Business Directory Ionic 1.2 Documentation

Quick start Guide - Installation and user support

Created: Jan 19, 2016

Updated: Mar 04, 2016

By: Stavros Kounis, about.me/stavros.kounis

Thank you for purchasing my product. If you have any question that are beyond of the scope of this help file, please feel free to email via my user page contact form. Thank you!

Table of contents:

- Preparing your local environment (first-time preparations)
  - NodeJS
  - PhoneGap and Ionic
  - Tools
- Run for the first time
  - Download and extract
  - Install libraries
    - Post installation
    - Windows Users
  - Plugins
- Run/Build the application (after the first-time preparations)
  - Run a local development server
  - Run in the emulator
  - Build the app for specific platforms
- Personalize the app
  - Keys
- Configuration
  - Data Sources Creation
    - Firebase
      - Create a Firebase app
      - Create/Add data
      - Firebase Data source Configuration
    - JSON data sources
      - Businesses
      - Common

News, Products, Services and Catalogs
Wordpress
JSON API plugin
Drupal
Services and JSON View
JSON Data Sources Configuration
Support
References / Links:
Thank you

Preparing your local environment (first-time preparations)

NodeJS

Ensure first that NodeJS is installed in your computer.

For choosing which version of NodeJS to install, please refer to the "Getting Started with Ionic" section, at the official Ionic site.

PhoneGap and Ionic

This is a PhoneGap and Ionic based application, so PhoneGap and Ionic should be installed on your computer for this application to run. Since Barebone 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 "Install PhoneGap" and "Getting Started with Ionic" sections in the official PhoneGap and Ionic sites respectively.

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:

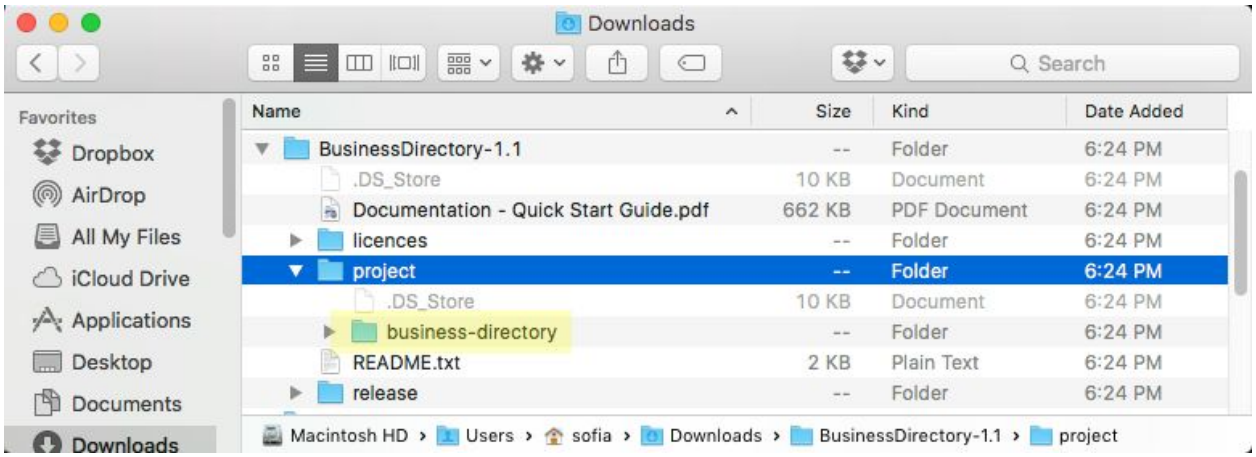
```
$ npm install -g generator-ionic
```

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 folder is the Ionic Project's directory.

Install libraries

Open a terminal window and navigate to `business-directory` folder under `project` directory, i.e. `project/business-directory` path.

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
```

Windows Users

Similarly, Windows users should execute:

```
$ install.bat
```

Plugins

Since Cordova comes with a minimum set of APIs, plugins are added in this project in order to be part of some features of the app. For example, plugins are used for enabling the app to access device's hardware and OS such as keyboard and location of the device. You can find the full list of the required plugins and their functionality in the `README.md` file under `project/business-directory` path.

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

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

eg:

