

Real Estate Ionic 1.0 Documentation

Quick start Guide - Installation and user support

Created: Apr 07, 2016

Updated: Apr 07, 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](#)

[Cordova 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](#)

[Themes - Customize the appearance](#)

[Activate a theme](#)

[Configuration](#)

[Data Sources Creation](#)

[Firebase](#)

[Create a Firebase app](#)

[Create/Add data](#)

[Firebase Data source Configuration](#)

[JSON Data sources](#)

[Properties](#)

[Common](#)

[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. If NodeJS is not installed, please install the latest LTS version of NodeJS.

For more information, please refer to the “[Getting Started with Ionic](#)” section, at the official Ionic site.

Cordova and Ionic

This is a [Cordova](#) and [Ionic](#) based application, so [Cordova](#) and [Ionic](#) should be installed on your computer for this application to run. Since Real Estate 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.

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:

★ Macintosh HD > Users > sofia > Downloads > RealEstateIonic-1.0		
Name	Size	Modified
▶ release	--	Friday 11 March 2016
▶ project	--	Friday 11 March 2016
▶ licences	--	Friday 12 February 20
▶ README.txt	1.8 KB	Friday 11 March 2016
▶ Documentation - Quick Start Guide.pdf	1.4 MB	Friday 11 March 2016

The highlighted folder is the Ionic Project's directory.

Install libraries

Open a terminal window and navigate to `project` directory and install NodeJS dependencies:

```
$ 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` folder.

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
```

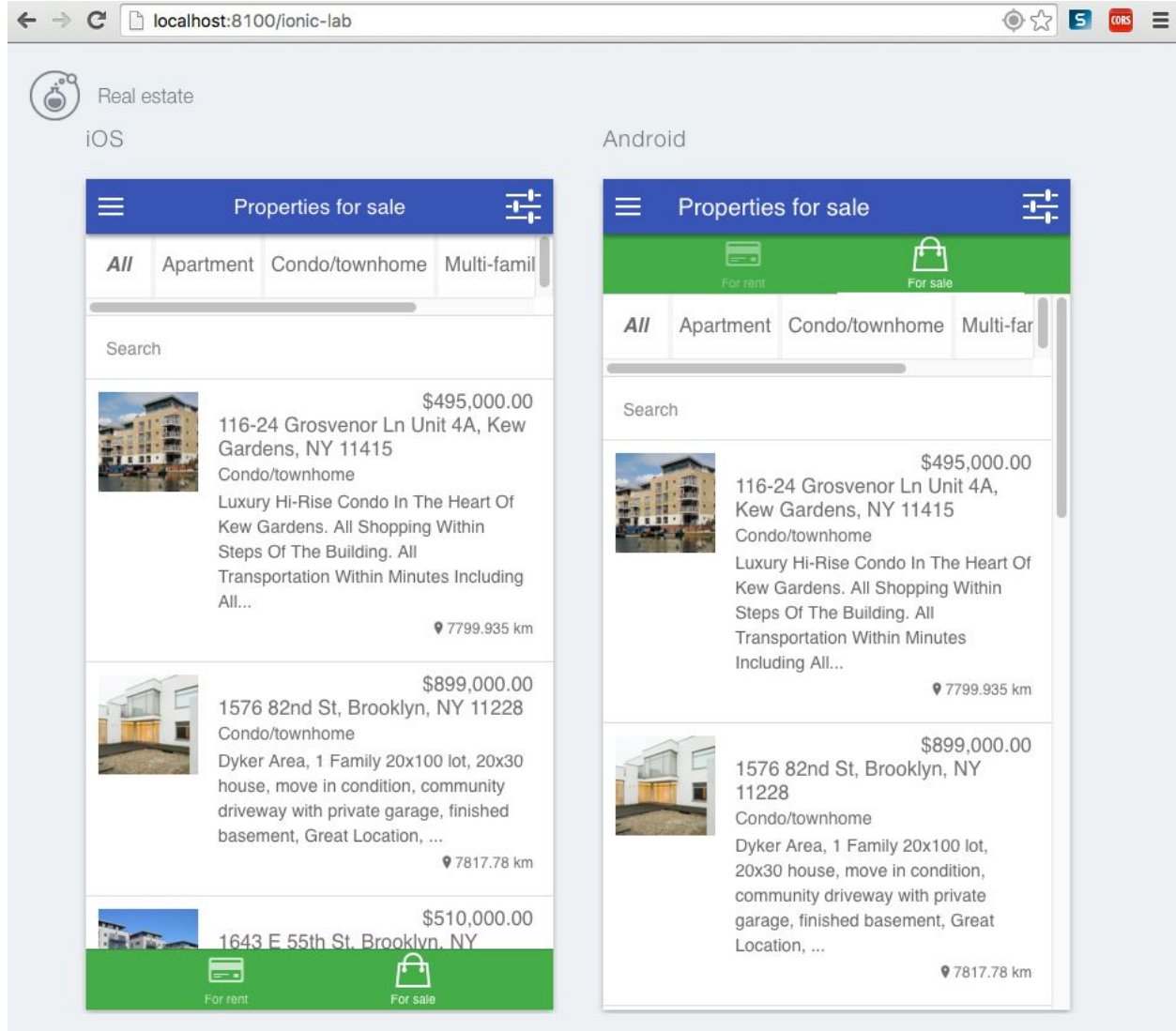
Run/Build the application (after the first-time preparations)

