Catalogue Ionic 1.4 Documentation

Quick start Guide - Installation and user support

Created: Nov 25, 2015

Updated: Oct 25, 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:

Prerequisites

PhoneGap and Ionic

This is a PhoneGap and Ionic based application, so the PhoneGap and Ionic should be installed in your computer. Since Catalogue 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.

Ensure first that NodeJS and GIT 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:

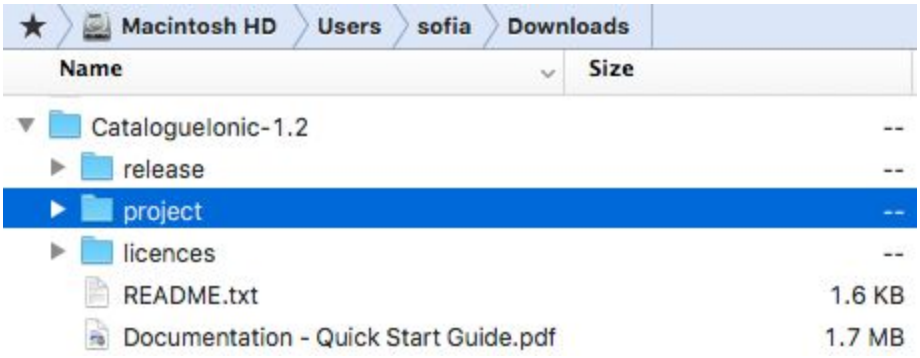
```
$ npm install -g generator-ionic
```

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:



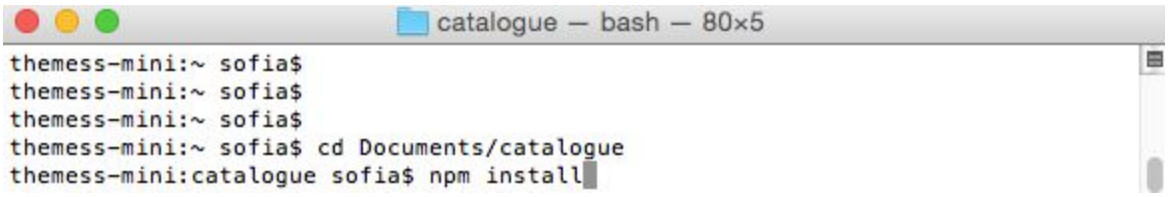
The highlighted folder is the Ionic Project's directory.

Install libraries

Open a terminal window and navigate to the directory where the “catalogue” 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
```

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:

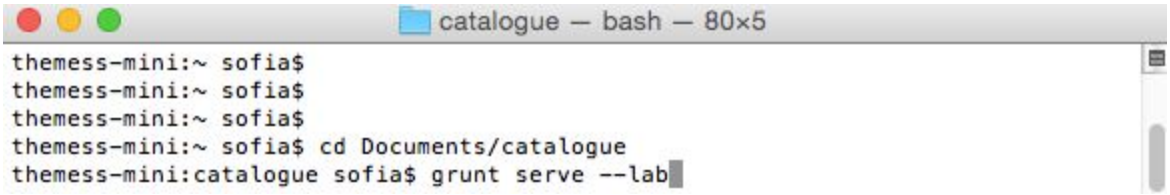
```
$ cordova plugin add cordova-plugin-inappbrowser
```

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.

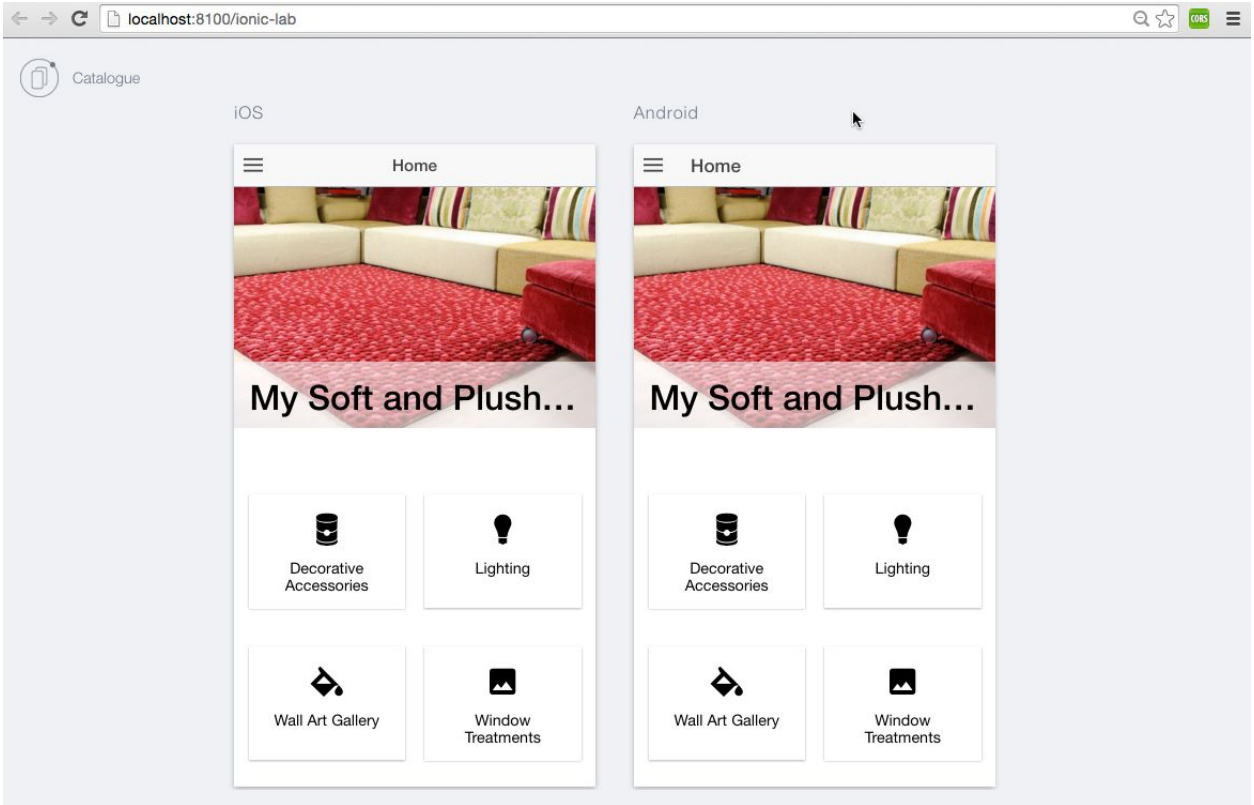
Run a local development server

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.



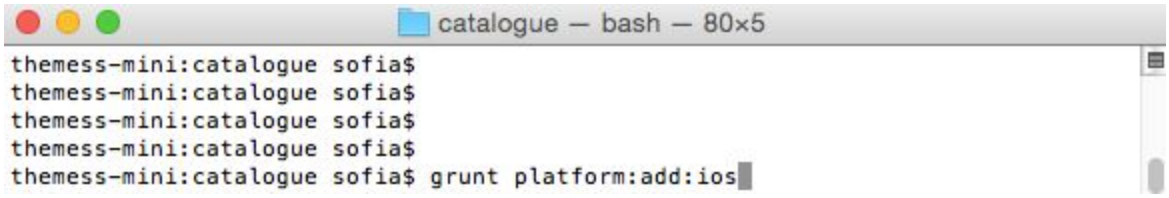
You could also open the application in a single browser window by starting it with the command:

```
$ grunt serve
```

Run in the emulator

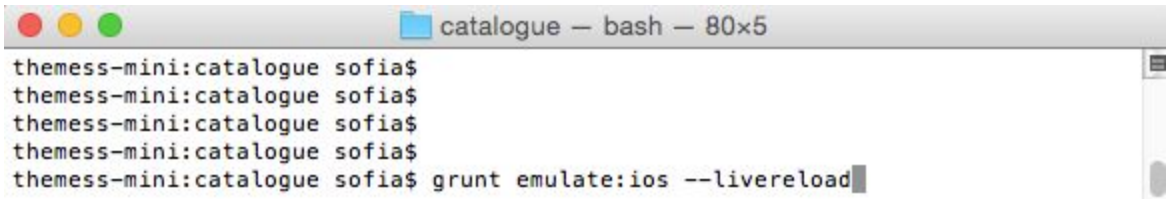
First the prefered platform should be added. In this case iOS:

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

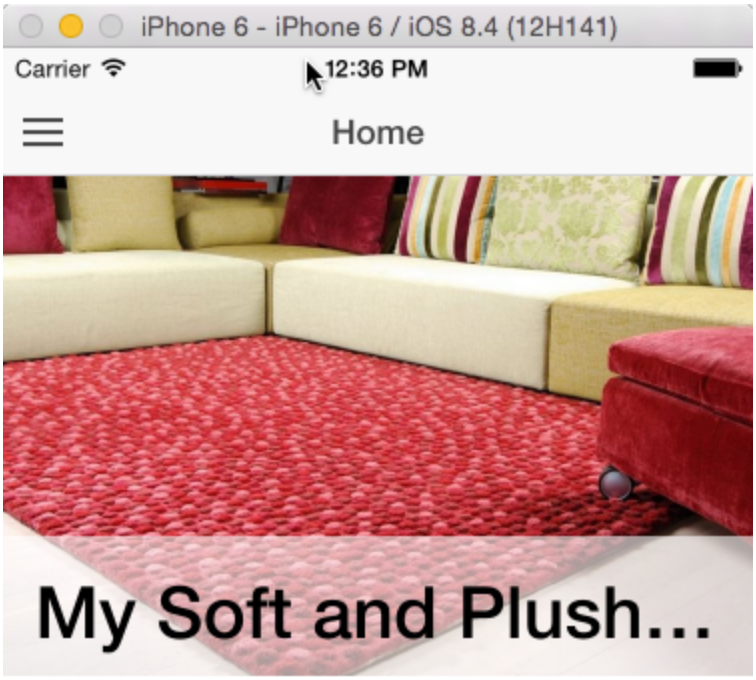


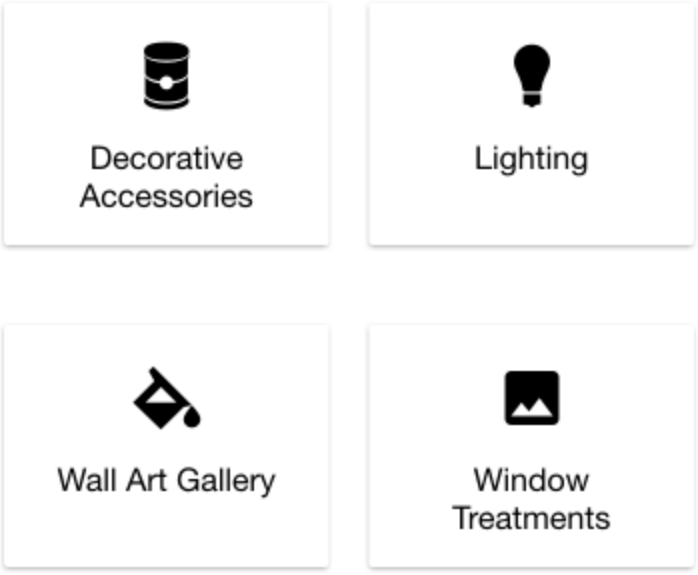
Now the application is ready to start inside a simulator:

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



The iPhone simulator will launch and the Catalogue 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

```
{
  "name": "catalogue",
  "app_id": "c52a7655"
}
```

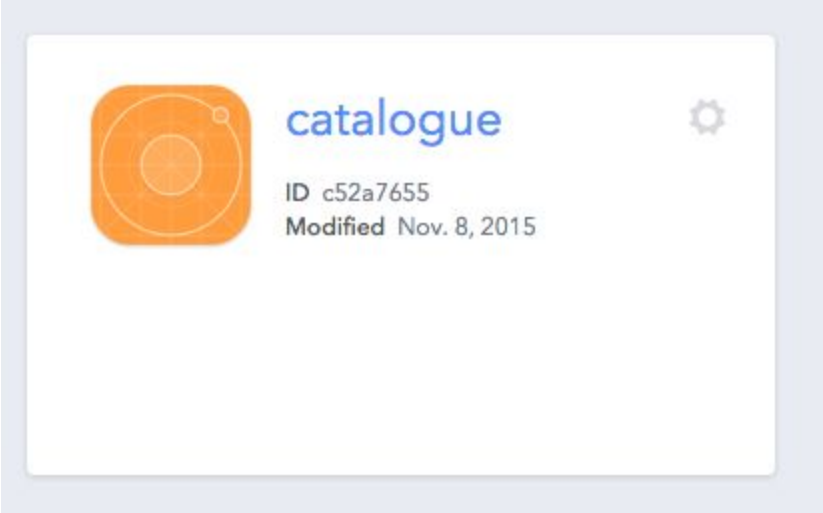
ionic.project

config.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<widget id="com.titaniumtemplates.catalogue" version="0.0.1" xmlns="http://www.w3.org/ns/wid
  <name>Catalogue</name>
  <description>
        Catalogue. The boostrap you need in order to build your next Ionic application.
    </description>
  <author email="skounis@gmail.com" href="http://about.me/stavros.kounis">
        Stavros Kounis
    </author>
  <content src="index.html"/>
```

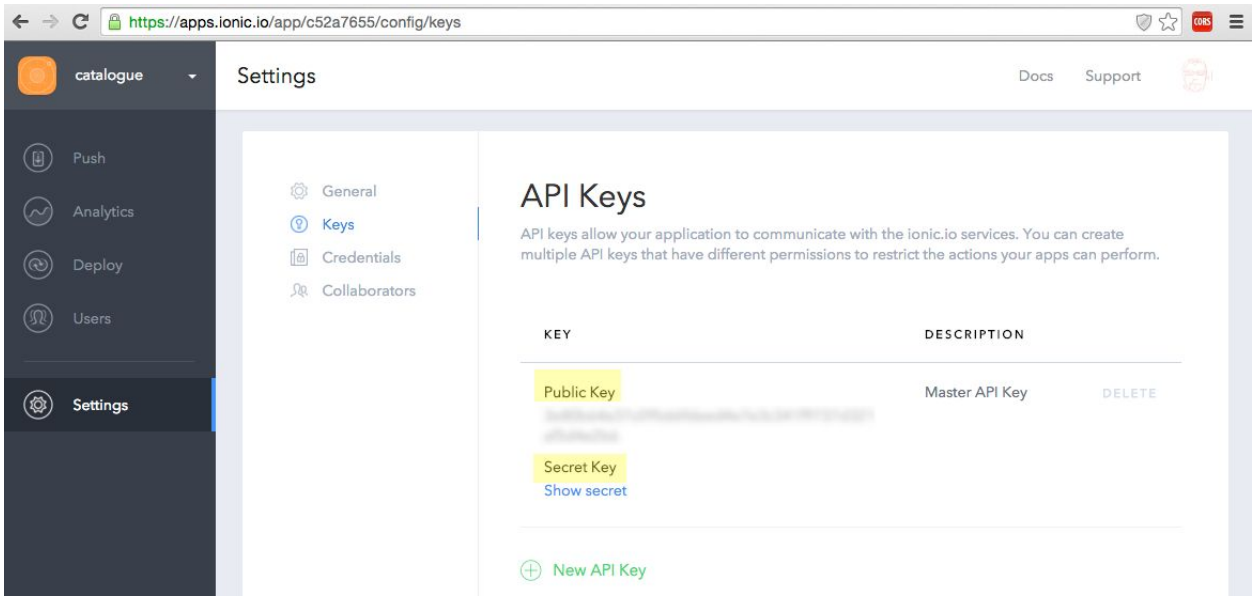
config.xml

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

Configuration

News

News screen is configured to fetch data from a remote source. JSON protocol is used for this purpose.

The remote data source that is already configured for demonstrating purposes is available in the URL below:

- http://skounis.s3.amazonaws.com/mobile-apps/local-business/news.json

Data Source configuration

In order to set the **Catalogue Ionic** application to work with your own news data source you should follow the next steps:

1. Create your JSON structures and put them online.
2. Open the related `news.service.js` file, locate the `url` variable and replace it with your source.

news.service.js

Categories and Products

The application can consume JSON files that you have created beforehand and provided to the application uploading them online (e.g. Amazon S3). Alternatively, in case you have a WooCommerce website, you can connect the application to consume data that already exist on your site.