```
$ cordova plugin add cordova-plugin-inappbrowser
```

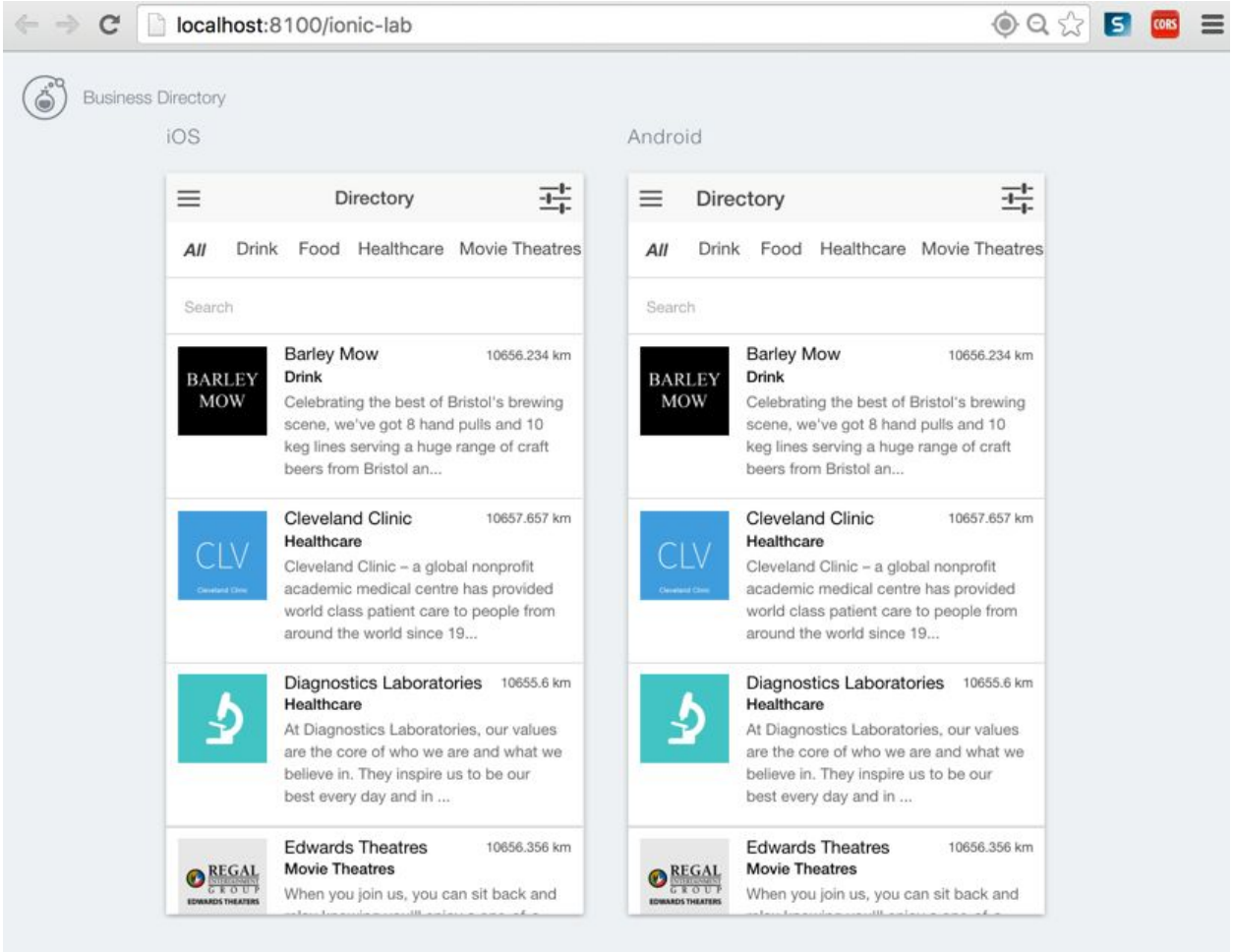
Run/Build the application (after the first-time preparations)

Run a local development server

Navigate to `project/barebone-ionic` path 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

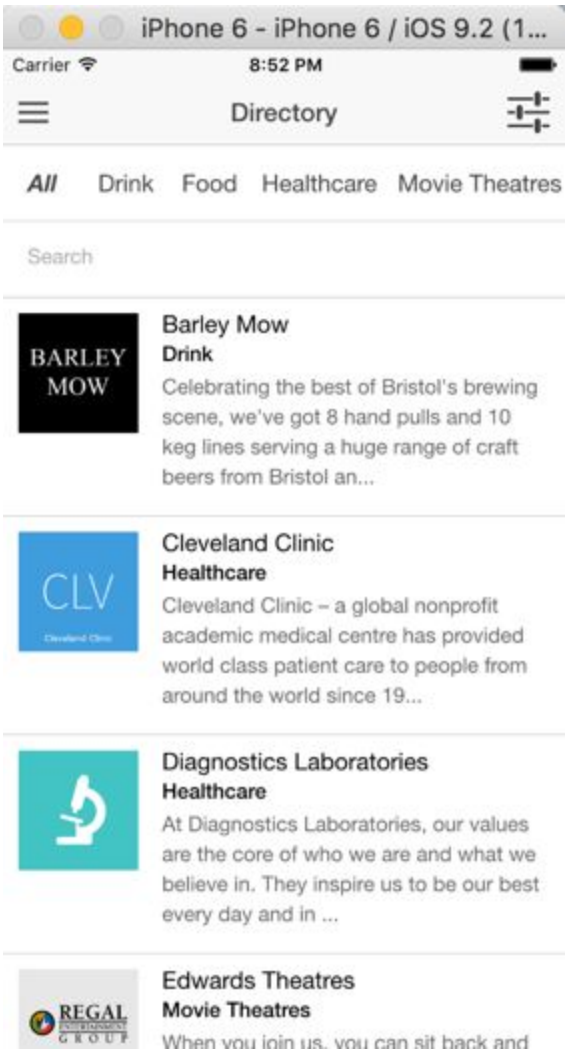
First the prefered platform should be added. In this case iOS:

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

Now the application is ready to start inside a simulator:

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

The iPhone simulator will launch and the Business Directory Ionic app will start.



Build the app for specific platforms

In order to build all the added platforms, run the command:

```
$ grunt build
```

In order to build Android platform, run the command:

```
$ grunt build:android
```

In order to build iOS platform, run the command:

```
$ grunt build:ios
```

After building your project, [www] folder will be created which is the actual cordova directory and where all the plugins and required libraries should be installed.

Personalize the app

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

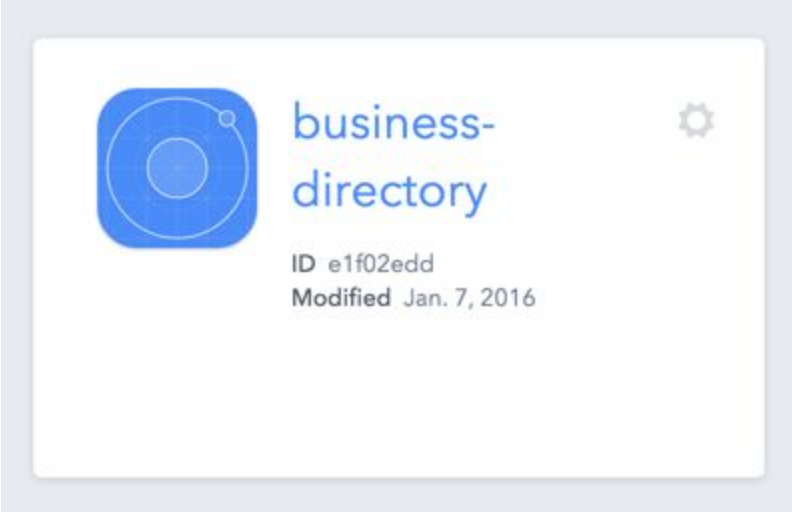
ionic.project

```
{
  "name": "business-directory",
  "app_id": "e1f02edd"
}
```

config.xml

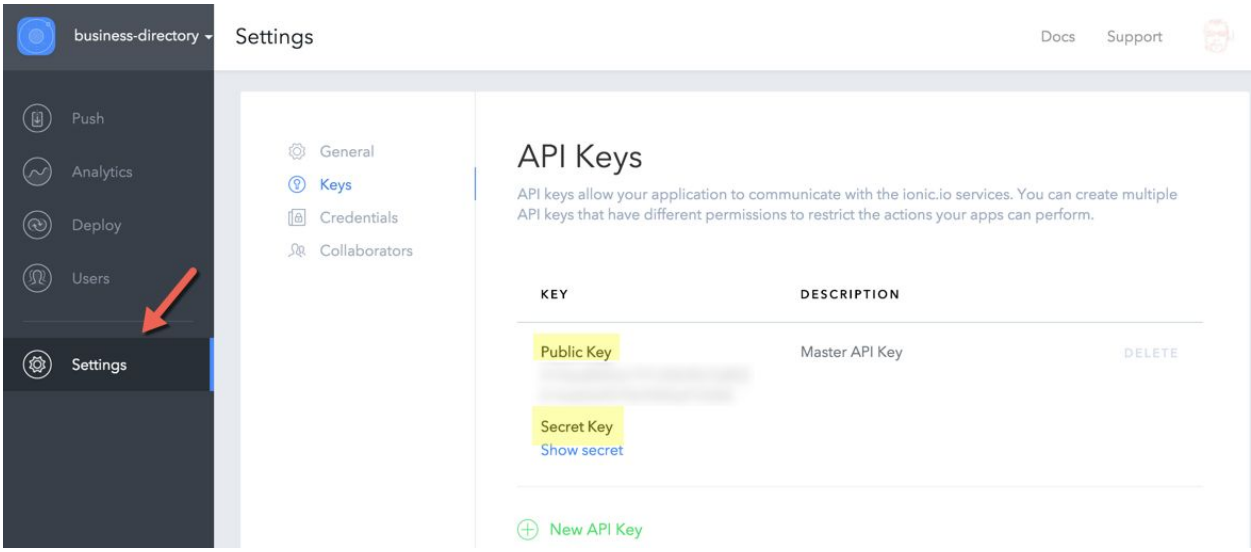
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<widget id="com.titaniumtemplates.business-directory" version="1.5.0" xmlns="http
  <name>Business Directory</name>
  <description>
        Business Directory. The boostrap you need in order to build your next Ior
  </description>
  <author email="skounis@gmail.com" href="http://about.me/stavros.kounis">
        Stavros Kounis
  </author>
  <content src="index.html"/>
```

