Music Band Ionic 1.11 Documentation

Quick start Guide - Installation and user support

Created: Jun 13, 2015

Updated: Jul 28, 2016

By: AppSeed, http://appseed.io/

Table of contents:

Prerequisites

Cordova and Ionic

This is a [Cordova](#) and [Ionic](#) based application, so the [Cordova](#) and [Ionic](#) should be installed in your computer. Since Music Band Ionic, targets iPhone and Android mobile devices, your environment should be properly configured and the corresponding SDK should be installed. If not you will be still able to run the application into a Browser.

Please check the "[Get started with Cordova](#)" and "[Getting Started with Ionic](#)" sections in the

official Cordova and Ionic sites respectively.

Ensure first that [NodeJS](#) is installed in your computer

Tools

This project is based on the popular "[Ionic Framework Generator](#)" that boosts the overall development process by integrating a couple of very popular automation tools like [Grunt](#) and [Bower](#).

Install these tools by following the instructions in their corresponding web pages:
1. [Install Bower](#)
2. [Getting started with Grunt - Install the CLI](#)
3. [Getting started with Yeoman](#)

Finally install the yeoman generator via:

```
$ npm install -g generator-ionic
```
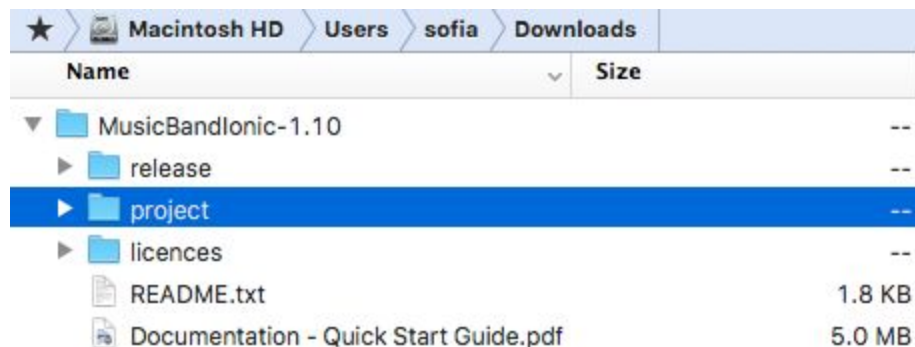
Run for the first time

In the screen captures that follow, we will demonstrate the process of preparing your environment and running the project for the first time.

Download and extract

Download the provided .zip file and extract it, you will see something similar to what is shown in the next screen:

The highlighted is used for all the instructions in this document.

Install libraries

Open a terminal window and navigate to the directory where the "music-band" folder is located.

Install NodeJS dependences:

```
$ npm install
```

Post installation

There is a post installation process under which required Cordova plugins and Javascript dependencies are installed. To simplify this process two scripts are already prepared for both platforms: Linux/MacOS and Windows

Linux/MacOX
Install all the required plugins and Javascript dependencies:

```
$ ./install.sh
```

Note: If you don't have iOS, you should comment:

```
grunt platform:add:ios
```
and comment out:

```
grunt platform:add:android
```

Windows Users
Similarly, Windows users should execute:

```
$ install.bat
```

Plugins

Build your project for the first time. This will create the [www] folder which is the actual cordova directory and where the plugins will be installed.

```
$ grunt build
```

Follow the same process as with "Libraries" and install the required plug ins by using the commands that follow:

```
$ cordova plugin add {plugin name or url}
```

eg:

```
$ cordova plugin add cordova-plugin-inappbrowser
```
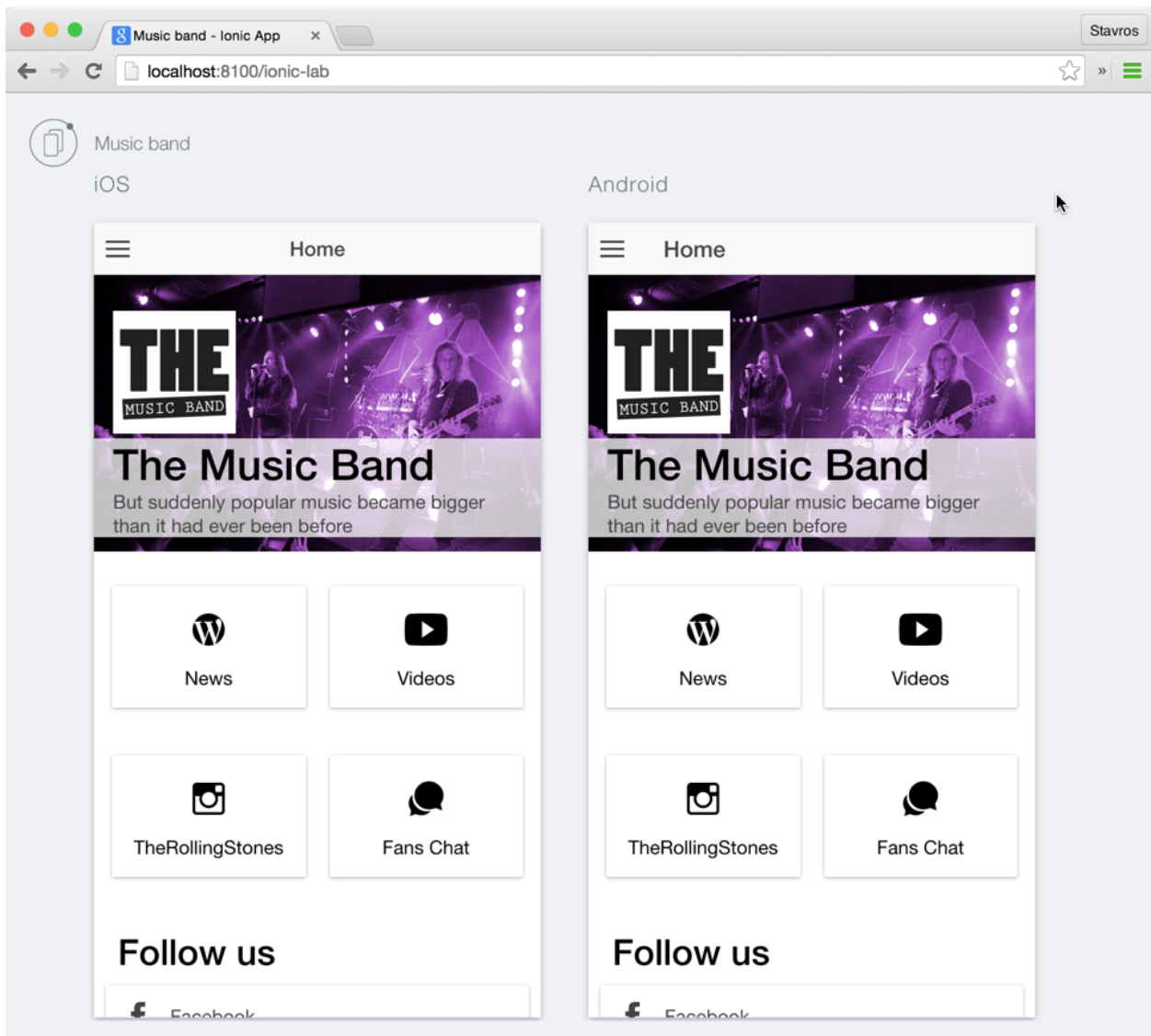
Please check the `Readme.md` file in the project folder for the list with all the required plugins the application is based on and should be installed.

Run a local development server

Start the local NodeJS server and run the application in the browser:

```
$ grunt serve --lab
```

A browser window will open with two virtual devices the one next to the other.

You could also open the application in a single browser window by starting it with the command:

```
$ grunt serve
```

Run in the emulator

First the prefered platform should be added. In this case iOS:

```
$ grunt platform:add:ios
```

Note: You can only add iOS platform if you have iOS.

Now the application is ready to start inside a simulator:

```
grunt emulate:ios --livereload
```

The iPhone simulator will launch and the Music Band Ionic app will start.
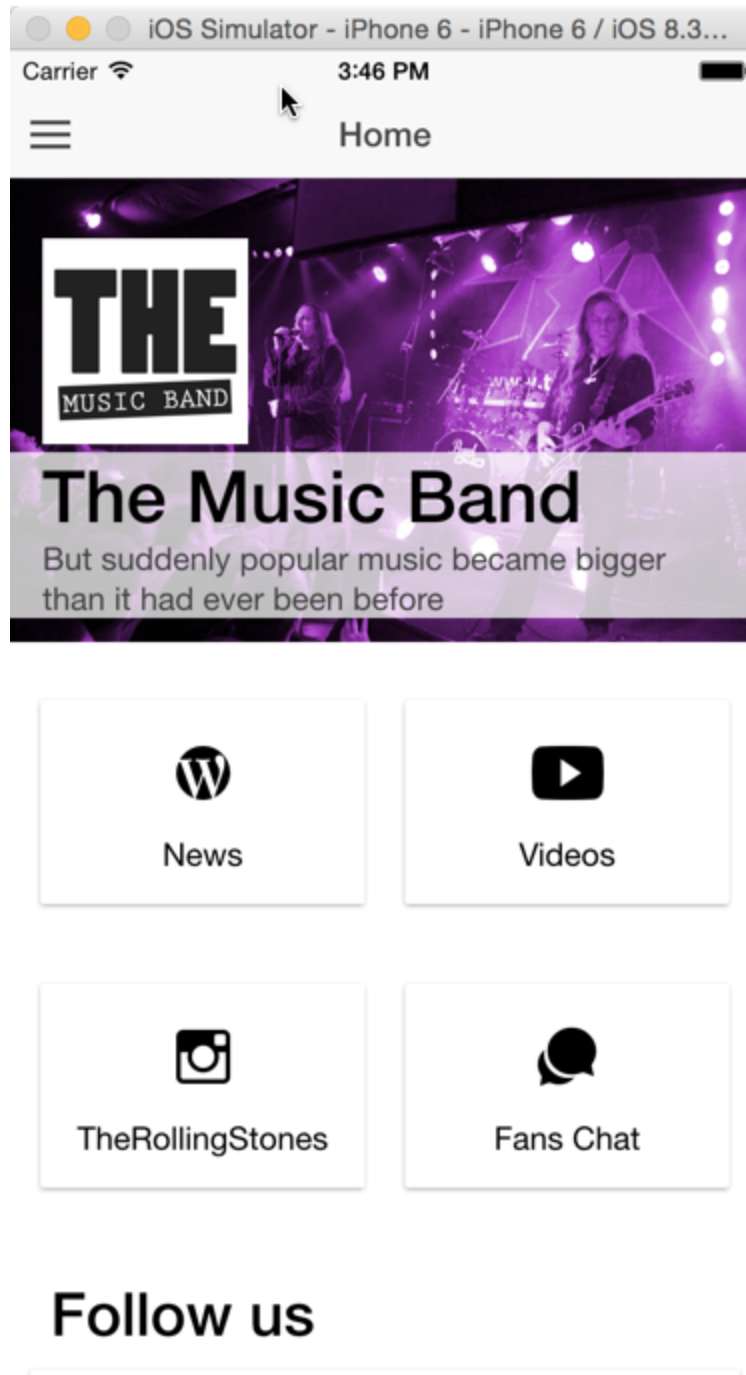
Personalize the app

The first step, once you get familiar with the application, is to personalize it. In order to do this, edit the `ionic.project` and `config.xml` files and replace the highlighted fields:

```
        ionic.project
1   {
2     "name": "music-band",
3     "app_id": ""
4   }
```

ionic.project


```
        config.xml
1   <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2   <widget id="com.titaniumtemplates.music-band" version="0.0.1" xmlns="http://www.w3.org/ns/widgets" xmlns:cdv="http://cordova.apache.org/ns/1.0">
3     <name>Music band</name>
4     <description>
5         Music Band
6     </description>
7     <author email="skounis@gmail.com" href="http://about.me/stavros.kounis">
8         Stavros Kounis
9     </author>
10    <content src="index.html"/>
```

config.xml

As for the app_id, use the ID of the coresponding application in your ionic.io account.
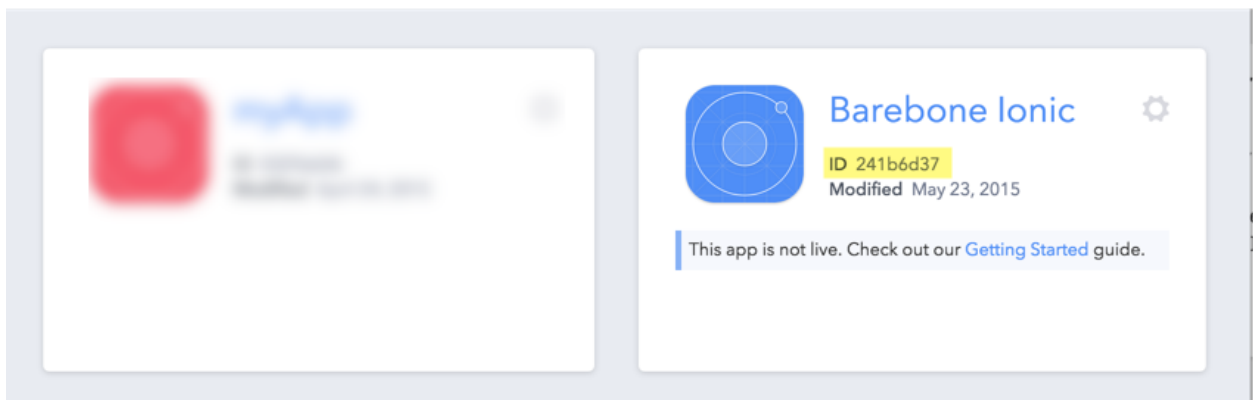
Keys

Create a set of Public and Secret key for this app in your ionic.io account.

