City Guide Ionic 1.0 Documentation

Quick start Guide - Installation and user support

Created: Jul 28, 2016

Updated: Jul 28, 2016

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

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

Table of contents:

What's in the Pack

After extracting it, the downloaded .zip file looks like this:

It contains:

1. **Quick Start Guide**: Documentation with instructions on how to install, configure and personalize the applications.
2. **Licences folder**: Terms and conditions for use, reproduction and distribution of this software piece.
3. **Project folder**: Includes a Grunt based ionic/cordova project. This is the project that is suggested to be used for development. Also, all the instructions in this document refer to this folder.
4. **Release folder**: Includes a generated Ionic project. This is a simplified version of the Grunt based project where all the grunt related automations and dependencies are stripped out.

Note: For the instructions in this document, we will use `project` folder.

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 this app 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:

The highlighted folder is the Ionic Project's directory.

Install libraries

Open a terminal window and navigate to `project` directory and 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` 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
```

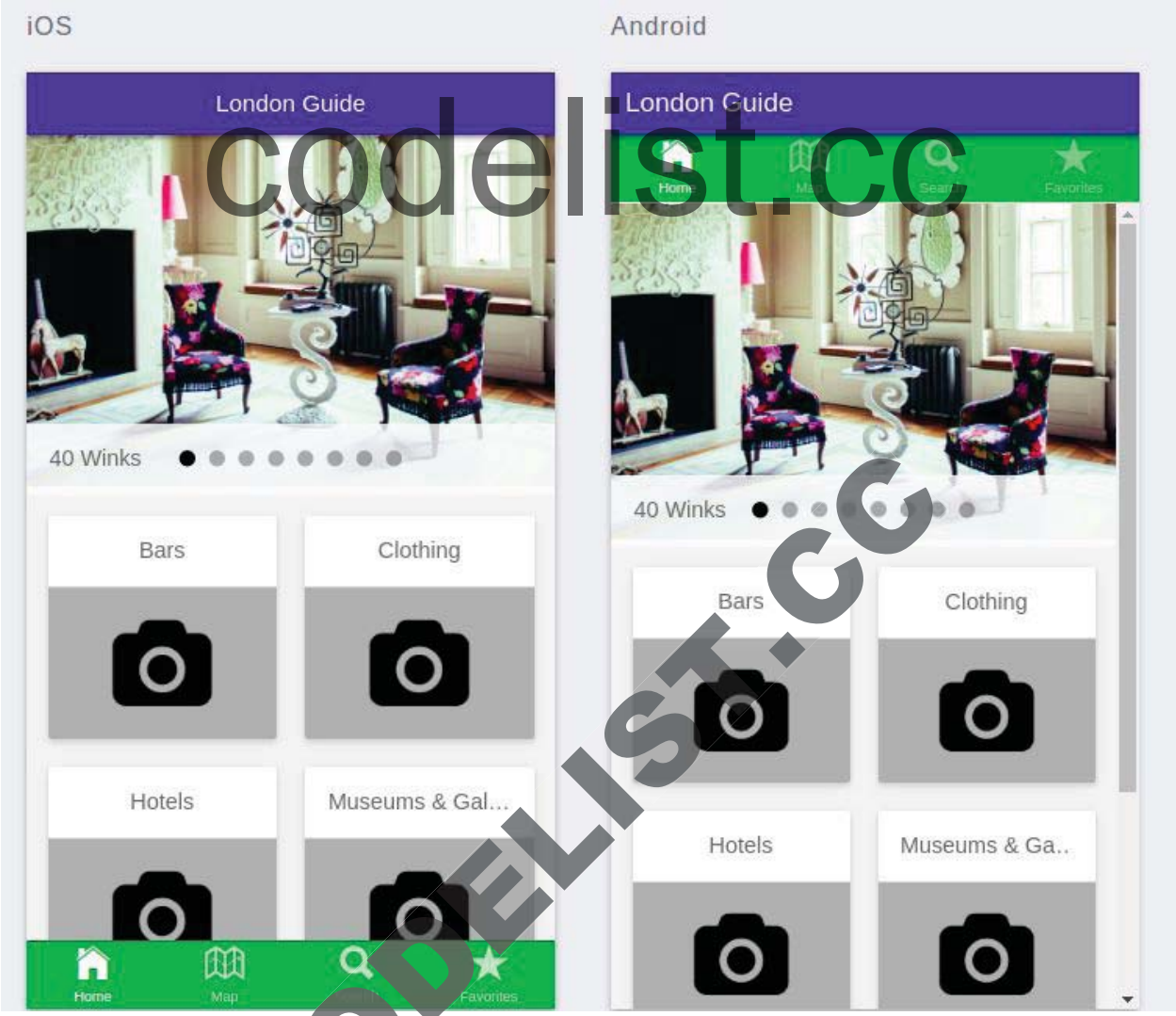
Run/Build the application (after the first-time preparations)

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

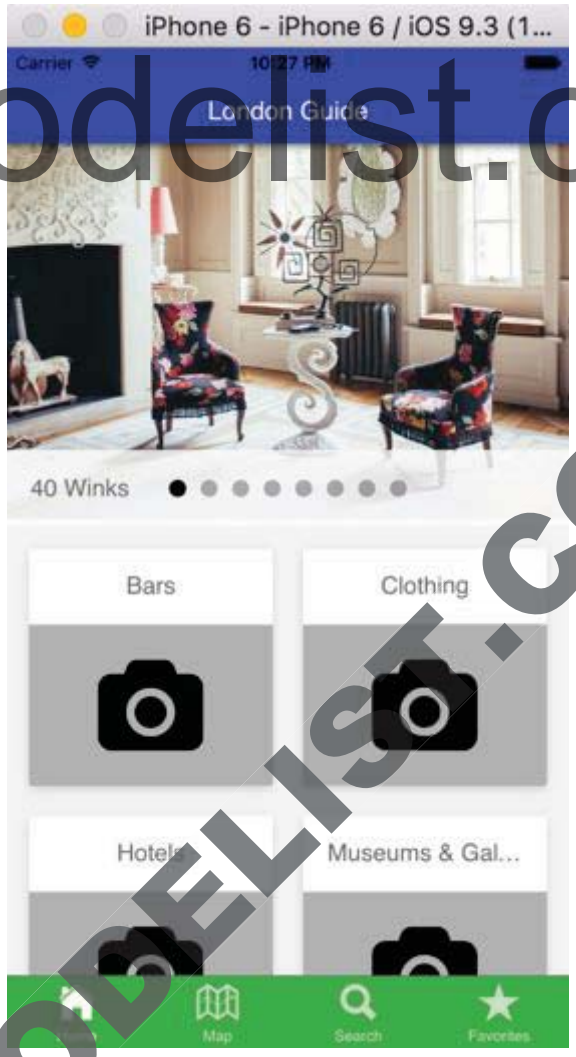
First the prefered platform should be added. In this case iOS:

```
$ 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 City Guide 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": "listing",
  "app_id": "db8af23d"
}
```

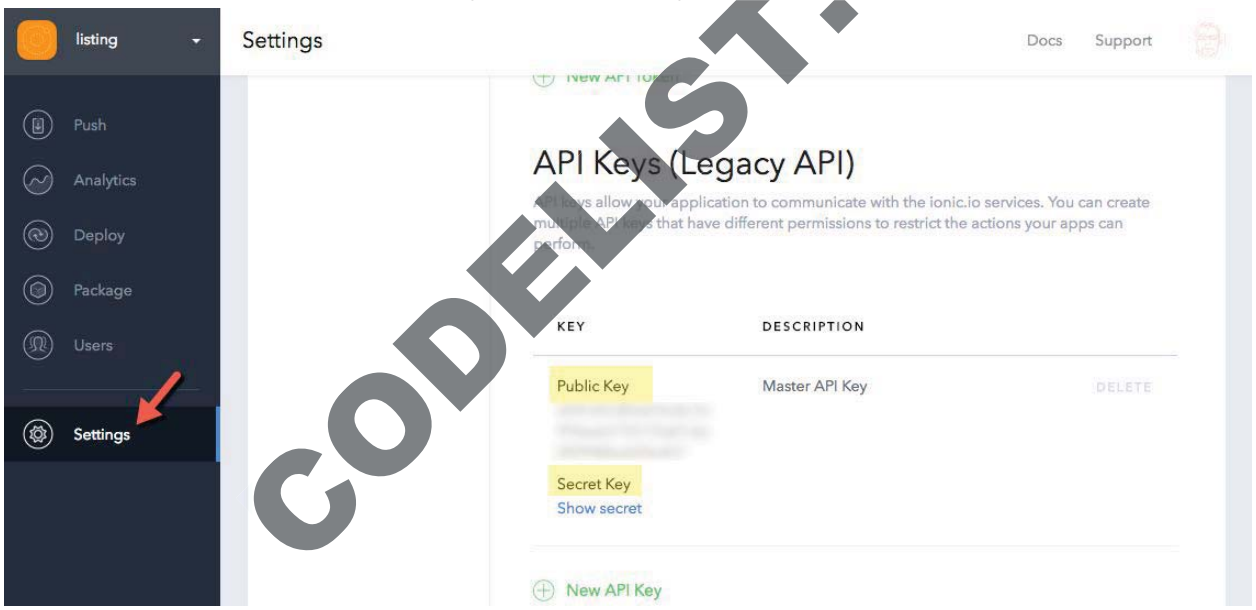

config.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<widget id="com.titaniumtemplates.business-directory" version="1.3.0" xmlns="ht
  <name>City Guide</name>
  <description>
        City Guide. The boostrap you need in order to build your next Ionic app
    </description>
  <author email="skounis@gmail.com" href="http://about.me/stavros.kounis">
        Stavros Kounis
    </author>
  <content src="index.html"/>
```