JSON Approach

Categories are fetched from a remote JSON file consisting of all the available categories of products, each one of them in the form of a URL where the JSON source is located. As an example, a categories JSON file is already configured and available in the URL below:

- http://skounis.s3.amazonaws.com/mobile-apps/catalogue/categories.json

Also, if a category has the “featured” property set as “true”, it will be displayed on the home screen. The default value of this property is “false”.
Therefore, you will need to construct a JSON file with all the URLs pointing to each category and a JSON file of the category itself. For demonstrating purposes, there are six categories, namely six more JSON files. The JSON sources for all the categories used as an example are the following:

- http://skounis.s3.amazonaws.com/mobile-apps/catalogue/cat-a.json
- http://skounis.s3.amazonaws.com/mobile-apps/catalogue/cat-b.json
- http://skounis.s3.amazonaws.com/mobile-apps/catalogue/cat-c.json
- http://skounis.s3.amazonaws.com/mobile-apps/catalogue/cat-d.json
- http://skounis.s3.amazonaws.com/mobile-apps/catalogue/cat-e.json
- http://skounis.s3.amazonaws.com/mobile-apps/catalogue/cat-f.json

Furthermore, at the very top of the home screen, there is a slide box where some of the products on sale or on promotion campaigns could be displayed. For that reason, a separate JSON file containing all those products is used:

- http://skounis.s3.amazonaws.com/mobile-apps/catalogue/featured.json

Finally, all the business related information such as address, email, name etc are fed by the following JSON file:

- http://skounis.s3.amazonaws.com/mobile-apps/catalogue/business.json

Data Sources configuration

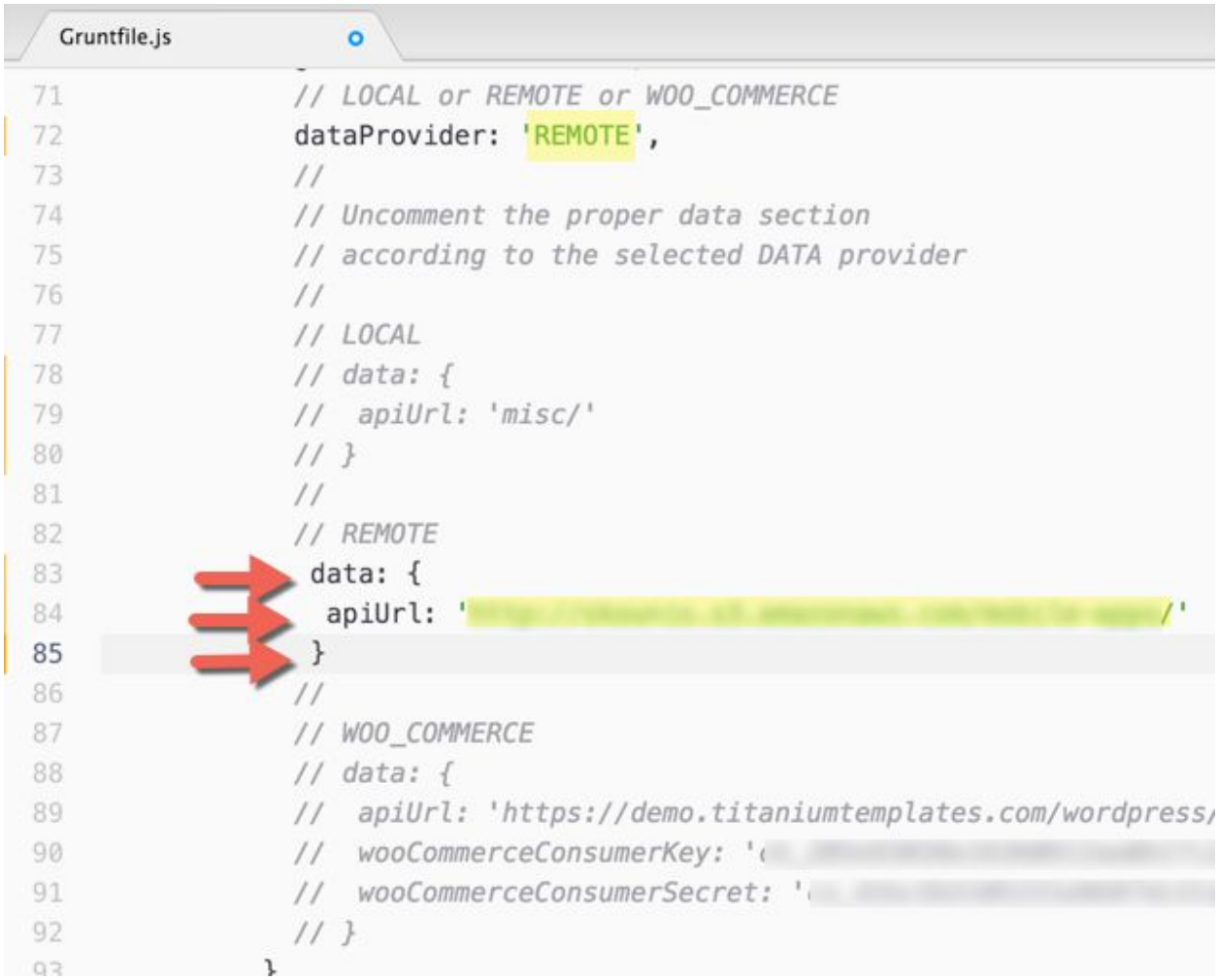
In order to set the categories data source, you should follow the next steps:

1. Create the JSON structure of each category and put it online.
2. Create the JSON structure containing the URLs of all the categories and put it online.
3. Create the JSON structure of the products that will be displayed on the home screen and put it online.
4. Create the JSON structure of the business related information and put it online.
5. Set the data provider:
    a. In `Gruntfile.js`, set the data provider to *LOCAL* if the data is pulled from the local `/misc` folder and make sure the related lines are uncommented:

Gruntfile.js

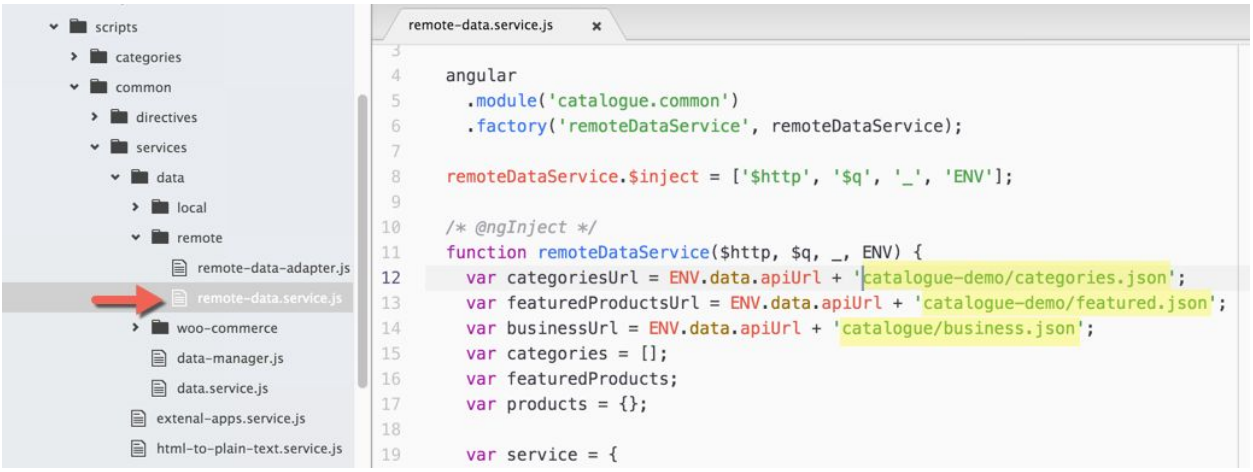
```
95        },
96        production: {
97          constants: {
98            ENV: {
99              name: 'production',
100             youtubeKey: 'AIzaSyDael5MmCQa1GKQNKQYypmBeB08GATgSEo',
101             ionicPrivateKey: 'a9265eaf15a20cc8516c770e8748aeed4891b28f453ce755',
102             //'c63b22cd7330a4a9d5e526bfcd74891a59fd5c23d1d81239',
103             ionicPublicKey: 'e30d4d540b8c75d1f167bbf242423c3fb23fe10275d1c016',
104             //'04dee953a91ef3857f1c8a7cf4748ecee375848681f7e833',
105             ionicAppId: '241b6d37', //'2113c758',
106             gcmId: '228071472080',
107             // LOCAL or REMOTE or WOO_COMMERCE
108             dataProvider: 'LOCAL',
109             //
110             // Uncomment the proper data section
111             // according to the selected DATA provider
112             //
113             // LOCAL
114             data: {
115               apiUrl: 'misc/'
116             }
117             //
118             // REMOTE
119             // data: {
```

b. In case that the data is pulled from a remote source, set the data provider to *REMOTE* and the related `apiUrl`, as shown below:



Gruntfile.js

Note that the `apiUrl` points to the S3 bucket's path where the JSON files are located. Additionally, open the `remote-data.service.js` file under the `app/scripts/common/services/data` path, locate the `categoriesUrl`, `featuredProductsUrl` and `businessUrl` variables and replace them with the sources created in step 2, 3 and 4 respectively. Note that the url should point to the exact JSON file.