Use these keys and configure the related properties in the Gruntfile.js


```javascript
      Gruntfile.js
56        wrap: '"use strict";\n\n {%= __ngModule %}',
57        name: 'config',
58        dest: '<%= yeoman.app %>/<%= yeoman.scripts %>/configuration.js'
59      },
60      development: {
61        constants: {
62          ENV: {
63            name: 'development',
64            apiEndpoint: 'http://dev.yoursite.com:10000/',
65            youtubeKey: 'AIza                              ',
66            ionicPrivateKey: '
67            ionicPublicKey: 'e
68            ionicAppId: '241b6d37', //'2113c758',
69            gcmId: '228071472080',
70            instagramAppId: '
71            facebookAppId: '7
72            // http://logicum.co/getting-a-facebook-page-permanent-access-token/
73            facebookPermanentAccessToken: '
74            firebaseUrl: 'music-band-ionic.firebaseio.com',
75            audioStream: 'http://stream-dc1.radioparadise.com/mp3-128'
76          }
77        }
78      },
79      production: {
80        constants: {
81          ENV: {
82            name: 'production',
83            apiEndpoint: 'http://api.yoursite.com/',
84            youtubeKey: ''
85          }
```
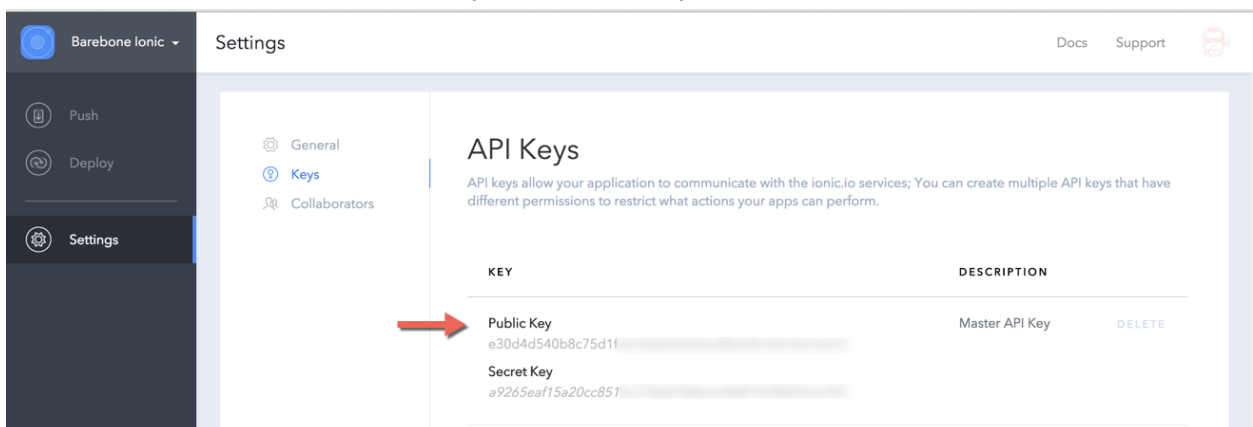
Gruntfile.js

Modules Configuration

Most of the needed configuration can be done through the `local-app-settings.service.js` file under the `app/scripts/common/services` path. The following sections will guide you regarding all the Music Band Ionic's modules and theirs configuration.

Home

Welcome area

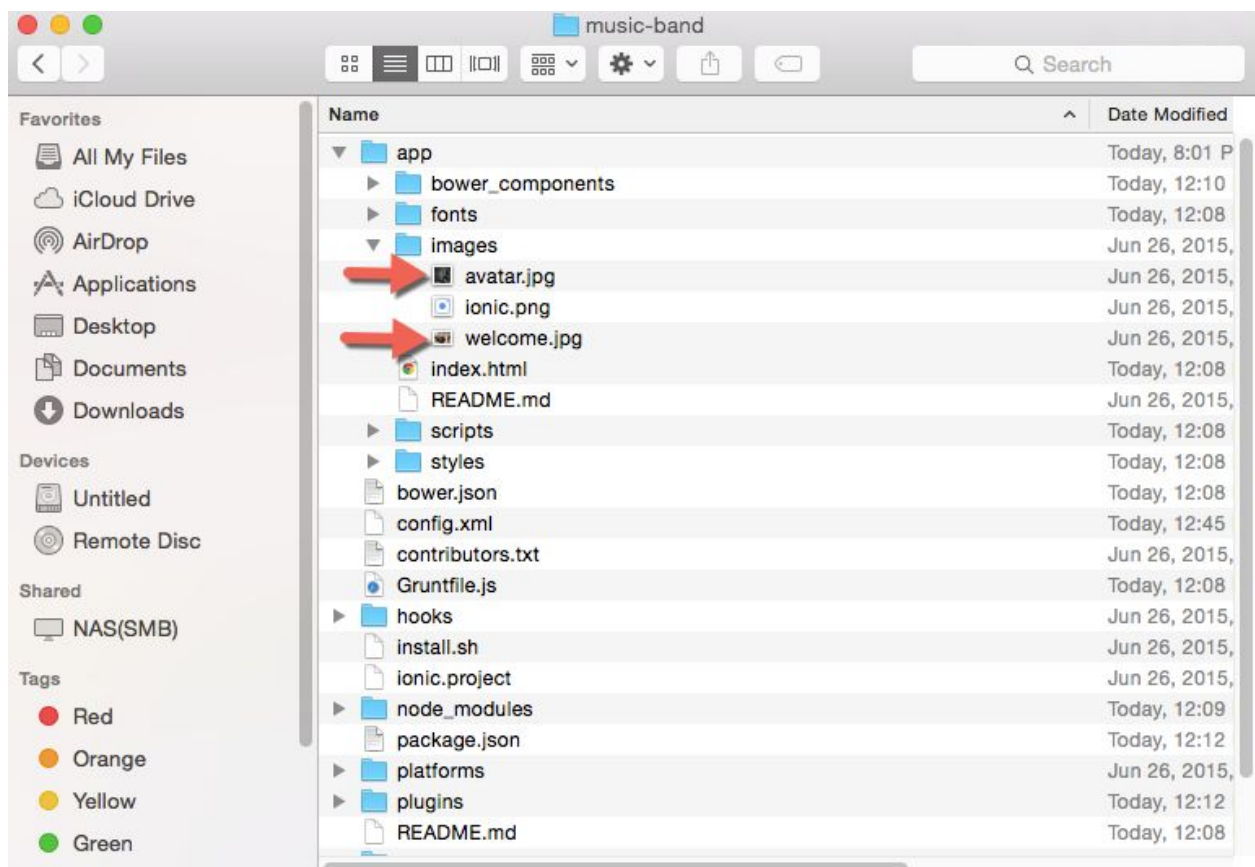To replace the **welcome title** and **slogan** showing on the **home screen**, open the `local-app-settings.service.js` file located under the `app/scripts/common/services/` path and adjust the highlighted area.

local-app-settings.service.js
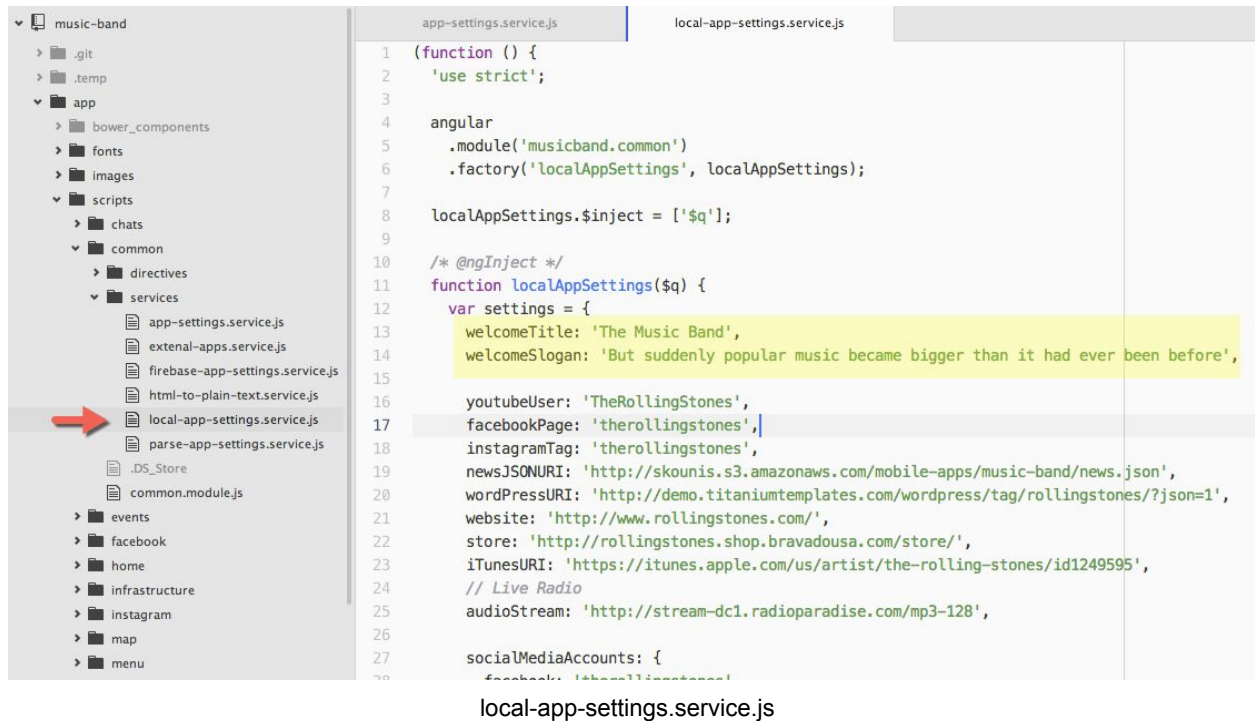
You can change the home screen's **avatar** and **welcome image** by replacing the files shown below:

Please, note that you need to keep the names and the location of the images the same with the ones above ("avatar" and "welcome" in `images` folder).

