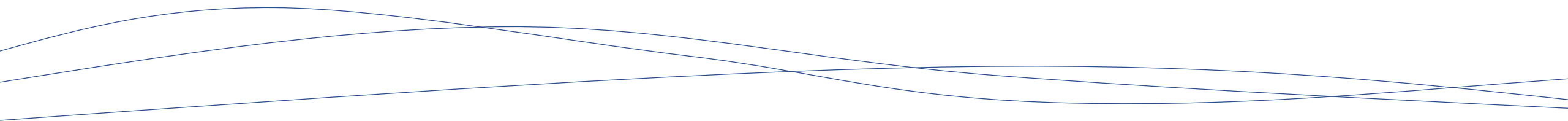
Access the Device with ngCordova

通过 ngCordova 来调用设备的系统原生接口



Ionic 框架的使用

Usage of Ionic Framework



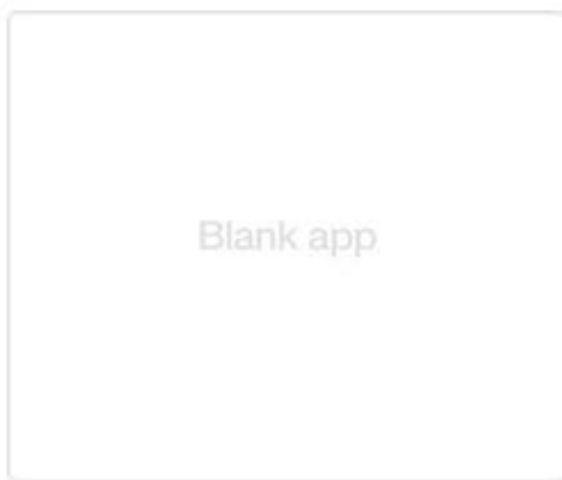
Getting Started

安装依赖与新建应用

安装依赖: `$ npm install -g cordova ionic`

新建应用: `$ ionic start myApp sidemenu` → 项目模板 (默认tabs)