Run a local development server

Navigate to `project` folder 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

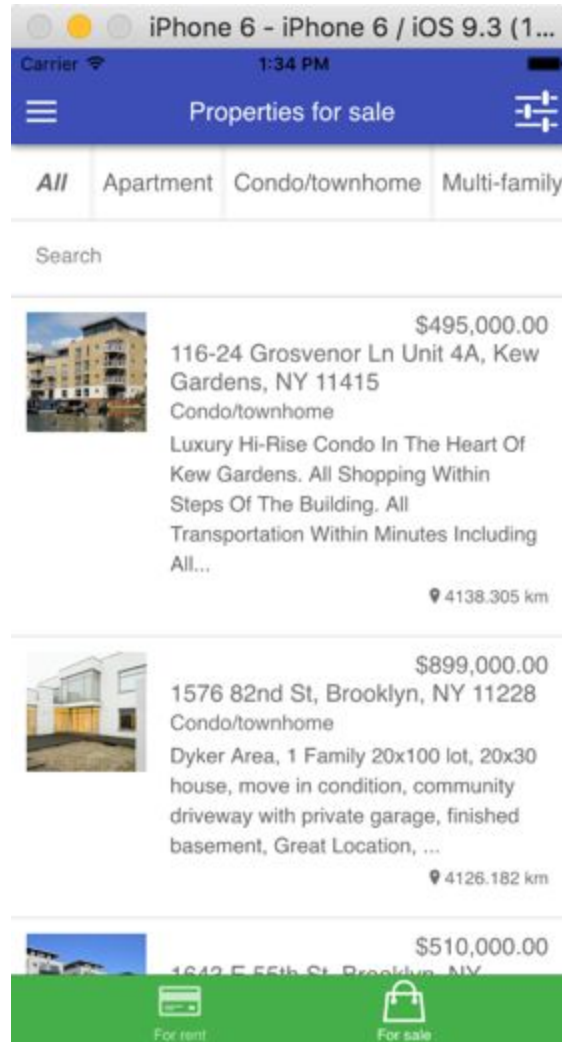
First the preferred platform should be added. In this case iOS:

```
$ 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 Real Estate 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:



```
1 {  
2   "name": "realestate",  
3   "app_id": "7c75efbd"  
4 }  
5
```

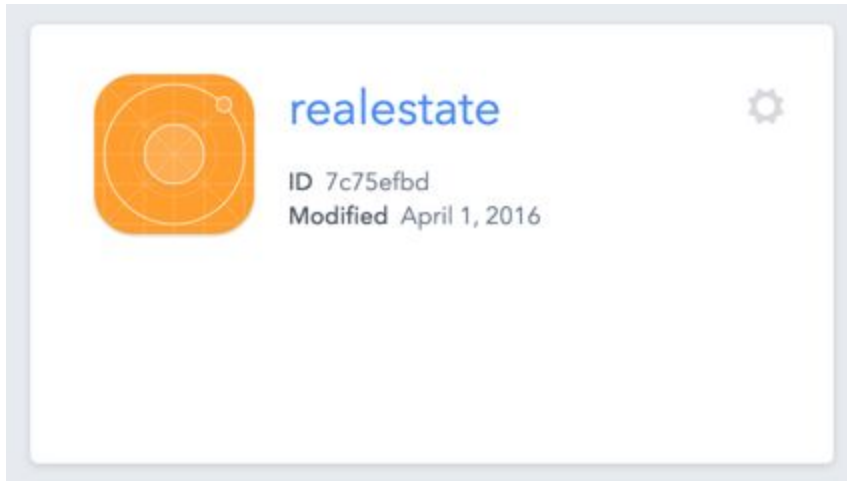
ionic.project



```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
2 <widget id="com.titaniumtemplates.realestate" version="1.0.0" xmlns="http://w  
3   <name>Real estate</name>  
4   <description>  
5     Real estate  
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 corresponding application in your ionic.io account.



Keys

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

A screenshot of the Ionic.io 'API Keys' settings page. The left sidebar shows a menu with options: Push, Analytics, Deploy, Users, and Settings (highlighted with a red arrow). The main content area is titled 'API Keys' and includes a brief explanation: 'API keys allow your application to communicate with the ionic.io services. You can create multiple API keys that have different permissions to restrict the actions your apps can perform.' Below this is a table with two columns: 'KEY' and 'DESCRIPTION'. The table contains two entries: a 'Public Key' (labeled 'Master API Key') and a 'Secret Key' (with a 'Show secret' link). A 'DELETE' button is visible next to the Public Key. At the bottom of the table is a '+ New API Key' button.

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

```
Gruntfile.js x
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/realestate/',
67             // dataProvider: 'LOCAL',
68             // apiUrl: 'misc/',
69             dataProvider: 'FIREBASE',
70             firebaseUrl: 'https://realestate-as.firebaseio.com/',
71
72             ionicPrivateKey: 'a9265eaf15a20cc8516c770',
73             ionicPublicKey: 'e30d4d540b8c75d1f167bb',
74             ionicAppId: '241b6d37', // '2113c758',
75             gcmId: '22807'
76         }
77     }
```

