IonFullApp Support Documentation

**Table of Contents**

[Overview](#h.c4yomrwzk2h8)

[Videos, tutorials, guides and even more help](#h.2vmo62a8afox)

[What is Ionic Framework?](#h.14s9ejcljt2t)

[How do I start?](#h.euv1tj2c6pqg)

[Installing Ionic Framework](#h.pvs5zl24gbhg)

[Install the app](#h.do8bppcgy3mr)

[Build the app](#h.o9gpdagkqd13)

[Building for iOS](#h.jso4crmfxs5)

[Building for android](#h.capmg2xz1de1)

[Configure your Wordpress site](#h.hrk1kicys1bl)

[Wordpress Integration](#h.dq25rvisp2an)

[Wordpress JSON API](#h.msk20srsa7js)

[Use the API](#h.ce67gd5gkp9q)

[Google Feed API](#h.k8gf8366mnmj)

[Bookmarks](#h.ljozn8ie1bhf)

[Email Composer](#h.jjya40ui61q0)

[Image picker](#h.l7nu9dqldwgn)

[Social Sharing](#h.932jicxvrfbk)

[AdMob](#h.b8noygwpm0xm)

[iAd](#h.cfm0hlp4hf76)

[App Rate](#h.m6cm7amcfmp0)

[Maps and GeoLocation](#h.omct0zmr7fj8)

[Cordova geolocation plugin](#h.wq734w3yv5lo)

[GoogleMap AngularJS Directive](#h.xpcgxexo889g)

[InAppBrowser](#h.bgzb9xgvb3fy)

[App Settings](#h.52n8t09iyevc)

[Action Sheet](#h.jcazdro7749u)

[Profile](#h.vsh4owvmjlbv)

[Slider](#h.fdav4b2bvjuu)

[Login/Signup](#h.3i8zcuyigh0t)

[Push Notifications](#h.b4tunlt42s9d)

[Android Pre-requisites](#h.r08jsnvarmvl)

[iOS Pre-requisites](#h.vmc3375k9wyx)

[Register the application with the Google Cloud Messaging service](#h.yqyyfuvjhbx3)

[Register the application with the Apple APN service](#h.3v5zusgq1y9g)

[Configure Push Server](#h.secy0ezbf8e6)

[Add Push Notifications to the app](#h.ee4o1oakfow2)

[Splash screen and app icon](#h.r4c98ygozqvw)

[Image Sizes](#h.9tts0ntnmil7)

[Let’s customize some styles](#h.mx0hn3kd70eo)

[Prepare ionic app for production](#h.cqy3u5qe9whd)

[Lint your javascript](#h.jcv6pkpxzwzk)

[HTML templates transformation](#h.f6ijw2m0tnuy)

[Enable ng-strict-di](#h.hvsz6wbx2sqm)

[Concatenate js and css files](#h.nl9lhibhn6hl)

[Uglify, minify and obfuscate](#h.ewd287rk3rak)

[Common issues](#h.n1mdmldw1chm)

[Errors when building for iOS](#h.2qmy9cwazujr)

[Errors with Wordpress JSON API](#h.9mkk4jcamcmc)

[Errors when running npm install on Windows](#h.kgjz32l0uj97)

[Ionic setup sass - error](#h.v8udu86k1pip)

[FAQS](#h.ttpw1znwwktv)

[How do I start?](#h.q6gm1hpdnmuv)

[What program do you use to edit the code?](#h.v53ay6bddz5b)

[Can I use Phonegap Build?](#h.k1ykmsdquj7a)

[RTL support](#h.usnsys52wiim)

[How to change the status bar color](#h.b0nr1o7ih0in)

[Disable login](#h.cctdi02vfzhk)

[What is dist/ folder? (removed on version 1.2)](#h.q153dzx8u2q1)

[Remote Debugging on Android with Chrome](#h.x5zuse4hfzzh)

[Remote Debugging on iOS with Safari](#h.jijuzvqigvtu)

[Older versions of Android devices (4.0-4.3)](#h.pmfosm13723)

[Changelog](#h.tlz6ve6y1dqz)

[Support](#h.aqwrd5sbu3ln)

# 

# Overview

**IonFullApp** is a PhoneGap / Cordova Full Hybrid App with AngularJS + [Ionic framework](http://ionicframework.com/)

PhoneGap is a very helpful hybrid mobile app development framework, which can drastically improve your HTML5 app development experience. It allows you to easily access native functionality, like camera and Geolocation via JavaScript. Ionic framework is an open source front-end SDK for developing awesome hybrid mobile apps with HTML5.

**IonFullApp** gives you the bootstrap you need in order to build your next PhoneGap / Cordova application based on the Ionic framework. We have put a lot of work for making this app beautiful so we hope you like it!

# Videos, tutorials, guides and even more help

If you are new to this technologies please take some time to look at this videos. They will help you to install and understand Ionic and its dependencies.

**MUST SEE:**

* [Installing Ionic and its dependencies](http://ionicframework.com/docs/guide/installation.html)
* [Getting started building mobile apps with the Ionic Framework](https://www.youtube.com/watch?v=C-UwOWB9Io4)
* [100+ Ionic Framework Resources](http://mcgivery.com/100-ionic-framework-resources/?utm_source=mobilewebweekly&utm_medium=email)
* [Installing Cordova, Android, and Ionic on Windows 7 and 8](http://learn.ionicframework.com/videos/windows-android/)
* [Installation and setup of Cordova, Ionic, Node.js, Android SDK](https://www.youtube.com/watch?v=LLX2FEg0TjU)
* [Introduction to Ionic for iOS](http://learn.ionicframework.com/videos/getting-started/)
* [Customizing the Status Bar](http://learn.ionicframework.com/formulas/customizing-the-status-bar/)
* [Written tutorials and how-to's on building hybrid apps with Ionic](http://learn.ionicframework.com/formulas/)

## What is Ionic Framework?

Ionic is a powerful HTML5 native app development framework that helps you build native-feeling mobile apps all with web technologies like HTML, CSS, and Javascript.

Ionic is focused mainly on the look and feel, and UI interaction of your app. That means it is not a replacement for [PhoneGap](http://phonegap.com/) or your favorite Javascript framework. Instead, Ionic simply fits in well with these projects in order to simplify one big part of your app: the front end. We recommend reading [Where does the Ionic Framework fit in?](http://ionicframework.com/blog/where-does-the-ionic-framework-fit-in/) to get a good understanding of the framework's goals.

Ionic currently requires AngularJS in order to work at its full potential. While you can still use the CSS portion of the framework, you'll miss out on powerful UI interactions, gestures, animations, and other things.

## How do I start?

1. [Install Ionic](#h.pvs5zl24gbhg)
2. [Install the app](#h.do8bppcgy3mr)
3. [Build the app](#h.o9gpdagkqd13)
4. [Configurations](#h.hrk1kicys1bl)

Please check [Common Issues](https://docs.google.com/document/d/1yUDhFlL1G6Gfz178Hys_7fxz85dFRBqehhOOrqxLj3k/edit#heading=h.n1mdmldw1chm) before contacting support.

Also read the [FAQS](#h.ttpw1znwwktv) to find more answers.

