Ionic 3 Toolkit - Core app - Documentation

-Quick start Guide - Installation and user support

Created: Apr 11, 2016
Updated: May 25, 2018
By: Appseed, 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 us to our Support Centre. Thank you!

Table of contents:

Upgrade and Save

What's in the Pack

Preparing your local environment (first-time preparations)
- NodeJS
- Sass
- Git
- Cordova and Ionic

Run for the first time
- Install libraries
- 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

Configuration
- Home
- Wordpress
  - JSON API plugin
- Drupal
  - Services and JSON View
- Youtube

- Vimeo
  - API Key
- News and Products
  - Data Sources configuration
- OAuth
  - Facebook
    - Facebook Application ID
  - Google
    - Google Application ID
  - Github
    - Create a GitHub Application
    - Set the Application ID and Secret
  - Twitter
    - Create a Twitter Application
    - Set the Application ID and Secret
  - Linked In
    - Create a LinkedIn Application
    - Set the Application ID and Secret
- Map and Annotations
- RSS Feeds
- Instagram
  - Register a client
  - Set client ID in app
- Google Places Autocomplete
- Slack Webhook
- PouchDB and Cloudant
- Firebase
  - Create a Firebase app
  - Set database permissions
  - Configuration in app
- Flickr
- Tumblr

Support
- References / Links:

Thank you

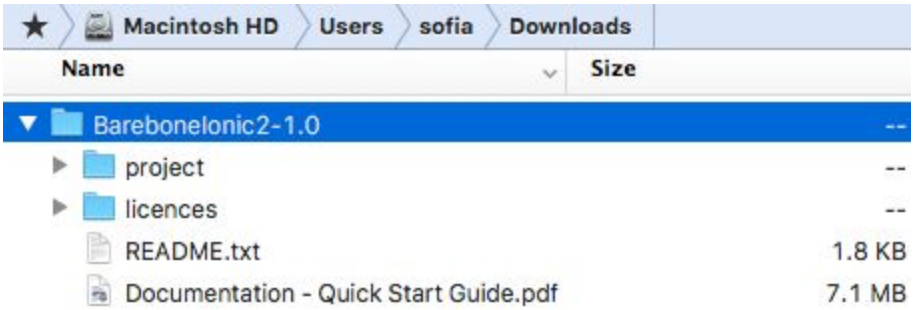
Upgrade and Save

*Are you a "**Barebone Ionic 1**" or "**Ionic 3 Toolkit - Personal**" client? Upgrade now to Ionic 3 Tookit Professional or Developer edition and save up to 60%. Contact us now and get your discount coupon.*

- *https://appseed.desk.com/customer/portal/emails/new*

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 application.
2. **Licences folder**: Terms and conditions for use, reproduction and distribution of this software piece.
3. **Project folder**: This is the ionic project that can be used to build a final mobile application or to create your own ionic project for any platform.

Preparing your local environment (first-time preparations)

NodeJS

First, make sure that NodeJS is installed on your computer. If NodeJS is not installed, please install version 6.x of NodeJS.

For more information about the NPM packages Ionic uses, refer to "Using NPM" section of Ionic official documentation.

Sass

Ionic 3 uses Sass. In order to install it on your Linux, Windows or Mac follow the given

instructions.

Git

GIT should be installed in your computer. Follow the instructions for your platform to install it:

https://www.atlassian.com/git/tutorials/install-git

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.

In order to install Ionic CLI for the creation/running of Ionic 3 projects, run the command:

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

In order to install Cordova, run the command:

```
$ npm install -g cordova
```

For more information, please, check the "Get Started Fast" and "Getting Started with Ionic" sections in the official Cordova and Ionic websites respectively.

Run for the first time

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

Install libraries

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

```
$ npm install
```

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.
Install the required plugins that are included in the `config.xml` file and add a platform by

using the command:

```
$ ionic cordova prepare ios
```

or install the plugins one by one by running the following command for each plugin:

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

eg:

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

You can find the full list of the required plugins and their functionality in the `README.md` file under `project` directory.

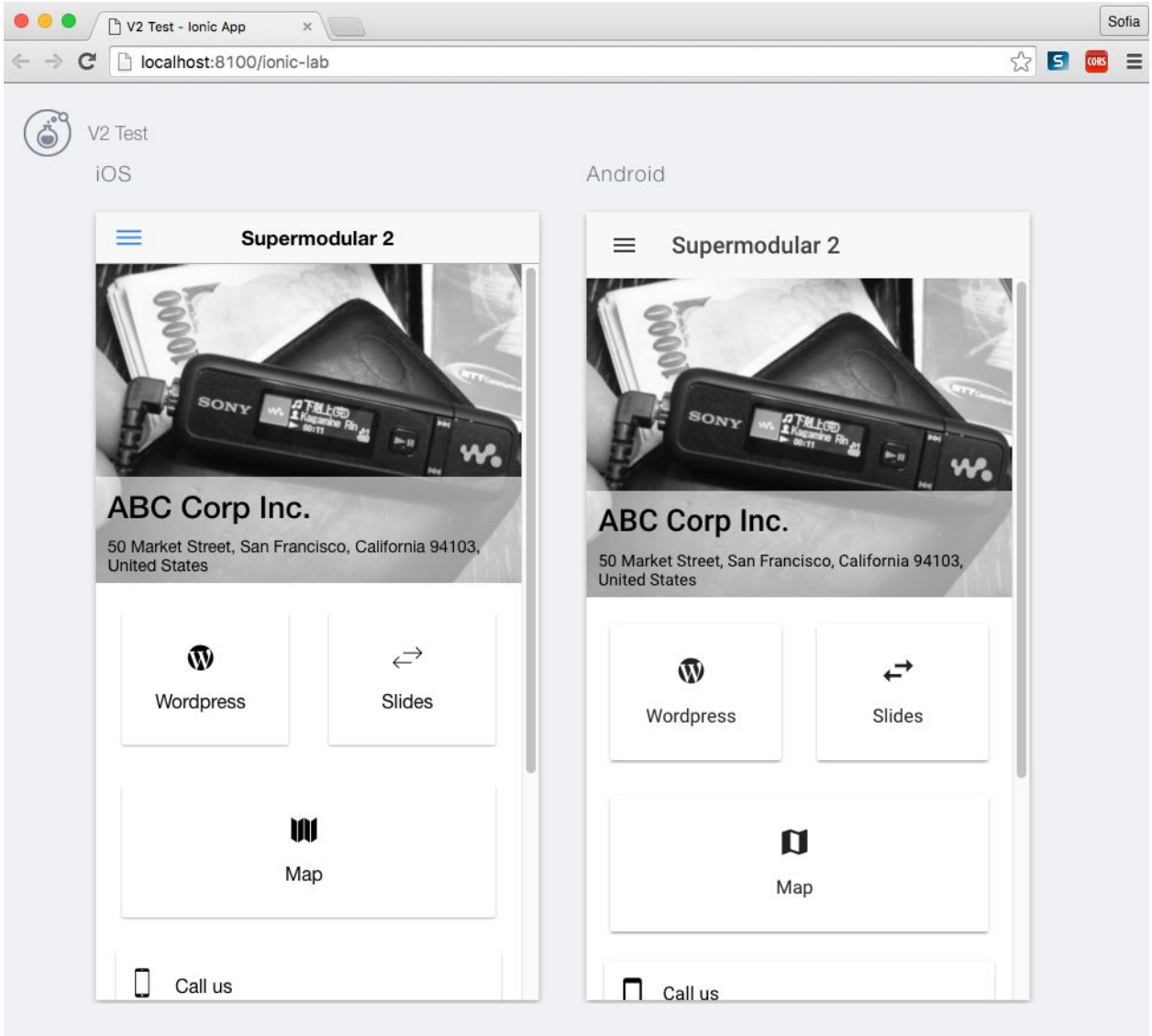
Run/Build the application (after the first-time preparations)

Run a local development server

Navigate to `project` directory and run the application in the browser:

```
$ ionic 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:

```
$ ionic serve
```

Run in the emulator

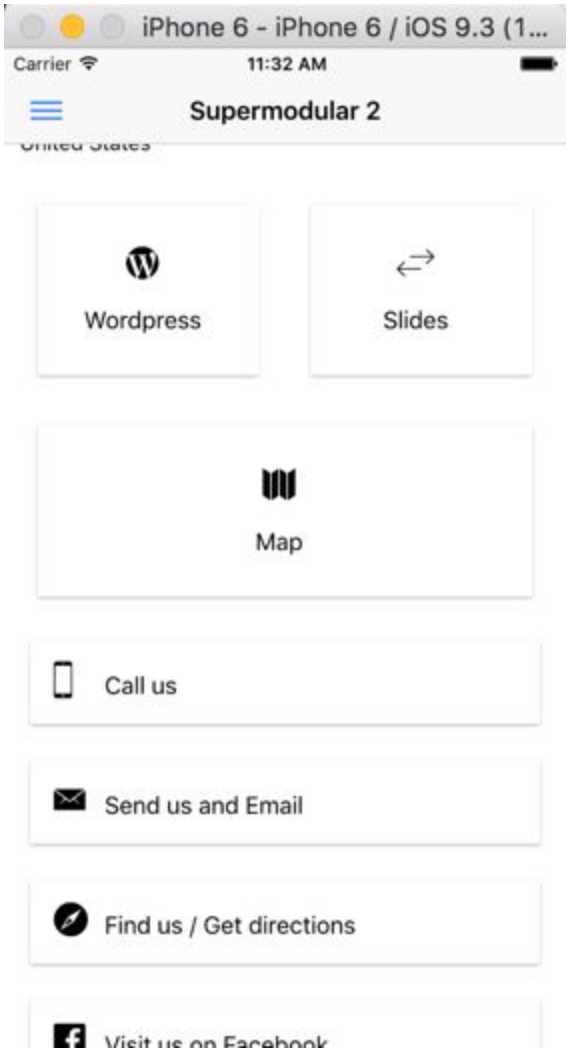
First the prefered platform should be added. In this case iOS:

```
$ cordova platform add ios
```

Now the application is ready to start inside a simulator:

```
$ cordova emulate ios
```

The iPhone simulator will launch and the app will start.



Build the app for specific platforms

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

```
$ cordova build
```

In order to build Android platform, run the command:

```
$ cordova build android
```

In order to build iOS platform, run the command:

```
$ cordova build ios
```

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.config.xml` and `config.xml` files and replace the highlighted fields:

```
ionic.config.json
1 {
2   "name": "barebone-ionic-2",
3   "app_id": "8b828c0d",
4   "v2": true,
5   "typescript": true
6 }
```

ionic.config.xml