Home buttons

On the home screen, there are 4 buttons. However, they could be replaced with other buttons by editing the highlighted area in `menu-items.service.js` file under the `app/scripts/home/` path:

menu-items.services.js

Also, the markup could change by modifying the `home.html` file under the under the `app/scripts/home/` path as shown below:
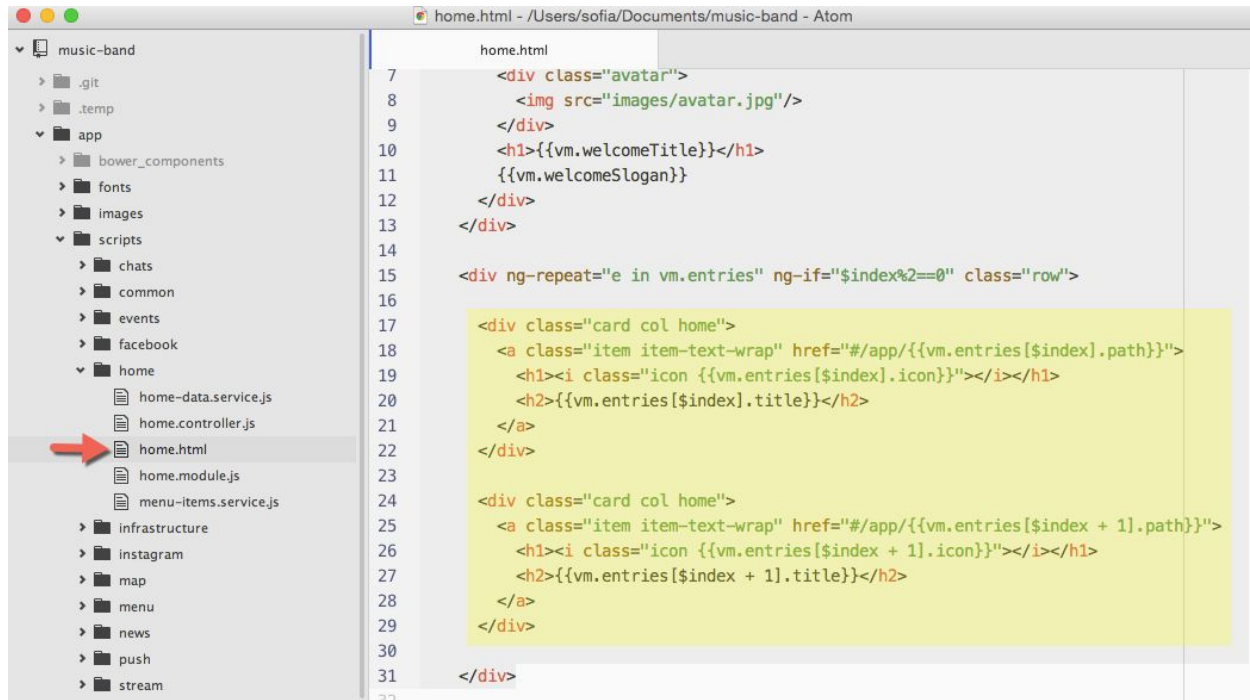
home.html

Social Links / Follows

Here, you can set the url which will be used to launch the browser installed in the device.

Open the `local-app-settings.service.js` file under the `app/script/common/services/` path and replace the highlighted areas.
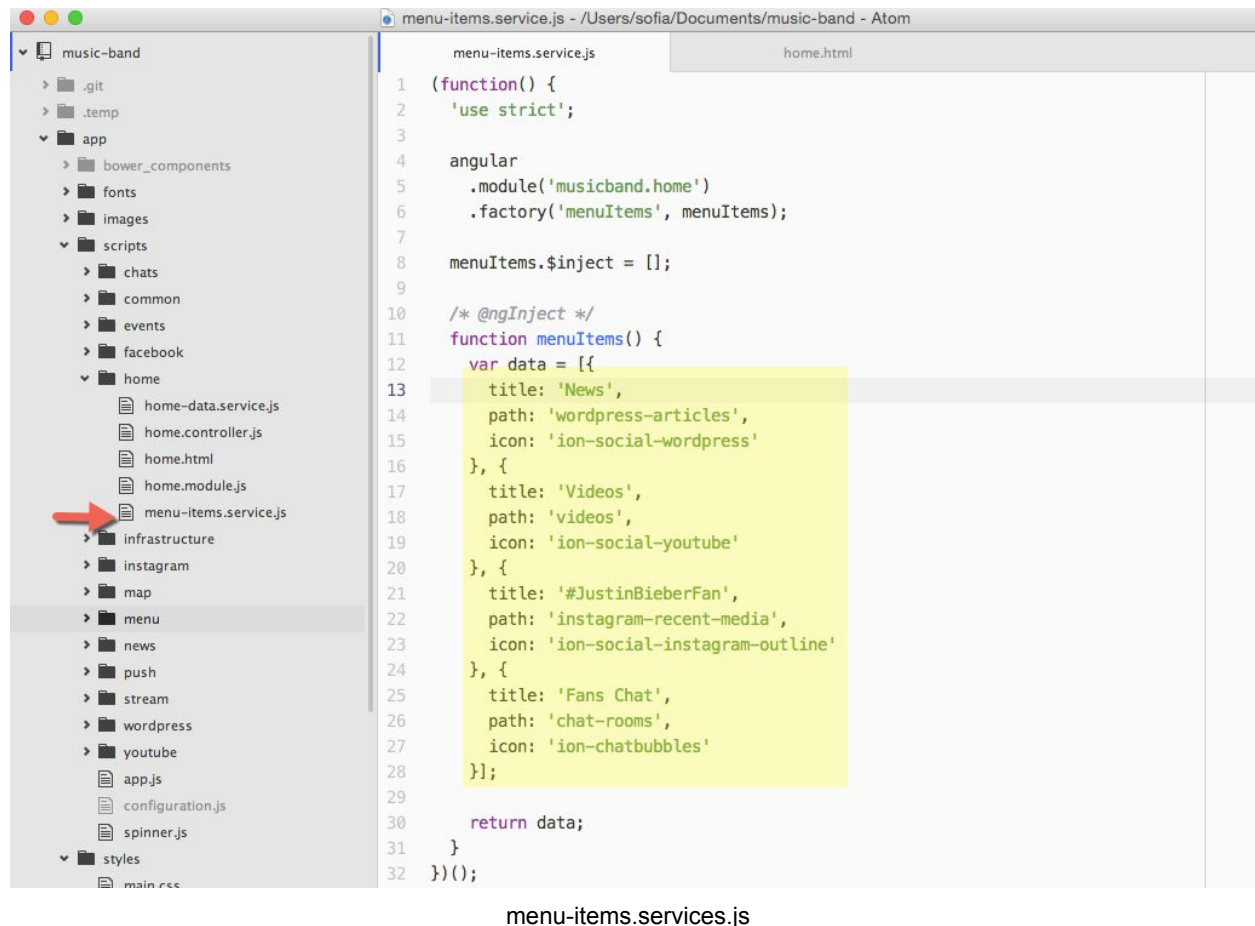
For example, in the highlighted area "instagram" you can define what follows "https://instagram.com/". Therefore, if you want to go to https://instagram.com/therollingstones address, you should set the "instagram" parameter "therollingstones".

The same applies for Facebook and Twitter. For the website, music and store just define the whole url as shown below:
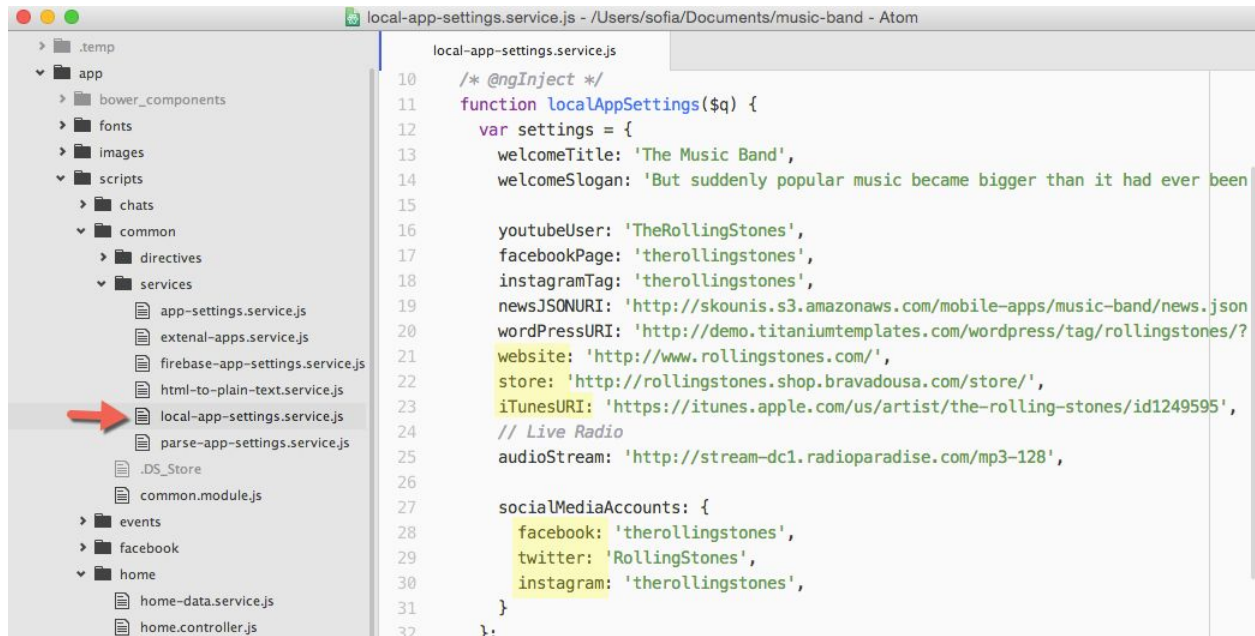
app-settings.service.js

News

News screen is configured to fetch data from remote sources. JSON is the protocol that is used for this purpose.

Its source is available in the URLs below:

http://skounis.s3.amazonaws.com/mobile-apps/music-band/news.json

Data Sources configuration

In order to set the **Music Band Ionic** application to work with your own data sources you should follow the next steps:

1. Create your JSON structures and put them online.
2. Open the `local-app-settings.service.js` file under the app/scripts/common/services locate the `newsJSONURI` variable and replace it with your sources.

app-settings.service.js

Wordpress

Wordpress screen is configured to fetch data from remote sources. JSON is the protocol that is used for this purpose.
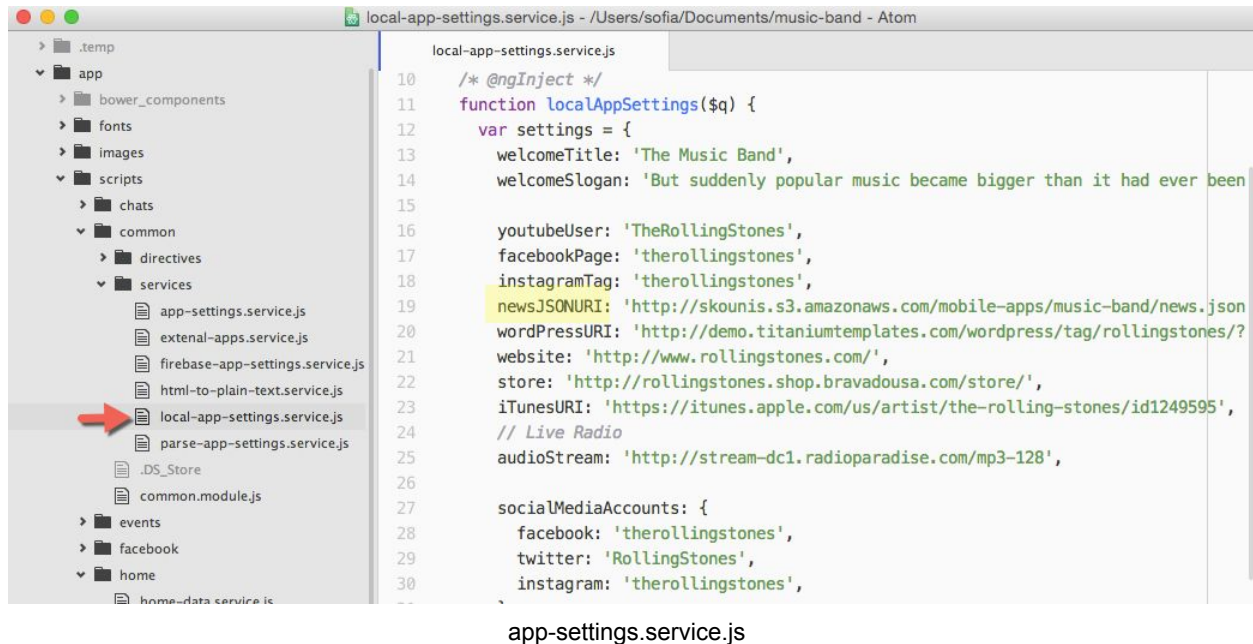
Its source is available in the URLs below:

http://demo.titaniumtemplates.com/wordpress/tag/rollingstones/?json=1

Data Sources configuration

In order to set the **Music Band Ionic** application to work with your own data sources you should follow the next steps:

3. Create your JSON structures and put them online.
4. Open the `local-app-settings.service.js` file under the app/scripts/common/services locate the `wordPressURI` variable and replace it with your sources.

app-settings.service.js
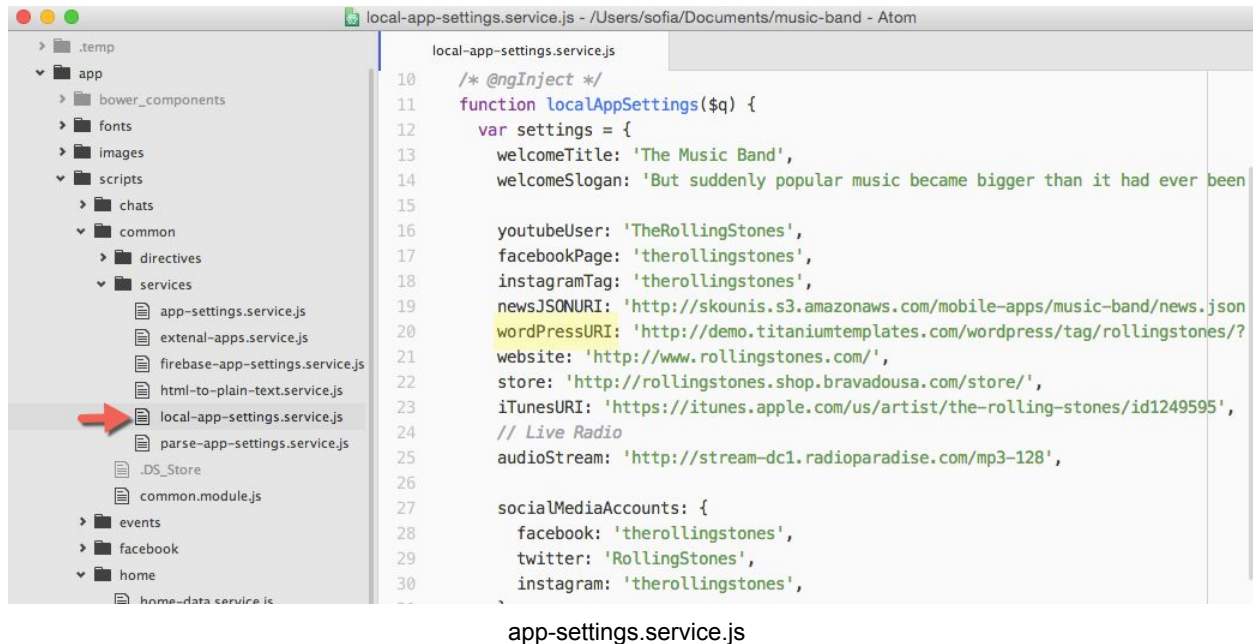
Regarding Wordpress's JSON creation, firstly, the JSON API plugin needs to be installed and activated. The related documentation is also available. According to that, the generation of the feed is done by finding the location on a website that you want to get a JSON feed and add "?json=1" at the end.

Instagram

This feature fetches the Recent Posts from the defined tag into Music Band Ionic. The Instagram tag should be defined in `local-app-settings.service.js` file under the `app/scripts/common/services/` path. Below, both of the tags are highlighted:
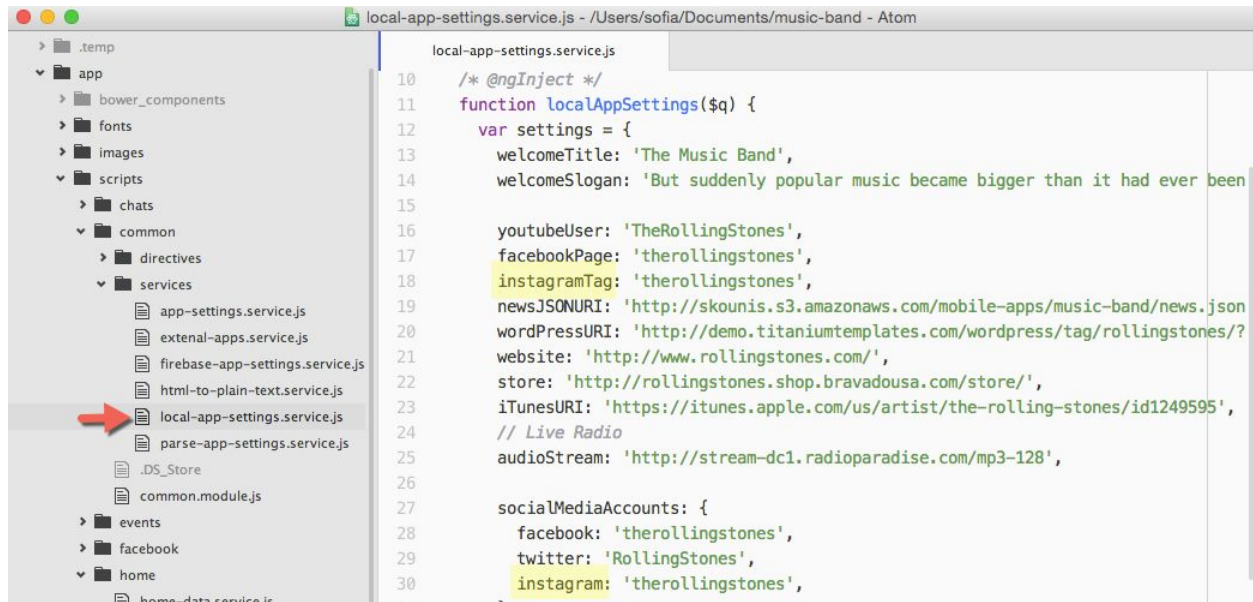
app-settings.service.js

[Instagram API](#) is used for this purpose. Everything related to the code for the communication with Instagram API is already implemented.

Instagram Application ID

Please note that you should create and register a new Instagram Client (application) by following the URL below:

https://instagram.com/developer/clients/manage/

Once the API Application is created navigate to the "Security" tab and check the "Disable implicit Auth" option. It should not be checked.
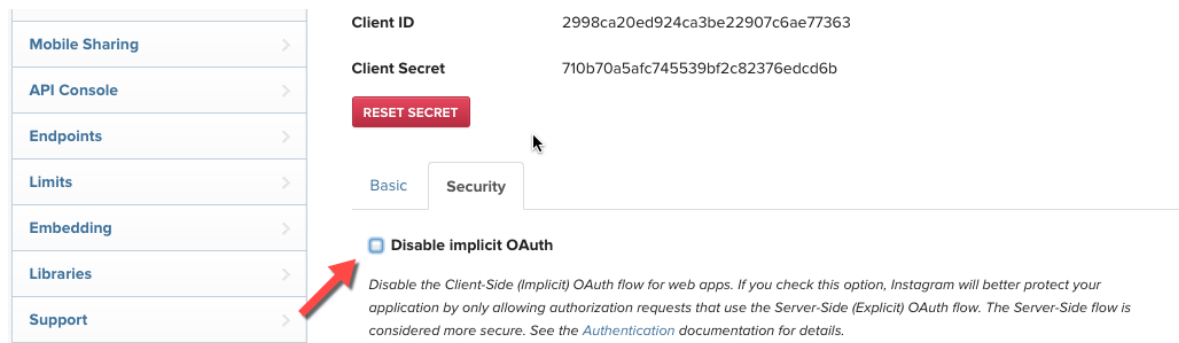
Keep also a note of the "Client ID". You need to define this App ID in the `Gruntfile.js` file, as shown below:

```
      app-settings.service.js    |              Gruntfile.js

56          wrap: '"use strict";\n\n {%= __ngModule %}',
57          name: 'config',
58          dest: '<%= yeoman.app %>/<%= yeoman.scripts %>/configuration.js'
59        },
60        development: {
61          constants: {
62            ENV: {
63              name: 'development',
64              apiEndpoint: 'http://dev.yoursite.com:10000/',
65              youtubeKey: 'A                                    ',
66              ionicPrivateKey: '                                  ,
67              ionicPublicKey: 'e                                  ,
68              ionicAppId: '241b6a37', // 2113c758',
69              gcmId: '228071472080',
70              instagramAppId: '2                                  '
71              facebookAppId: '7                                  ,
72              // http://logicum.co/getting-a-facebook-page-permanent-access-token/
73              facebookPermanentAccessToken: '
74              firebaseUrl: 'music-band-ionic.firebaseio.com',
75              audioStream: 'http://stream-dc1.radioparadise.com/mp3-128'
```

Gruntfile.js

Facebook Albums

This feature integrates Facebook Albums into Music Band Ionic. [Facebook Graph API](#) is used for this purpose.

Everything related to the code for the communication with Facebook API is already implemented.

Facebook permanent Page Access token

For accessing the Albums and their images of a Facebook page the authorization of the user is not required. It would be a bad user experience either.

For cases like this the "Permanent Facebook Access token" could be used. This allows the application to establish an authenticated connection with the API of Facebook by using a token (key) stored in the source code of the application.

For instructions on how to create your permanent token please check the following web page:

http://logicum.co/getting-a-facebook-page-permanent-access-token/

You need to define this `PermanentAccessToken` in the `Gruntfile.js` file, as shown below:

Facebook Page

Music Band Ionic is already configured in order to pull and display the photos in the albums of a particular Facebook page.

The only information you need to set is the id of the Facebook page you would like to be used. The **page id** can be found in the browser's URL when you are visiting a Facebook Page

Once you have created your **PermanentAccessToken** and chosen the **page id**, set them in the `Gruntfile.js` and `local-app-settings.service.js` files respectively as shown below:

Gruntfile.js

app-settings.service.js

Live Radio

In order to set the audio stream, open the `local-app-settings.service.js` file and change the highlighted value:

app-settings.service.js
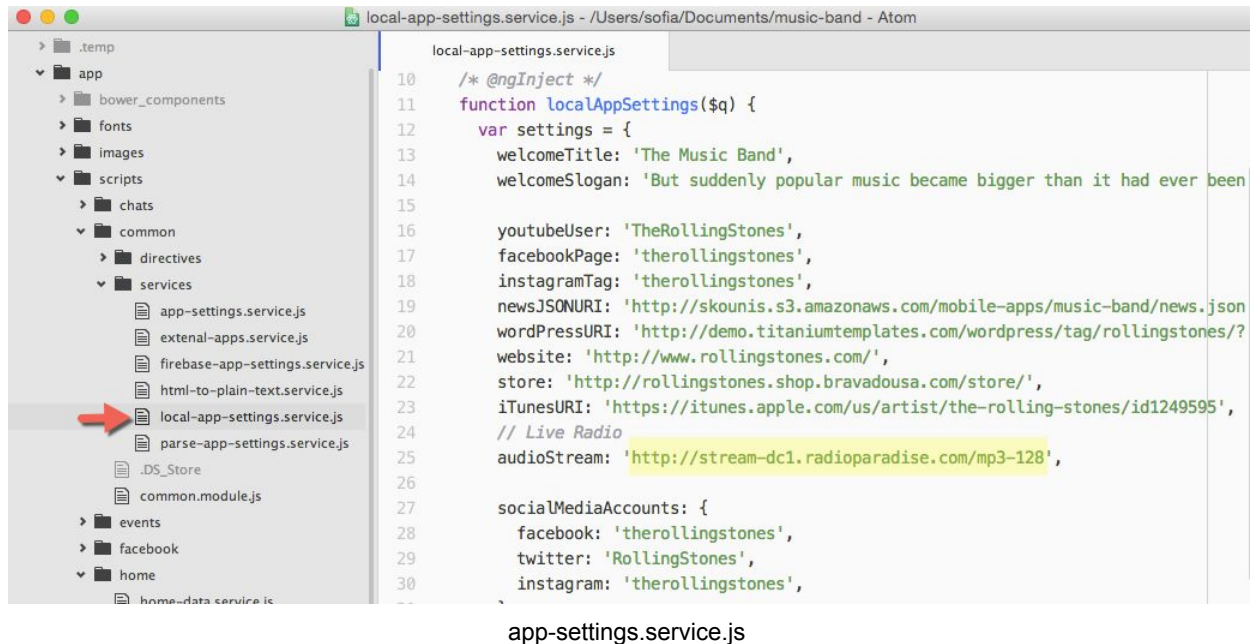
SHOUTcast Metadata

When a SHOUTcast sound stream is set in the application's settings, the song title, artist name and cover photo, that are part of the SHOUTcast metadata, are fetched into the app while the live radio is playing. Therefore, Music Band app will extract the song artist-title and use it for the construction of an iTunes API request in order to get the song's cover image.