# 

# Installing Ionic Framework

Please refer to the Ionic offical installation documentation in order to install Ionic and cordova on your machine:<http://ionicframework.com/docs/guide/installation.html>

**Ionic Getting started guide**

[ionicframework.com/getting-started](http://ionicframework.com/getting-started/)

**Ionic Documentation**

[ionicframework.com/docs](http://ionicframework.com/docs/)

**Visit the Ionic Community Forum**

[forum.ionicframework.com](http://forum.ionicframework.com/)

# Install the app

After you have installed Ionic in your computer you are almost ready to start working on this app, but we need to install the app dependencies:

1. Unzip the folder you have downloaded from Envato
2. Move the unzipped folder to your preferred path inside your computer
3. Open your terminal and go to your selected path
4. Run **npm install -g bower** in order to install bower
5. Run **npm install** in order to install all the node modules required by the app
6. Run **bower install** in order to install all the javascript libraries required by the app
7. Run **ionic setup sass**
8. Run **ionic serve** to start a local dev server and watch/compile Sass to CSS

# Build the app

## Building for iOS

1. Add ios platform
   1. **ionic platform add ios**
2. Build your app
   1. **ionic build ios**
3. Open ***yourAppName.xcodeproj*** in the ***yourProjectName/platforms/ios*** directory
4. In Xcode, run the application on a device connected to your computer or in the iOS emulator

## Building for android

1. Make sure the Android SDK and the ant build tool are available on your system. The Android SDK is available [here](http://developer.android.com/sdk/index.html). **Both the android and ant tools must be available in your path**. To test your configuration, you should be able to execute both **android** and **ant** from the command line. Make sure ANDROID\_HOME is exported and adding the SDK tool directories to PATH. Check [this](http://stackoverflow.com/a/26701273/1330740) for more information.
2. Add android platform
   1. **ionic platform add android**
3. Build your app
   1. **ionic build android**
4. To build and run the application on an Android device connected to your computer using a USB cable and then run
   1. **ionic run android**
5. To build and run the application in the Android emulator
   1. **ionic emulate android**

Or you can check it on the web with the command: **ionic serve**

# 

# Configure your Wordpress site

In www/js/config.js you will find the following configurations that you have to replace with your own values:

angular.module('your\_app\_name.config', [])

.constant('WORDPRESS\_API\_URL', 'http://wordpress.startapplabs.com/blog/api/')

.constant('GCM\_SENDER\_ID', '574597432927')

* 'WORDPRESS\_API\_URL' is your wordpress site url
* 'GCM\_SENDER\_ID' is your Google Cloud Messaging ID required for push notifications to Android

Please enable permalinks in your Wordpress.

Run **ionic serve** to make this changes effective.

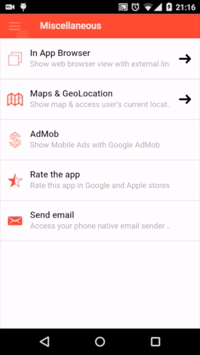
# 

# Wordpress Integration

# 

Features:

* Wordpress JSON API
* Loading sign when bringing posts
* Pull to refresh new posts
* Infinite scroll (improves user experience in mobile)
* “Read more” button to have access to the entire post
* Bookmark icon
* Share a post with Social Sharing tool
* Like a post
* Increase and Decrease font size control
* Wordpress post categories



## Wordpress JSON API

JSON API allows you to retrieve and manipulate WordPress content using HTTP requests. There are three main goals:

1. Provide a simple, consistent external interface
2. Create a stable, understandable internal implementation
3. Enable new types of extensions for WordPress

**Note:** in order to be able to access the data from your Wordpress you need to install the JSON API Wordpress plugin to your Wordpress site. You can follow these steps:

1. Download the plugin from this url: <https://downloads.wordpress.org/plugin/json-api.1.1.1.zip>
2. Upload the json-api folder to the /wp-content/plugins/ directory or install directly through the plugin installer.
3. Activate the plugin through the 'Plugins' menu in WordPress or by using the link provided by the plugin installer.

## Use the API

The API is extremely easy to use. The standard response format for JSON API is [JSON](http://json.org/).

With this few lines of code you can make an http request to get the recent posts of your site:

|  |
| --- |
| $http.jsonp("http://yoursitehere.com/api/get\_recent\_posts?callback=JSON\_CALLBACK")  .success(function(data) {  $scope.posts = data.posts;  })  .error(function(data) {  }); |

Here is an example response from ***http://yoursitehere.com/api/get\_recent\_posts*** called on a default WordPress installation:

|  |
| --- |
| {  "status": "ok",  "count": 1,  "count\_total": 1,  "pages": 1,  "posts": [  {  "id": 1,  "type": "post",  "slug": "hello-world",  "url": "http:\/\/localhost\/wordpress\/?p=1",  "title": "Hello world!",  "title\_plain": "Hello world!",  "content": "<p>Welcome to WordPress. This is your first post. Edit or delete it, then start blogging!<\/p>\n",  "excerpt": "Welcome to WordPress. This is your first post. Edit or delete it, then start blogging!\n",  "date": "2009-11-11 12:50:19",  "modified": "2009-11-11 12:50:19",  "categories": [],  "tags": [],  "author": {  "id": 1,  "slug": "admin",  "name": "admin",  "first\_name": "",  "last\_name": "",  "nickname": "",  "url": "",  "description": ""  },  "comments": [  {  "id": 1,  "name": "Mr WordPress",  "url": "http:\/\/wordpress.org\/",  "date": "2009-11-11 12:50:19",  "content": "<p>Hi, this is a comment.<br \/>To delete a comment, just log in and view the post's comments. There you will have the option to edit or delete them.<\/p>\n",  "parent": 0  }  ],  "comment\_count": 1,  "comment\_status": "open"  }  ] } |

**Examples of usage**

* You can call [http://www.example.org/api/get\_recent\_posts](http://www.example.org/?json=get_recent_posts) to get the recent posts from your blog
* You can call [http://www.example.org/api/get\_post?post\_id=47](http://www.example.org/?json=get_post&post_id=47) to get the data of the post with ***id=47***

**You can find the full documentation with more examples of how to use this API here:** <https://wordpress.org/plugins/json-api/other_notes/>

If you are having trouble please see this section [#heading=h.ttpw1znwwktv](https://docs.google.com/document/d/1Oh0EfqUCzzPoUdHiSFcxykTy65afY7g4mHz3wXtBabo/edit#heading=h.ttpw1znwwktv)

**Custom fields**

If your posts have custom\_fields, you can easily access them on the field **post.custom\_fields** for example:

**<div ng-if="post.custom\_fields.myCustomFieldName">**

**//show your custom field**

**<p>{{post.custom\_fields.myCustomFieldName}}</p>**

**</div>**

# 

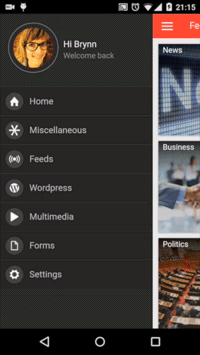
# 

# Google Feed API

We use [Google Feed API](https://developers.google.com/feed/). With the Feed API, you can download any public Atom, RSS, or Media RSS feed using only JavaScript, so you can easily mashup feeds with your content. The Google Feed API takes the pain out of developing mashups in JavaScript because you can now mashup feeds using only a few lines of JavaScript, rather than dealing with complex server-side proxies. This makes it easy to quickly integrate feeds on your website.