```
config.xml
1 <?xml version='1.0' encoding='utf-8'?>
2 <widget id="com.titaniumtemplates.bareboneionic2" version="0.1.0" xmlns="http://www.w3.org/n
3     <name>Barebone Ionic 2</name>
4     <description>Barebone Ionic. The boostrap you need in order to build your next Ionic app
5     <author email="skounis@gmail.com" href="http://about.me/stavros.kounis">Stavros Kounis</
6     <content src="index.html" />
```

config.xml

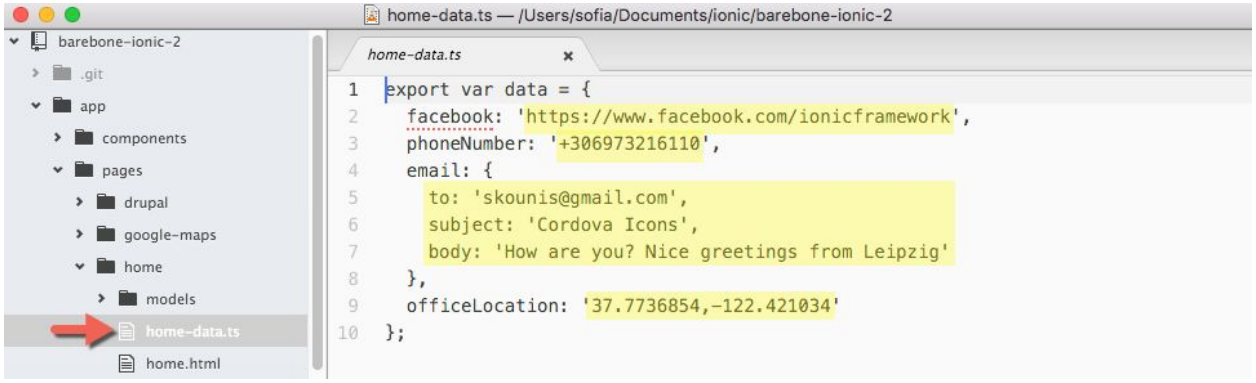
As for the app_id, use the ID of the corresponding application in your ionic.io account.

Configuration

All the app configuration is done in the `config.ts` file under the `src/` folder.

Home

The phone number, the email address, the email subject, the email body, the location and the facebook page that are going to be used for the “Call us”, “Send us an Email”, “Find us / Get directions” and “Visit us on Facebook” buttons should be set in the `home-data.ts` file under the `app/pages/home` path as shown:



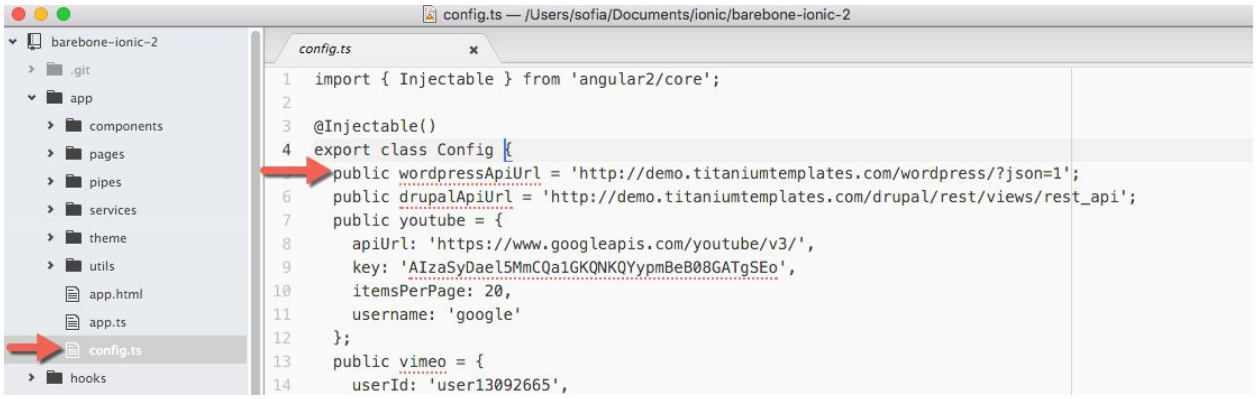
Wordpress

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 `config.ts` file and place your link where the arrow shows.



config.ts

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.

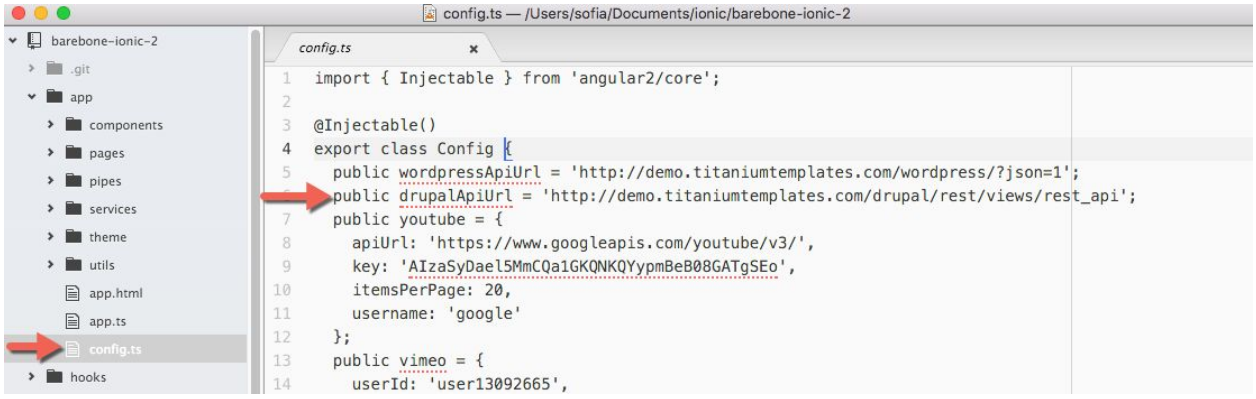
Drupal

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 `config.ts` file and place your link where the arrow shows.



config.ts

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/

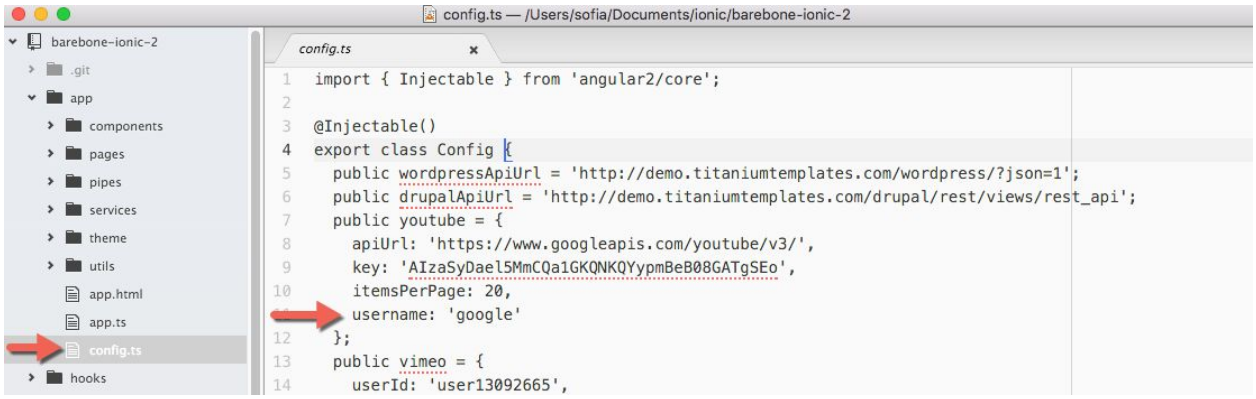
All the modules that are needed for this are listed below:

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