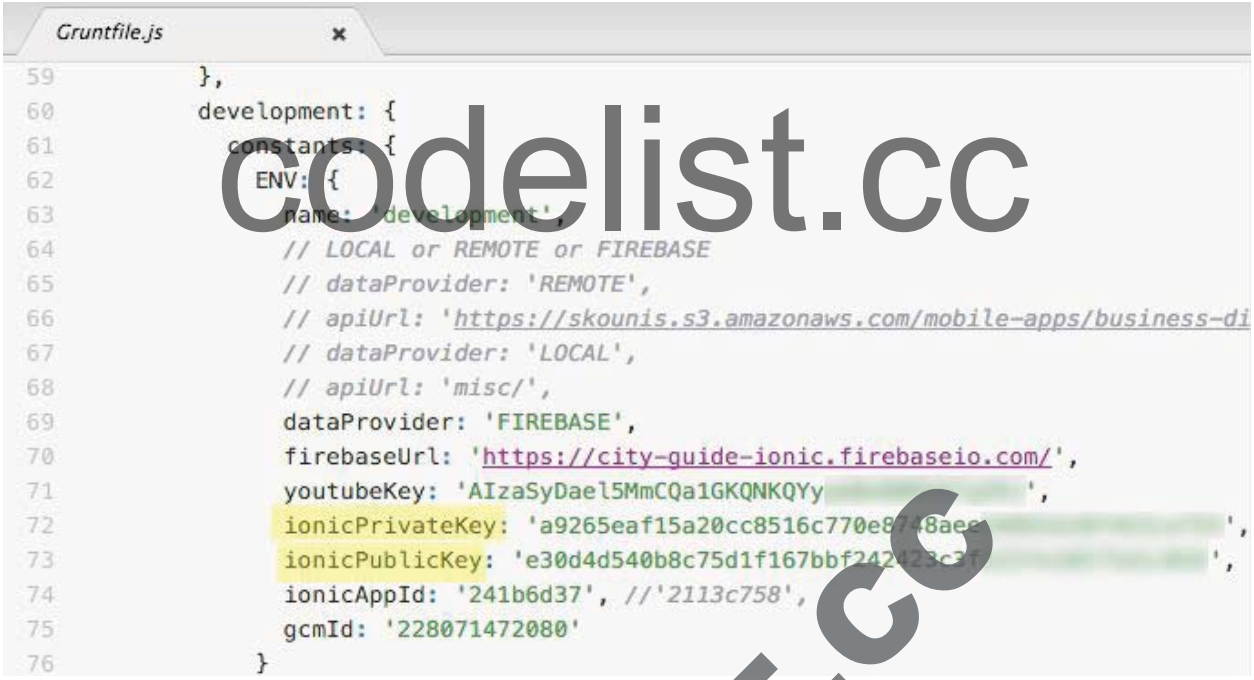
As for the app_id, use the ID of the coresponding application in your ionic.io account.