Features:

* Google Feed API
* Different categories
* Very easy to customize
* Loading sign when bringing posts
* Pull to refresh new posts
* “Read more” button to have access to the entire post using the InAppBrowser plugin.
* Bookmark icon



To make it easy, we use a file named ***/feeds-categories.json*** where we define the RSS categories of and its source providers. This JSON file has the following structure where ***news*** is a feed category and ***bbc*** and ***reuters*** are feed sources:

|  |
| --- |
| [  {  "id":"news",  "title":"News",  "image":"img/feeds/news.jpg",  "feed\_sources":[  {  "id":"bbc",  "title":"BBC",  "description":"The latest stories from the World section of the BBC News web site.",  "image":"img/feeds/logos/bbc.jpg",  "url":"http://feeds.bbci.co.uk/news/world/rss.xml"  },  {  "id":"reuters",  "title":"Reuters",  "description":"Reuters.com is your source for breaking news, business, financial and investing news, including personal finance and stocks. Reuters is the leading global provider of news, financial information and technology solutions to the world's media, financial institutions, businesses and individuals.",  "image":"img/feeds/logos/reuters.jpg",  "url":"http://feeds.reuters.com/Reuters/worldNews"  }  ]  }  ] |

You can find this file in ***/www/feeds-categories.json***

**If you want to change the feeds categories and its source providers, you only have to modify this file.**

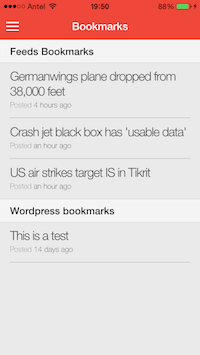
# 

# 

# Bookmarks

(added in version 1.1)

There is a “Saved for later” section which allows your users to save the posts they care in a separate section of the app. The bookmarks id’s are saved in the phone’s local storage.



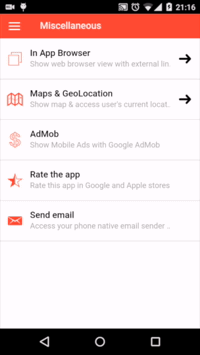
# 

# 

# 

# Email Composer

The plugin provides access to the standard interface that manages the editing and sending an email message. You can use this view controller to display a standard email view inside your application and populate the fields of that view with initial values, such as the subject, email recipients, body text, and attachments. The user can edit the initial contents you specify and choose to send the email or cancel the operation.



You can add this functionality to your app by adding the following plugin:

**cordova plugin add** [**https://github.com/katzer/cordova-plugin-email-composer.git**](https://github.com/katzer/cordova-plugin-email-composer.git)

The Email service is only available on devices capable which are able to send emails. E.g. which have configured an email account and have installed an email app. You can use this function to hide email functionality from users who will be unable to use it.

|  |
| --- |
| cordova.plugins.email.isAvailable(  function (isAvailable) {  // alert('Service is not available') unless isAvailable;  } ); |

A pre-filled email draft can be opened through the email.open or email.openDraft interface.

|  |
| --- |
| cordova.plugins.email.open({  to: Array, // email addresses for TO field  cc: Array, // email addresses for CC field  bcc: Array, // email addresses for BCC field  attachments: Array, // file paths or base64 data streams  subject: String, // subject of the email  body: String, // email body (for HTML, set isHtml to true)  isHtml: Boolean, // indicats if the body is HTML or plain text }, callback, scope); |

Example

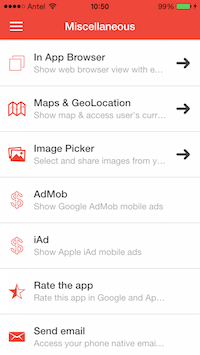
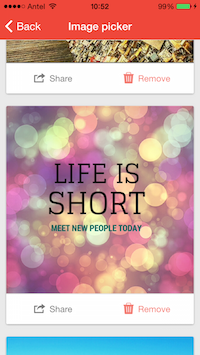
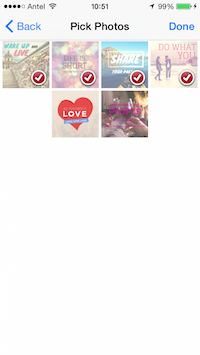
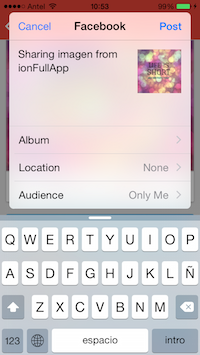
|  |
| --- |
| cordova.plugins.email.open({  to: 'max@mustermann.de',  cc: 'erika@mustermann.de',  bcc: ['john@doe.com', 'jane@doe.com'],  subject: 'Greetings',  body: 'How are you? Nice greetings from Leipzig' }); |

You can find more examples and the full documentation of this plugin here: <https://github.com/katzer/cordova-plugin-email-composer>

# Image picker

(added in version 1.1)

Many times you need to let your users select images from their phones gallery.

We are using image picker plugin which lets you select multiple images from your phone. It has an incredibly simple API allowing for a maximum number of pictures and automatic resizing/quality changes too.

**To install this plugin** just type the following command on your terminal window inside your project’s path:

**cordova plugin add com.synconset.imagepicker**

You can check the full documentation here:<https://github.com/wymsee/cordova-imagePicker>

# Social Sharing

Adding social media to your PhoneGap app is always a good idea – you can draw attention to your app by letting your users do the promotional work for you. We use a plugin that allows you to use the native sharing window of your mobile device.

* Works on Android, version 2.3.3 and higher (probably 2.2 as well).
* Works on iOS 6 and up.
* Share text, a link, a images (or other files like pdf or ics). Subject is also supported, when the receiving app supports it.
* Supports sharing files from the internet, the local filesystem, or from the www folder.
* You can skip the sharing dialog and directly share to Twitter, Facebook, or other apps.
* Compatible with [Cordova Plugman](https://github.com/apache/cordova-plugman).
* Officially supported by [PhoneGap Build](https://build.phonegap.com/plugins).

You can add this functionality to your app by adding the following plugin:

**cordova plugin add https://github.com/EddyVerbruggen/SocialSharing-PhoneGap-Plugin.git**

You can check the full documentation of this plugin here: <https://github.com/EddyVerbruggen/SocialSharing-PhoneGap-Plugin>

You can share text, a subject (in case the user selects the email application), (any type and location of) file (like an image), and a link. However, what exactly gets shared, depends on the application the user chooses to complete the action.

To share a link just use this code:

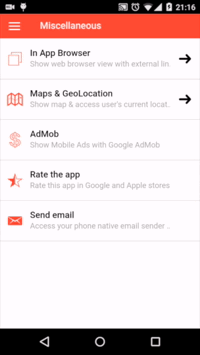
|  |
| --- |
| $scope.sharePost = function(link){  window.plugins.socialsharing.share('Check this post here: ', null, null, link);  } |

You can find more examples of how to use the share sheet here: <https://github.com/EddyVerbruggen/SocialSharing-PhoneGap-Plugin#using-the-share-sheet>

# AdMob

Show Mobile Ads with Google AdMob.