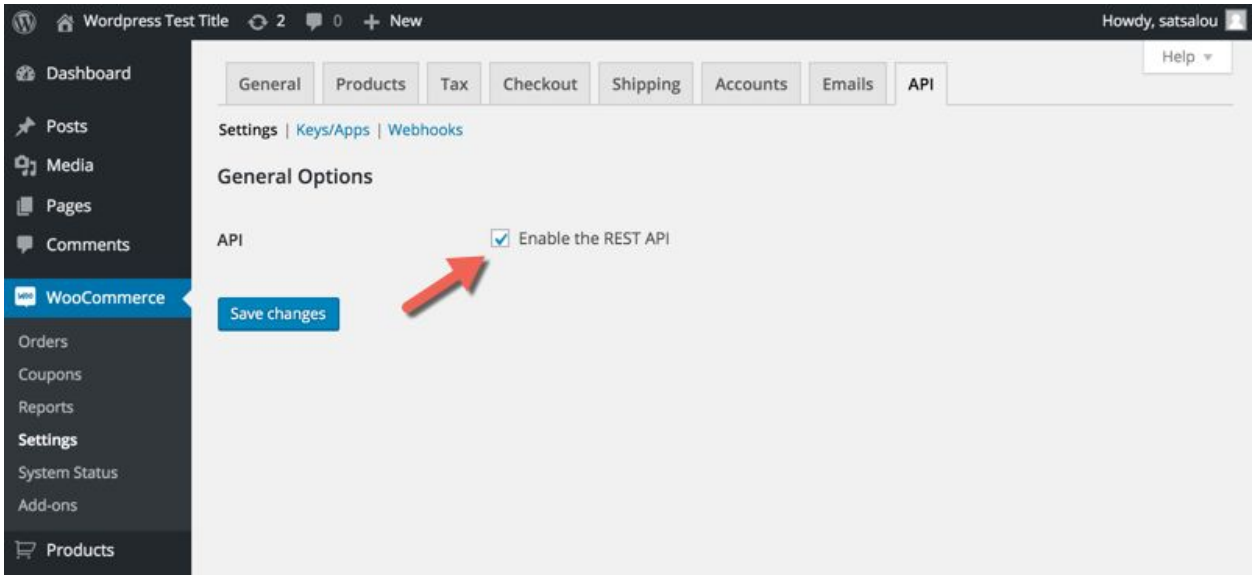


remote-data.service.js

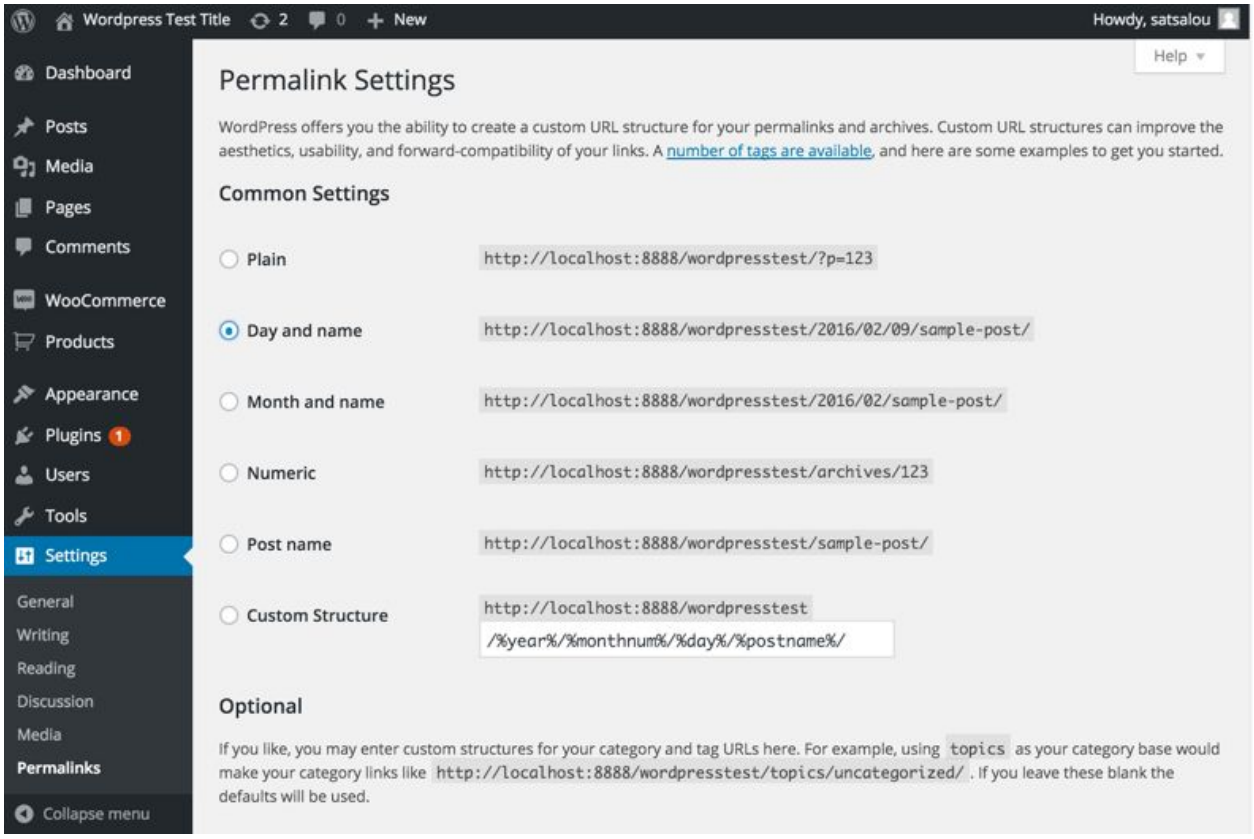
WooCommerce

Enable REST API

Firstly, you should enable the WooCommerce REST API. In order to do that, visit WooCommerce **>** Settings **>** API tab and tick the “Enable REST API” checkbox.

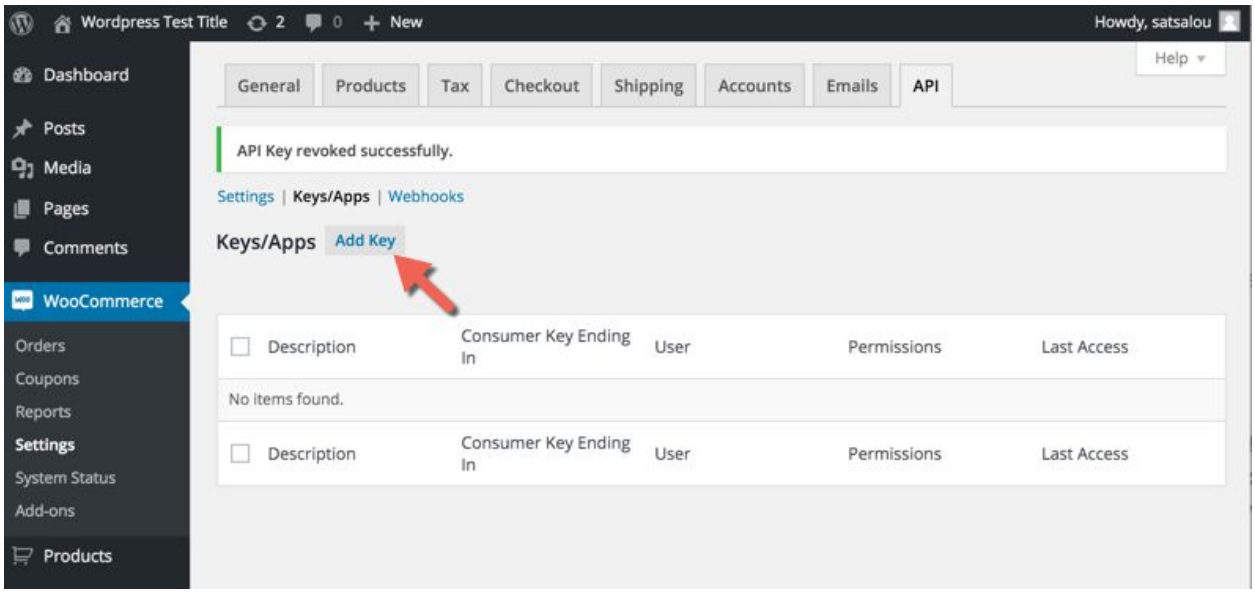


Please, note that you should have turned on the Wordpress Permalinks (Settings **>** Permalinks).