Configuration

First, make sure you have set an audio stream url that is SHOUTcast in the app settings as described in the previous section.

An example of a SHOUTcast URL is:

- http://198.100.125.242:80/
  This URL consists of the server IP and the server port. All SHOUTcast streams use port 80.

Then, you need to define if you want the app to pull metadata from that stream or not. This can be done by modifying the value of the highlighted variable in the `local-app-settings.service.js` file under the `app/scripts/common/services` path. Thus, in case of pulling metadata set it to `true`:

local-app-settings.service.js
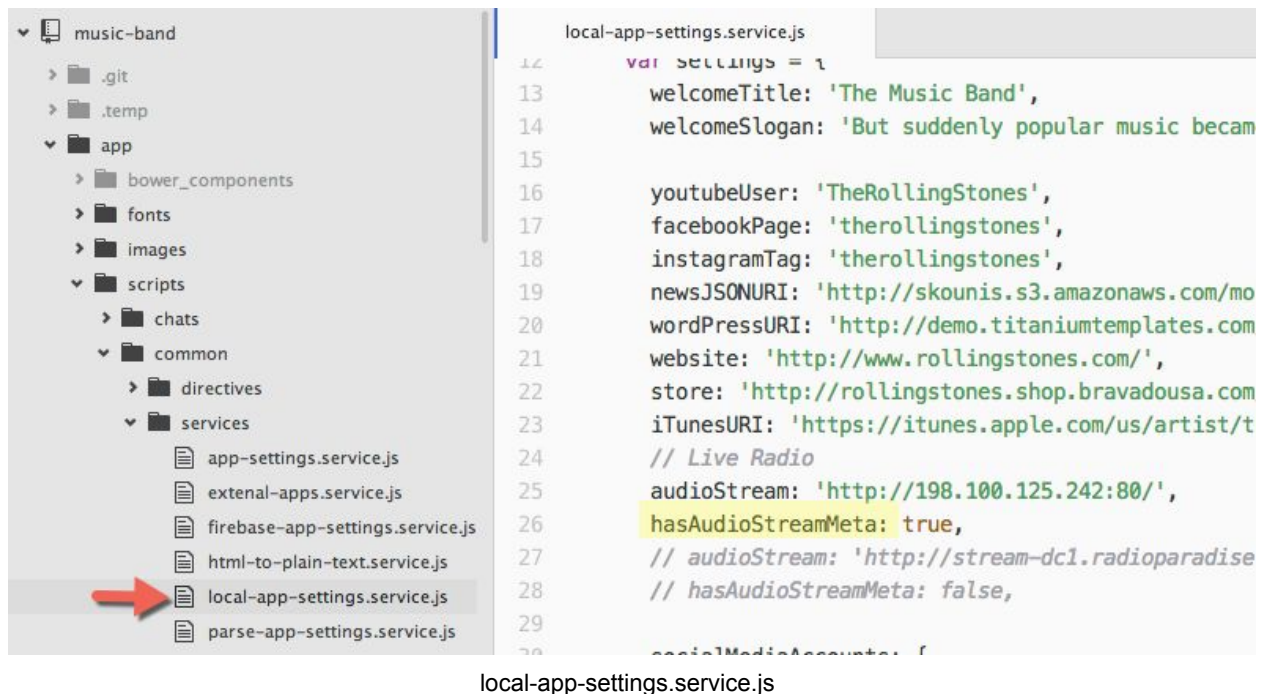
About Shoutcast

Read more about Shoutcast in the following resources:

- Wikipedia: https://en.wikipedia.org/wiki/SHOUTcast
- Official Shoutcast site: http://www.shoutcast.com/
- Samples of radio stations:
  - Radio Caprice Chicago Blues http://79.111.14.76:9091/
  - Blues After Hours http://46.23.68.170:8108/

Videos

This feature shows videos by integrating Youtube playlists into Music Band Ionic. YouTube DATA API (v3) is used for this purpose.

Everything related to the code for the communication with YouTube DATA API is already implemented. The user should only define the YouTube account username by replacing the variable "youtubeUser" as shown below. The respective file, local-app-settings.service.js is located app/scripts/common/services/path.

app-settings.service.js
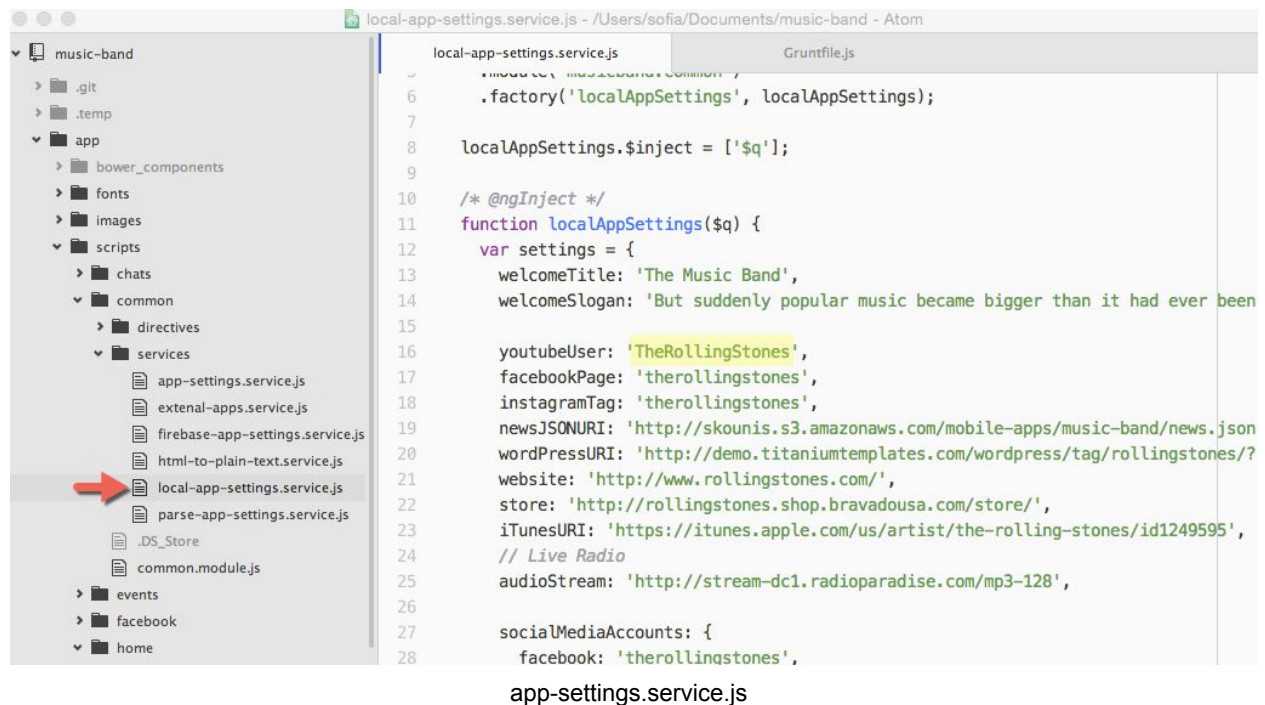
API key

Please, note that you will need a developer key in order to make YouTube API calls. To create that key, you will need a registered Google account.

Therefore, using the [Google API Console](#), you will need to enable the YouTube Data API service and generate your API key.
You can find more instructions on how create a YouTube key in the video below:
https://www.youtube.com/watch?v=zOYW7FO9rzA

As shown below, you need to define this key in the `Gruntfile.js` file.

```
        app-settings.service.js              Gruntfile.js
52       // https://github.com/diegonetto/generator-ionic/blob/master/docs/FAQ.md#how-do-i-add-constants
53       ngconstant: {
54         options: {
55           space: ' ',
56           wrap: '"use strict";\n\n {%= __ngModule %}',
57           name: 'config',
58           dest: '<%= yeoman.app %>/<%= yeoman.scripts %>/configuration.js'
59         },
60         development: {
61           constants: {
62             ENV: {
63               name: 'development',
64               apiEndpoint: 'http://dev.yoursite.com:10000/',
65               youtubeKey: '                                    ',
66               ionicPrivateKey: '
67               ionicPublicKey: '
68               ionicAppId: '241bbd37', //'2113c758',
69               gcmId: '228071472080',
70               instagramAppId: '
71               facebookAppId: '
72               // http://logicum.co/getting-a-facebook-page-permanent-access-token/
73               facebookPermanentAccessToken: '
74               firebaseUrl: 'music-band-ionic.firebaseio.com',
75               audioStream: 'http://stream-dc1.radioparadise.com/mp3-128'
76             }
77           }
```

Gruntfile.js

For more information, please, checkout the following references:
1.   YouTube Data API (v3)
2.   Registration of the application with Google

Events

Events screen is configured to fetch data from an array in
`local-app-settings.service.js` file under the `app/scripts/common/services/`
path. In order to add an event, follow the highlighted structure below:

app-settings.service.js

For more information, please visit https://github.com/mattlewis92/angular-bootstrap-calendar.
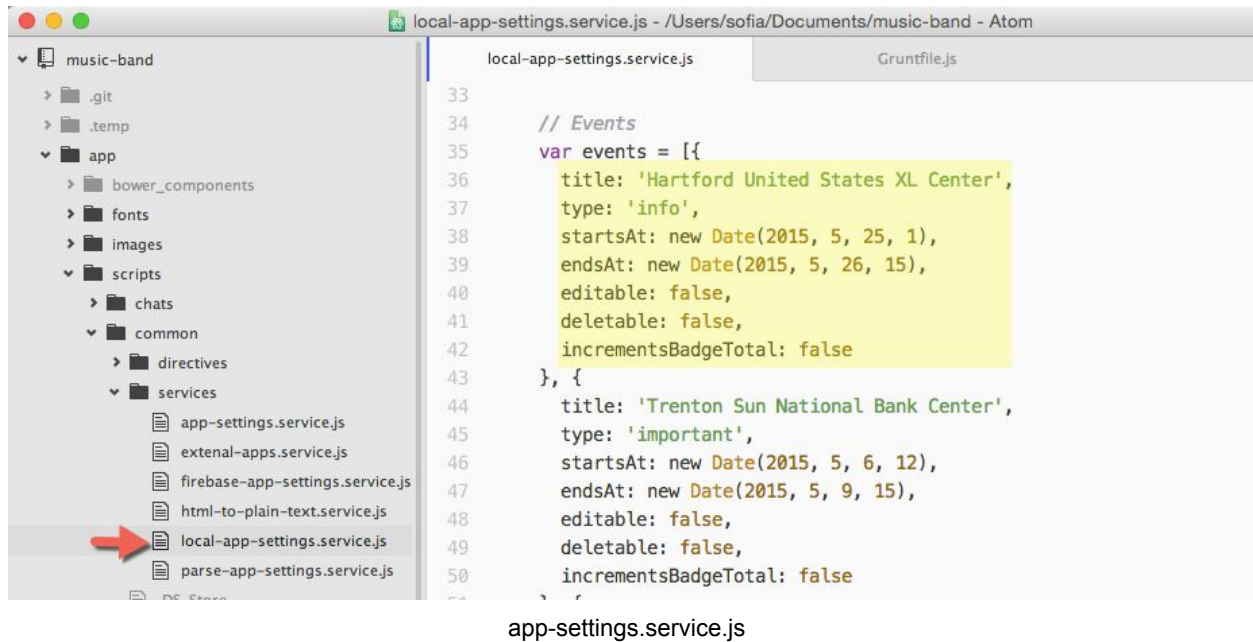
Events Map

The Map Window annotations are automatically created based on the corresponding information in the `app/scripts/map/map.service.js` file.