Gruntfile.js

Themes - Customize the appearance

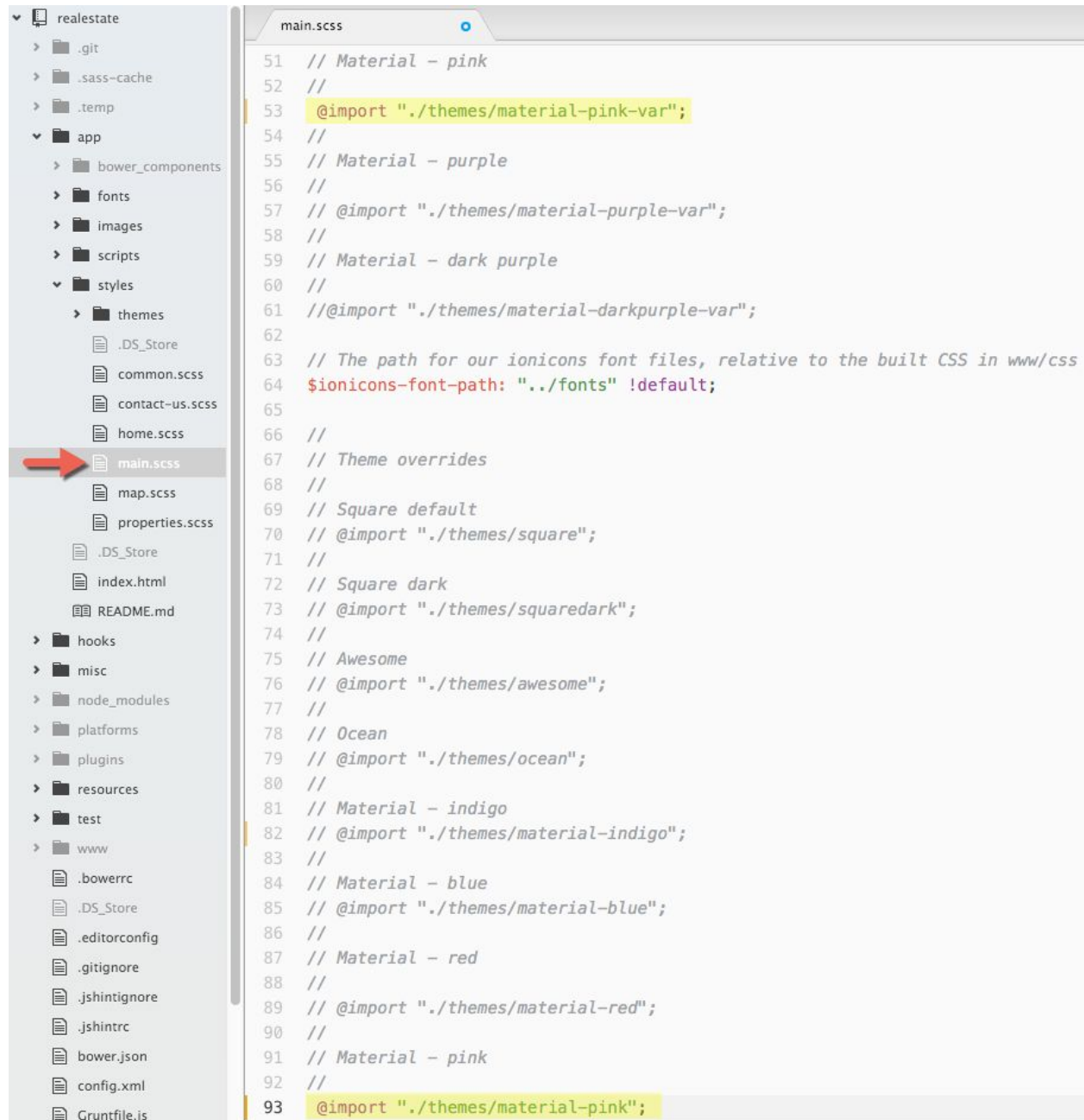
Real Estate Ionic app comes with theme options ready to be used. Each theme consists of two `.scss` files:

1. Variables: `{theme}-var.scss`
2. Overrides: `{theme}.scss`

These files are located under `app/styles/` path and are used in `main.scss` file where a theme can be activated as described in the next section.

Activate a theme

In order to activate a theme, edit `main.scss` file under `app/styles/` path and uncomment the theme you opt for. For example, in case of selecting “Ocean” theme, you should uncomment the lines showing below:



main.scss

Save the changes and the theme is all set up.

Configuration

All the configuration of the Real Estate app is done through the creation of the JSON files which include information about the properties, their features, amenities, contact info etc.