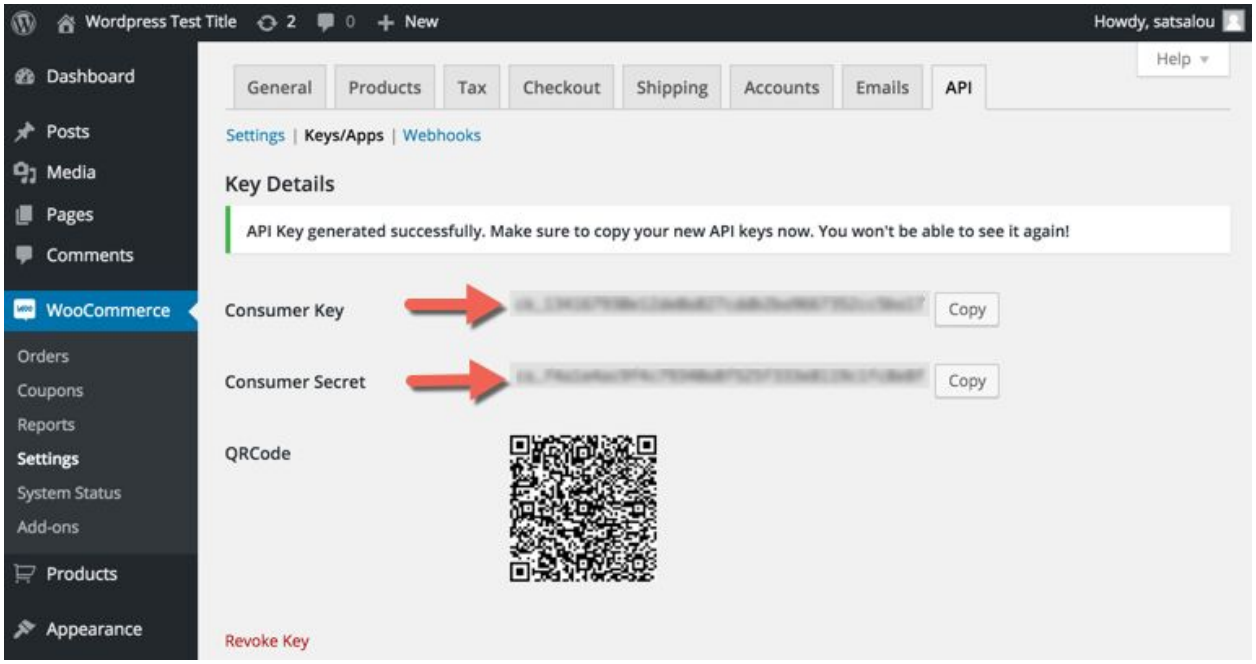


Generate API Keys

To create the keys, go to WooCommerce **>** Settings **>** API **>** Keys/Apps.

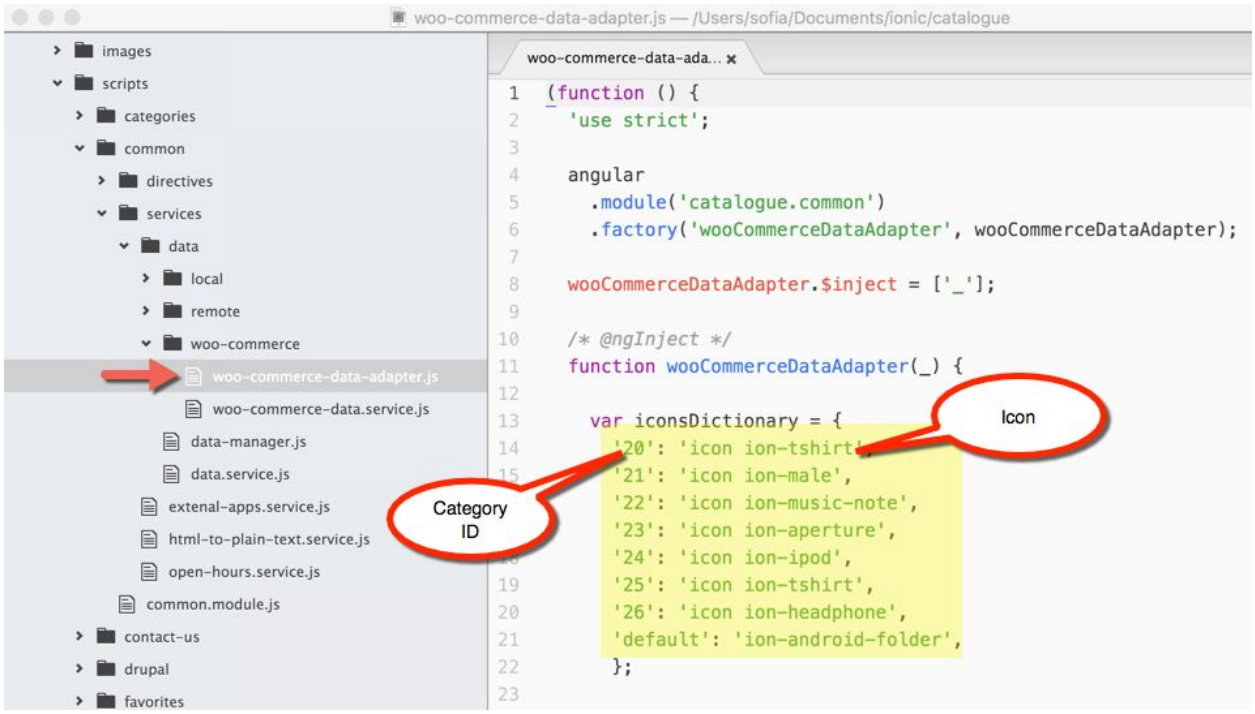


Take a note of the “Consumer Key” and the “Consumer Secret” as they will be used later.



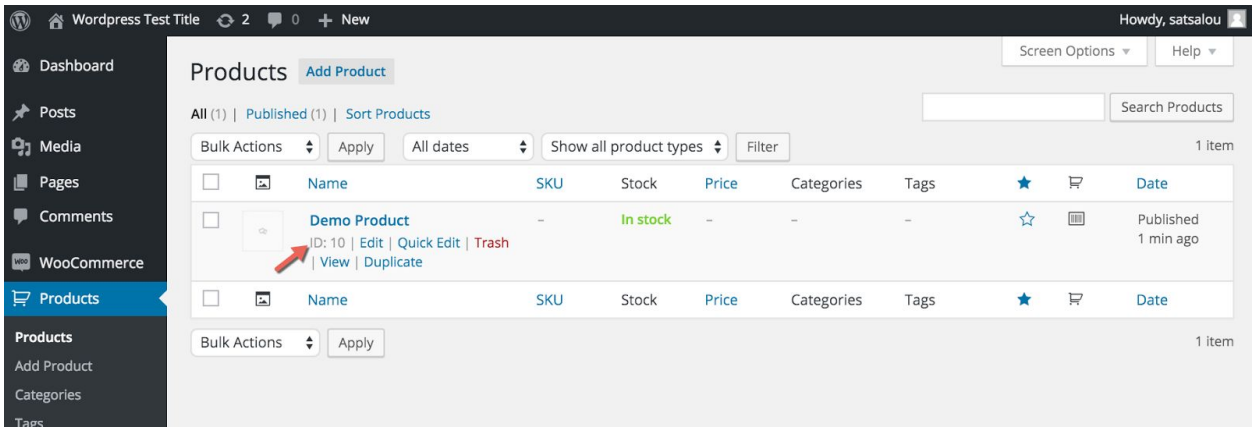
Set icons for product categories

In order to set the icons for each product category showing on Home screen and in menu, you should declare the product category ID and then the icon you wish to be displayed for this category. This is taking place in `woo-commerce-data-adapter.js` file under `scripts/common/services/data/woo-commerce/` path:



woo-commerce-data-adapter.js

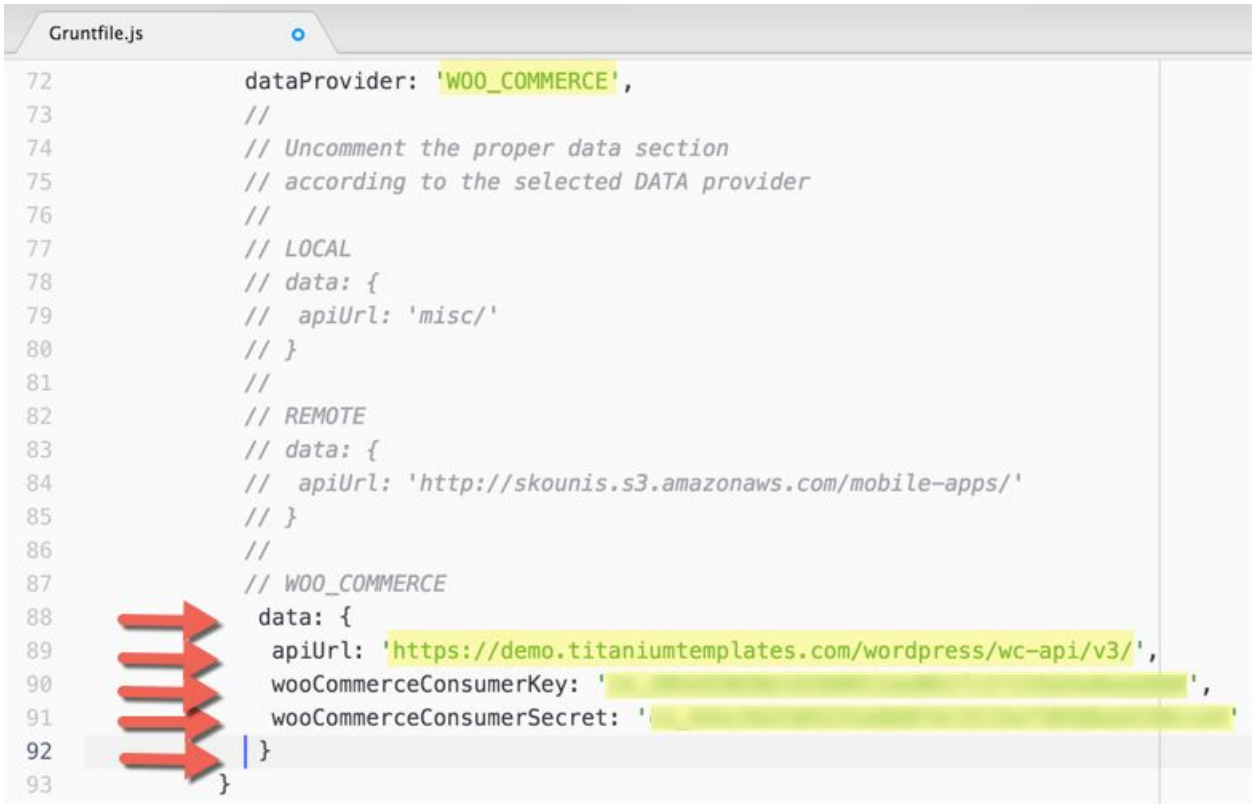
Note that the can find the category ID under the “Products” tab by hovering on a particular product:



Also, notice that you can set a default icon in case that you want to show a default icon for some product categories which do not have a defined icon.

Set the data provider

In `Gruntfile.js` file, set the data provider to *WOO_COMMERCE,* uncomment the indicated lines and replace the variables with your own api url and the keys mentioned earlier:



Gruntfile.js

Note that your api url should consist of your home url (in this example https://demo.titaniumtemplates.com/wordpress/) and followed by `wc-api/` and the version of the WooCommerce API you use (in this example `v3/`).

Wordpress

This feature loads articles from a remote Wordpress site. Wordpress JSON API is used for the creation of JSON feed of the posts.

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.

To replace this JSON source with yours, open the `wordpress.service.js` file located under the `app/scripts/wordpress/` path and place your link where the arrow shows.

wordpress.service.js

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.

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.

To replace this JSON source with yours, open the `drupal.service.js` file located under the `app/scripts/drupal/` path and place your link where the arrow shows.

drupal.service.js

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:

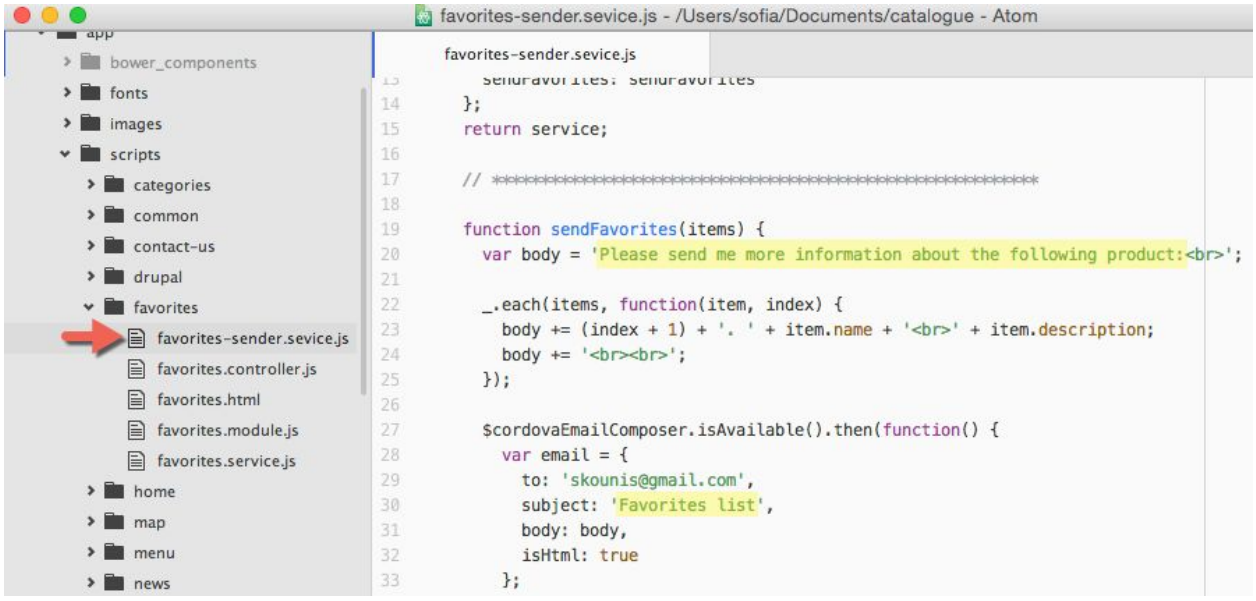
1. https://www.drupal.org/project/services
2. https://www.drupal.org/project/ctools
3. https://www.drupal.org/project/libraries
4. https://www.drupal.org/project/views
5. https://www.drupal.org/project/services_views

Favorites

On Favorites screen, the user is able to send their favorites products to an email address. For example, a reason to do this could be asking for more information about the products.

Email subject and body

In `favorites-sender.service.js` file located under the `app/scripts/favorites/` path, you can edit the **body** and the **subject** of the email as shown below:



favorites-sender.service.js

Business Information

All the business information such as store name, address, description, email, office location, phone number, social networks and map annotations can be configured in the `business.json` file and stored either locally in the `misc` folder or online, for example Amazon S3. The structure of the JSON file should look similar to the following:

business.json

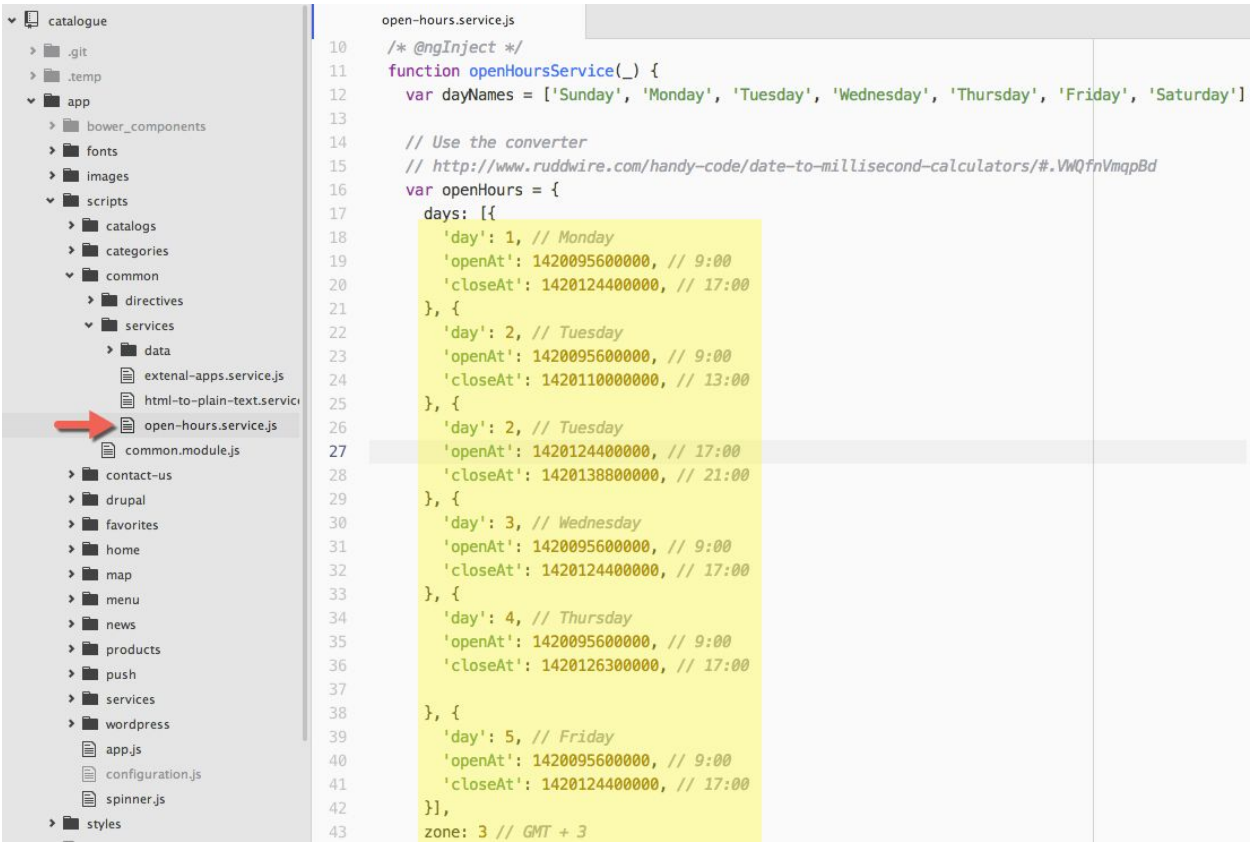
```
{
  "id": 1,
  "jsonrpc": "2.0",
  "result": {
    "storeName": "Home Decoration",
    "address": "50 Market Street, San Francisco, California 94103, United States",
    "desc": "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
    "phoneNumber": "+306973216110",
    "email": "skounis@gmail.com",
    "officeLocation": "37.7736854,-122.421034",
    "facebookPage": "https://www.facebook.com/ionicframework",
    "instagramPage": "https://instagram.com/phonegap/",
    "twitterPage": "https://twitter.com/titemplates/",
    "pinterestPage": "https://www.pinterest.com/",
    "map": {
      "origin": {
          "latitude" : 37.407,
          "longitude" : -122.1
      },
      "zoomLevel": 15,
      "annotations" : [{
          "title" : "Molestie et wisi.",
          "latitude" : 37.407,
          "longitude" : -122.1
      }, {
          "title" : "Ullamcorper eros.",
          "latitude" : 37.41,
          "longitude" : -122.1
      }]
    }
  }
}
```