Before you can serve any AdMob ads you must sign up for an AdMob account at [www.admob.com](http://www.admob.com/). AdMob does not charge a fee for creating an account or for serving up AdMob ads (be aware that some AdMob plugins vendors do charge a fee for the use of their plugin). If you have an AdMob account you must create a set of Ad Unit IDs that identify your ad impressions and provide these IDs as part of the AdMob API initialization sequence within your app.



**IMPORTANT**: each application should have its own set of Ad Unit IDs! If you do not yet have an app in an app store, you can use the "manual" method to identify your app for the purpose of obtaining Ad Unit IDs.

**To install this plugin** just type the following command on your terminal window inside your project’s path:

**cordova plugin add com.google.cordova.admob**

You can check the full documentation here: <https://github.com/floatinghotpot/cordova-admob-pro>

**Note:** If use this plugin in commercial project and need email/skype support, please [buy a license](http://rjfun.github.io/), you will be supported with high priority.

You can follow this tutorial to see how to add banner ad show up on every screen at the bottom: <https://blog.nraboy.com/2014/06/using-admob-ionicframework/>

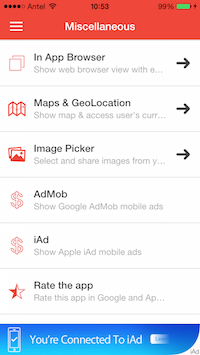
# iAd

(added in version 1.1)

Show Mobile Ads with Apple's advertising platform.

You can check Apple’s documentation here: <http://advertising.apple.com/get-started/>

**Notice**: iAd interstitial Ad only supports iPad.



**To install this plugin** just type the following command on your terminal window inside your project’s path:

**cordova plugin add com.rjfun.cordova.iad**

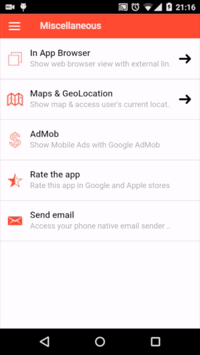
You can check the full documentation here: <https://github.com/floatinghotpot/cordova-iad-pro>

**Note:** If use this plugin in commercial project and need email/skype support, please [buy a license](http://rjfun.github.io/), you will be supported with high priority.

# 

# App Rate

This app provides the “rate this app” functionality. We use a the AppRate Cordova/Phonegap plugin. You can find the full documentation of this plugin here: <https://github.com/pushandplay/cordova-plugin-apprate>



**To install this plugin** just type the following command on your terminal window inside your project’s path:

**cordova plugins add org.pushandplay.cordova.apprate**

You can use this code to prompt the rate dialog:

|  |
| --- |
| $scope.rateApp = function(){  if(ionic.Platform.isIOS()){  AppRate.preferences.storeAppURL.ios = '<my\_app\_id>';  AppRate.promptForRating(true);  }else if(ionic.Platform.isAndroid()){  AppRate.preferences.storeAppURL.android = 'market://details?id=<package\_name>';  AppRate.promptForRating(true);  }  } |

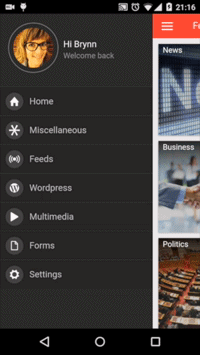
For more examples and customization please check [this documentation.](https://github.com/pushandplay/cordova-plugin-apprate)

# Maps and GeoLocation

In order to access the user current location you can use [the cordova geolocation plugin.](https://github.com/apache/cordova-plugin-geolocation)

## Cordova geolocation plugin

This plugin provides information about the device's location, such as latitude and longitude. Common sources of location information include Global Positioning System (GPS) and location inferred from network signals such as IP address, RFID, WiFi and Bluetooth MAC addresses, and GSM/CDMA cell IDs. There is no guarantee that the API returns the device's actual location.



**WARNING**: Collection and use of geolocation data raises important privacy issues. Your app's privacy policy should discuss how the app uses geolocation data, whether it is shared with any other parties, and the level of precision of the data (for example, coarse, fine, ZIP code level, etc.). Geolocation data is generally considered sensitive because it can reveal user's whereabouts and, if stored, the history of their travels. Therefore, in addition to the app's privacy policy, you should strongly consider providing a just-in-time notice before the app accesses geolocation data (if the device operating system doesn't do so already). That notice should provide the same information noted above, as well as obtaining the user's permission (e.g., by presenting choices for OK and No Thanks). For more information, please see the Privacy Guide.

This plugin defines a global navigator.geolocation object (for platforms where it is otherwise missing). Although the object is in the global scope, features provided by this plugin are not available until after the deviceready event.

**To install this plugin** just type the following command on your terminal window inside your project’s path:

**cordova plugin add org.apache.cordova.geolocation**

You can find the full documentation of this plugin here: <https://github.com/apache/cordova-plugin-geolocation/>

|  |
| --- |
| //with this function you can get the user's current location  navigator.geolocation.getCurrentPosition(function(position) {  var pos = new google.maps.LatLng(position.coords.latitude, position.coords.longitude);  $scope.current\_position = {lat: pos.k,lng: pos.D};  }); |

## GoogleMap AngularJS Directive

In order to show a map we are going to use an angular directive called ***ngMap.*** Please check the documentation here: <http://ngmap.github.io/>. It is very simple to use.

To install this directive use [bower](http://bower.io/)

**bower install ngmap**

Include ***ng-map.min.js*** as well as google maps.

|  |
| --- |
| <script src="//maps.google.com/maps/api/js"></script>  <script src="//rawgit.com/allenhwkim/angularjs-google-maps/master/build/scripts/ng-map.min.js">  </script> |

Name angular app as ***ngMap,*** or add it as a dependency:

|  |
| --- |
| var myApp = angular.module('myApp', ['ngMap']); |

# 

# 

# InAppBrowser

The InAppBrowser window behaves like a standard web browser, and can't access Cordova APIs. For this reason, the InAppBrowser is recommended if you need to load third-party (untrusted) content, instead of loading that into the main Cordova webview. The InAppBrowser is not subject to the whitelist, nor is opening links in the system browser.

The InAppBrowser provides by default its own GUI controls for the user (back, forward, done).

This plugin provides a web browser view that displays when calling ***window.open()***

|  |
| --- |
| var ref = window.open('http://apache.org', '\_blank', 'location=yes'); |

**To install this plugin** just type the following command on your terminal window inside your project’s path:

**ionic plugin add org.apache.cordova.inappbrowser**

You can find the full documentation of this plugin here: <https://github.com/apache/cordova-plugin-inappbrowser/blob/master/doc/index.md>

**How to disable URL and Navigation Bar?**

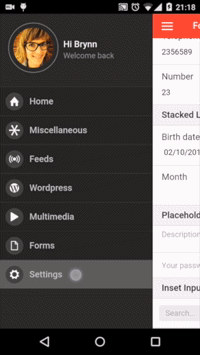
Please check the answer here: <http://stackoverflow.com/a/15597029/1330740>

# 

# 

# App Settings

In this section you can see examples of how to use the common controllers needed on a regular Settings view such as: radio buttons, toggles, checkboxes, ranges, and a Logout button with an action sheet.