Youtube

This feature integrates Youtube playlists into the app. 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 his YouTube account username by replacing the variable "`username`" as shown below:



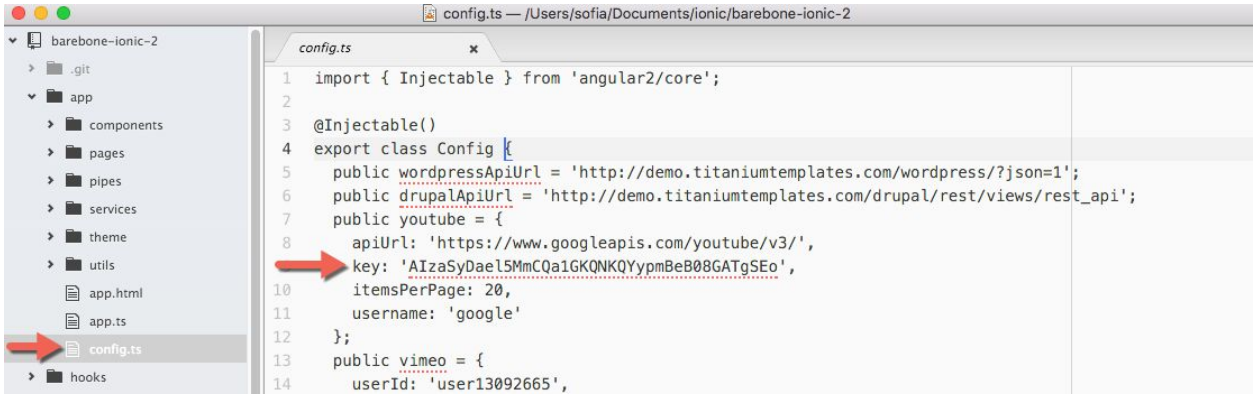
config.ts

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.

As shown below, you need to define this key in the `config.ts` file too.



config.ts

For more information, please, checkout the following references:

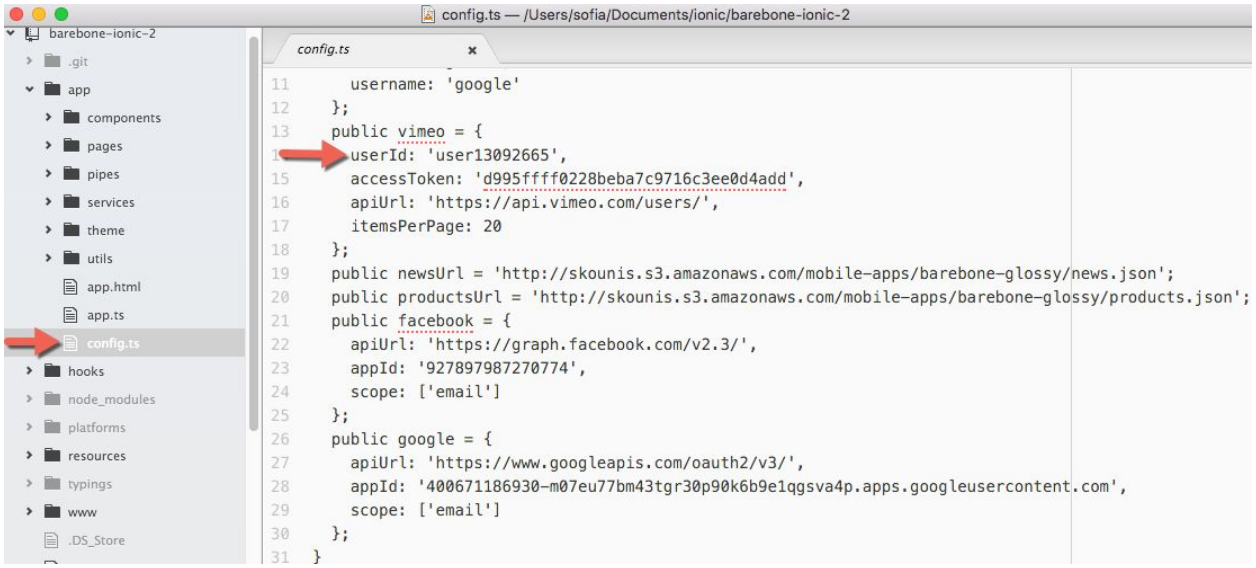
1. YouTube Data API (v3)
2. Registration of the application with Google

Vimeo

This feature integrates Vimeo playlists into the app. Vimeo Dev API is used for this purpose.

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

The user should only define his Vimeo account username by replacing the variable "`userId`" as shown below:



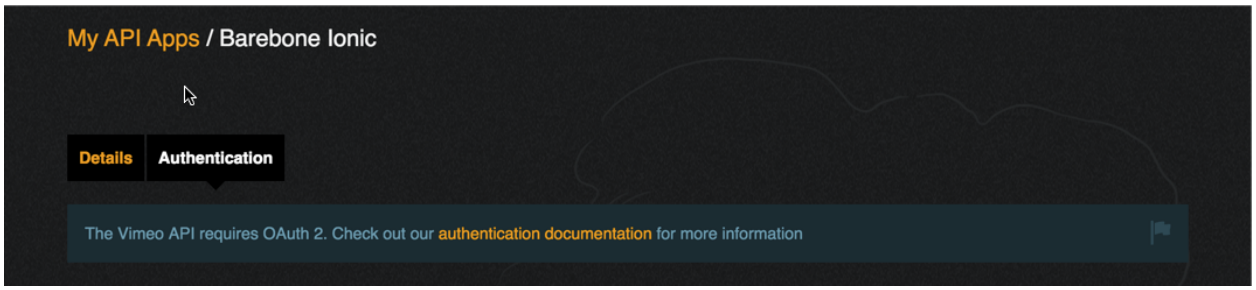
config.ts

API Key

Please note that you should create and register an API App by following the URL below:

https://developer.vimeo.com/apps/new

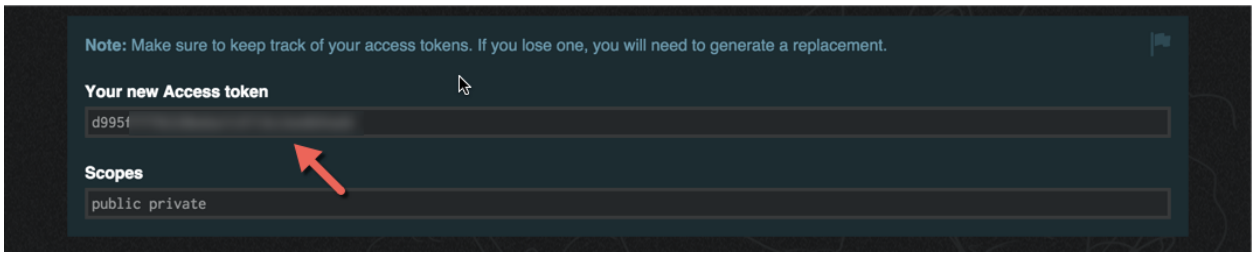
Once the API Application is created navigate to the “Authentication” tab and generate a token by using the corresponding button that is located there.



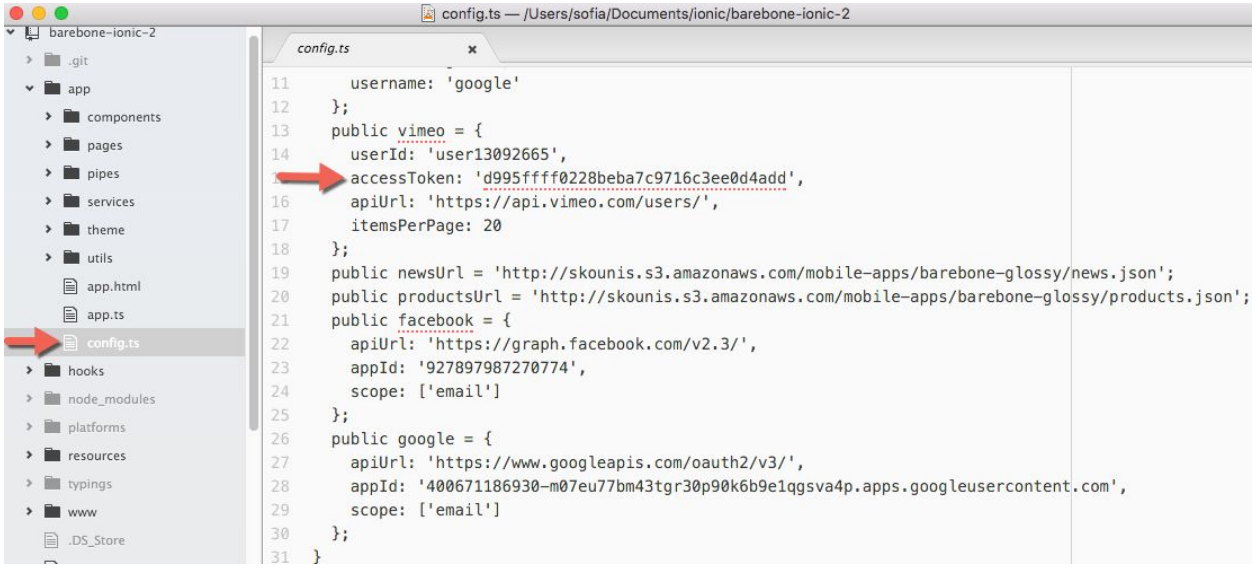
Authentication tab

Generate token button

As shown below, you need to define this key in the `config.ts` file.



Access Token



Vimeo Access Token field in config.ts

News and Products

News and Products screens are configured to fetch data from remote sources. JSON is the structure that is used for this purpose.