business.json

Note that the `origin` property of map annotations 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.

In order to set the hours at which the company is open, open the `open-hours.service.js` file located under the `app/scripts/common/services` path and adjust the `openHours array`. The open hours are going to be displayed on the *`Contact us`* screen too. Note that the hours should be represented in milliseconds since 1970/01/01 in GMT timezone.



open-hours.service.js

Push Notifications

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

https://docs.ionic.io/services/push/

Create Security Profiles

Firstly, you should create a security profile with push credentials for Android. For more information, please, visit Ionic official documentation. Similarly, you should create a security profile for iOS as the Ionic documentation describes.

Plugins

Install the required plugins by using the following command where SENDER_ID is your GCM project number:

```
$ cordova plugin add phonegap-plugin-push --variable SENDER_ID=12341234
```

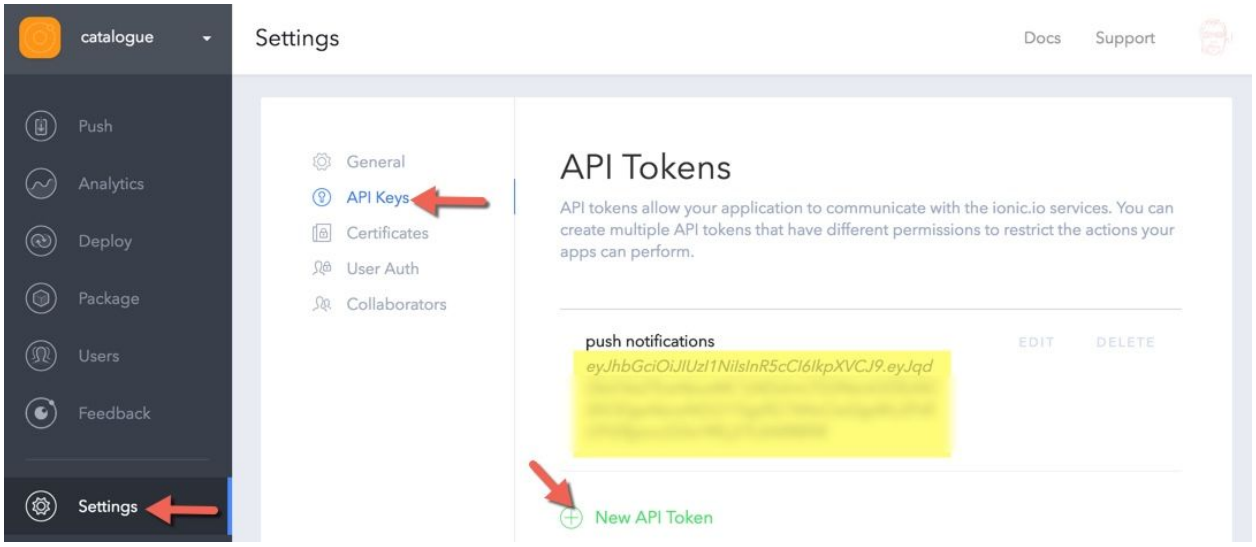
Set Ionic Platform's credentials

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

Gruntfile.js

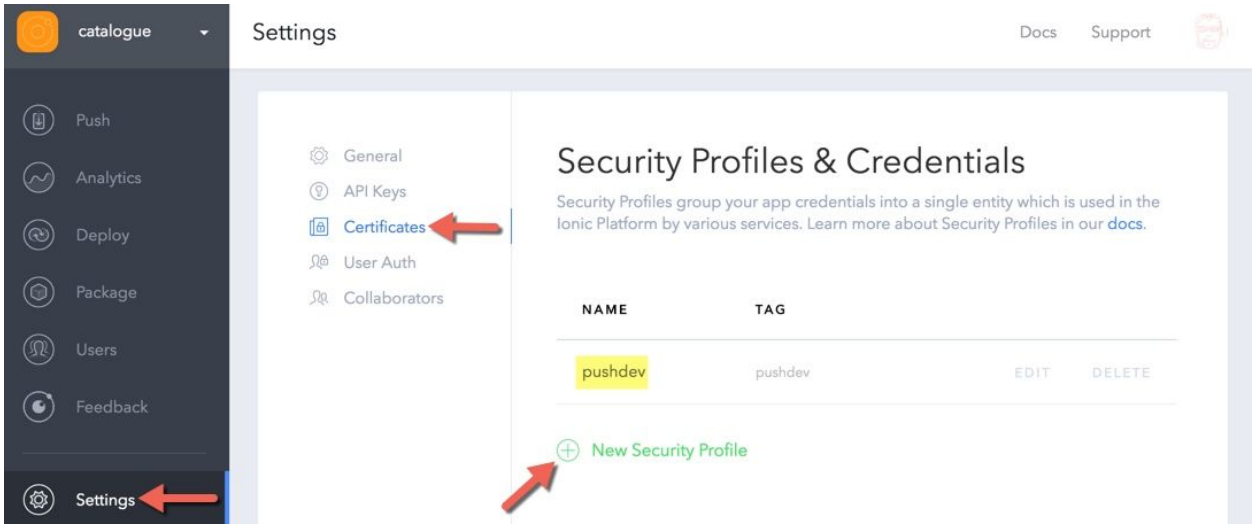
```
59          },
60          development: {
61            constants: {
62              ENV: {
63                name: 'development',
64                youtubeKey: 'AIzaSyDael5MmCQa1[redacted]',
65                ionicCloudApiToken: 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOi
66                ionicSecurityProfile: 'pushdev', // https://apps.ionic.io/app/
67                ionicAppId: 'eca7b2b5', //'2113c758',
68                gcmId: '773185234778', // Google Developers Console
69                // LOCAL or REMOTE or WOO_COMMERCE
70                dataProvider: 'LOCAL',
```

Also, you will need an API token so you can use Ionic API to send push notifications. Simply, go to the [Ionic Platform] and hit settings to create a token. Please, make a note of this token:



Additionally, define your iOS and Android security profiles by creating a certificate on the [Ionic]

Platform. Please, make a note of your certificate name:



Finally, set your `ionicCloudApiToken` and `ionicSecurityProfile` with your API token and certificate name in `Gruntfile.js`:

```
Gruntfile.js
59          },
60          development: {
61            constants: {
62              ENV: {
63                name: 'development',
64                youtubeKey: 'AIzaSyDael5MmCQa1[redacted]',
65                ionicCloudApiToken: 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOi
66                ionicSecurityProfile: 'pushdev', // https://apps.ionic.io/app/
67                ionicAppId: 'eca7b2b5', //'2113c758',
68                gcmId: '773185234778', // Google Developers Console
69                // LOCAL or REMOTE or WOO_COMMERCE
70                dataProvider: 'LOCAL',
```