## Action Sheet

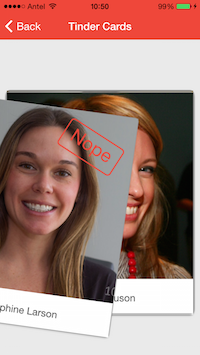
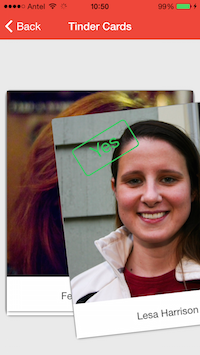
The Action Sheet is a slide-up pane that lets the user choose from a set of options. Dangerous options are highlighted in red and made obvious.

You can find the documentation and an example of how to use it here: <http://ionicframework.com/docs/api/service/$ionicActionSheet/>

Tinder cards

(added in version 1.1)

Swipeable card based layout for Ionic and Angular as seen in apps like [Tinder](http://www.gotinder.com/).



We are using this library <https://github.com/driftyco/ionic-ion-tinder-cards>.

**To install this library** just type the following command on your terminal window inside your project’s path:

**bower install ionic-contrib-tinder-cards**

Then include `ionic.tdcards.js` and `ionic.tdcards.css` after the rest of your Ionic and Angular includes in your www/index.html. Add `ionic.contrib.ui.tinderCards` as a module dependency of your app.

# 

# 

# Profile

(added in version 1.1)

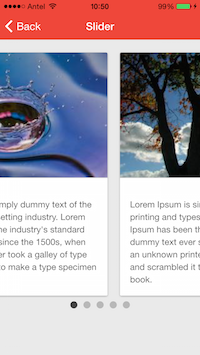
If you press on the menu item with the face of the user you will see this profile view.

# 

# Slider

(added in version 1.1)

Many times we need a slider in order to show some content/pictures. We included and example to help you building it.

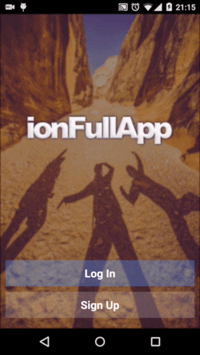


# 

# Login/Signup

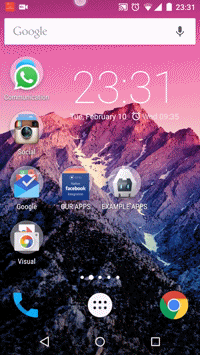
Features:

* Login tabbed options (login with email or phone)
  + Let your users choose between logging in using their email and password or with their phone and a PIN
* Form validation
  + PIN custom validation (using AJAX)
  + Submit button disabled if form is invalid
* Show/Hide password functionality (why is this important: <http://www.lukew.com/ff/entry.asp?1941>)
  + Masking passwords makes complex input even harder while doing little to improve security -especially on mobile where there’s a visible touch keyboard directly below every input field that highlights each key as you press it. These bits of feedback show the characters in your password at a larger size than most input fields. So in reality, masking characters isn’t really hiding your password from prying eyes. And if you do suspect an eavesdropper is peering at your screen, just move your mobile device out of their line of sight!
* Forgot password
* Complete flow
  + Walkthrough
  + Login
    - Forgot password
    - Signup



# Push Notifications

Android devices receive push notifications through the [Google Cloud Messaging (GCM)](http://developer.android.com/google/gcm/) service, whereas iOS devices receive them from the Apple Push Notifications (APN) Service. Though there are differences in the two services in terms of the size of the payload that can be sent, certificates required etc, both act as a store and forward type of service where they receive a message from a 3rd party, identify the recipient and pass it along. Upon receipt, your application that has registered to receive them can examine the returned contents and act accordingly. The way the notifications are received (by sound, alert etc) is a combination of the options set in the application code upon registration as well as the user’s device settings for notifications.



## Android Pre-requisites

* Google Cloud Messaging Project ID
* Google Cloud Messaging API Key for above Project ID (need for server)

## iOS Pre-requisites

* App ID configured for Push Notifications on Apple Developer Portal
* SSL Certificate and private key (need for server)

## Register the application with the Google Cloud Messaging service

**Architectural Overview**

A GCM implementation includes a Google-provided connection server, a 3rd-party app server that interacts with the connection server, and a GCM-enabled client app running on an Android device:



Figure 1. GCM Architecture.

**Explanation of Workflow**

This is how these components interact:

* Google-provided GCM Connection Servers take messages from a 3rd-party application server and send these messages to a GCM-enabled Android application (the "client app") running on a device.
* The 3rd-Party Application Server is a component that you implement to work with your chosen GCM connection server(s). App servers send messages to a GCM connection server; the connection server enqueues and stores the message, and then sends it to the device when the device is online.
* The Client App is a GCM-enabled Android application running on a device. To receive GCM messages, this app must register with GCM and get a registration ID.

**Follow these steps**

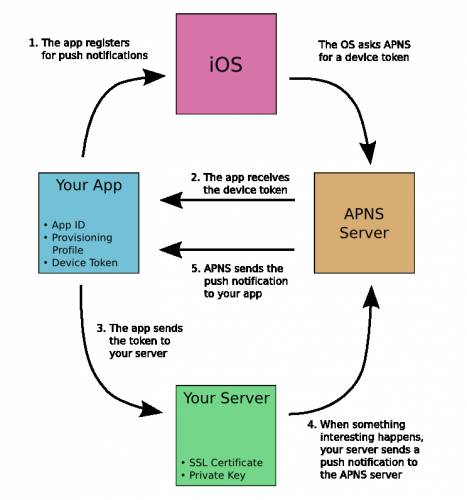
In this step we will register our application with the [Google Cloud Messaging service](http://developer.android.com/google/gcm/index.html) to receive push notifications.

1. Start by creating an account and project with the GCM service [here](http://developer.android.com/google/gcm/gs.html). You will need to:
   1. Create a Google API project
   2. Enable the GCM Service
   3. Obtain an API Key

## Register the application with the Apple APN service

**APN Workflow Overview**

(credit to [Ray Wenderlich](http://www.raywenderlich.com/))



**Explanation of Workflow**

* An app enables push notifications. The user has to confirm that he wishes to receive these notifications.
* The app receives a “device token”. You can think of the device token as the address that push notifications will be sent to.
* The app sends the device token to your server.
* When something of interest to your app happens, the server sends a push notification to the Apple Push Notification Service, or APNS for short.
* APNS sends the push notification to the user’s device.

**Follow these steps**

In this step we will register our application with the Apple APN to receive push notifications.

1. Start by setting Provisioning Profiles and Certificates (follow this tutorial: <http://www.raywenderlich.com/32960/apple-push-notification-services-in-ios-6-tutorial-part-1>). You will need to:
   1. Generating the Certificate Signing Request (CSR)
   2. Making the App ID and SSL Certificate
   3. Making a PEM File
   4. Making the Provisioning Profile

## Configure Push Server

With the configuration done for Google and Apple in the previous steps we can now start working on our 3rd party application PUSH server.