To set your annotations, open the `local-app-settings.service.js` file located under the `app/scripts/common/services/` path and adjust the highlighted array.

app-settings.service.js
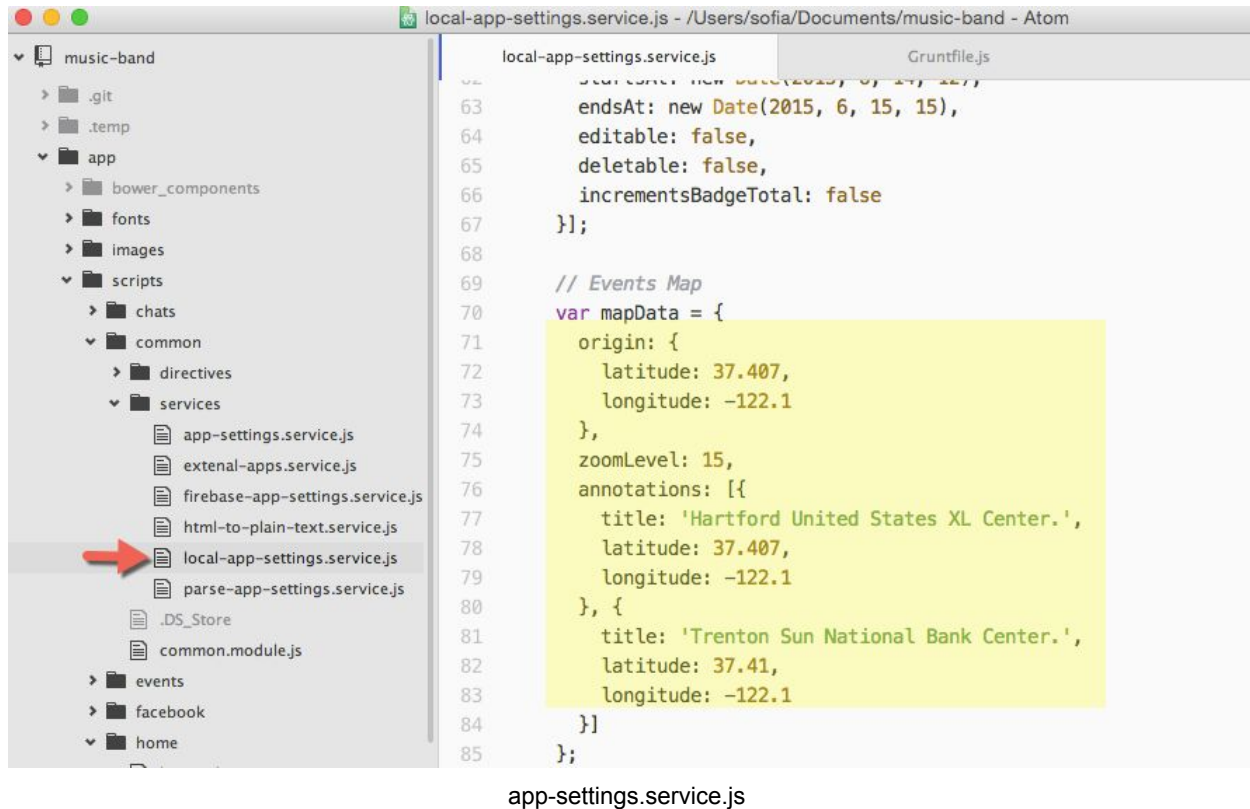
The `origin` property sets the point where the map will be centered.

You are able to set as many annotation points as you wish. To do this, you need to extend the existing JSON structure.

Push Notification

Before you proceed further please check the related documentation provided by Ionic:

http://docs.ionic.io/v1.0/docs/push-from-scratch

Plugins

Install the required plugins by using the following commands:

```
$ ionic plugin add https://github.com/phonegap-build/PushPlugin.git
$ ionic add ngCordova
$ ionic add ionic-service-push
```

Ionic Platform

Please, confirm that you have already created the coresponding Application in your ionic.io account and you have properly set the ID of you application in the `Gruntfile.js`

```
     Gruntfile.js
54        options: {
55          space: '   ',
56          wrap: '"use strict";\n\n {%= __ngModule %}',
57          name: 'config',
58          dest: '<%= yeoman.app %>/<%= yeoman.scripts %>/configuration.js'
59        },
60        development: {
61          constants: {
62            ENV: {
63              name: 'development',
64              apiEndpoint: 'http://dev.yoursite.com:10000/',
65              youtubeKey: '                                      ',
66              ionicPrivateKey: '                                   ', //
67              ionicPublicKey: 'e                                 ', //
68              ionicAppId: '241b6d37', //'2113c758',
69              gcmId: '228071472080',
70              instagramAppId: '                              ', //
71              facebookAppId: '7                        ', //
72              // http://logicum.co/getting-a-facebook-page-permanent-access-token/
73              facebookPermanentAccessToken: '                                   
74              firebaseUrl: 'music-band-ionic.firebaseio.com',
75              audioStream: 'http://stream-dc1.radioparadise.com/mp3-128'
76            }
77          }
```

Gruntfile.js

iOS Configuration

Follow the instructions provided by Ionic and create your PFX (.p12) certificates.

http://docs.ionic.io/v1.0/docs/push-ios-setup

Remember to make ionic aware of your development and production certificates:

```
$ ionic push --ios-dev-cert
```

```
$ ionic push --ios-prod-cert
```

Android Configuration

Follow the instructions provided by Ionic and create a Google Console project and the related API key.

http://docs.ionic.io/v1.0/docs/push-android-setup

Remember to send the API key to ionic.io:

```
ionic push --google-api-key your-google-api-key
```

Use the number of your project and set it as your "`GCM ID`" in the `Gruntfile.js`

```
                Gruntfile.js
53              ngconstant: {
54              options: {
55                space: ' ',
56                wrap: '"use strict";\n\n {%= __ngModule %}',
57                name: 'config',
58                dest: '<%= yeoman.app %>/<%= yeoman.scripts %>/configuration.js'
59              },
60              development: {
61                constants: {
62                  ENV: {
63                    name: 'development',
64                    apiEndpoint: 'http://dev.yoursite.com:10000/',
65                    youtubeKey: '                                    ',
66                    ionicPrivateKey: '                                                  ,
67                    ionicPublicKey: 'e                                                  ,
68                    ionicAppId: '241b6d37', //'2113c758',
69                    gcmId: '228071472080',
70                    instagramAppId: '                                          ,
71                    facebookAppId: '7                              ,
72                    // http://logicum.co/getting-a-facebook-page-permanent-access-token/
73                    facebookPermanentAccessToken: '
74                    firebaseUrl: 'music-band-ionic.firebaseio.com',
75                    audioStream: 'http://stream-dc1.radioparadise.com/mp3-128'
76                  }
77                }
78              },
```

Gruntfile.js

Firebase Backend

The application is already ready to work with Firebase as its back-end. In this case all the Application settings like Welcome message, Social Media Accounts, Wordpress or Drupal URL, etc, are stored in Firebase and dynamically are loaded from there.

Create a Firebase App
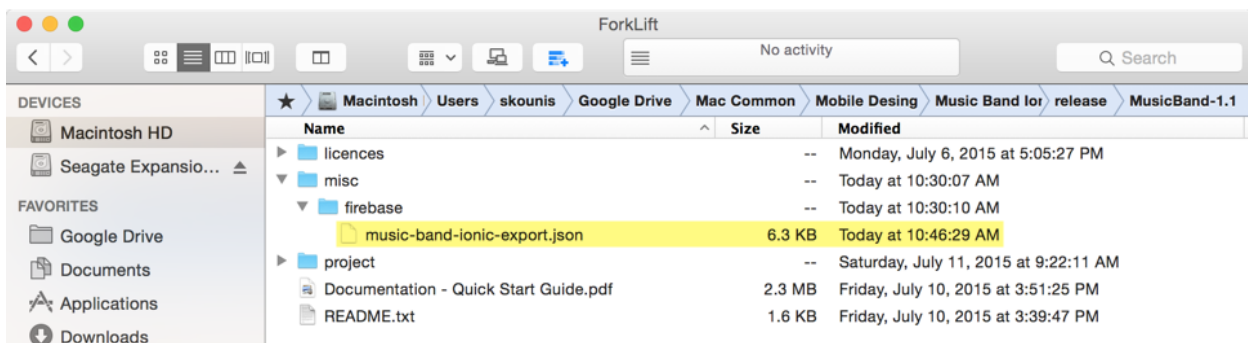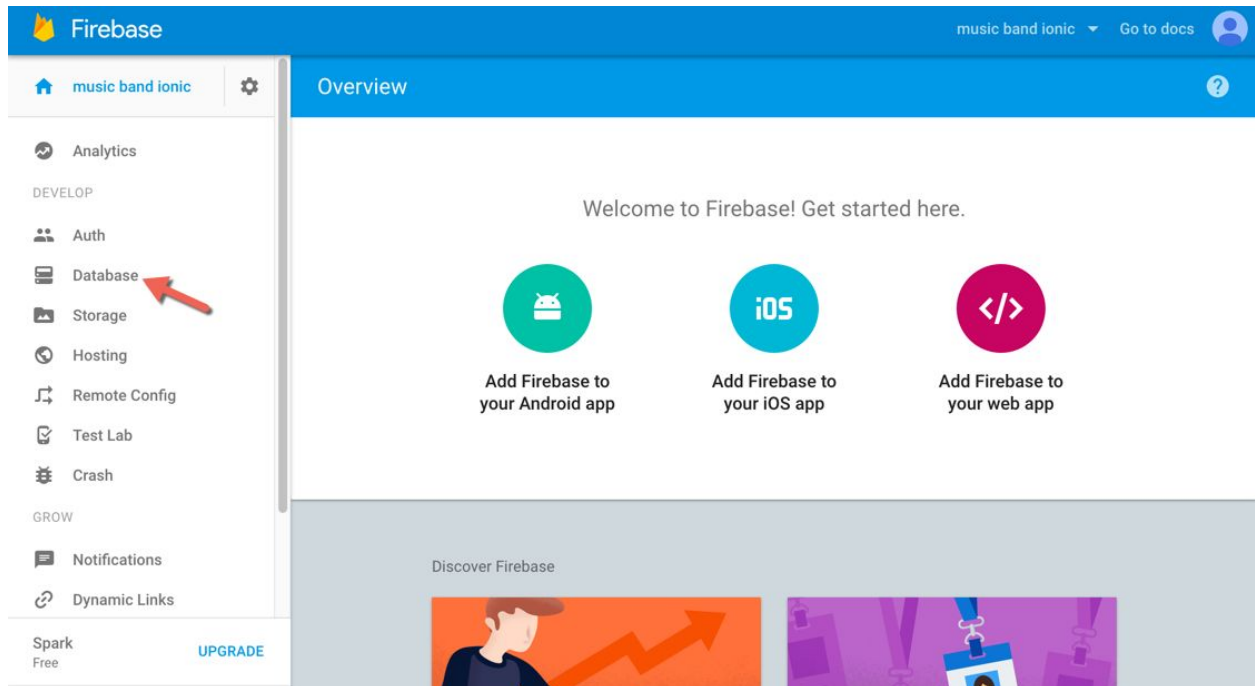First you should create a free Firebase account and create a new app there.