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

```
},
development: {
  constants: {
    ENV: {
      name: 'development',
      // LOCAL or REMOTE or FIREBASE
      // dataProvider: 'REMOTE',
      // apiUrl: 'https://skounis.s3.amazonaws.com/mobile-apps/business-di
      // dataProvider: 'LOCAL',
      // apiUrl: 'misc/',
      dataProvider: 'FIREBASE',
      firebaseUrl: 'https://city-guide-ionic.firebaseio.com/',
      youtubeKey: 'AIzaSyDael5MmCQa1GKQNKQYy                    ',
      ionicPrivateKey: 'a9265eaf15a20cc8516c770e8748aes                    ',
      ionicPublicKey: 'e30d4d540b8c75d1f167bbf242423c3f                    ',
      ionicAppId: '241b6d37', //'2113c758',
      gcmId: '228071472080'
    }
```


Themes - Customize the appearance

City Guide 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

Category Images and City Name

Each of the business/listing belongs to a certain category, e.g. Museums. The category images and the city name are displayed on the home screen and can be set in the `home.service.js` file under the `app/scripts/home` path.

home.service.js

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

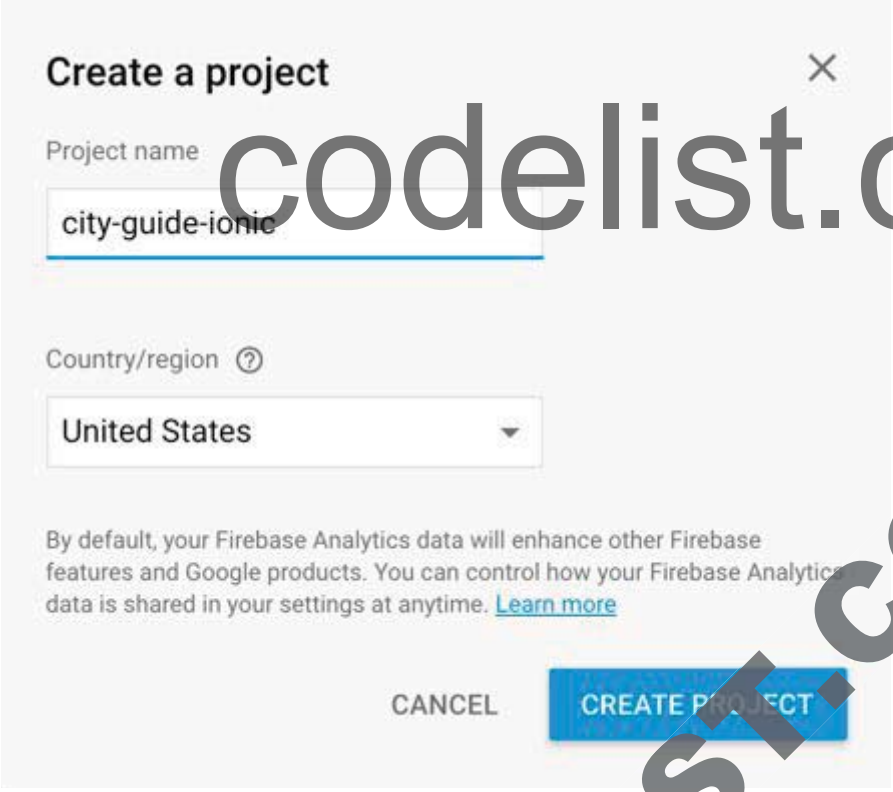
Data Sources Creation

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.