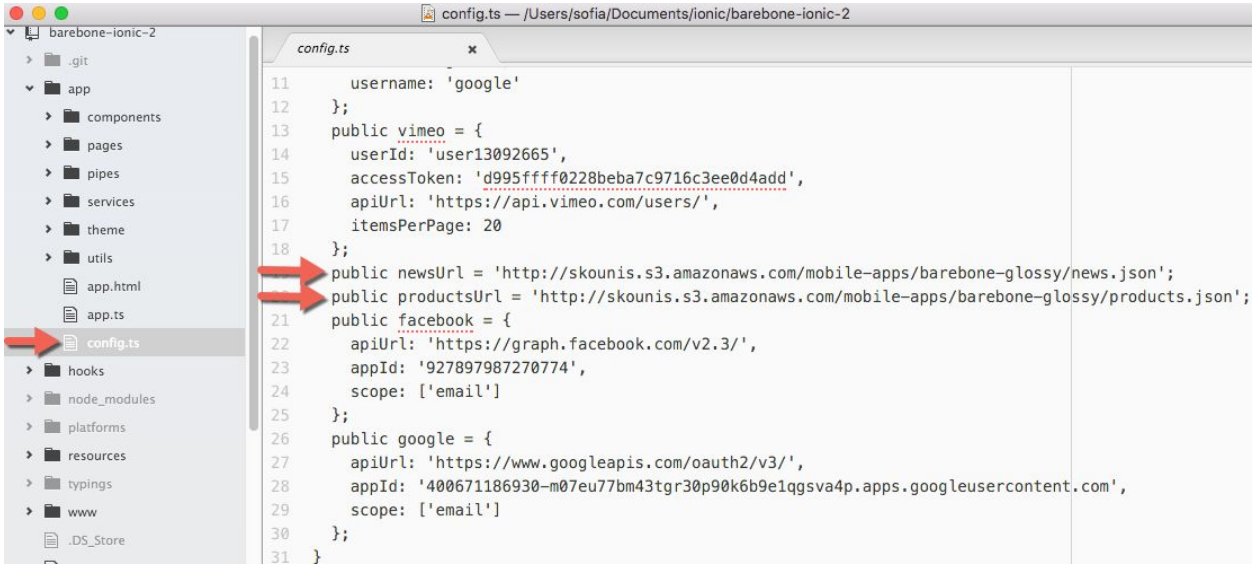
The remote data sources already configured for demonstration purposes are available in the URLs below:

- News
  http://skounis.s3.amazonaws.com/mobile-apps/barebone-glossy/news.json
- Products
  http://skounis.s3.amazonaws.com/mobile-apps/barebone-glossy/products.json

Data Sources configuration

In order to set the 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 related `config.ts` file and replace "`newsUrl`" and "`productsUrl`" with your sources.



config.ts

OAuth

This module demonstrates the usage of the ng2CordovaOauth by supporting at the same time 2 service providers:

1. Google
2. Facebook

All the code you will need is already there and only minor configuration steps will be required.

Facebook

This feature integrates Facebook Login into the app. Facebook Graph API is used for this purpose.

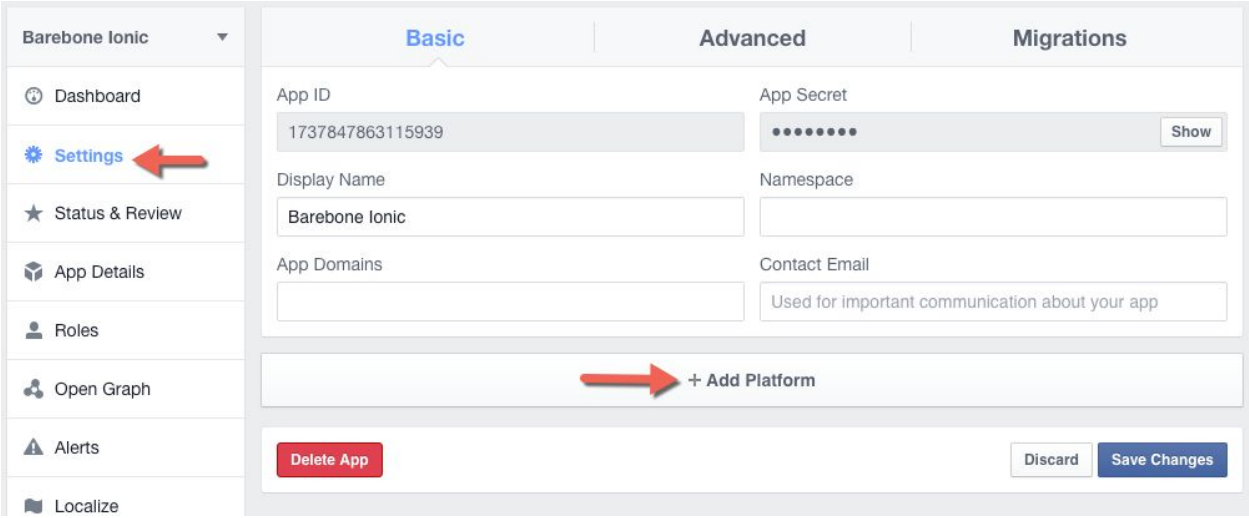
Everything related to the code for the communication with Facebook API is already implemented.

Facebook Application ID

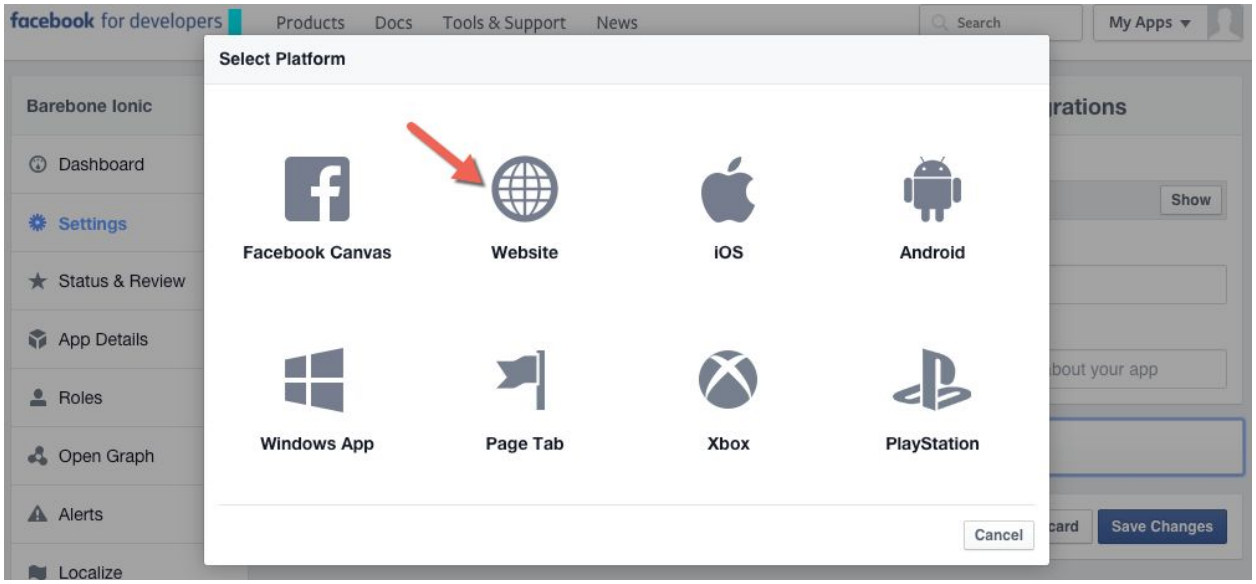
Firstly, you should create and register a new Facebook Application by following the URL below:

https://developers.facebook.com/apps/

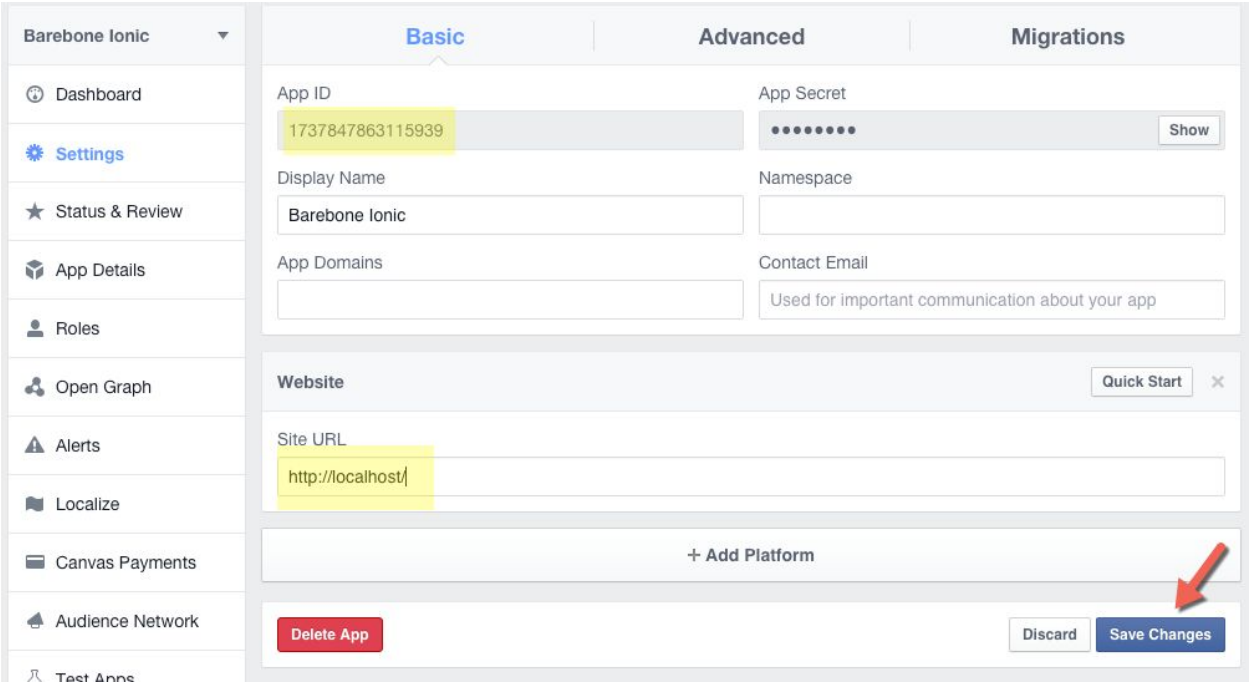
Once the Application is created navigate to the “Settings” page and select “Add Platform”.



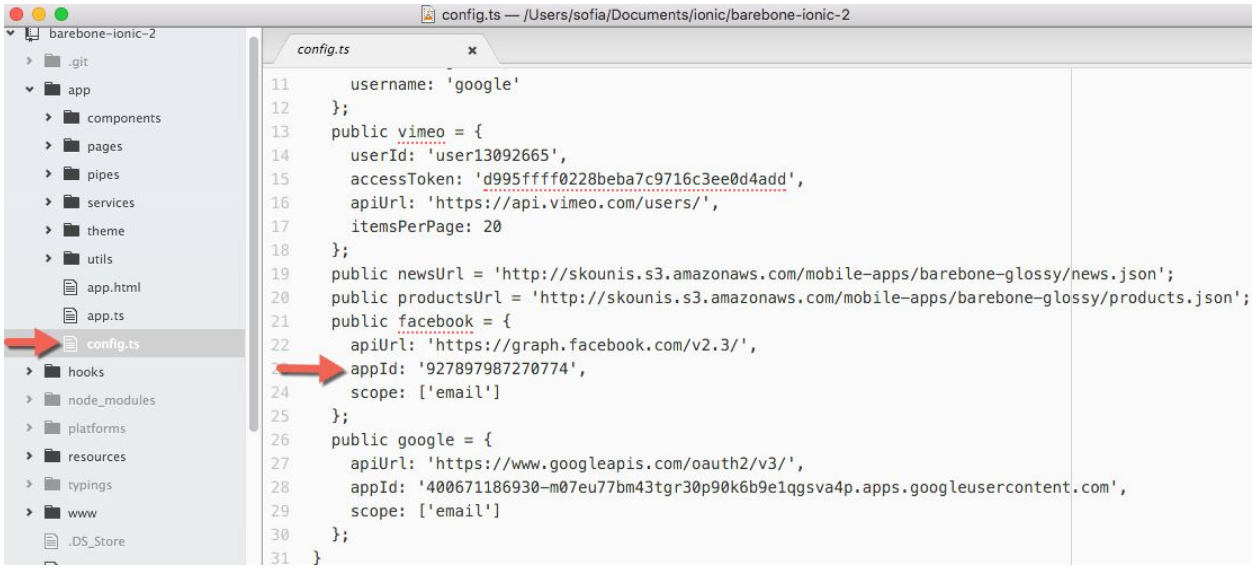
Then, select “Website”



and set as “Site URL” the value “http://localhost/”. Also, keep a note of the “App ID” and hit “Save changes”.

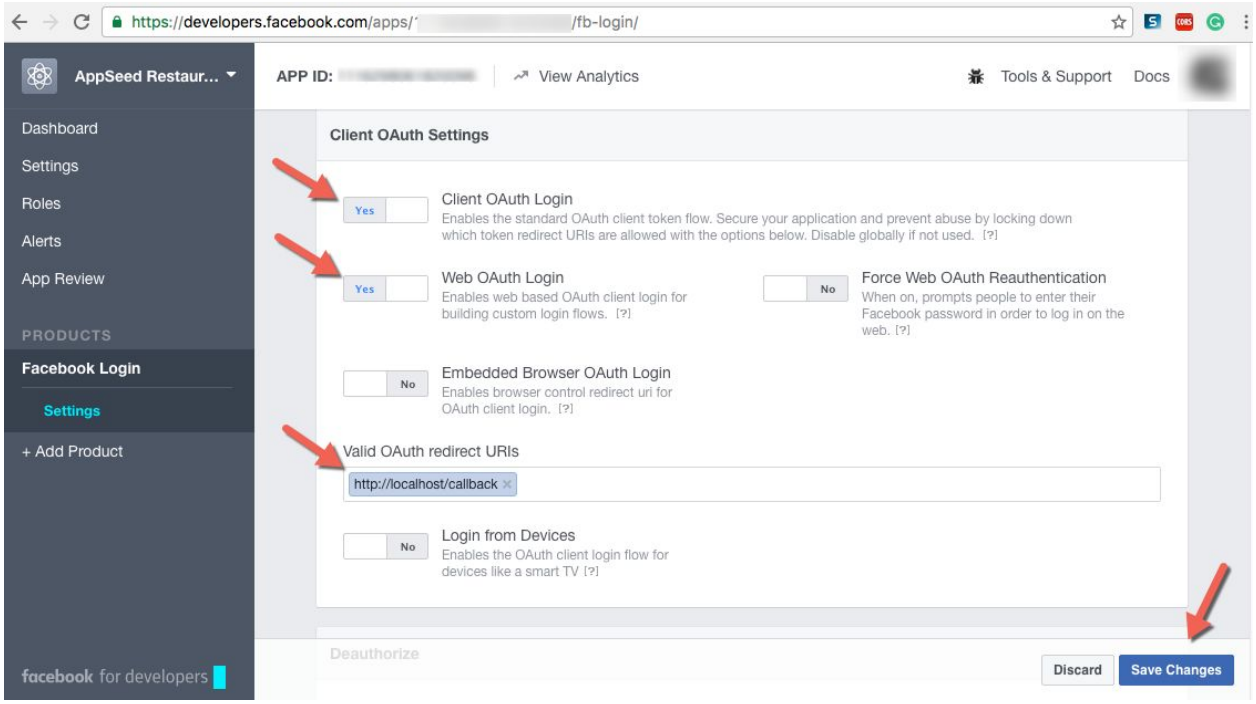