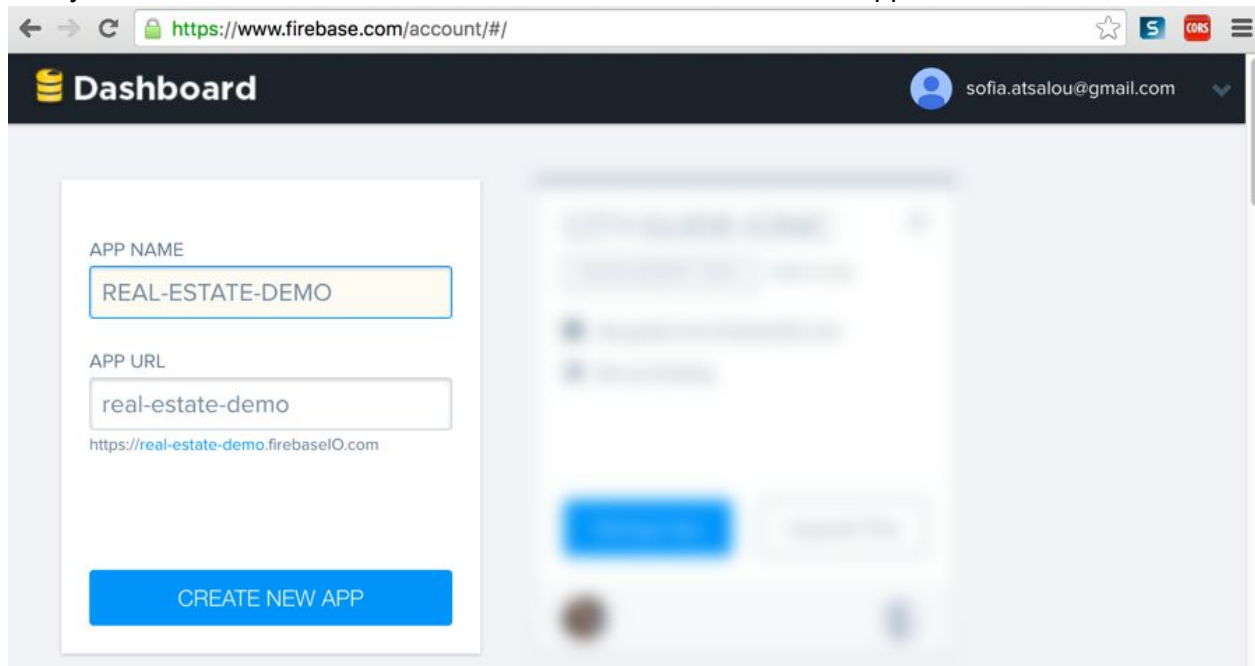
Data Sources Creation

Firestore

This app uses Firestore 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 Firestore.