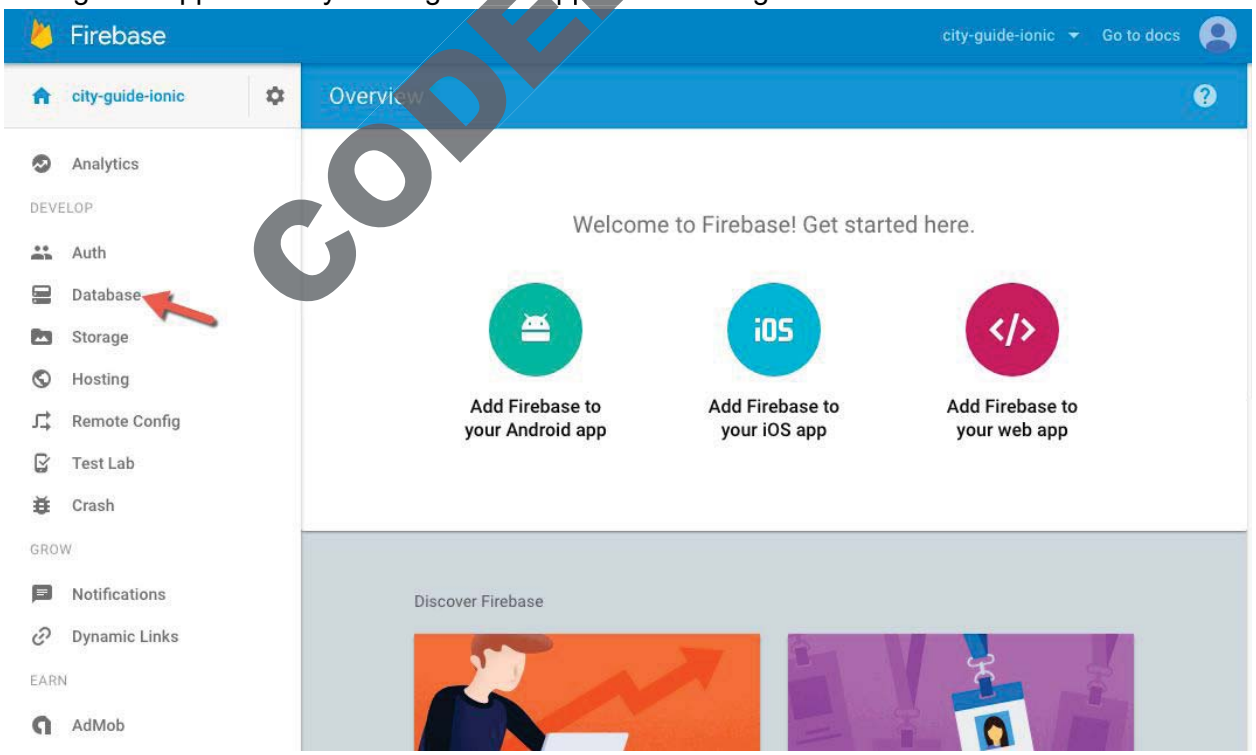
Create a Firebase app

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

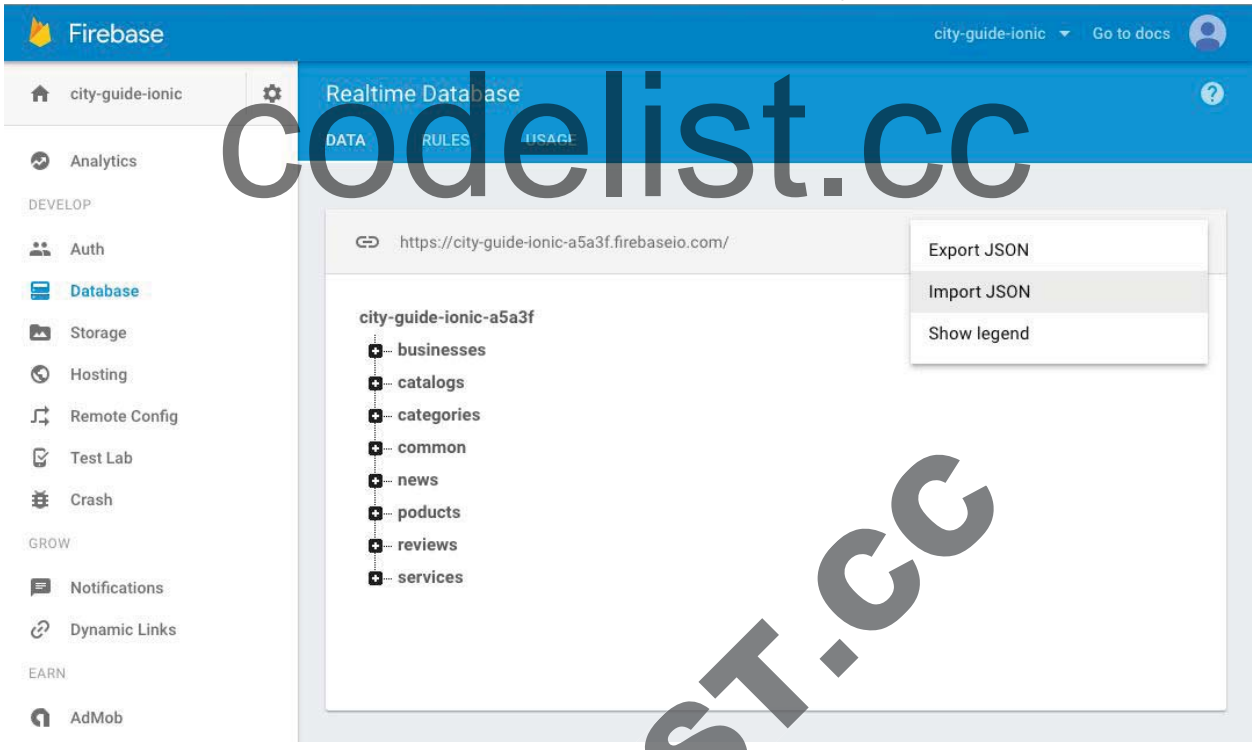


Create/Add data

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



Then, choose the “Import JSON” option from the menu 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 City Guide.

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