Once you create your Facebook App, set its ID in the `config.ts` file.



config.ts

On the Facebook Developers platform, select “Add Product” from the side menu and, then, choose “Facebook Login”. Make sure you have enabled “Client OAuth Login” and “Web OAuth Login” and set your “Valid OAuth redirect URIs” to http://localhost/callback. At the end, do not forget to hit the “Save changes” button.

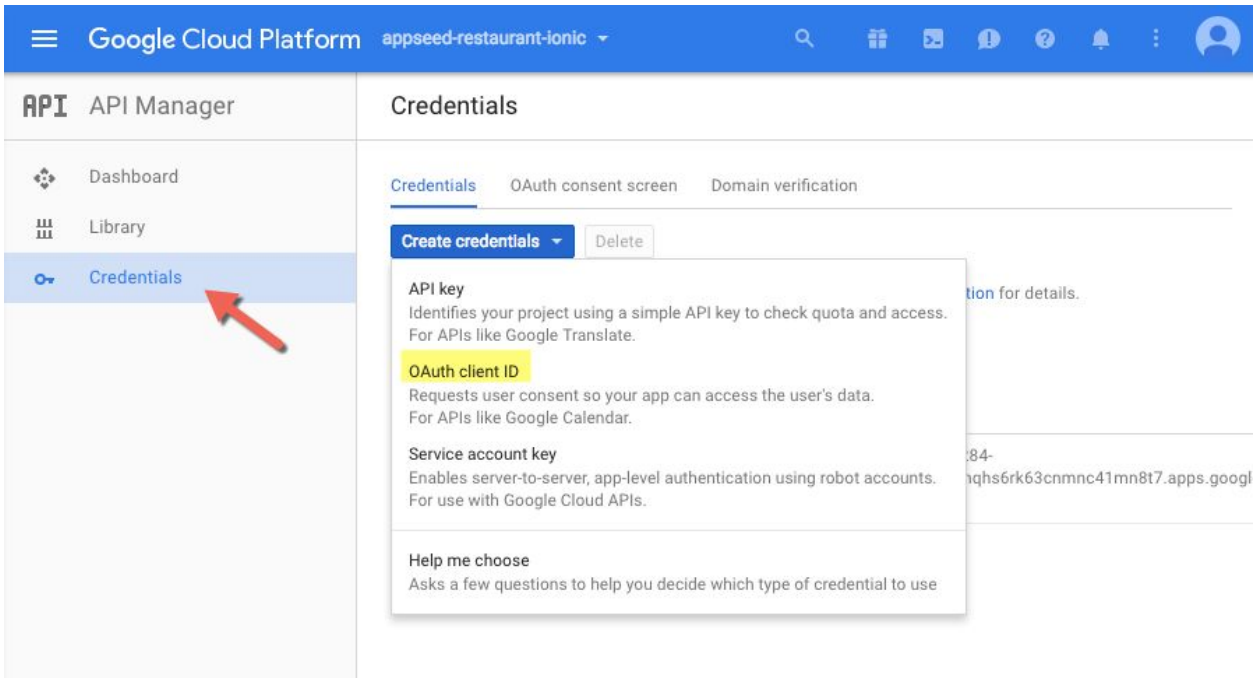


Google

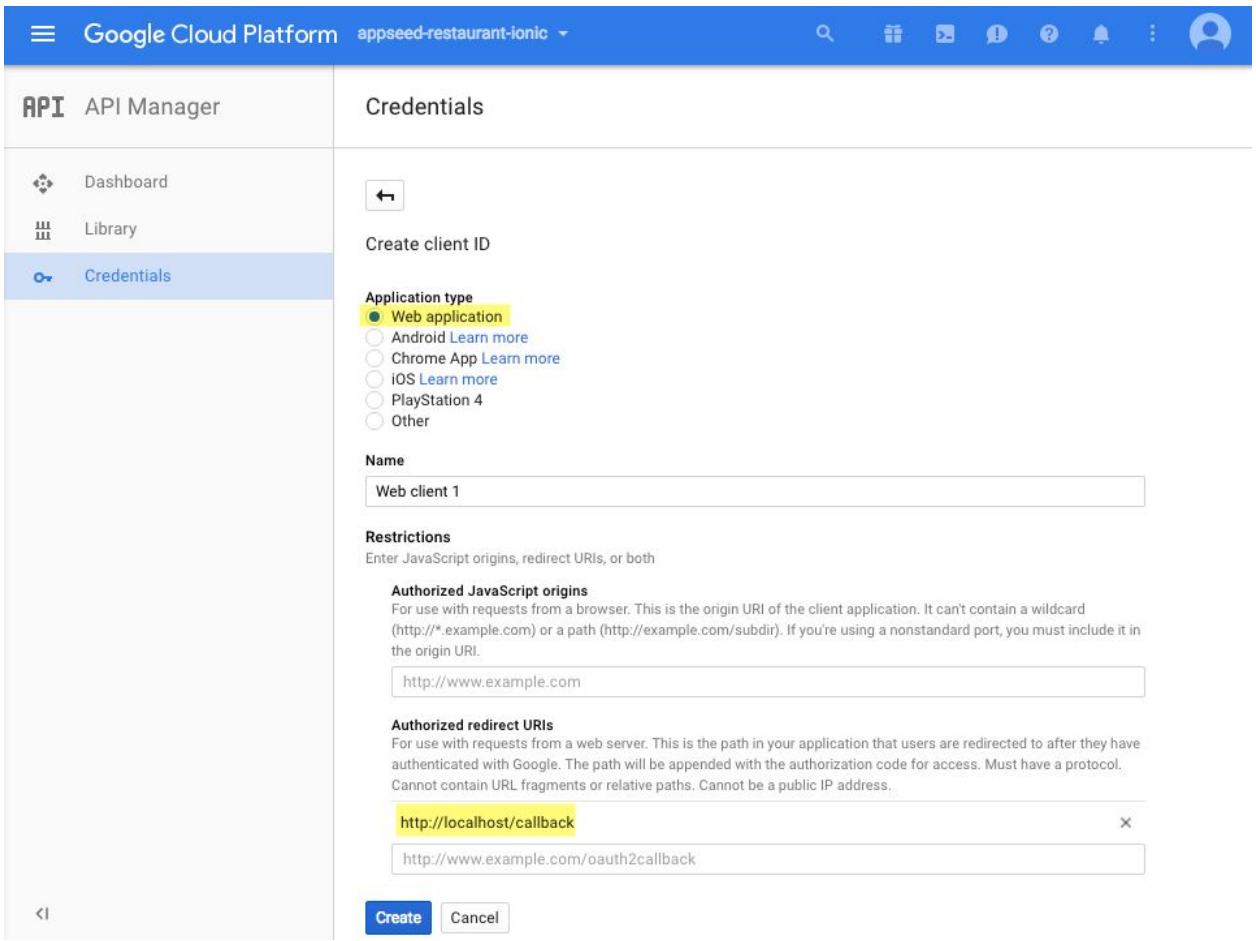
This feature integrates Google Login into the app. Google OAuth2 API is used for this purpose.

Google Application ID

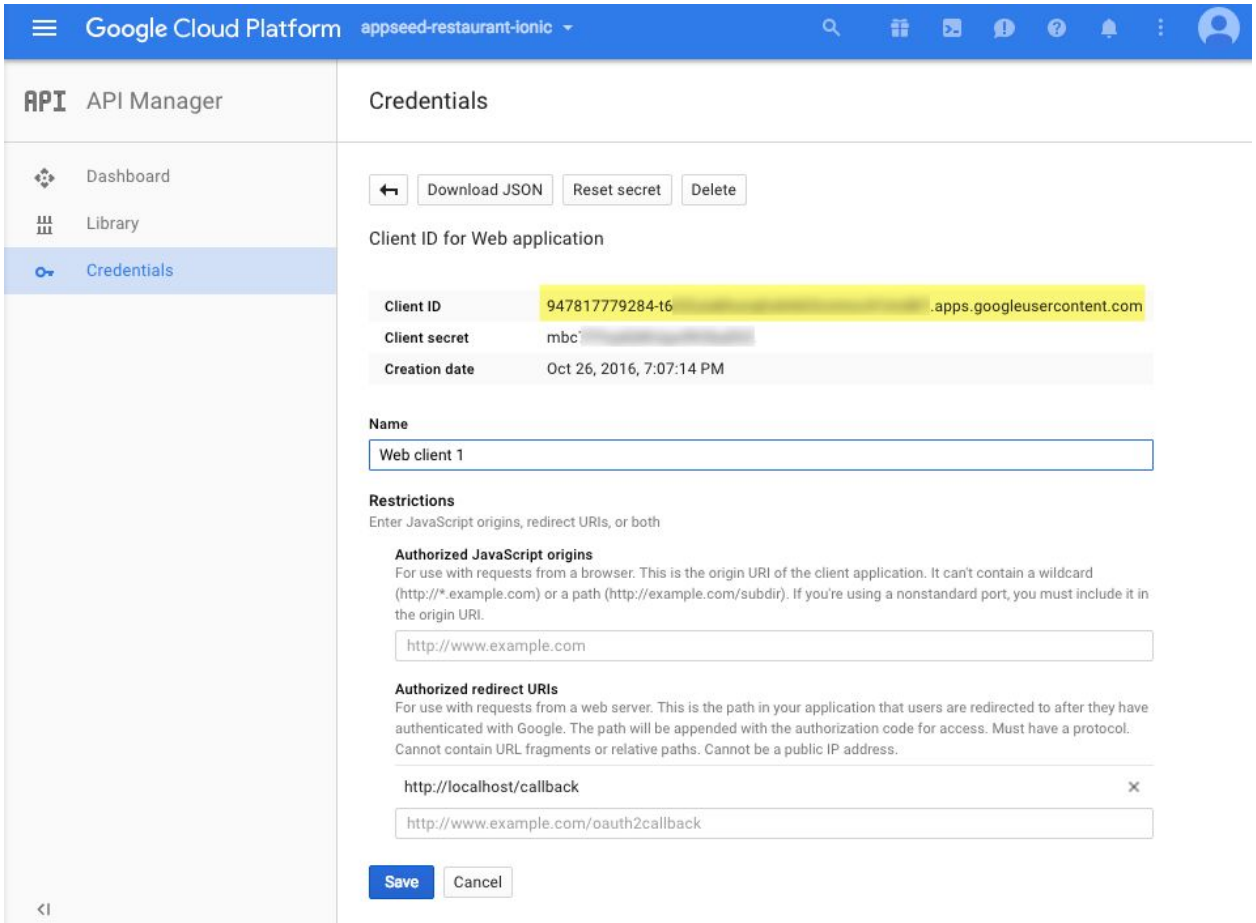
You should create and register a new Google App on Google Developer Console. In the credentials section, choose to create credentials for OAuth client ID.



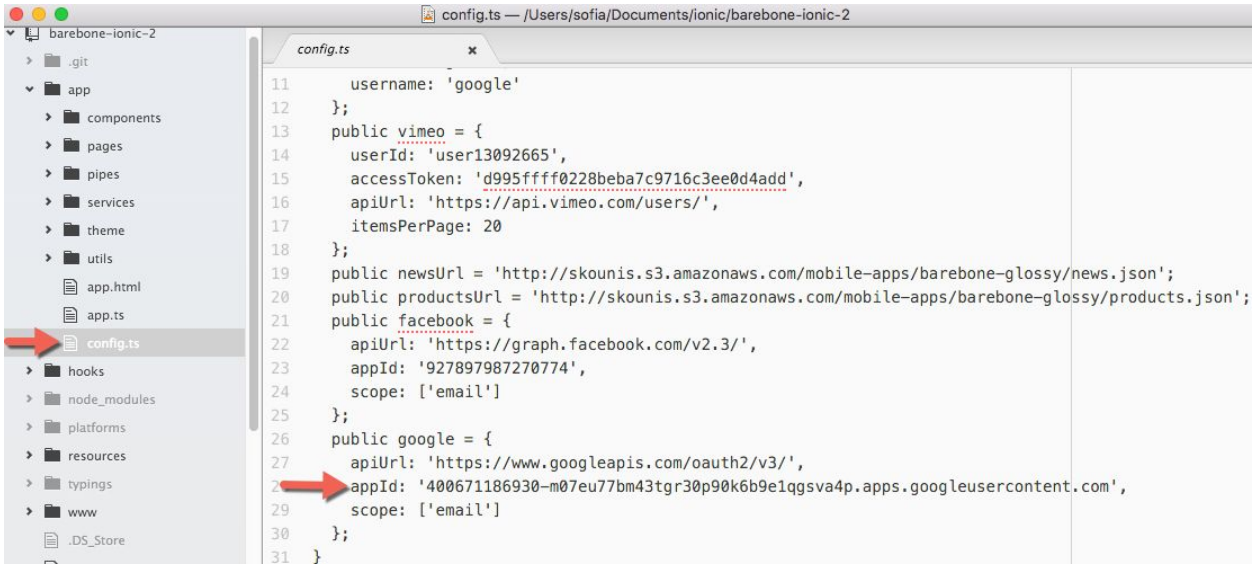
Then, select “Web application” and set the “Authorized redirect URIs” to http://localhost/callback.



Finally, copy the client ID



and set it in the `config.ts` file.



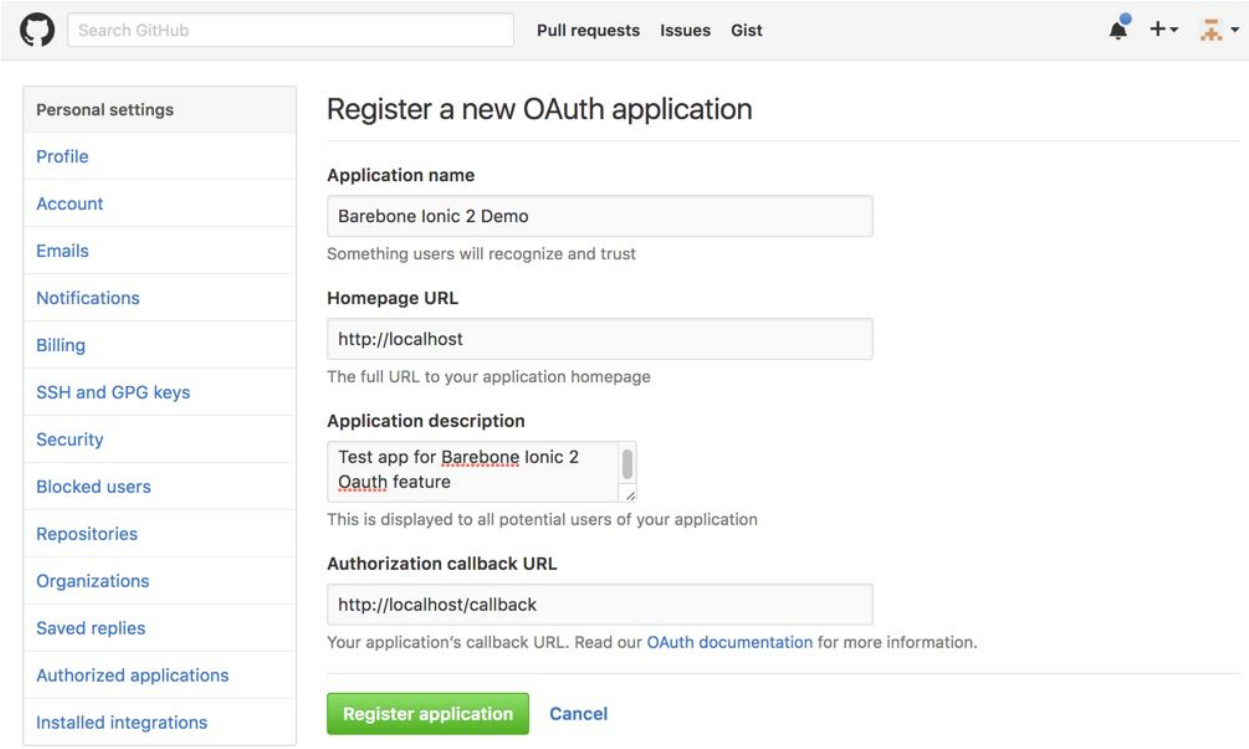
Config.ts

Github

This feature integrates GitHub Login into the app. GitHub API v3 is used for this purpose.

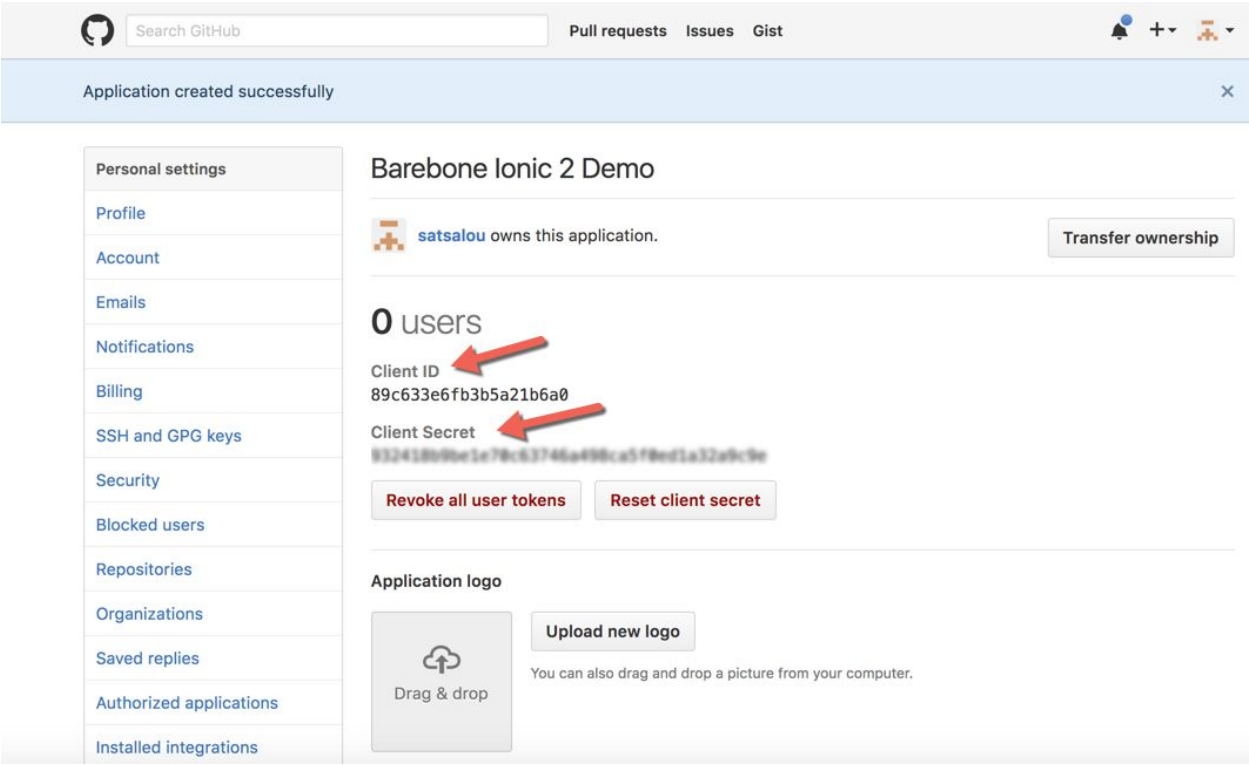
Create a GitHub Application

Before getting started, you need to register your application visiting https://github.com/settings/applications/new in order to get a unique Client ID and Client Secret. Fill in the form with your application name and application description as you like. As for the "Homepage URL" and "Authorization callback URL", set them to "http://localhost" and "http://localhost/callback" respectively. Finally, you should end up with something as the following:

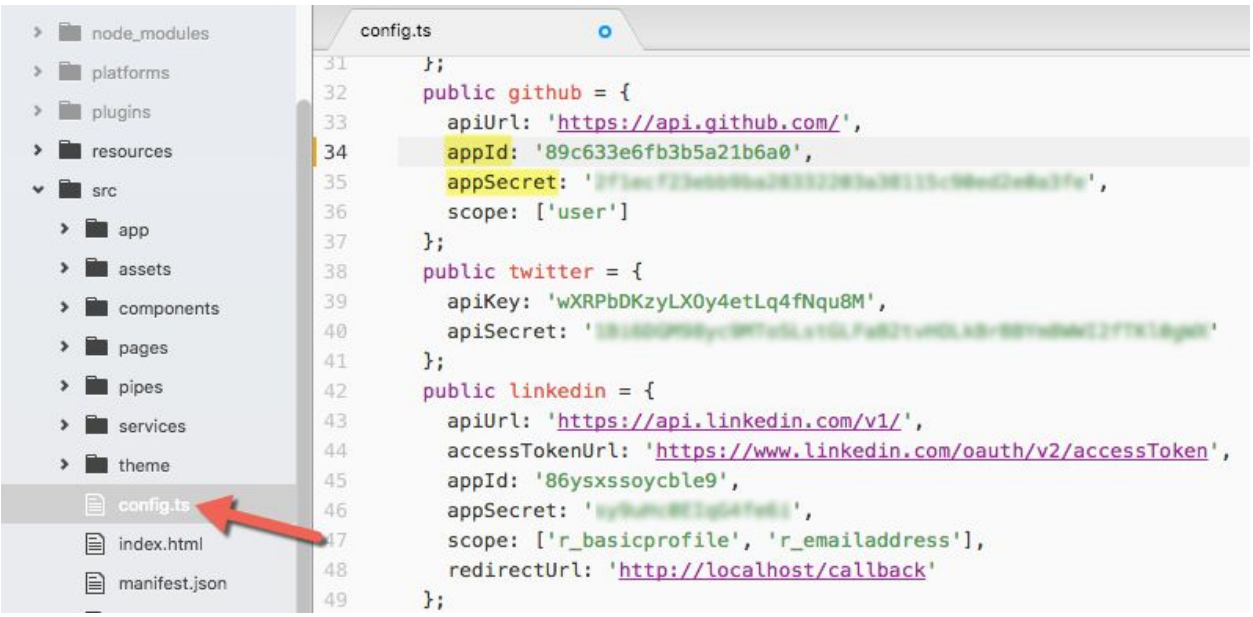


Set the Application ID and Secret

After registering your application, you should see the following page:



Make a note of your Client ID and Client Secret and open the `config.ts` file in `src` folder where you need to place them as highlighted:

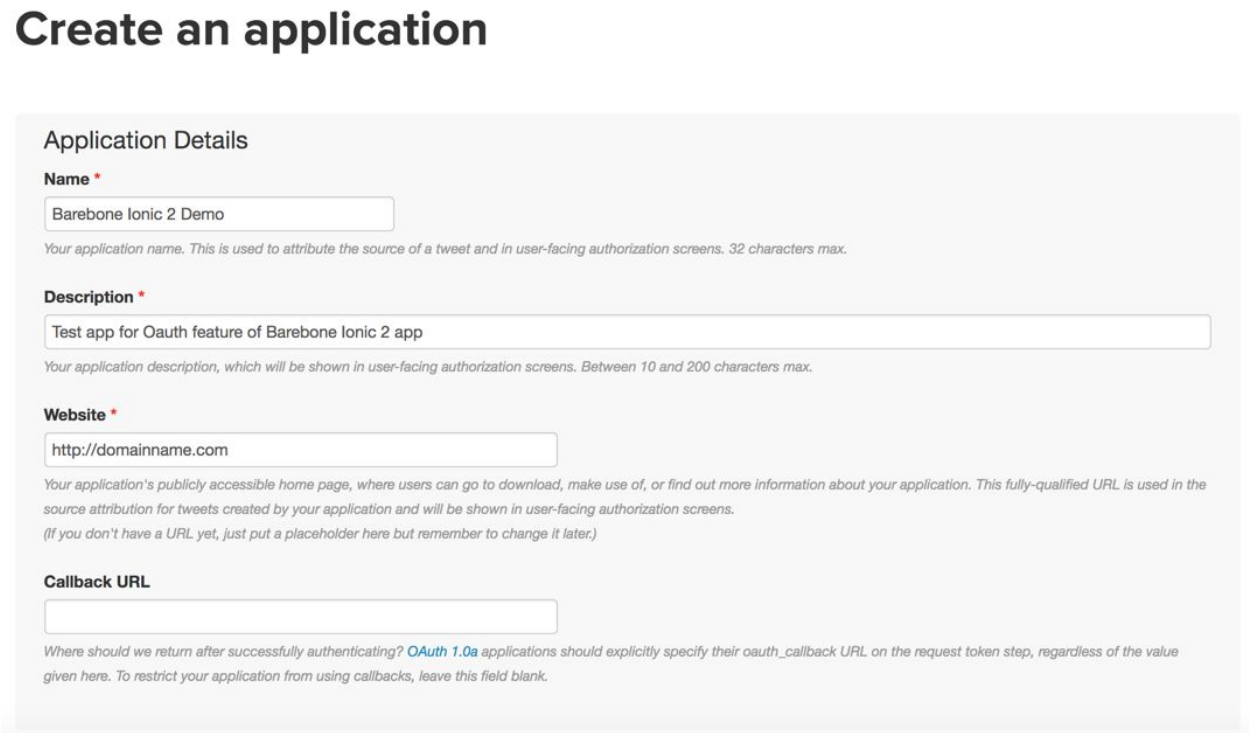


Twitter

This feature integrates Twitter Login into the app. Twitter's REST API is used for this purpose.

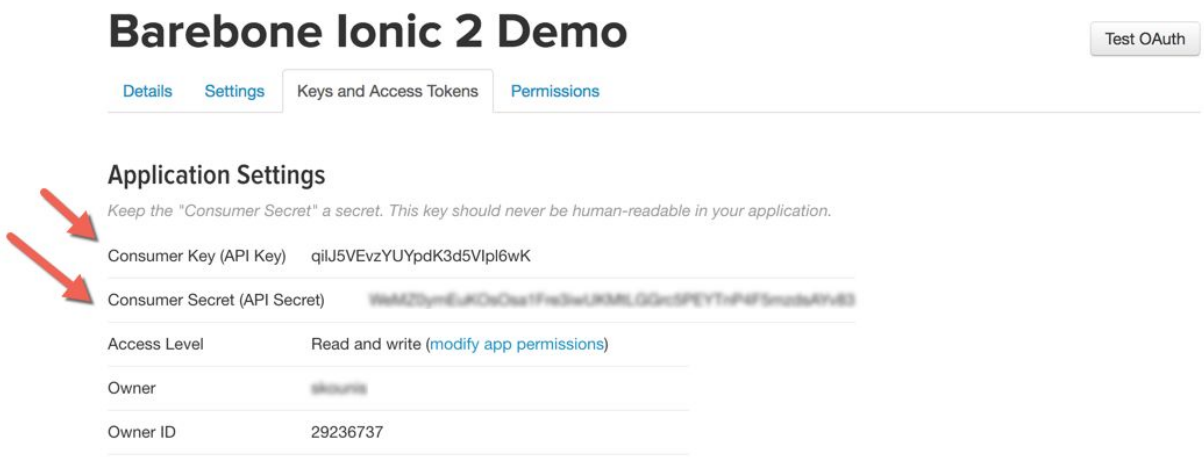
Create a Twitter Application

Firstly, go to https://apps.twitter.com/ and create a new app. Fill in the form with your application name and application description as you like. As for the “Website”, set it to “http://domainname.com”. Finally, you should end up with something as the following:

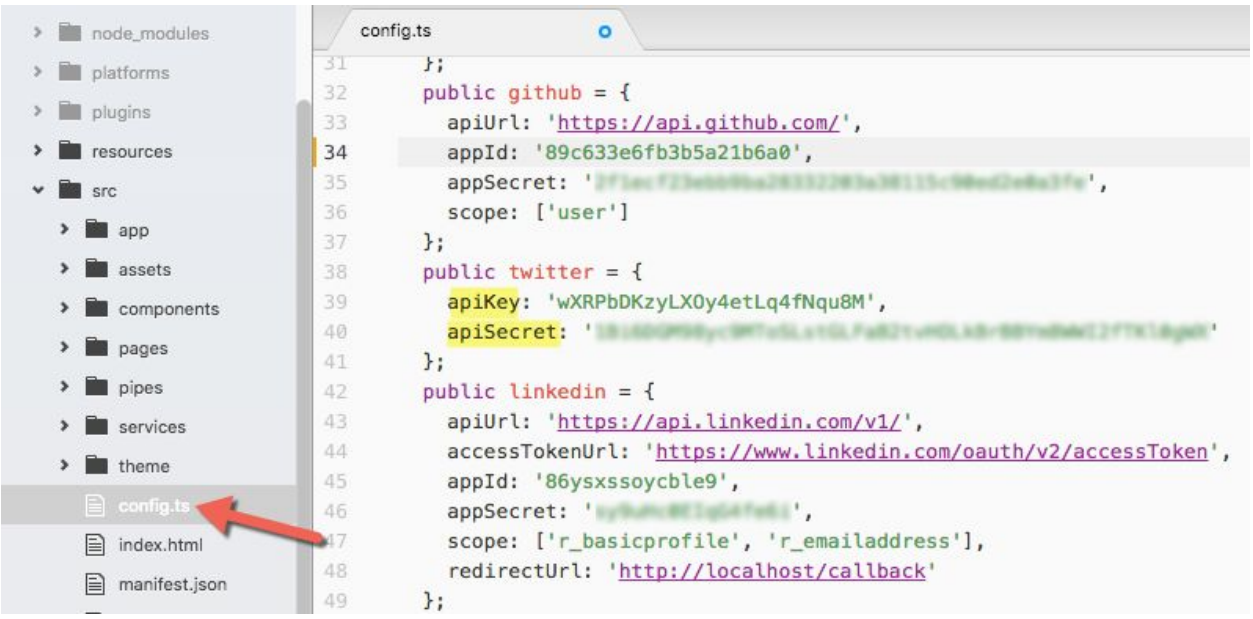


Set the Application ID and Secret

After registering your application, hit the “Keys and Access Tokens” tab and make a note of your API Key and API Secret:



Finally, open the `config.ts` file in `src` folder and place them as highlighted below:

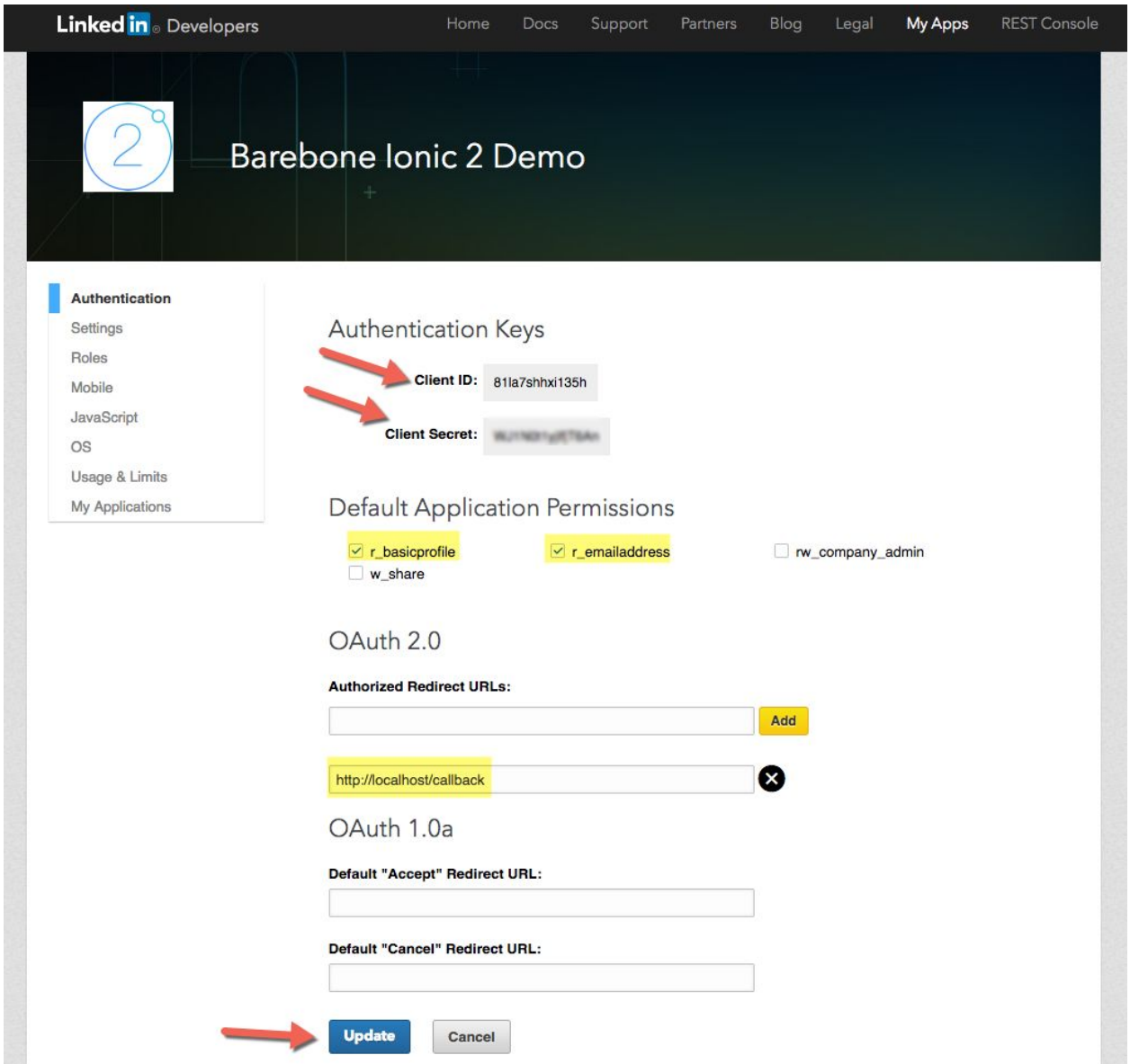


Linked In

This feature integrates LinkedIn Login into the app. LinkedIn's REST API is used for this purpose.

Create a LinkedIn Application

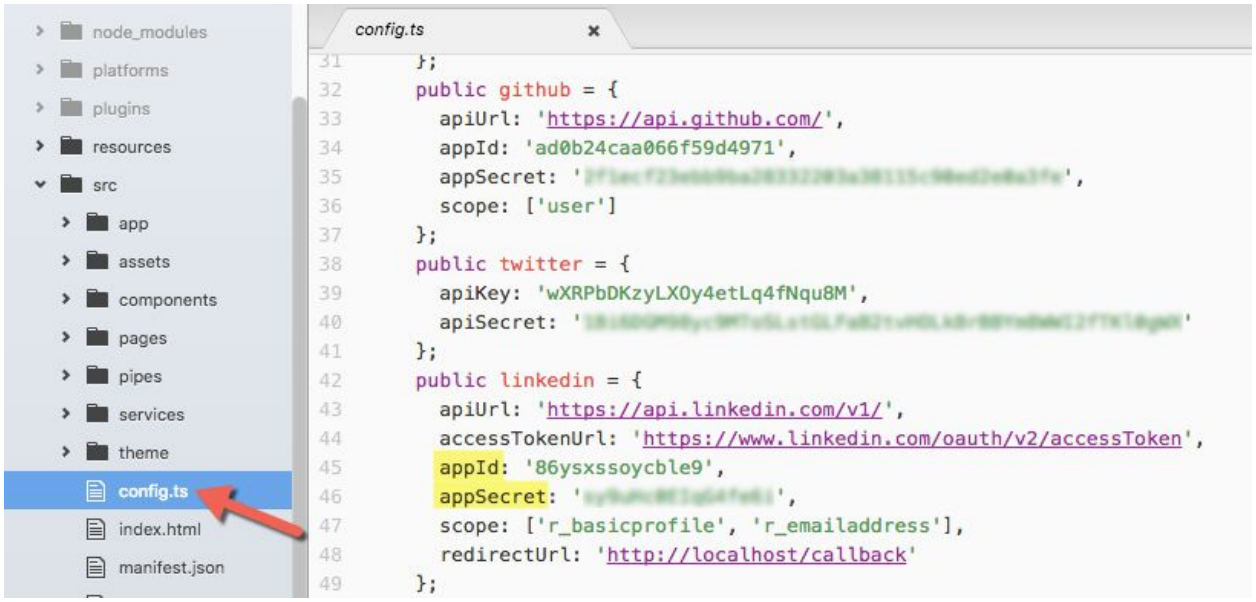