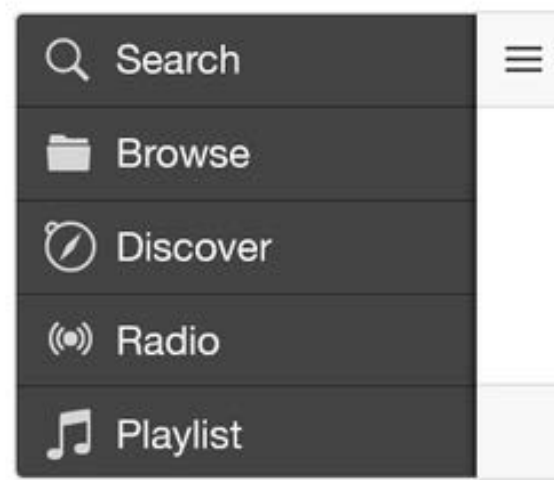
App 名称



`$ ionic start myApp blank`



`$ ionic start myApp tabs`



`$ ionic start myApp sidemenu`

Getting Started

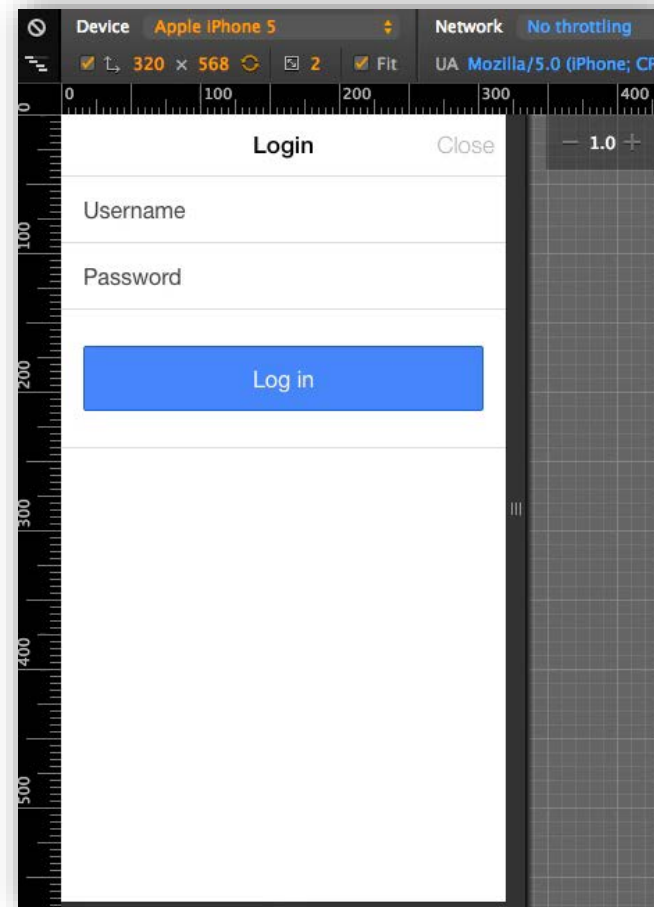
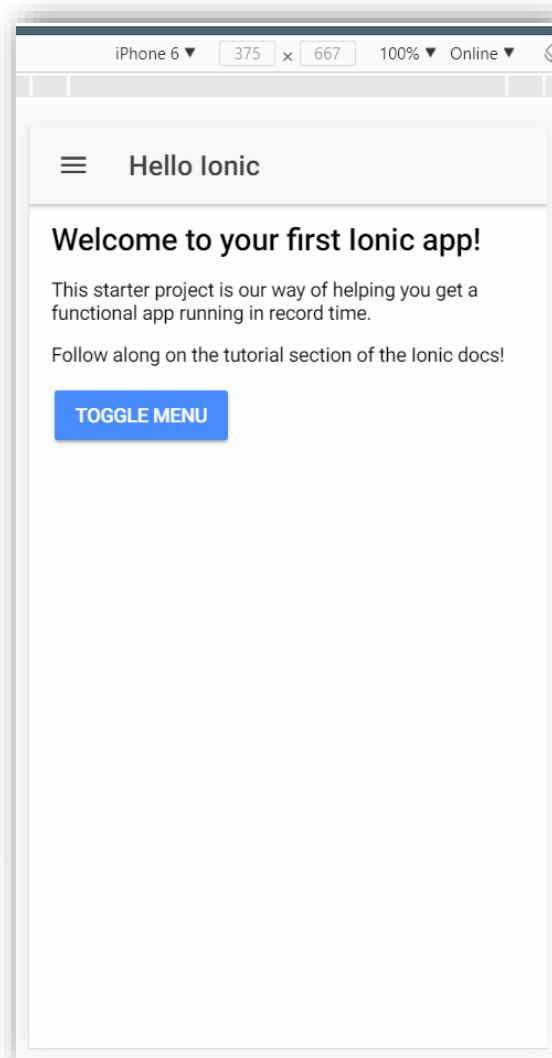
通过浏览器查看与调试应用

启动应用: `$ cd myApp; ionic serve`

```
PS E:\Coder\MyIonicProject> ionic serve
Starting app-scripts server: --address 0.0.0.0 --port 8100 --livereload-port 35729 --dev-logger-port 53703 - Ctrl+C to
cancel
[11:04:23] watch started ...
[11:04:23] build dev started ...
[11:04:23] clean started ...
[11:04:24] clean finished in 10 ms
[11:04:24] copy started ...
[11:04:24] deeplinks started ...
[11:04:24] deeplinks finished in 32 ms
[11:04:24] transpile started ...
[11:04:36] transpile finished in 12.02 s
[11:04:36] preprocess started ...
[11:04:36] preprocess finished in 1 ms
[11:04:36] webpack started ...
[11:04:37] copy finished in 13.50 s
[11:04:45] sass started ...
[11:04:48] sass finished in 3.55 s
[11:04:48] postprocess started ...
[11:04:48] postprocess finished in 25 ms
[11:04:48] lint started ...
[11:04:48] build dev finished in 24.99 s
[11:04:49] watch ready in 25.66 s
[11:04:49] dev server running: http://localhost:8100/

[OK] Development server running!
    Local: http://localhost:8100
    External: http://192.168.30.1:8100, http://192.168.153.1:8100

[11:04:51] lint finished in 2.95 s
```



Getting Started

使用 ionic UI 组件

```
<ion-list>
  <ion-item
    nav-clear
    class="item-icon-left"
    menu-close
    ui-sref="^.menu.home">

    <i class="icon ion-home"></i>Home
  </ion-item>

  <!--...-->

  <ion-item class="item-divider">
    Resources
  </ion-item>
</ion-list>
```