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

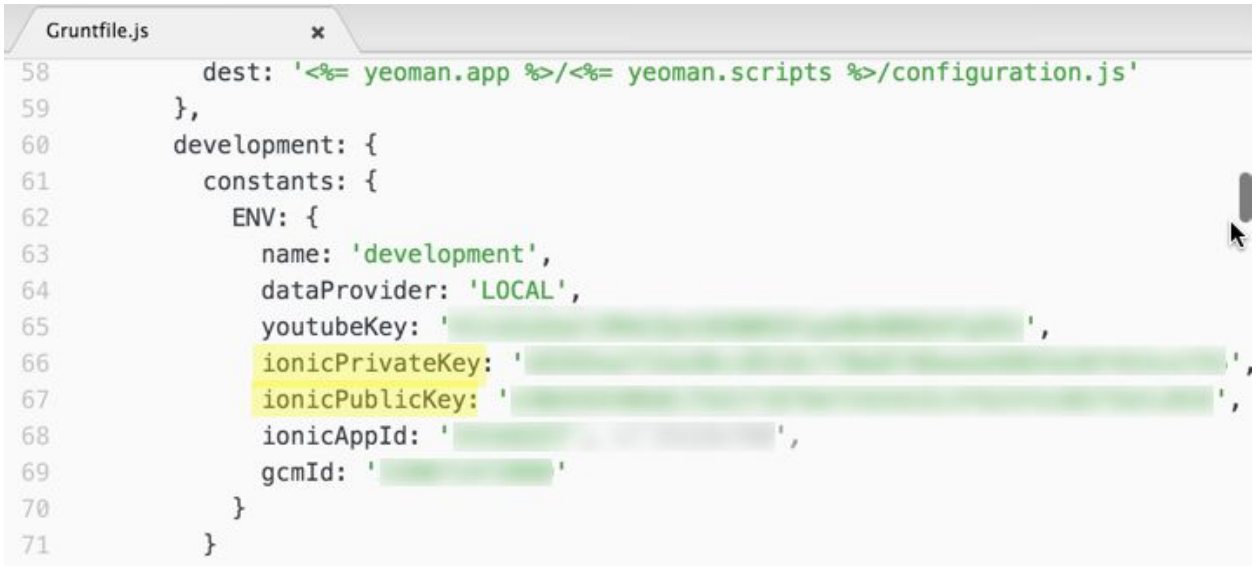


Keys

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



Gruntfile.js

Configuration

All the configuration of Business Directory app is done through the creation of the JSON files which include information about businesses, their news, products, catalogs, services and many more.

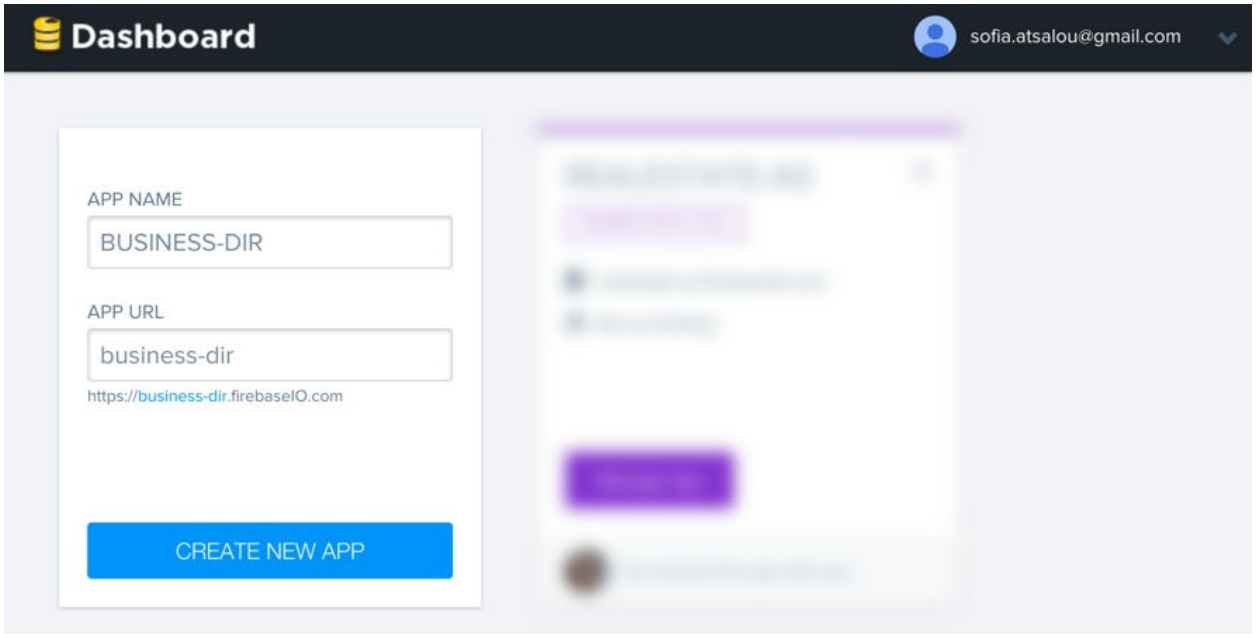
Data Sources Creation

Firebase

This app uses Firebase as a backend to read the data content it needs. In this section, you will find all the steps required for the app setup in order to be connected with Firebase.