Firstly, visit https://www.linkedin.com/secure/developer?newapp= and create a new app. Fill in the form as you like. As for the "Website URL", set it to "http://localhost". After hitting the "Submit" button, you will the following screen where you should check the "r_basicprofile" and "r_emailaddress" under the "Default Application Permissions" section and add "http://localhost/callback" under the "OAuth 2.0" section. Finally, you should end up with something as shown in the following screen capture:



Do not forget to hit the “Update” button and make a note of the Client ID and Client Secret.

Set the Application ID and Secret

In order to set the client Id and Secret in the app, open the `config.ts` file in `src` folder and place them as highlighted below:



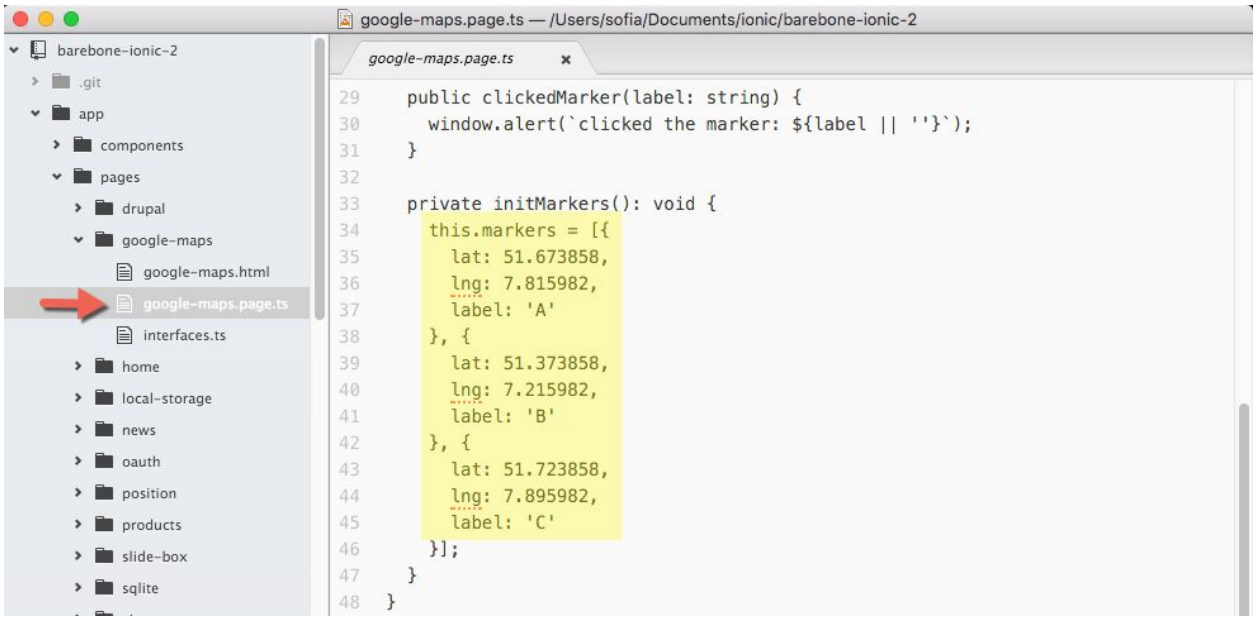
Map and Annotations

The Map screen annotations are automatically created based on the corresponding information in the `app/pages/google-maps/google-maps.page.ts` file.

Open the `google-maps.page.ts` file and adjust the `origin` property which sets the point where the map will be centered and the zoom level of the map.

google-maps.page.ts

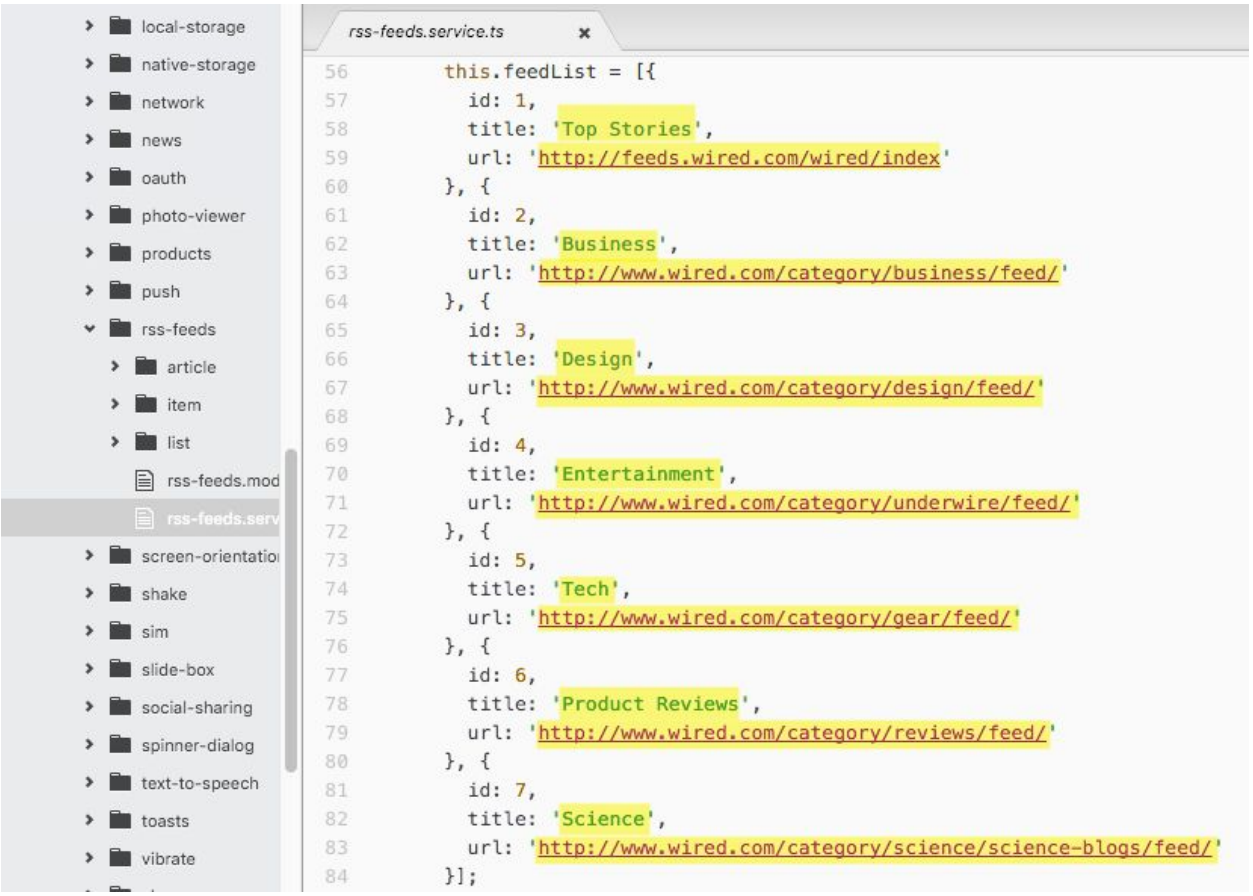
To set the annotations and their labels, adjust the highlighted code accordingly:



You are able to set as many annotation points as you wish.

RSS Feeds

In order to set the titles and the URLs of the topics that are going to be displayed in the app, open the `rss-feeds.service.js` file under the `src/pages/rss-feeds` path and define them as shown:

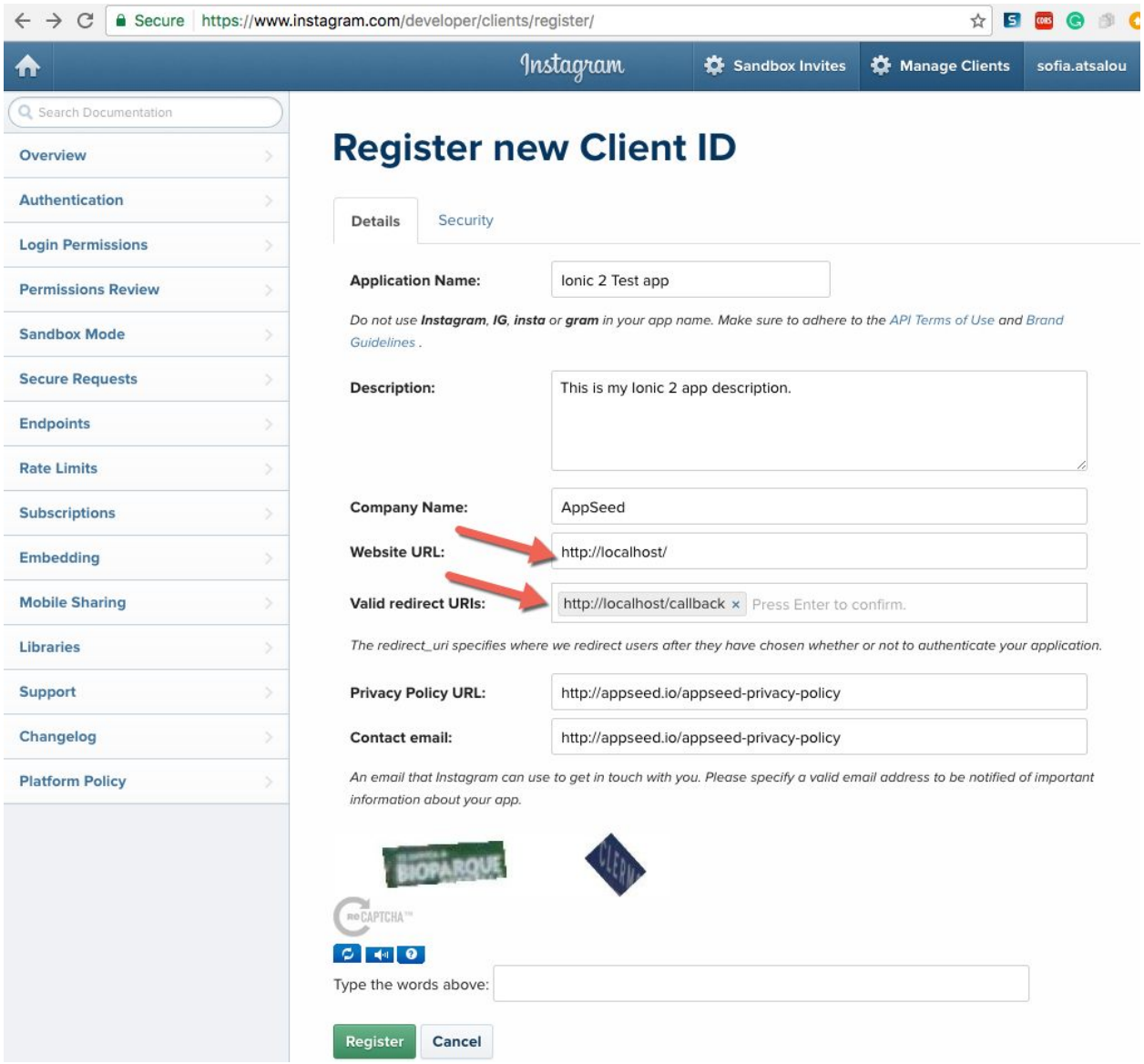


Instagram

This feature get the posts of an Instagram account. Follow these instructions to configure the app with an Instagram client.

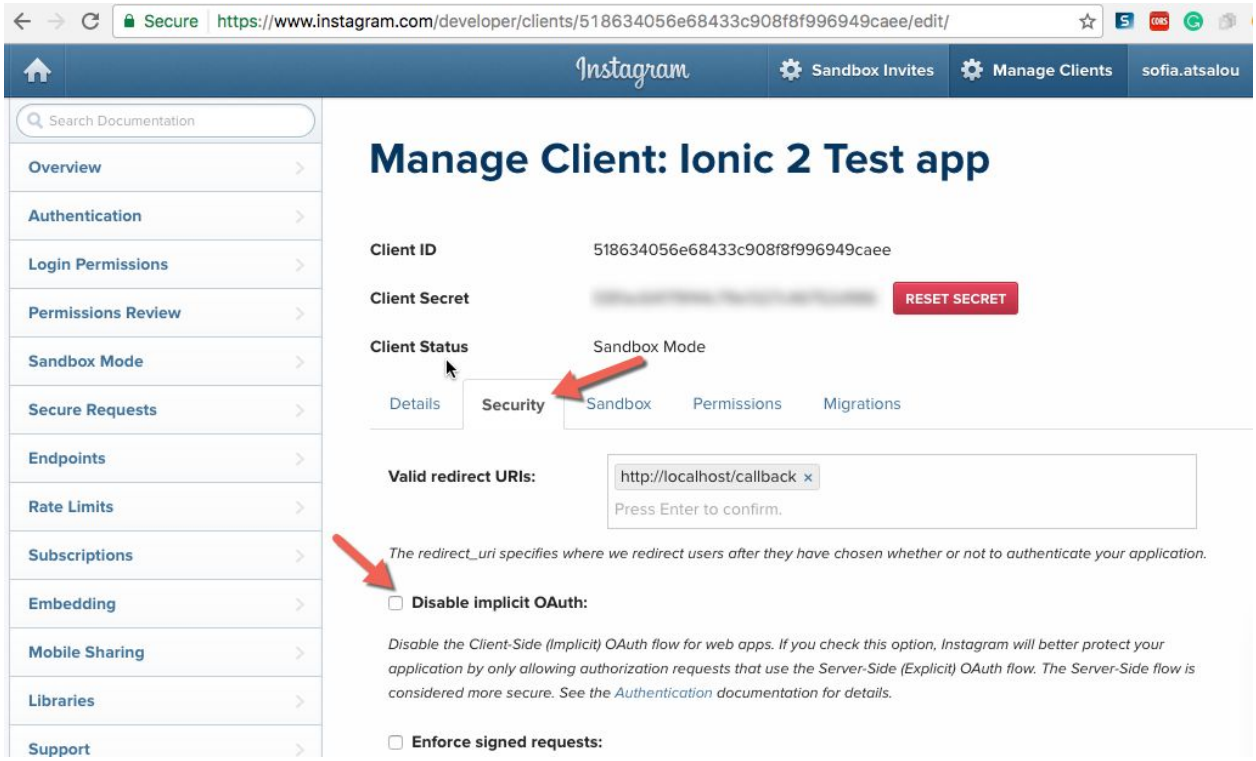