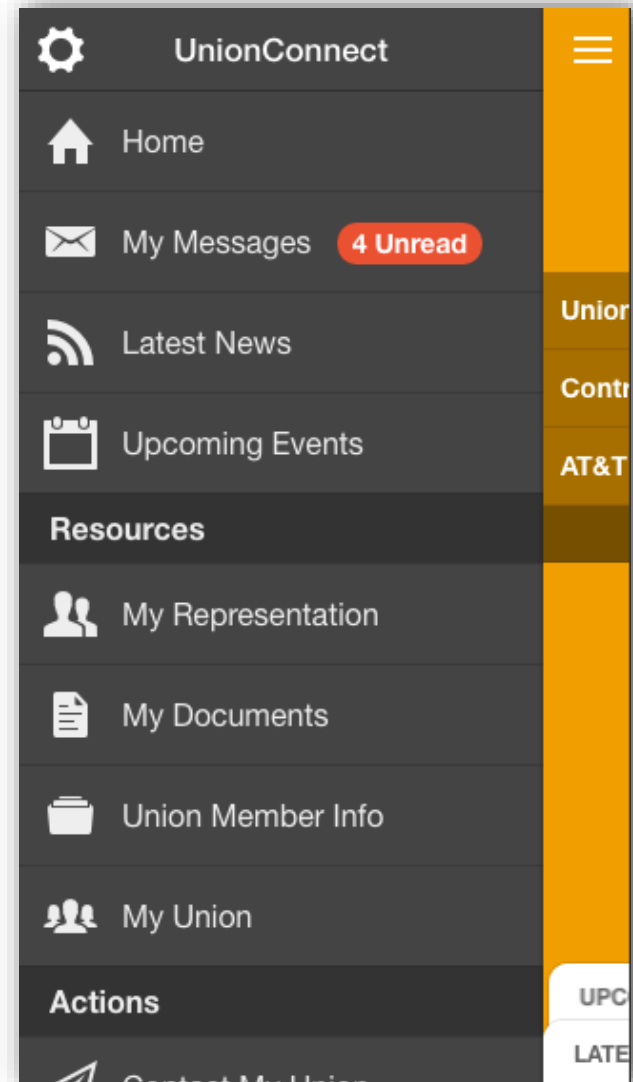
THUMBNAILS



Nirvana
Nevermind

```
<a class="item item-thumbnail-left" href="#">
  
  <h2>Nirvana</h2>
  <p>Nevermind</p>
</a>

</a>
```



Getting Started

使用 Angular 语法进行数据绑定

.html

```
<ion-list ng-controller="ListController">
  <ion-item ng-repeat="item in items">
    <i class="icon ion-{{item.icon}}"></i>{{item.title}}
  </ion-item>
</ion-list>
```

.js / .ts

```
angular.module("IonicApp")
  .controller("ListController", function ($scope) {
    $scope.items = [
      {icon: "home", title: "Home"},
      {icon: "envelope", title: "My Messages"},
    ];
  });
```

{{item.title}}

Home

My Messages

Getting Started

通过 ngCordova 调用系统原生API

```
$ ionic cordova plugin add cordova-sms-plugin
```

```
$ npm install --save @ionic-native/sms
```

```
import { SMS } from '@ionic-native/sms';
```

```
constructor(private sms: SMS) { }
```

```
...
```

```
// Send a text message using default options  
this.sms.send('416123456', 'Hello world!');
```

```
$ ionic cordova plugin add cordova-plugin-camera
```

```
$ npm install --save @ionic-native/camera
```

```
import { Camera, CameraOptions } from '@ionic-native/camera';
```

```
constructor(private camera: Camera) { }
```

```
...
```

```
const options: CameraOptions = {  
  quality: 100,  
  destinationType: this.camera.DestinationType.DATA_URL,  
  encodingType: this.camera.EncodingType.JPEG,  
  mediaType: this.camera.MediaType.PICTURE  
}
```

```
this.camera.getPicture(options).then((imageData) => {  
  // imageData is either a base64 encoded string or a file URI  
  // If it's base64:  
  let base64Image = 'data:image/jpeg;base64,' + imageData;  
}, (err) => {  
  // Handle error  
});
```

Getting Started

通过 Cordova 生成不同平台的 App

Android

```
$ ionic cordova platform add android
```

实体机测试

```
$ ionic cordova build
```

```
$ ionic cordova run android
```

虚拟机测试

```
$ ionic cordova emulate android
```

iOS

```
$ ionic cordova platform add ios
```

实体机测试

```
$ ionic cordova build
```

```
$ ionic cordova run ios
```

虚拟机测试

```
$ ionic cordova emulate ios
```

```
PS E:\Coder\Ionic\superApp> ionic cordova emulate android
```

Ionic 官网:

- <http://ionicframework.com/>

Ionic UI 组件:

- <http://ionicframework.com/docs/components/>

Ionic Native API:

- <http://ionicframework.com/docs/api/>
- <http://ionicframework.com/docs/native/>

THANKS