Set GCM id

Note that in order to push notifications to the registered Android devices, you will need to set the `gcmId` (GCM project number) in `Gruntfile.js` as shown below:

Gruntfile.js

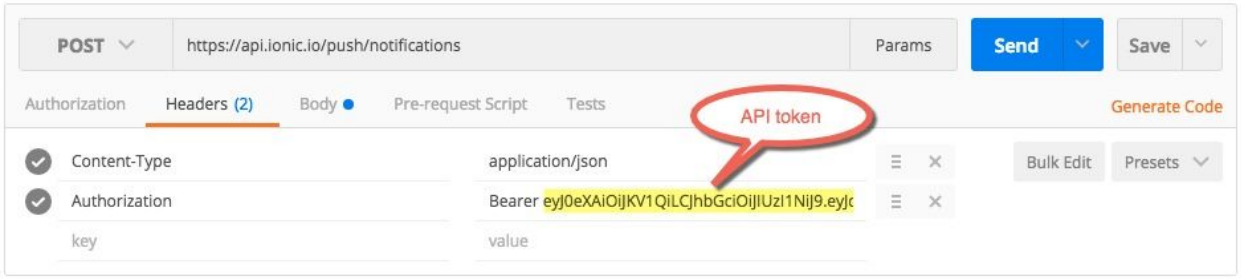
```
59          },
60          development: {
61            constants: {
62              ENV: {
63                name: 'development',
64                youtubeKey: 'AIzaSyDael5MmCQa1[redacted]',
65                ionicCloudApiToken: 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOi
66                ionicSecurityProfile: 'pushdev', // https://apps.ionic.io/app/
67                ionicAppId: 'eca7b2b5', //'2113c758',
68                gcmId: '773185234778', // Google Developers Console
69                // LOCAL or REMOTE or WOO_COMMERCE
70                dataProvider: 'LOCAL',
```

Send a Push Notification

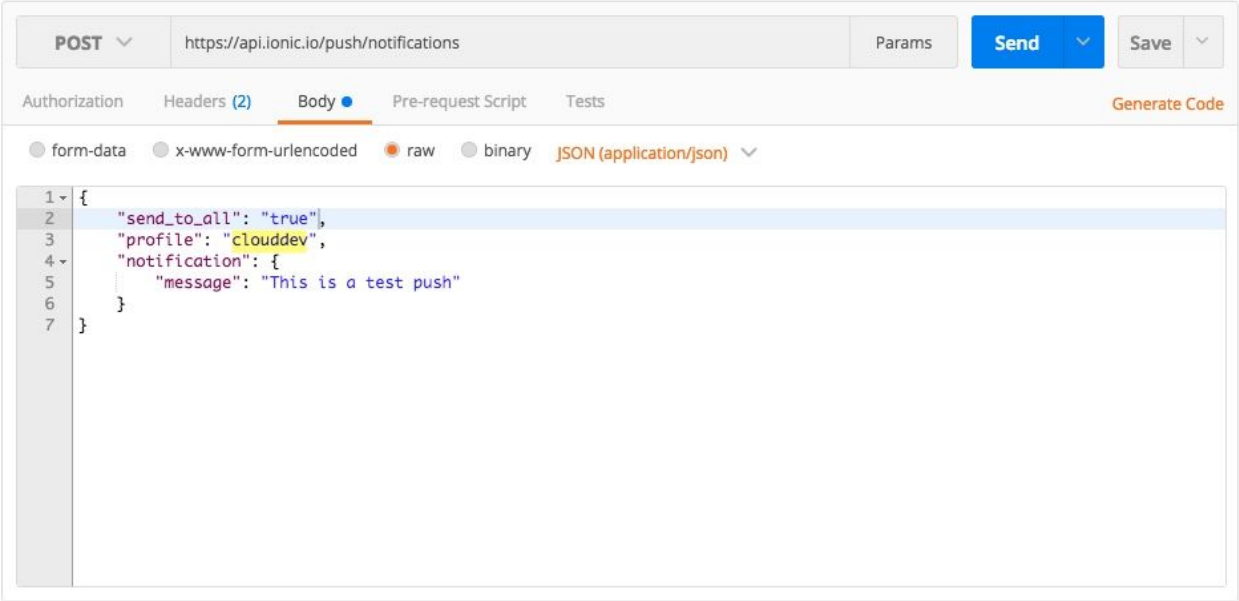
Please, bear in mind that you can send push notifications only to real devices. In order to do this, you can use POSTMAN or the UI of the Ionic Platform.

POSTMAN

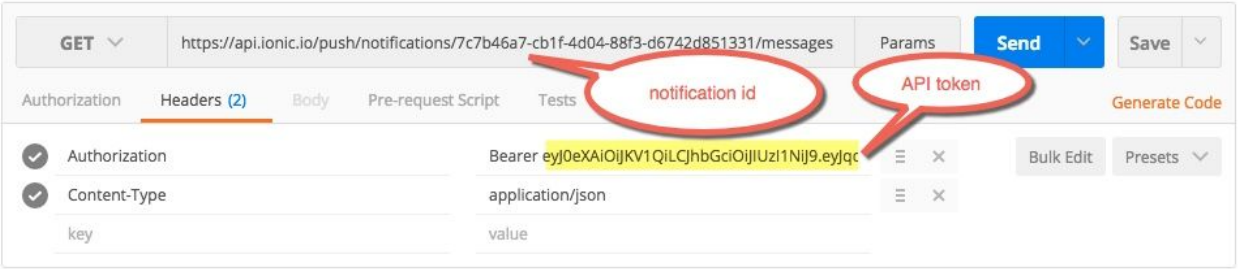
To send a notification to all the registered devices, you should use the related API endpoint and make a POST call. Also, you should have the following headers. For the authorization use the API token you created previously.



Set the body and place your certificate name where highlighted:

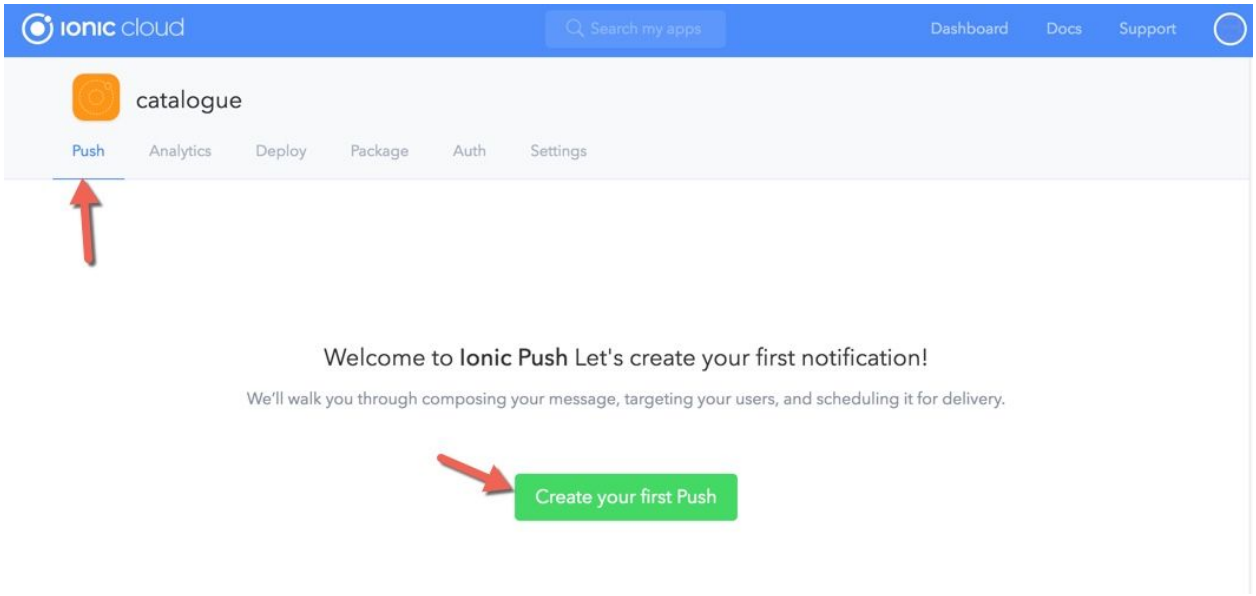


You may need to troubleshoot in case of a failed push notification. In order to see the state of your push notification and any error messages use the related API endpoint. The GET request should look similar to the following:



Ionic Platform UI

To send a notification using the UI of Ionic Platform, simply, create a new push notification.



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 us by using our 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 us to our Support Centre.

--- *The Appseed team. appseed.io*