We are going to use a simple node.js application to perform this actions (<https://github.com/Smile-SA/node-pushserver>).

Follow these steps:

* Run these commands:
  + **git clone git://github.com/Smile-SA/node-pushserver.git**
  + **cd node-pushserver**
  + **npm install -g**
* Change the configuration file (***config.json***)
  + Set Google and Apple credentials from previous steps
  + Also add your PEM pass phrase in the ***config.json***

|  |
| --- |
| "gcm": {  "apiKey": "YOUR\_KEY\_HERE"  },  "apn": {  "connection": {  "gateway": "gateway.sandbox.push.apple.com",  "cert": "apple\_assets/ionFullAppPushCert.pem",  "key": "apple\_assets/ionFullAppPushKey.pem",  "passphrase":"YOUR\_PASS\_PHRASE\_HERE"  },  "feedback": {  "address": "feedback.sandbox.push.apple.com",  "cert": "apple\_assets/ionFullAppPushCert.pem",  "key": "apple\_assets/ionFullAppPushKey.pem",  "passphrase":"YOUR\_PASS\_PHRASE\_HERE",  "interval": 43200,  "batchFeedback": true  } |

* Start the push server (the server is going to be running at ***localhost:8000*** but you can change the port in the ***config.json***). Run this command
  + **pushserver -c "$HOME"/node-pushserver/config.json**

## Add Push Notifications to the app

After the Google and Apple configurations and setting up our push server we need to configure our app so it can receive push notifications.

1. Install cordova plugin
   1. **cordova plugin add** [**https://github.com/phonegap-build/PushPlugin.git**](https://github.com/phonegap-build/PushPlugin.git)
2. Install ngCordova
   1. bower install ngCordova --save
   2. add ‘ngCordova’ to you app module
      1. angular.module('your\_app\_name', [......., 'ngCordova'])
   3. add ng-cordova.min.js to you index.html
3. Change bundle identifier in your ionic ***config.xml*** (must match with the one used when configuring Apple APN)
   1. (something like this: ***<widget id="com.startapplabs.ionFullApp" ...>***)

# 

# 

# 

# Splash screen and app icon

An app icon and splash screen (launch image) are important parts of any app, yet making them used to be incredibly tedious. You needed numerous icons for iOS and Android, and then you had to deal with splash screens and all their different sizes.

To save you the stress of dealing with all that, ionic enables you to generate app icons and splash screens via the Ionic CLI.

With the ionic CLI, all you need is a resource directory and two images. These images can be .png files, Photoshop .psd files, or Illustrator .ai files, named ***icon.png*** and ***splash.png***. With the images in a resources directory, .***/resources***, the **ionic resources** command will generate the icons and splash screen images for each platform setup in the project, sending them to Ionic's image resizing and cropping server, so you don't have to install extra dependencies.

If you only need to update one of the resources, or you only want to generate icons and not both, the ionic resources command has two flags that allow you to target each asset, instead of generating both.

**ionic resources --icon**  
**ionic resources --splash**

## Image Sizes

In order to be able to crop and resize images to fit the various needs of each platform, icons and splash screen images should meet a minimum size requirement.

The icon image's minimum dimensions should be ***192x192 px*** and should have no rounded corners. Note that each platform will apply its own mask and effects to the icons. For example, iOS will automatically apply its custom rounded corners, so the source file should not already come with rounded corners.

Splash screen dimensions vary for each platform, device, and orientation, so a square source image is required to generate each of the various screen sizes. The source image's minimum dimensions should be ***2208x2208 px***, and the artwork should be centered within the square, because each generated image will be center cropped into landscape and portrait images. The splash screen's artwork should roughly fit within a center square (***1200x1200 px***). This [template](http://code.ionicframework.com/resources/splash.psd) provides the recommended size and guidelines about artwork's safe zone.

If you need any further assistance with this, please go to <http://ionicframework.com/blog/automating-icons-and-splash-screens/> for more information.

# 

# 

# 

# Let’s customize some styles

(added in version 1.1)

The app uses [SASS](http://sass-lang.com/), which basically is CSS with super powers :).

We built the app main styles based on variables. The file scss/ion-full-app.scss is where this variables live. For example you will find something like this:

// App colors

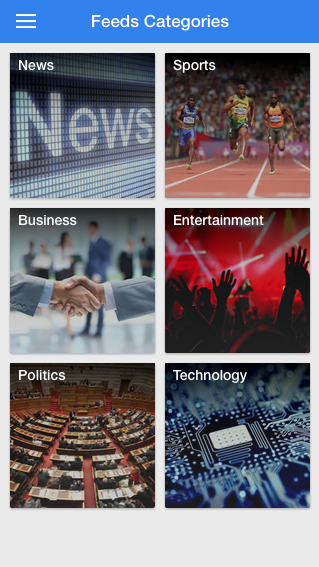
$top-bar-bg: $assertive;

$top-bar-color: #FFFFFF;

$loading-color: #FFFFFF;

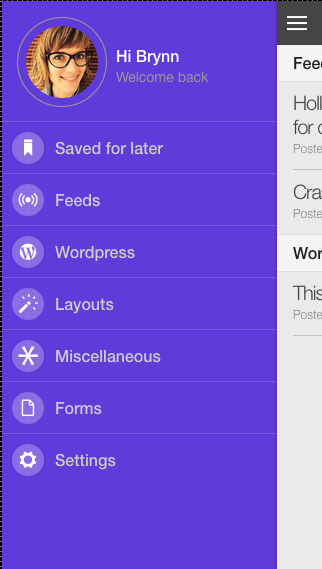
So if you want to change the top bar color you just need to change $top-bar-bg value.

Want to change it to blue?



Want to change the menu background color to blue?

Just change $main-menu-bg value to the color you want and magic will happen.



# 

# 

# Prepare ionic app for production

In this post i will explain how to prepare your project for production. We are going to:

1. **(cordova hook)** [**Lint your javascript**](#h.jcv6pkpxzwzk) (this step is needed because before minifying and obfuscating we need to ensure that there are no javascript errors)
2. **(gulp task)** [**Load html templates as javascript angular templates**](https://docs.google.com/document/d/1yUDhFlL1G6Gfz178Hys_7fxz85dFRBqehhOOrqxLj3k/edit#heading=h.i7bpjitg90r5) (this adds more obfuscation as your html won’t be visible for others)
3. **(gulp task)** [**Enable angular strict dependency injection**](https://docs.google.com/document/d/1yUDhFlL1G6Gfz178Hys_7fxz85dFRBqehhOOrqxLj3k/edit#heading=h.jr2cjwzfg02r) (this step is needed because before obfuscating we need to ensure angular dependency injection won’t brake)
4. **(gulp task)** [**Concat js and css files**](https://docs.google.com/document/d/1yUDhFlL1G6Gfz178Hys_7fxz85dFRBqehhOOrqxLj3k/edit#heading=h.gyyrmq4nu9za) (this hides more details about your code)
5. **(cordova hook)** And finally [**uglify, minify and obfuscate your code**](https://docs.google.com/document/d/1yUDhFlL1G6Gfz178Hys_7fxz85dFRBqehhOOrqxLj3k/edit#heading=h.pgu0x1hhbm1q).