Then you should initialize this application by using the `music-band-ionic-export.json` file which is provided and is located under the `misc/firebase/` path
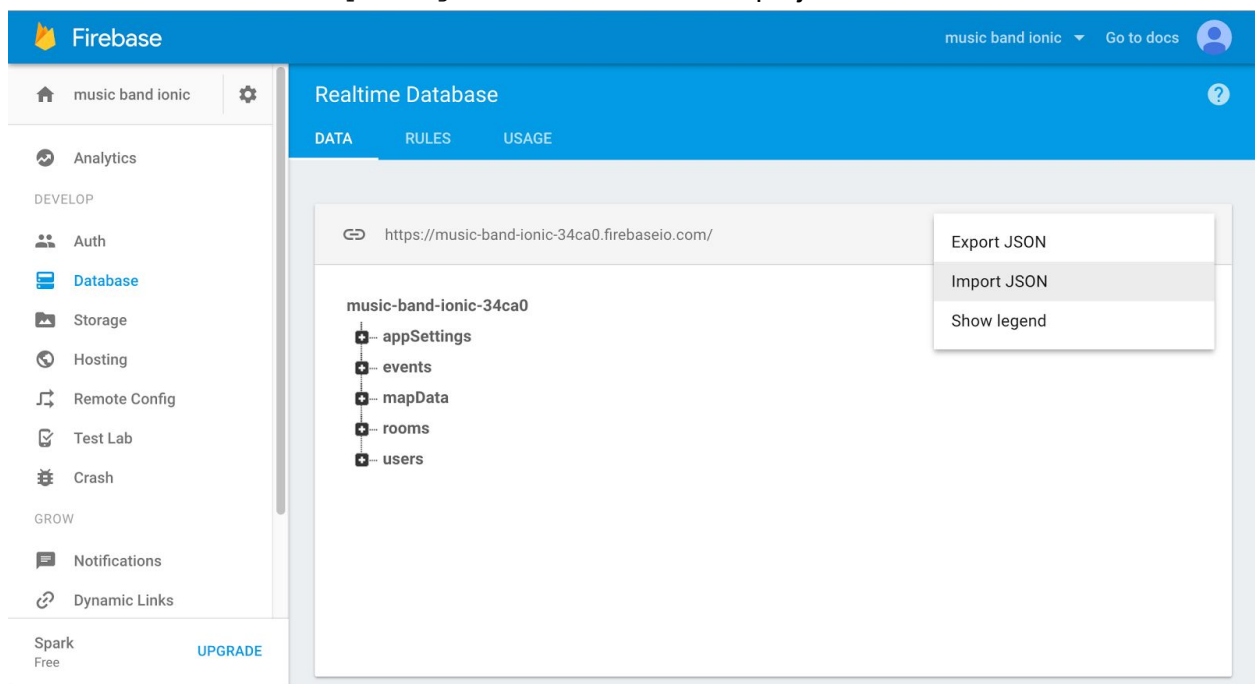

Create/Add data

Manage the app's data by clicking on the app and selecting "Database" from the side menu.

Then, choose the "Import JSON" option from the menu to import your data. Select the `music-band-ionic-export.json` file to initialize the project.

Now you are ready to configure the Music Band Ionic application.

Configuration of the application

To set the Firebase project the app will work with, you should set the "settingsSource" variable as "FIREBASE" and define the "apiKey", "authDomain" and "databaseURL" in `Gruntfile.js`:

```
    Gruntfile.js                    ✕
59                },
60             development: {
61               constants: {
62                 ENV: {
63                   name: 'development',
64                   youtubeKey: 'AIzaSyDael5MmCQa1GKQNKQY                ',
65                   ionicPrivateKey: 'a9265eaf15a20cc8516c770e8748                      ',
66                   ionicPublicKey: 'e30d4d540b8c75d1f167bbf242423c3fb23fe10275d1c016',
67                   ionicAppId: '241b6d37', //'2113c758',
68                   gcmId: '228071472080',
69                   instagramAppId: '2998ca20ed924                 ', //'ab4ccebff87a4
70                   // http://logicum.co/getting-a-facebook-page-permanent-access-token/
71                   facebookPermanentAccessToken: 'CAANL6xXrSHYBANNHhMUDugVZBHXfVQBMeWG6
72                   firebaseConfig: {
73                     apiKey: "AIzaSyDDK5YUttkgNoY97HALSXkU7S39bFA2rTQ",
74                     authDomain: "music-band-ionic-34ca0.firebaseapp.com",
75                     databaseURL: "https://music-band-ionic-34ca0.firebaseio.com"
76                   },
77                   parse: {
78                     applicationId: 'sidmrbO9OqG3pe4iErva408MHFysJZ2zChYPAXlU',
79                     key: 'e49Rnlja6llKnFW5p0OOF8dkHvJi5o1hrVzFGBnc'
80                   },
81                   //
82                   // Where the settings values will come from.
83                   // Available option:
84                   //    LOCAL    : hardcoded withing the app code
85                   //    FIREBASE : values are retrieved from Firebase
86                   //    PARSE : values are retrieved from Parse.com
87                   settingsSource: 'FIREBASE'
88                 }
```

Your application is now ready to pull its data from Firebase.

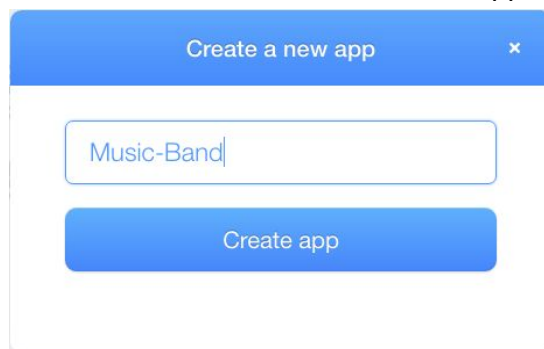## Parse Backend

Except from Firebase, the application can also retrieve the data it needs from Parse. If you choose this option, the Music Band app will use Parse as its backend to store all the app's settings, events for the Calendar and map data.
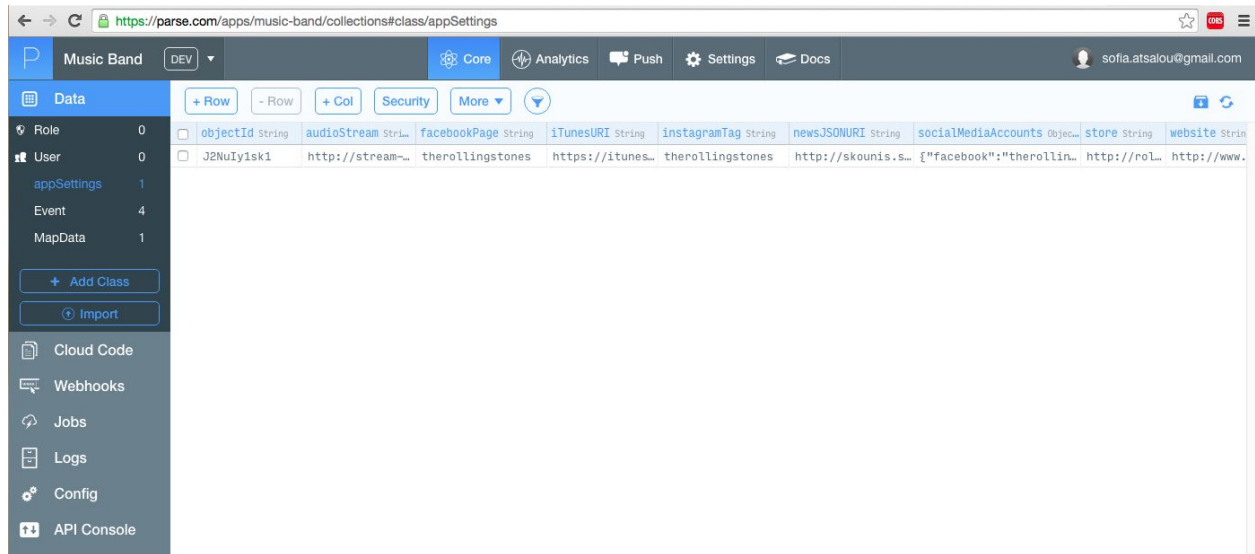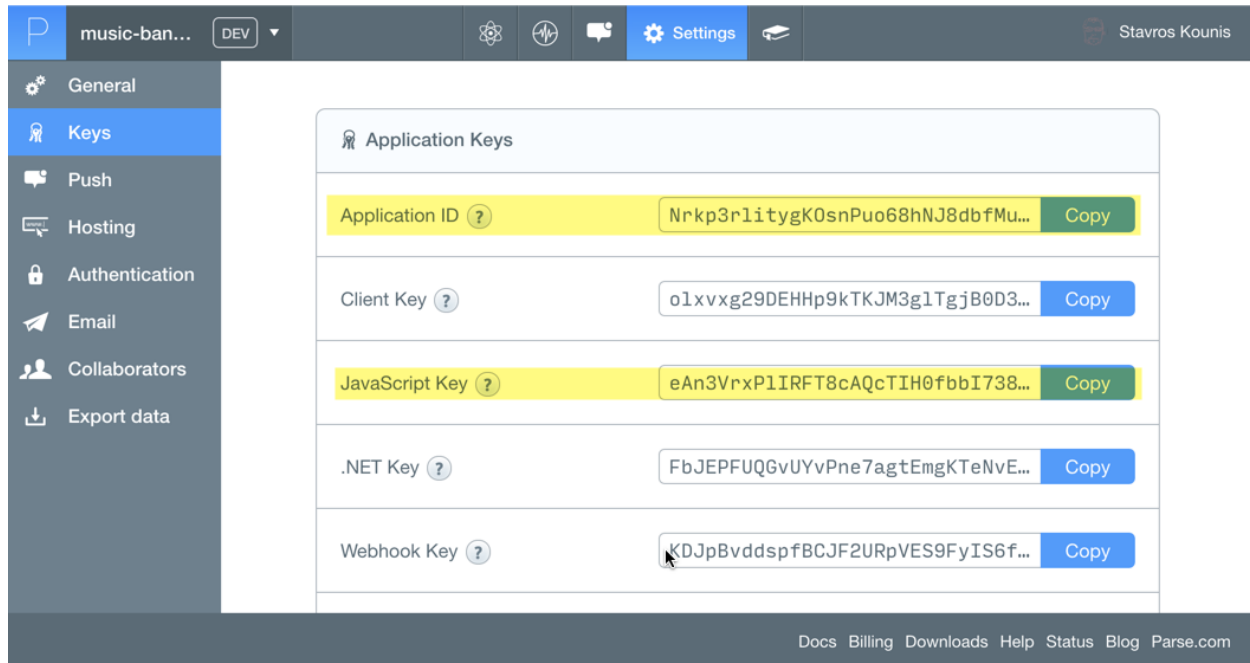
Parse App

First you should create a free Parse account and create a new app there.

Once the app is created in Settings on Keys tab, make a note of the **Application ID** and **Javascript Key**:

These will be used later for the configuration of your Parse app.

For now you should initialize this application and create the data structures that will be used by the mobile application.

Everything is already prepared for you and a set of initial values is already prepared ready to be imported in your newly created Parse application.