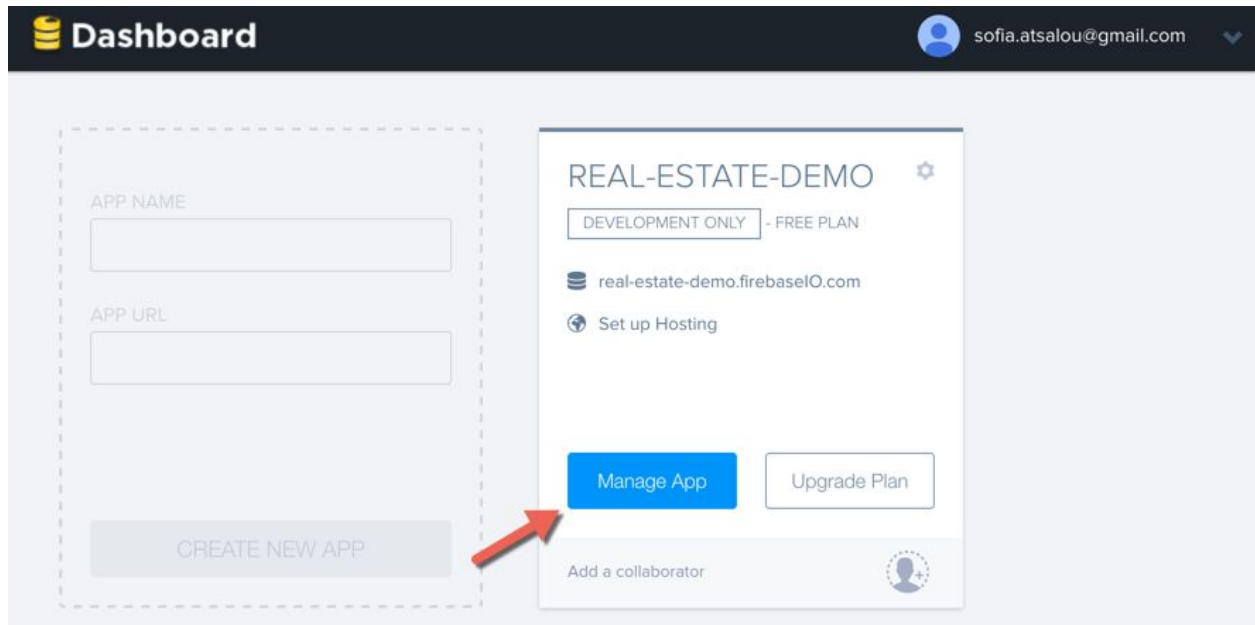
Create a Firestore app

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

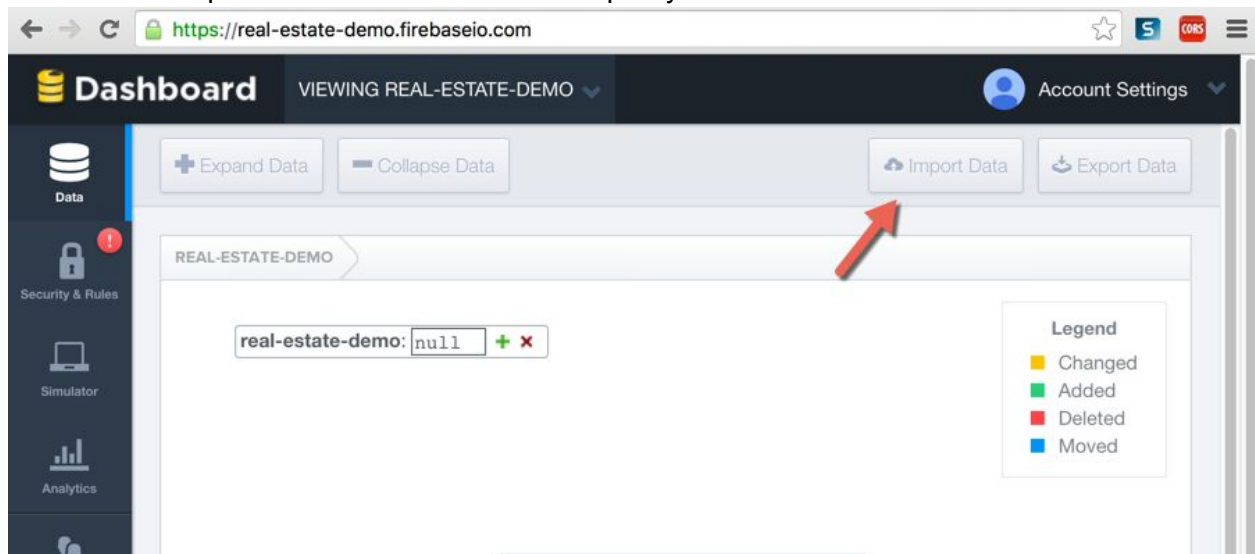
A screenshot of the Firebase console's 'Create new app' form. The browser address bar shows 'https://www.firebase.com/account/#/'. The page has a dark header with the 'Dashboard' title and a user profile for 'sofia.atsalou@gmail.com'. The main content area contains a form with two input fields: 'APP NAME' with the value 'REAL-ESTATE-DEMO' and 'APP URL' with the value 'real-estate-demo'. Below the URL field, the text 'https://real-estate-demo.firebaseio.com' is displayed. A large blue button labeled 'CREATE NEW APP' is at the bottom of the form. To the right of the form, there is a blurred preview of the app's configuration page.

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.