For all the previous task we are going to use a mixture between ***gulp tasks*** and ***cordova hooks***. Gulp tasks will be “running” all the time after you run **ionic serve**. Cordova hooks will run each time you build or run your project with **ionic build ios [android]** or **ionic run android [ios]**

## Lint your javascript

1. For this procedure we are going to use these npm packages. Run these commands to install them.
   1. **npm install jshint --save-dev**
   2. **npm install async --save-dev**
2. Copy the following cordova hooks
   1. In ***before\_prepare*** folder: <https://gist.github.com/agustinhaller/5e489e5419e43b11d7b7>
   2. Give execution permissions to all of them, run
      1. **chmod +x file\_name**
3. We are ready to test javascript linting, run this command and you will see that jshint is working
   1. **ionic build ios [android]**

# 

# 

## HTML templates transformation

Part of the obfuscation is transforming all the html templates in angular js templates (compressed inside a javascript file)

1. For this we are going to use ***gulp-angular-templatecache***. Run this command to install the npm package
   1. **npm install gulp-angular-templatecache --save-dev**
2. Add the following lines to ***gulpfile.js***

|  |
| --- |
| var templateCache = require('gulp-angular-templatecache'); |

|  |
| --- |
| var paths = {  templatecache: ['./www/templates/\*\*/\*.html']  }; |

|  |
| --- |
| gulp.task('templatecache', function (done) {  gulp.src('./www/templates/\*\*/\*.html')  .pipe(templateCache({standalone:true}))  .pipe(gulp.dest('./www/js'))  .on('end', done);  }); |

|  |
| --- |
| gulp.task('default', ['sass', 'templatecache']); |

|  |
| --- |
| gulp.task('watch', function() {  gulp.watch(paths.sass, ['sass']);  gulp.watch(paths.templatecache, ['templatecache']);  }); |

1. Also we need to add this to ***ionic.project***

|  |
| --- |
| "gulpStartupTasks": [  "sass",  "templatecache",  "watch"  ] |

1. Add ***templates*** module in your ***app.js***

|  |
| --- |
| angular.module('starter', ['ionic', 'starter.controllers', **'templates'**]) |

1. Add reference to ***templates.js*** file in your ***index.html***

|  |
| --- |
| <script src="js/templates.js"></script> |

1. Run
   1. **ionic serve**
   2. This will add a ***templates.js*** file inside ***www/js*** with all the html templates as angular js templates
   3. **Note:** remember to update the angular ***templateUrl*** (for example in ***app.js*** and your ***directives***). By these I mean matching each ***templateUrl*** to the name in ***templates.js*** file
      1. Before it was

|  |
| --- |
| .state('intro', {  url: "/",  templateUrl: "templates/intro.html",  controller: 'IntroCtrl'  }); |

* + 1. Now in ***templates.js*** is

|  |
| --- |
| $templateCache.put("intro.html", …. |

* + 1. So now we need to change it to

|  |
| --- |
| .state('intro', {  url: "/",  templateUrl: "intro.html",  controller: 'IntroCtrl'  }); |

# 

## Enable ng-strict-di

Before minifying we need to enable angular strict dependency injection (see: <https://github.com/olov/ng-annotate#highly-recommended-enable-ng-strict-di-in-your-minified-builds>). This will save us from breaking angular dependency injection when minifying.

1. For that we are going to use ***gulp-ng-annotate***. Run this command to install the npm package
   1. **npm install gulp-ng-annotate --save-dev**
2. Add the following lines to ***gulpfile.js***

|  |
| --- |
| var ngAnnotate = require('gulp-ng-annotate'); |

|  |
| --- |
| var paths = {  ng\_annotate: ['./www/js/\*.js']  }; |

|  |
| --- |
| gulp.task('ng\_annotate', function (done) {  gulp.src('./www/js/\*.js')  .pipe(ngAnnotate({single\_quotes: true}))  .pipe(gulp.dest('./www/dist/dist\_js/app'))  .on('end', done);  }); |

|  |
| --- |
| gulp.task('default', ['sass', 'templatecache', 'ng\_annotate']); |

|  |
| --- |
| gulp.task('watch', function() {  gulp.watch(paths.sass, ['sass']);  gulp.watch(paths.templatecache, ['templatecache']);  gulp.watch(paths.ng\_annotate, ['ng\_annotate']);  }); |

1. Also we need to add this to ***ionic.project***

|  |
| --- |
| "gulpStartupTasks": [  "sass",  "templatecache",  "ng\_annotate",  "watch"  ] |

1. Change the path of our angular js files in the ***index.html*** as follows

|  |
| --- |
| <script src="**dist/dist\_js/**app/app.js"></script> |

1. Add ***ng-strict-di*** directive in ***ng-app*** tag (inside ***index.html***)

|  |
| --- |
| <body ng-app="your-app" ng-strict-di> |

1. Run
   1. **ionic serve**

This will create a ***dist*** folder inside ***www*** folder with all our js files with strict dependency injection fixed

## Concatenate js and css files

1. For the concatenation of files we are going to use ***gulp-useref***. Run this command to install the npm package
   1. **npm install gulp-useref --save-dev**
2. Add the following lines to ***gulpfile.js***

|  |
| --- |
| var useref = require('gulp-useref'); |

|  |
| --- |
| var paths = {  useref: ['./www/\*.html']  }; |

|  |
| --- |
| gulp.task('useref', function (done) {  var assets = useref.assets();  gulp.src('./www/\*.html')  .pipe(assets)  .pipe(assets.restore())  .pipe(useref())  .pipe(gulp.dest('./www/dist'))  .on('end', done);  }); |

|  |
| --- |
| gulp.task('default', ['sass', 'templatecache', 'ng\_annotate', 'useref']); |

|  |
| --- |
| gulp.task('watch', function() {  gulp.watch(paths.sass, ['sass']);  gulp.watch(paths.templatecache, ['templatecache']);  gulp.watch(paths.ng\_annotate, ['ng\_annotate']);  gulp.watch(paths.useref, ['useref']);  }); |

1. Also we need to add this to ***ionic.project***

|  |
| --- |
| "gulpStartupTasks": [  "sass",  "templatecache",  "ng\_annotate",  "useref",  "watch"  ] |

1. Add the following to ***index.html*** to bundle ***css*** and ***js*** as you want

|  |
| --- |
| <!-- build:css dist\_css/styles.css -->  <link href="css/ionic.app.css" rel="stylesheet">  <!-- endbuild --> |

|  |
| --- |
| <!-- build:js dist\_js/app.js -->  <script src="dist/dist\_js/app/app.js"></script>  <script src="dist/dist\_js/app/controllers.js"></script>  <!-- endbuild --> |

* 1. **Note**: if you require an external script/file don’t include it inside a bundle. For example:

|  |
| --- |
| <script src="http://maps.google.com/maps/api/js"></script> |

1. Run
   1. **ionic serve**
   2. This will create the bundled files inside you ***www/dist*** folder also a new ***index.html*** with the new path to the bundled files.