The files you will need are:
- `AppSettings.json,`
- `Event.json`
- `MapData.json`
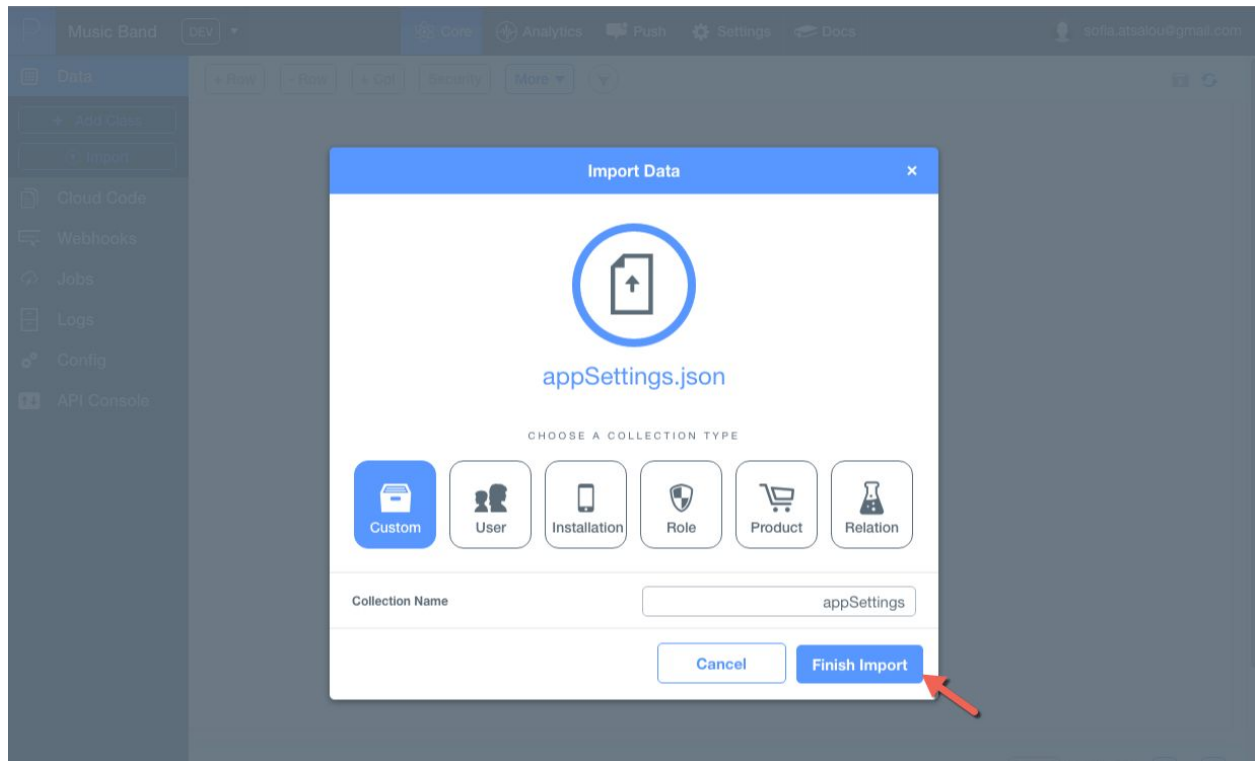
All these files are are located under the `misc/parse/` path
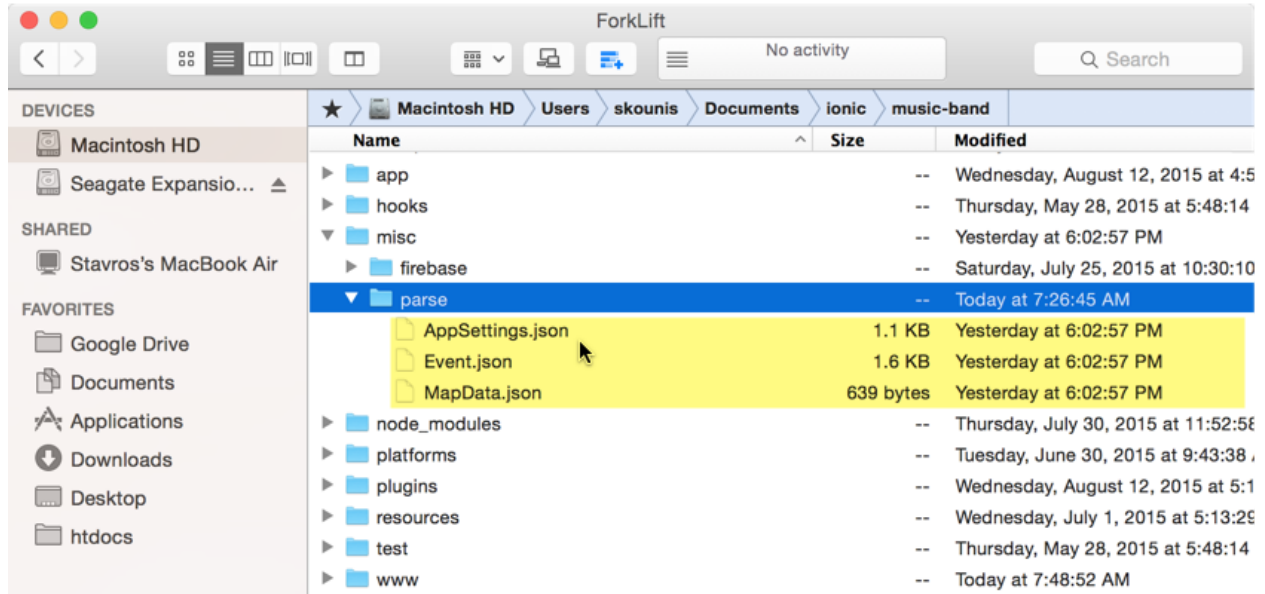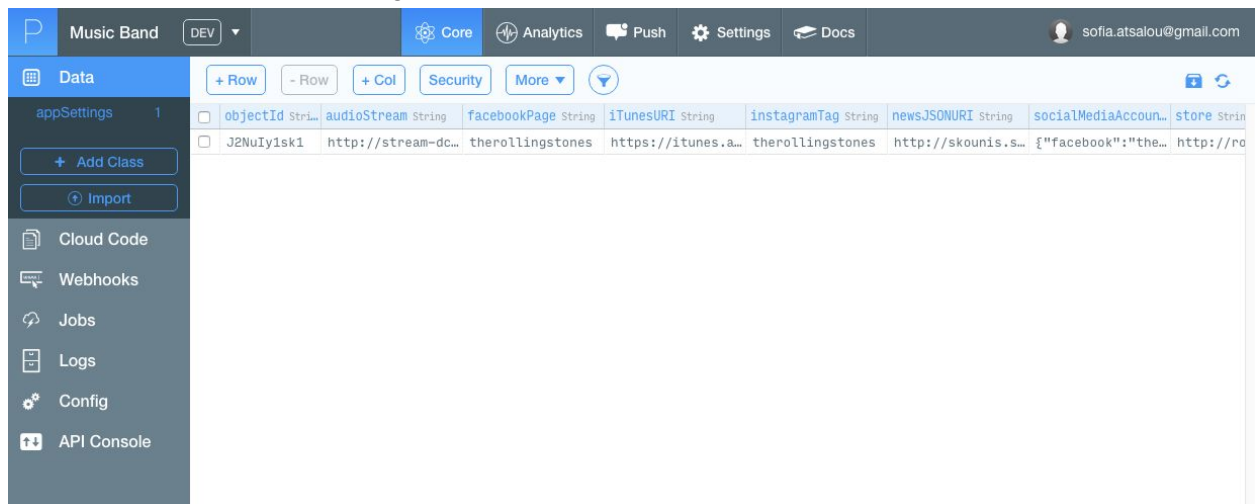
Initial import

Navigate to the Data Browser of your Parse application and click the "Import" button which is located in the top left corner of your screen.
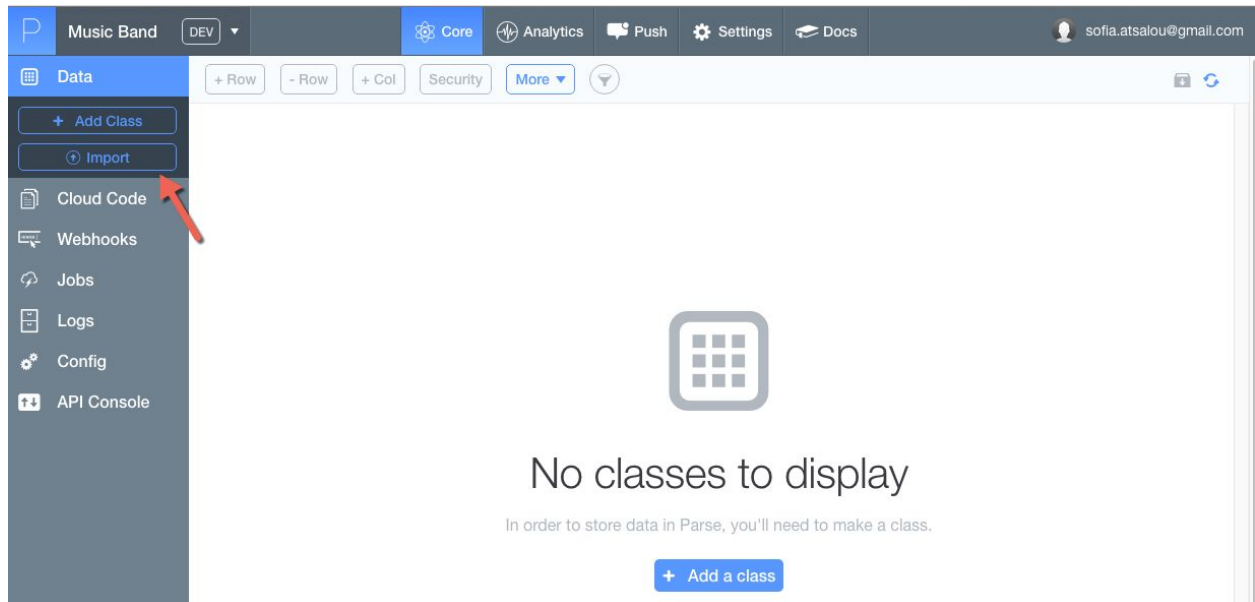
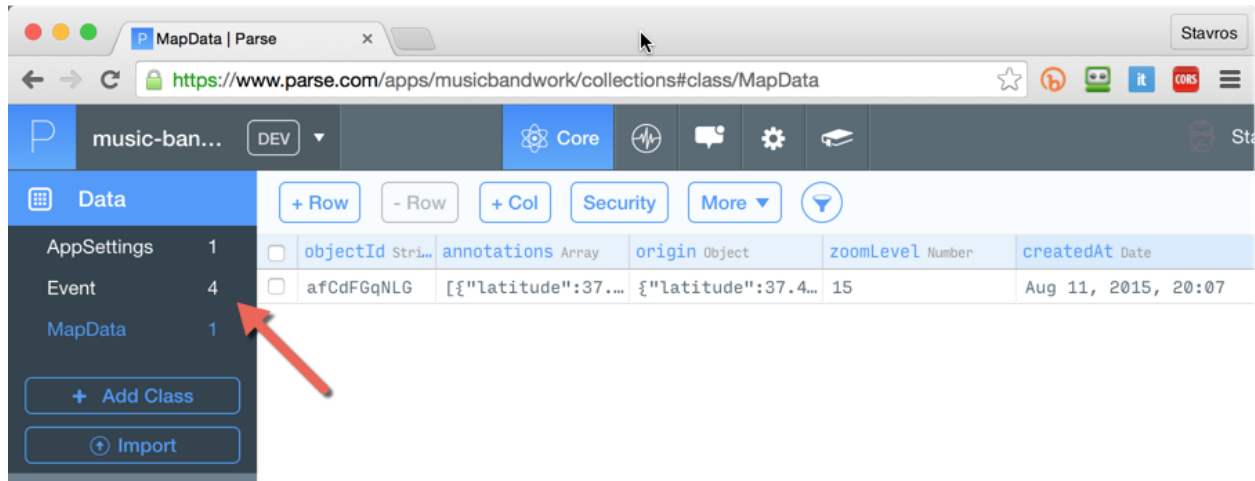Select the `AppSettings.json` file and click the "Finish Import" button.

You will end up with something similar to the screen that follows:

Similarly, you will import the rest of the JSON files:
- `Event.json`
- `MapData.json`

If everything will be successfully completed you should end up with three custom data classes as shown below:

Now you are ready to configure the Music Band Ionic application.

Configuration of the Ionic application

Open the `Gruntfile.js` file and locate the ENV property under the `development >` `constant` path.

Application ID and Javascript Key

Check the `applicationId` and `key` properties and make sure you replace them with the ones of your Parse application. Remember that the values you should use are the "**Application ID**" and the "**Javascript Key**"

Settings Source

Locate the `settingsSource` property and set its value to PARSE

```
         Gruntfile.js          •
73                firebaseurl: 'music-band-ionic.firebaseio.com',
74                parse: {
75                  applicationId: 'sidmrbO9OqG3pe4iErva408MHFysJZ2zChYPAXlU',
76                  key: 'e49Rnlja6llKnFW5p0OOF8dkHvJi5o1hrVzFGBnc'
77                },
78                //
79                // Where the settings values will come from.
80                // Available option:
81                //    LOCAL    : hardcoded withing the app code
82                //    FIREBASE : values are retrieved from Firebase
83                //    PARSE : values are retrieved from Parse.com
84                settingsSource : 'PARSE'
85              }
86            }
87          },
```

Your application is now ready to pull its data from Parse.

Support

With regard to technical questions, new ideas and suggestions, you may use the dedicated form and choose the product your enquiry refers to:

http://support.appseed.io/customer/portal/questions/new

References / Links:

- [YouTube channel](#)
  Periodically, video demonstrations and tutorials related to this product will be published in my YouTube channel.
- [Codecanyon User page](#)
  You may contact me by using my user page on Codecanyon.
- [Titanium Templates Forum](#)
  The Google Group that has been created for this product.
- [Quick Start Guide](#)
  The online version of this document.

Thank you

Thank you again for purchasing my product. If you have any questions that are beyond of the scope of this help file, please feel free to email also via [my user page](#) contact form.

--- Stavros.