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

These are explained in more detail in “JSON Data sources” section.

Firebase Data source Configuration

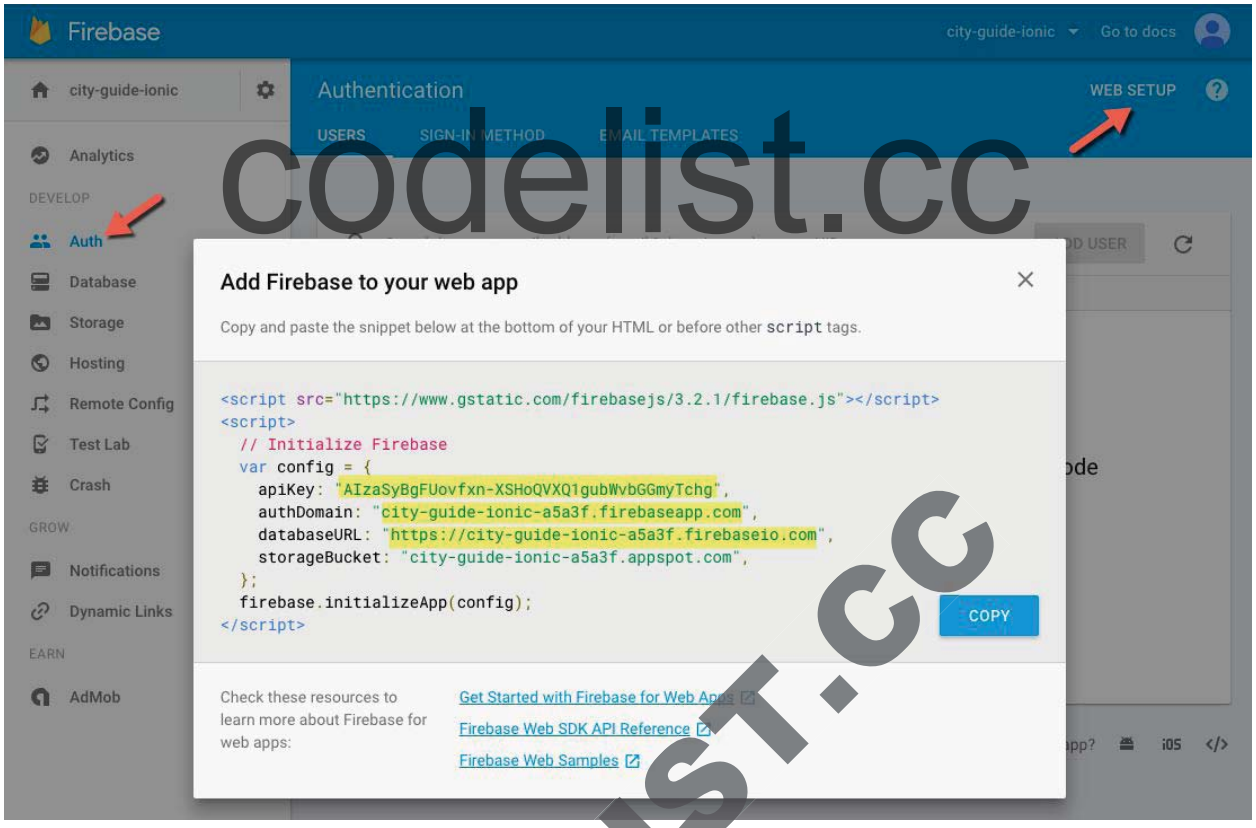
To set the Firebase project the app will work with, you should set the “dataProvider” variable as “FIREBASE” and define the “apiKey”, “authDomain” and “databaseURL” in `Gruntfile.js`:

Gruntfile.js

```
59          },
60          development: {
61            constants: {
62              ENV: {
63                name: 'development',
64                // LOCAL or REMOTE or FIREBASE
65                // dataProvider: 'REMOTE',
66                // apiUrl: 'https://skounis-dev.s3.amazonaws.com/mobile-apps/city-gu
67                // dataProvider: 'LOCAL',
68                // apiUrl: 'misc/',
69                dataProvider: 'FIREBASE',
70                firebaseConfig: {
71                  apiKey: "AIzaSyBgFUovfxn-XSHoQVXQ1gubWvbGGmyTchg",
72                  authDomain: "city-guide-ionic-a5a3f.firebaseapp.com",
73                  databaseURL: "https://city-guide-ionic-a5a3f.firebaseio.com"
74                },
75                youtubeKey: 'AIzaSyDael5MmCQa1GKQNKQYymBeB08GATgSEo',
76                ionicPrivateKey: 'a9265eaf15a20cc8516c770e8748aeed4891b28f453ce755',
77                ionicPublicKey: 'e30d4d540b8c75d1f167bbf242423c3fb23fe10275d1c016',
78                ionicAppId: '241b6d37', //'2113c758'
79                gcmId: '228071472080'
80              }
```

Gruntfile.js

In order to find the “apiKey”, “authDomain” and “databaseURL” select “Auth” from the side menu on the Firebase console and, then, click on “WEB SETUP”:



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 at the following path:

```
project/misc/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 `businesses.json` file, there is information such as business name, business description, open hours, pictures etc.

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 located under the path:

`project/misc/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 JSON files located under the following paths:

- News
  `project/misc/01/news.json`
- Products
  `project/misc/01/products.json`
- Services
  `project/misc/01/services.json`
- Catalogs
  `project/misc/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:

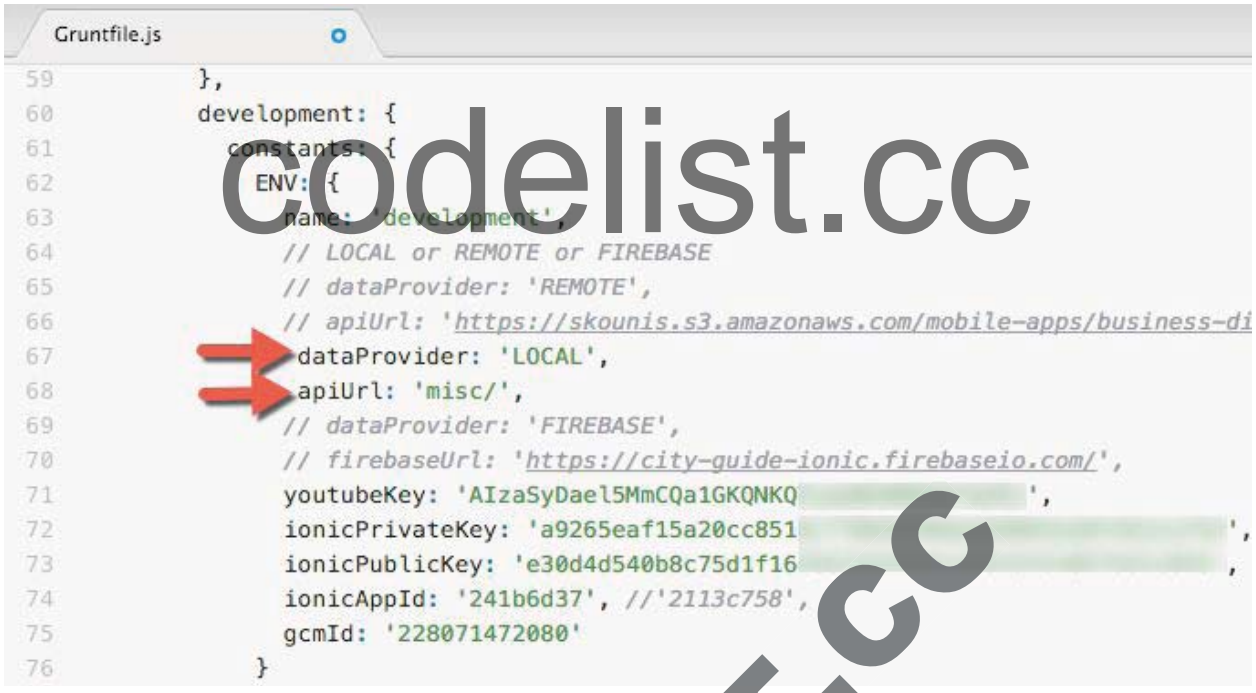
- A Beginners Guide to the Drupal Services Module
  https://www.ostraining.com/blog/drupal/services/

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

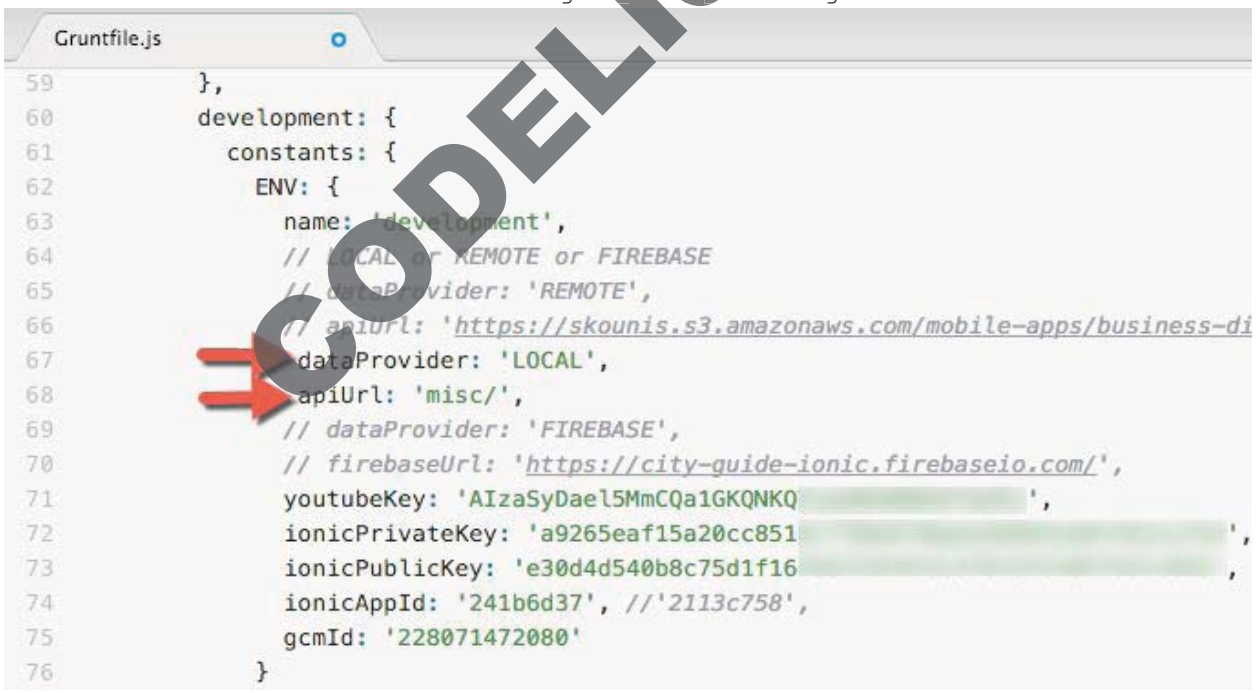
1. https://www.drupal.org/project/services
2. https://www.drupal.org/project/ctools
3. https://www.drupal.org/project/libraries
4. https://www.drupal.org/project/views
5. https://www.drupal.org/project/services_views

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

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 this YouTube channel.
- Codecanyon User page
  You may contact us by using our user page on Codecanyon.
- Quick Start Guide
  The online version of this document.

Thank you

Thank you again for purchasing our product. If you have any questions that are beyond of the scope of this help file, please feel free to email also via our support center form.

--- *The Appseed team.*