# 

## Uglify, minify and obfuscate

1. For this procedure we are going to use these npm packages. Run these commands to install them.
   1. **npm install cordova-uglify --save-dev**
   2. **npm instal mv --save-dev**
2. Copy the following cordova hooks
   1. In ***after\_prepare*** folder: <https://gist.github.com/agustinhaller/426351993c70a0329ad0>
   2. Give execution permissions to all of them, run
      1. **chmod +x file\_name**
3. We are ready to get the obfuscated/minified/compressed apk, run this command and you will see your production ready app
   1. **ionic build ios [android]**

# 

# 

# 

# Common issues

## Errors when building for iOS

If you build for ios and find yourself having cordova plugin errors (like ***iOS unable to find plugins***) do the following:

1. Delete ***./plugins/ios.json***
2. Then delete ***./platforms/ios***
3. Run
   1. **ionic platform add ios**
   2. **ionic build ios**

Deleting ***ios.json*** helps to force it to recompile the plugins.

## Errors with Wordpress JSON API

If you get this error when doing the http request to you wordpress site stay calm, I will help you solving it.

XMLHttpRequest cannot load [http://you\_web\_page?json=get\_recent\_posts&page=1](http://japanize.sakura.ne.jp/website-test/?json=get_recent_posts&page=1). No 'Access-Control-Allow-Origin' header is present on the requested

It means that you server doesn't have CORS (cross origin resource sharing). You can solve this problem by enabling CORS. For example in a php server side you could fix this adding this header: <?php header('Access-Control-Allow-Origin: \*'); ?>

There is another way to fix this and is using JSONP. "JSONP or "JSON with padding" is a communication technique used in JavaScript programs running in web browsers to request data from a server in a different domain, something prohibited by typical web browsers because of the same origin policy."

You can try to change the following code when making an http request to your Wordpress site.

Replace:

$http.get(rootURL + '&page='+$scope.page)

for:

$http.jsonp( rootURL + '&page='+$scope.page +"&callback=JSON\_CALLBACK").

success(function(data) {

//the same existing code

}).

Note that the callback function parameters changed from (data, status, headers, config) to (data). If you choose this way, you will need to make this change in all the requests to your wordpress site.

## Errors when running npm install on Windows

Please make sure you have node version 0.10.36

You should solve your errors by reading this links:

* <http://stackoverflow.com/questions/21365714/nodejs-error-installing-with-npm/21366601#21366601>
* <http://forum.ionicframework.com/t/ionic-setup-sass-error/17843>

### Ionic setup sass - error

Please check this topic: <http://forum.ionicframework.com/t/ionic-setup-sass-error/17843>

# 

# 

# FAQS

## How do I start?

1. [Install Ionic](#h.p5y9ryb8vw5b)
2. [Install the app](#h.do8bppcgy3mr)
3. [Build the app](#h.o9gpdagkqd13)
4. [Configurations](#h.hrk1kicys1bl)

Note: Each time you make a change in your code you need to [build](#h.o9gpdagkqd13) the app again in order to see those changes reflected in your ios or android app, or if you just want to see it in your browser you can just run: **ionic serve**

## What program do you use to edit the code?

Our preferred editors are:

1. Atom <https://atom.io/>
2. Sublime <http://www.sublimetext.com/>

## Can I use Phonegap Build?

Yes, you can use PhoneGap Build in order to build this app. Ionic is a front-end framework built on top of Cordova so you can use PhoneGap Build in the same way you would use it with other phonegap/cordova application.

Adobe PhoneGap provides a way for users to create mobile applications using technologies such as HTML, CSS, and Javascript. <https://build.phonegap.com>

You can find the whole documentation for the config.xml here: <http://docs.phonegap.com/en/3.5.0/config_ref_index.md.html>

## RTL support

The HTML5 attribute dir=auto sets the directionality of the element according to the first characters with strong directionality.

The only way to make an element’s directionality depend on its own content in this sense is to set the dir attribute to the value auto on the element itself. You cannot make this attribute inherited.

## How to change the status bar color

Follow this tutorial: <http://learn.ionicframework.com/formulas/customizing-the-status-bar/>

## Disable login

Go to the file www/js/app.js and change the last line of code from:

$urlRouterProvider.otherwise('/');

to:

$urlRouterProvider.otherwise('/app/feeds-categories');

With this you are saying that if none of the above states are matched, go to the feeds-categories as the fallback.

And also delete from the app.js the following states:

* walkthrough
* login
* signup
* forgot-password

You also have to remove the option “Home” from the menu. Go to www/templates/side-menu.html and remove this:

<ion-item class="item-icon-left" nav-clear menu-close href="#/app/">  
 <i class="icon ion-home" />  
 <h2 class="menu-text">Home</h2>  
</ion-item>

## What is dist/ folder? (removed on version 1.2)

www/dist/ folder is created and updated automatically when you run **ionic serve** so you should NOT edit files here.

We recommend you to include www/dist/ folder in your .gitignore

“dist” comes from “distribution”. All this folder hierarchy is created to manage minification, concatenation and obfuscation when building the app for production. For more information about what we do to prepare the app for production please read our tutorial on how to [Prepare ionic app for production](#h.cqy3u5qe9whd).

## Remote Debugging on Android with Chrome

<https://developer.chrome.com/devtools/docs/remote-debugging>

## Remote Debugging on iOS with Safari

<http://www.jasenlew.com/2014/07/28/web-inspect-an-ionic-app-on-your-iphone-ios-6/>

## Older versions of Android devices (4.0-4.3)

Older versions of Android devices (4.0-4.3) use Android’s default browser, which has significantly less performance and standards compliance than modern Chrome.

Using Crosswalk gives you a specific and more performant version of Chrome to use on all Android devices, in order to reduce fluctuations and fragmentation among devices.

Crosswalk is an open source project that allows you to specify a version of Chrome to use as your web browser in Android.

The compiled app will have your code hosted inside of this Chrome webview.

Please read more here <http://blog.ionic.io/crosswalk-comes-to-ionic/>

# 

# 

# Changelog

**Version 1.2- released 03 June 2015**

## Misc  
- Improved and ease build process  
- Update Ionic Version to v.1.0.0

**Version 1.1- released 27 March 2015**

## New Features

- Bookmark functionality to feeds and wordpress posts

- Tinder cards

- Profile page

- Image picker

- iAd - Apple mobile ad service

## Others

- Update Ionic version to 1.0.0-rc.1

- Added DOCUMENTATION.md with the documentation link

- Refactoring of some controlleres and services

- Added config.js with the app configurations

- Changed href for ui-sref

- Created Layouts section

- Remove ionicons dependency as they are now part of ionic 1.0.0-rc.1

## Bugs

- Fixed push notifications bug for iOS

- iframe content scrolling unstable in ionic, waiting for a fix until enabling it again

## Wordpress:

- Support for embeding audio posts

- Changed WP blog source

- Improves to avoid CORS issues

- Removed fixed image on posts

# Support

Once again, you very much for purchasing **ionFullApp**. I would be glad to help you if you have any questions relating this app. No guarantees, but I will do my best to assist with questions directly related to the app, however please note that code support is completely voluntary for CodeCanyon authors. Therefore before you contact me, please consider finding an answer to your question in this documentation.

If you like the app, please, please, please give us your feedback in Envato. You can [find here](https://help.market.envato.com/hc/en-us/articles/203269490-How-Do-I-Give-Feedback-On-Items-) the steps to do it :)