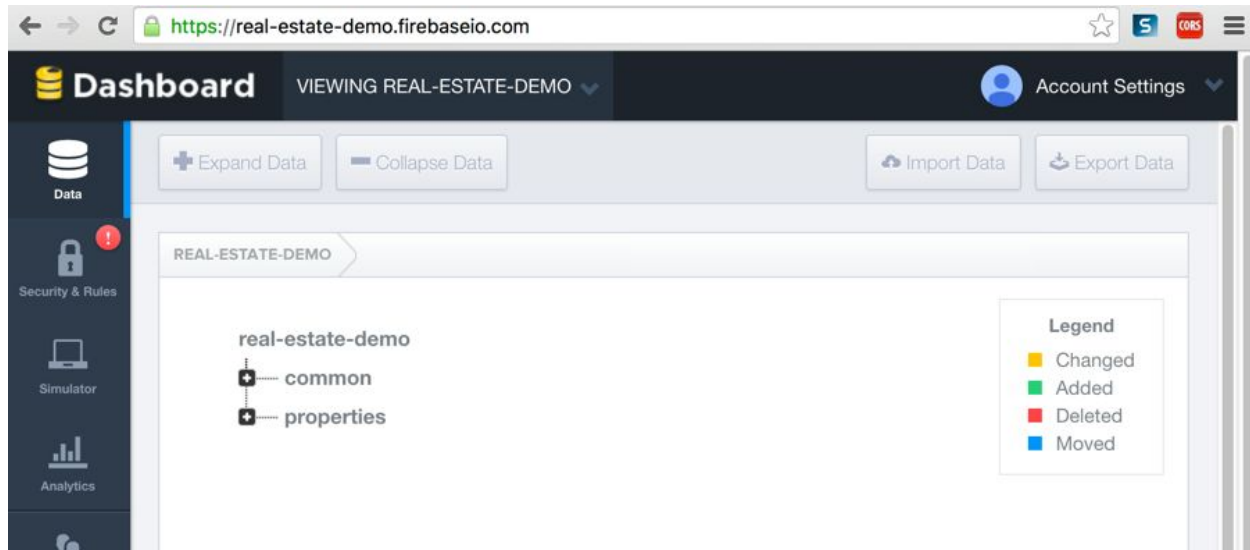
There is an example JSON file under `misc/_firebase/release` path which can be imported to the Firebase project created in the previous step. This file indicates the appropriate data structure for Real Estate.

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

- Properties
- 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
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-dev.s3.amazonaws.com/mobile-apps/realestate/',
67         // dataProvider: 'LOCAL',
68         // apiUrl: 'misc/',
69         dataProvider: 'FIREBASE',
70         firebaseUrl: 'https://realestate-as.firebaseio.com/',
71
72         youtubeKey: 'AIzaSyDael5MmCQa1GKQNKQYypmBeB08GATgSEo',
73         ionicPrivateKey: 'a9265eaf15a20cc8516c770e8748aeed4891b28f453ce755', //'c6
74         ionicPublicKey: 'e30d4d540b8c75d1f167bbf242423c3fb23fe10275d1c016', //'04c
75         ionicAppId: '241b6d37', //'2113c758',
76         gcmId: '228071472080'
77       }

```

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.

Properties

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

- <http://skounis.s3.amazonaws.com/mobile-apps/realestate/properties.json>

In this file, information that is going to be shown on **Home** screen, such as property address and price, as well as **Contact us** screen, such as property description, telephone number, email etc. is also included.

Additionally, `properties.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 properties.

The example of `common.json` file that is used for demonstrational purposes is the following:

- <http://skounis.s3.amazonaws.com/mobile-apps/realestate/common.json>

JSON Data Sources Configuration

In case you choose to fetch the data from a local location, you should locate the `properties.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:


```

Gruntfile.js
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-dev.s3.amazonaws.com/mobile-apps/realestate/',
67         dataProvider: 'LOCAL',
68         apiUrl: 'misc/',
69         //dataProvider: 'FIREBASE',
70         //firebaseUrl: 'https://realestate-as.firebaseio.com/',
71
72         youtubeKey: 'AIzaSyDael5Mr...',
73         ionicPrivateKey: 'a9265eaf15a20cc8516c77...', // 'c6
74         ionicPublicKey: 'e30d4d540b8c75d1f167bb...', // '04c
75         ionicAppId: '241b6d37', // '2113c758',
76         gcmId: '2280...'
77       }
78     }
79   }

```

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.

```

Gruntfile.js
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', // 'c63b
73         ionicPublicKey: 'e30d4d540b8c75d1f167bbf242423c3fb23fe10275d1c016', // '04dee
74         ionicAppId: '241b6d37', // '2113c758',
75         gcmId: '228071472080'
76       }
77     }
78   },

```


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.

--- *The Appseed team.*