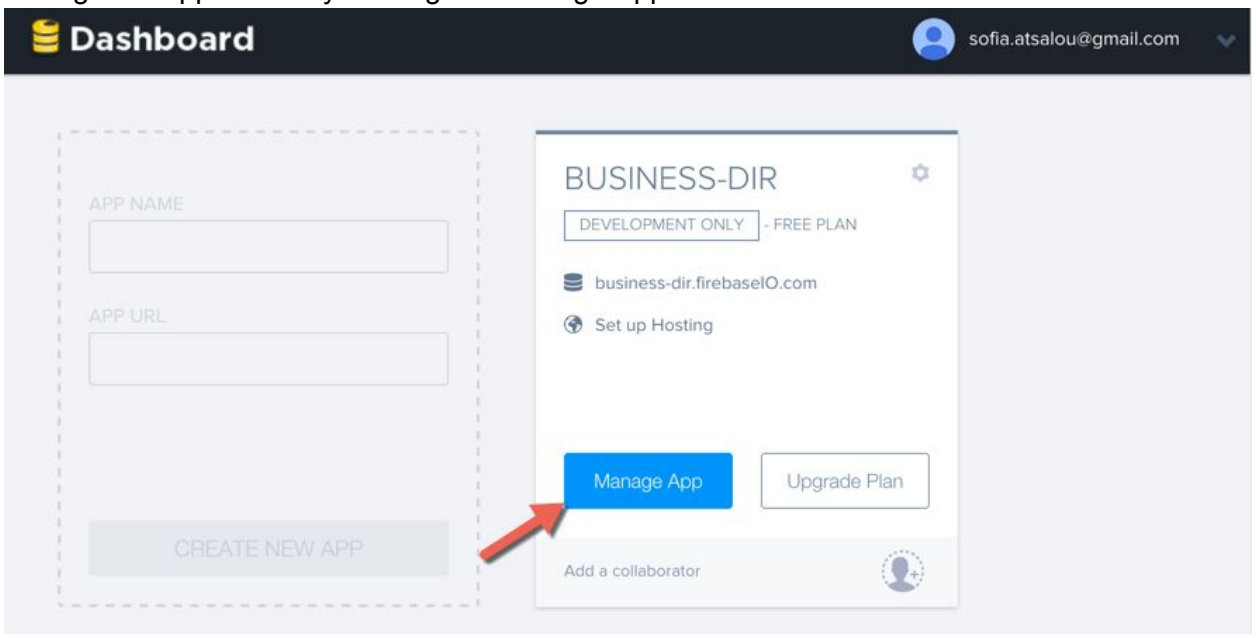
Create a Firebase app

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

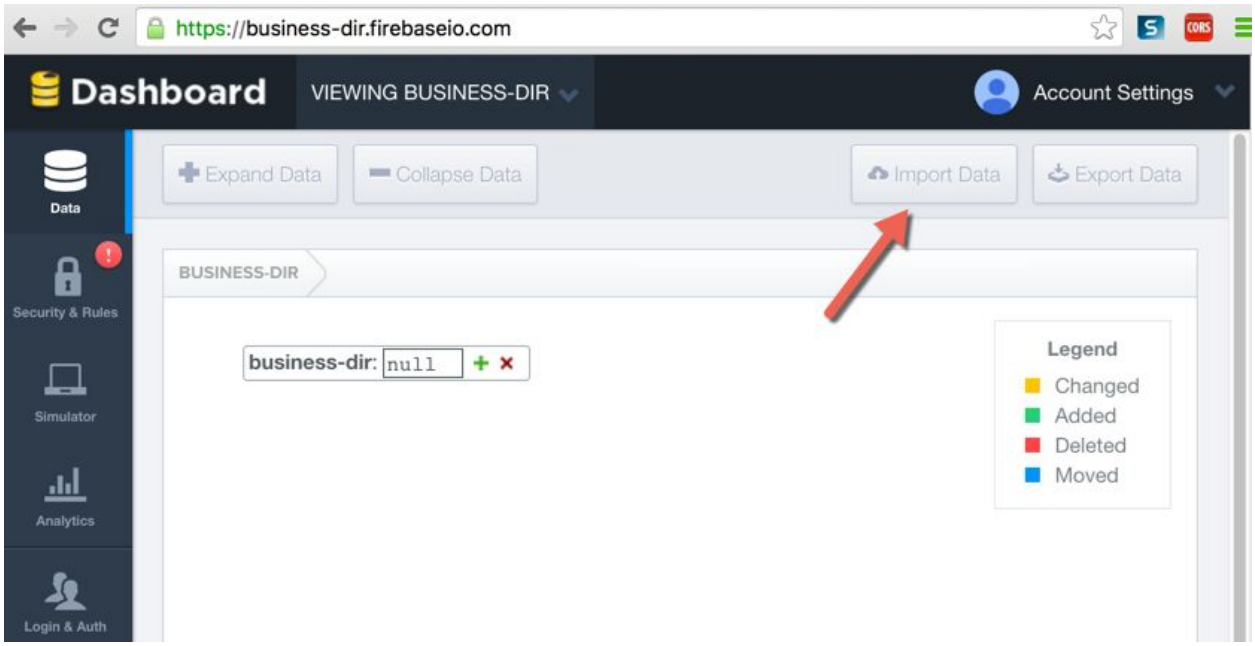


Create/Add data

Manage the app's data by clicking on "Manage App" button:



Click on the "Import Data" button in order to import your data.



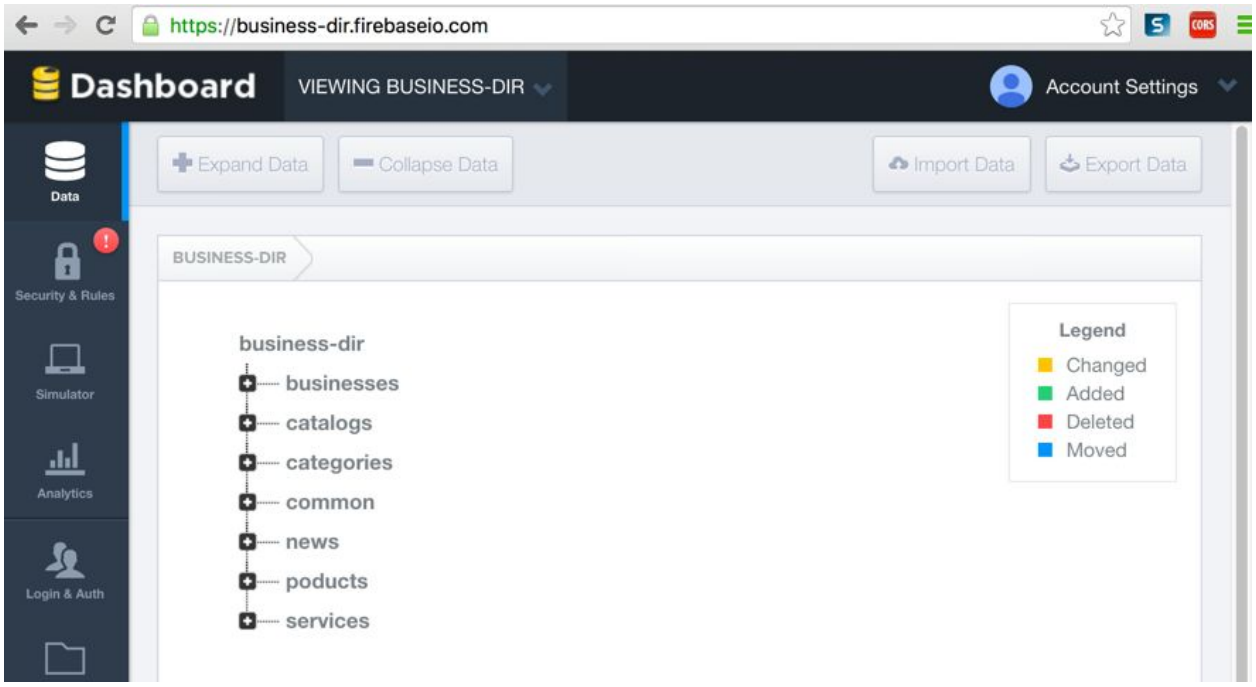
There is an example JSON file under `misc/_firebase/release` path which can be imported to the Firebase project created in the previous step. This file indicates the appropriate data structure for Business Directory.
The final JSON file for the Firebase data import should be similar to the following:

- https://drive.google.com/file/d/0B5d8Lqm1ViE1cW5mRjY3YjdvbUU/view?usp=sharing

It is worth noting that this JSON file represents a large JSON object which includes the following data:

- Businesses
- News
- Products
- Services
- Catalogs
- Common

These are explained in more detail in “JSON data sources” section.
At the end, the Firebase project should look like this:



Firebase Data source Configuration

To set the Firebase project our app will work with, you should set the “dataProvider” variable as “FIREBASE” and define the URL to your project in `Gruntfile.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          // LOCAL or REMOTE or FIREBASE
65          // dataProvider: 'REMOTE',
66          // apiUrl: 'https://skounis.s3.amazonaws.com/mobile-apps/business-directory/',
67          // dataProvider: 'LOCAL',
68          // apiUrl: 'misc/',
69          dataProvider: 'FIREBASE',
70          firebaseUrl: 'https://business-directory.firebaseio.com/',
71          youtubeKey: 'AIzaSyDael5MmCQa1GKQNKQYypmBeB08GATgSEo',
72          ionicPrivateKey: 'a9265eaf15a20cc8516c770e8748aeed4891b28f453ce755', //'c63b22
73          ionicPublicKey: 'e30d4d540b8c75d1f167bbf242423c3fb23fe10275d1c016', //'04dee95
74          ionicAppId: '241b6d37', //'2113c758',
75          gcmId: '228071472080'
76        }
```

JSON data sources

In case you chose Firebase as the app's data provider, you can skip this section. Otherwise, this section will guide you in order to make the app read the data it needs from a local or a remote source other than Firebase. In this case the sources should be JSON files located either locally or on a remote server.

Businesses

The example of the JSON file used in order to provide all the business related information is the following:

➔ http://skounis.s3.amazonaws.com/mobile-apps/business-directory/businesses.json

Notice that this is the file for all the businesses that are going to be displayed in the app. In this file, the addresses to some other JSON files included. These files are:

- `news.json`,
- `catalogs.json`,
- `products.json`
- `services.json`.

The next section describes more the mentioned JSON files.

Additionally, in this file information that is going to be shown on **Home** screen, such as business name, as well as **Contact us** screen, such as business description, open hours etc. is also included.

What's more, `businesses.json` file should contain map annotations that are going to be displayed on the **Map** screen. You are able to set as many annotation points as you wish.

Common

The `origin` property of the map sets the point where it will be centered. This is the **Map** where all the annotations are going to be displayed on.

Origin property together with the zoom level of the map are included in the `common.json` file as it is a piece of information that is in common for all the businesses.
The example of common.json file that is used for demonstrational purposes is the following:

➔ http://skounis.s3.amazonaws.com/mobile-apps/business-directory/common.json

News, Products, Services and Catalogs

Each business that will be displayed in the app should have its own news, products, services and catalogs JSON files.

As a point of reference, the app uses the following:

- News
  https://skounis-dev.s3.amazonaws.com/mobile-apps/business-directory/01/news.json
- Products
  https://skounis-dev.s3.amazonaws.com/mobile-apps/business-directory/01/products.json
- Services
  https://skounis-dev.s3.amazonaws.com/mobile-apps/business-directory/01/services.json
- Catalogs
  https://skounis-dev.s3.amazonaws.com/mobile-apps/business-directory/01/catalogs.json

These correspond to one business and, thus, they should be created for all the businesses.

Wordpress

This feature loads articles from a remote Wordpress site. The Wordpress JSON API is used for the creation of JSON feed of the posts. This should be included in the `businesses.json` file.

For demonstration purposes a Wordpress website has been installed. Its URL and JSON feed that is used in this application are provided by the following links:

- Site: http://demo.titaniumtemplates.com/wordpress/
- JSON: http://demo.titaniumtemplates.com/wordpress/?json=1

JSON API plugin

Please note that, 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.

Drupal

This feature loads articles from a remote Drupal web site. Services module is used for the creation of JSON feed of the posts which should be included in the `businesses.json` file.

For demonstration purposes a Drupal website has been installed. Its URL and JSON feed that is used in this application are provided by the following links:

- Site: http://demo.titaniumtemplates.com/drupal
- JSON: http://demo.titaniumtemplates.com/drupal/rest/views/rest_api

Services and JSON View

Using Drupal, initially, you should install the Services module and enable it. A View that exposes a JSON feed of the articles should also be created.

The following tutorial describes this process:

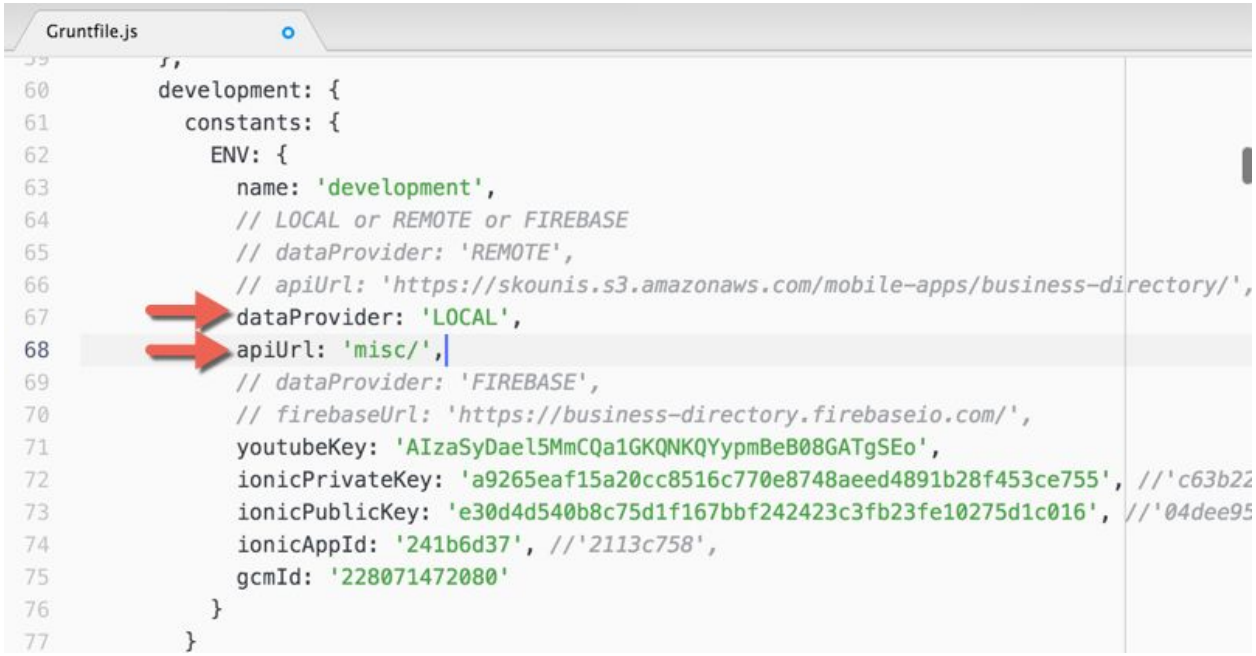
- A Beginners Guide to the Drupal Services Module
  https://www.ostraining.com/blog/drupal/services/

All the modules that are needed for this are listed below:

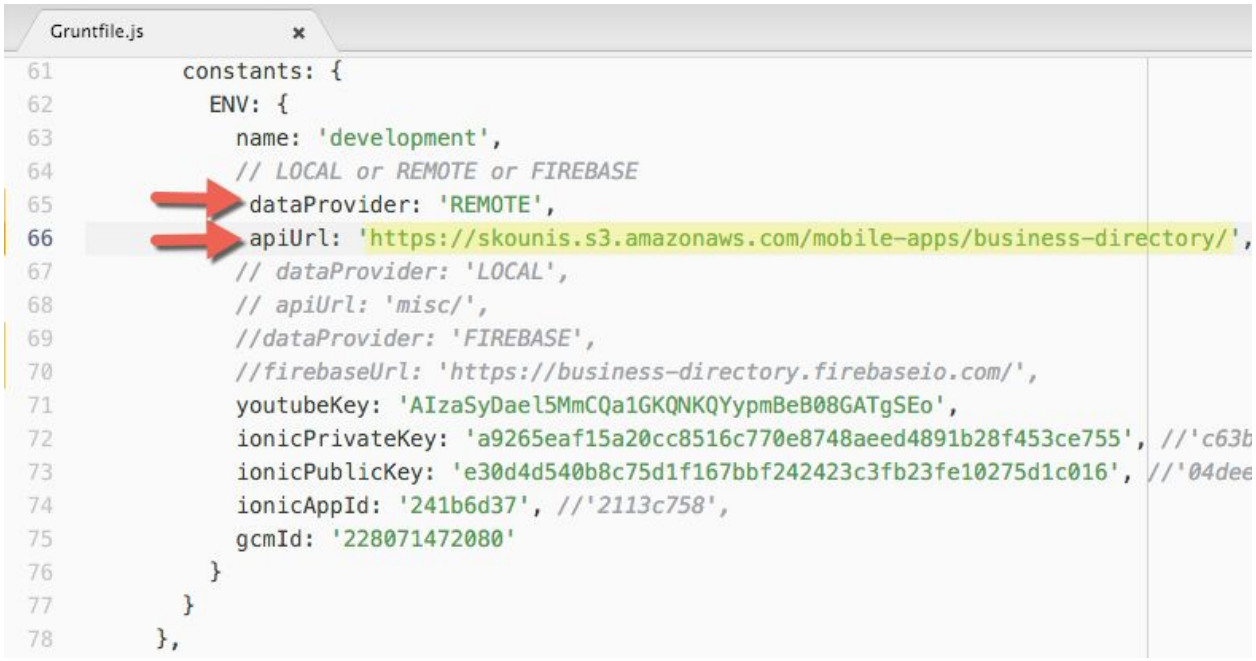
1. https://www.drupal.org/project/services
2. https://www.drupal.org/project/ctools
3. https://www.drupal.org/project/libraries
4. https://www.drupal.org/project/views
5. https://www.drupal.org/project/services_views

JSON Data Sources Configuration

In case you choose to fetch the data from a local location, you should locate the `businesses.json` and `common.json` files in `misc` folder and set the "dataProvider" variable as "LOCAL" in the `Gruntfile.js`. Make sure you uncomment the related lines as shown below:



In case you choose to fetch the data from a remote location, in `Gruntfile.js` file set the "dataProvider" variable as "REMOTE". Make sure you uncomment the related lines and set the URL to the folder where the `businesses.json` and `common.json` files are located.



Support

In regard to technical questions, new ideas and suggestions, you may use the dedicated Google Group that has been created for this product:

https://groups.google.com/forum/#!forum/titemplates-support

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.

--- *The Appseed team.*