Register a client

Visit https://www.instagram.com/developer/clients/register/ and register a client by filling in the form.



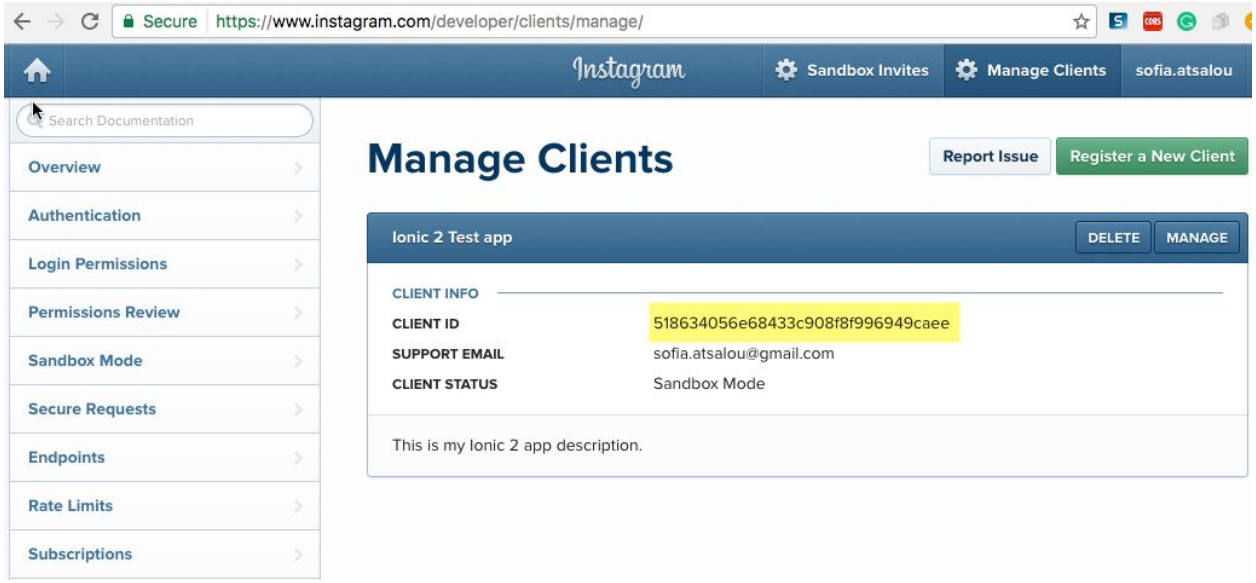
As shown above, set the website URL to http://localhost/ and the valid redirect URIs to http://localhost/callback.

After you hit the “Register” button, go to the “Manage Clients” page and hit the “Security” tab. There, uncheck the “Disable implicit OAuth” option.

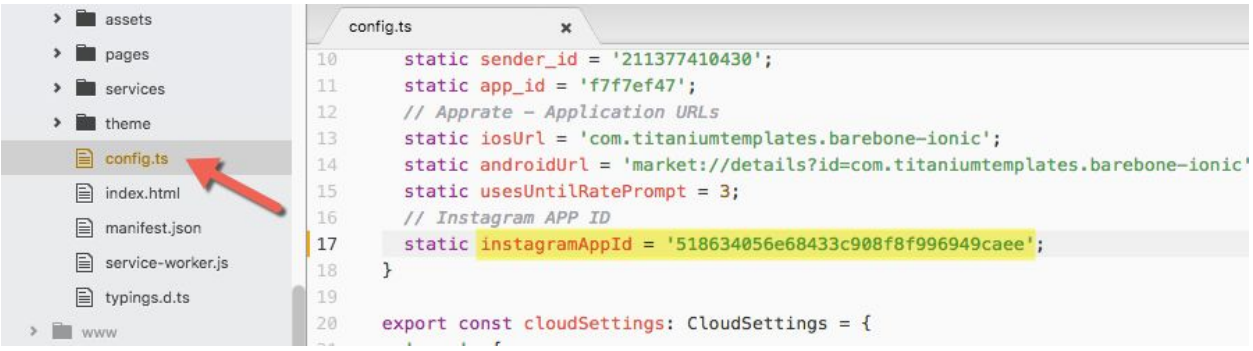


Set client ID in app

Once you have created your client you should be able to view it on the “Manage Clients” page. Copy the Client ID.

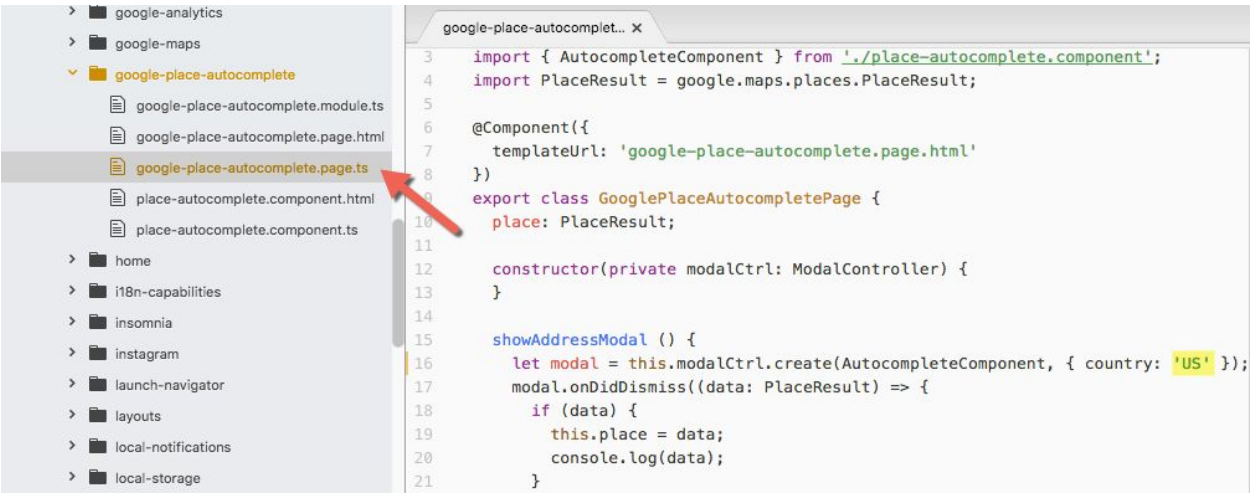


Finally, set it in your `config.ts` file under the `app/` folder.



Google Places Autocomplete

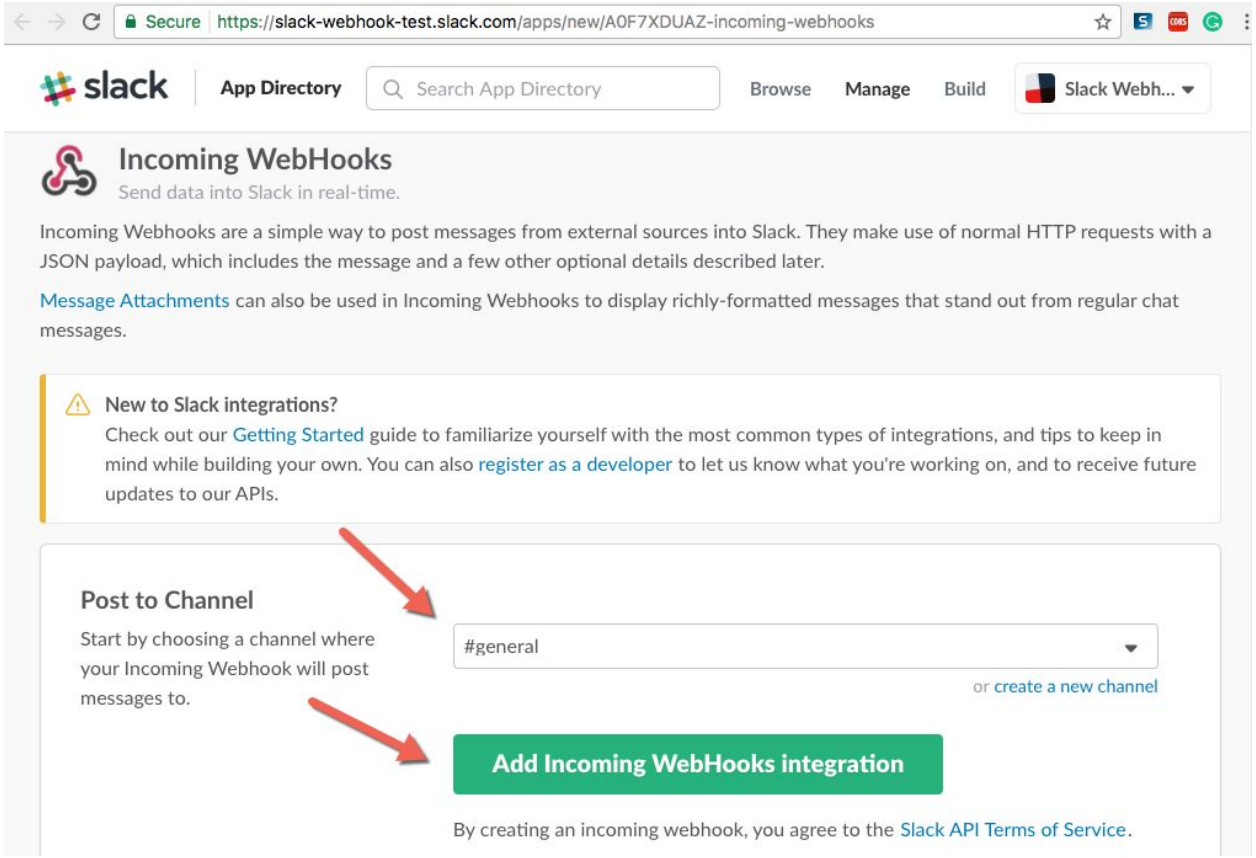
This app demonstrates how to use Google Places API for place predictions. The Google Places API returns places such as businesses, addresses and points of interest as a user types in the search bar. In order to restrict result to places of a certain country, open the `google-place-autocomplete.page.ts` file under the `project/src/pages/google-place-autocomplete` path and change the country code to your preference.



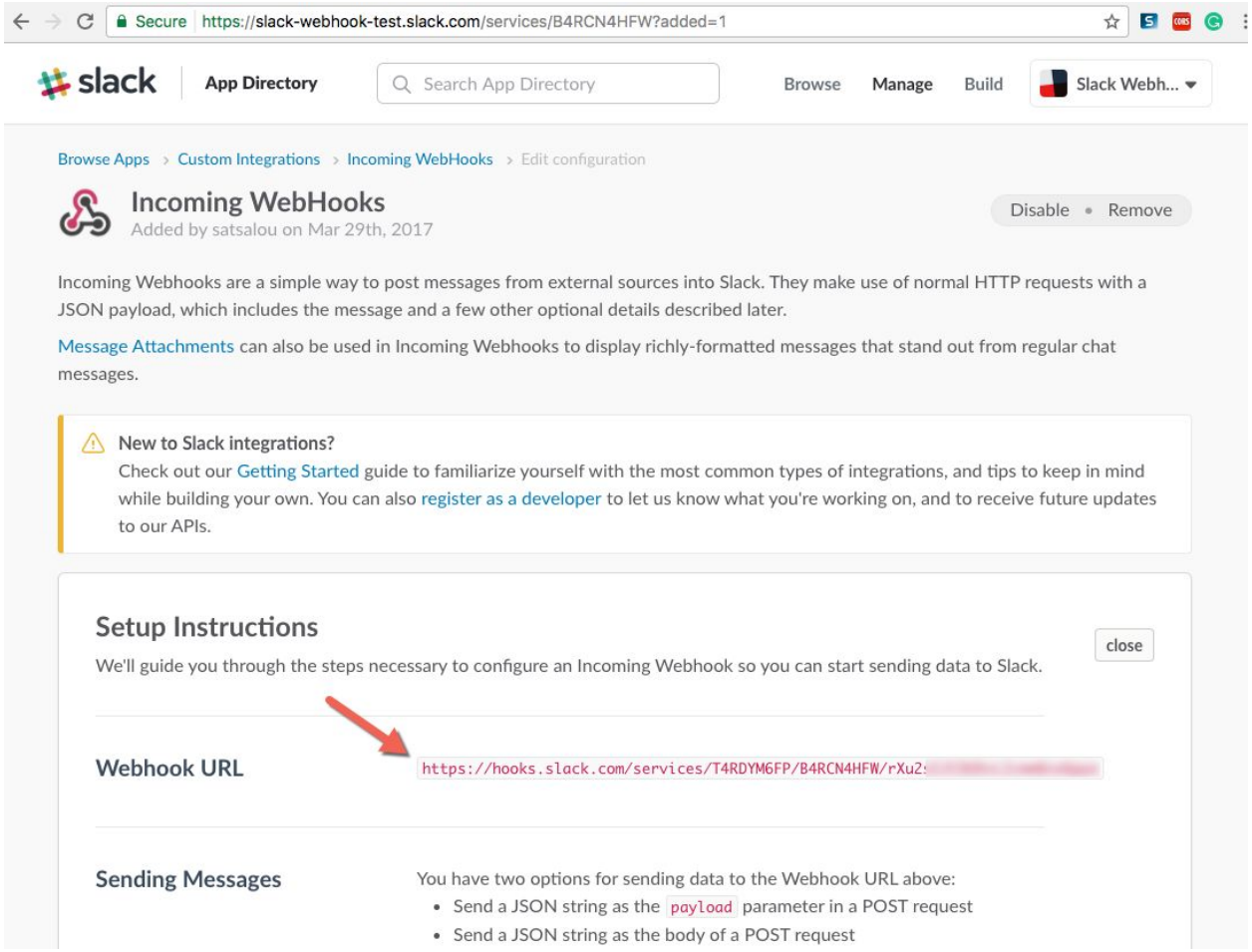
Slack Webhook

This feature allows the app users to send messages to a Slack channel without leaving the app. Before proceeding to the following configuration steps, make sure you have created a team and a channel into Slack. If you have not created yet, then you can do it by visiting https://slack.com/create#email.

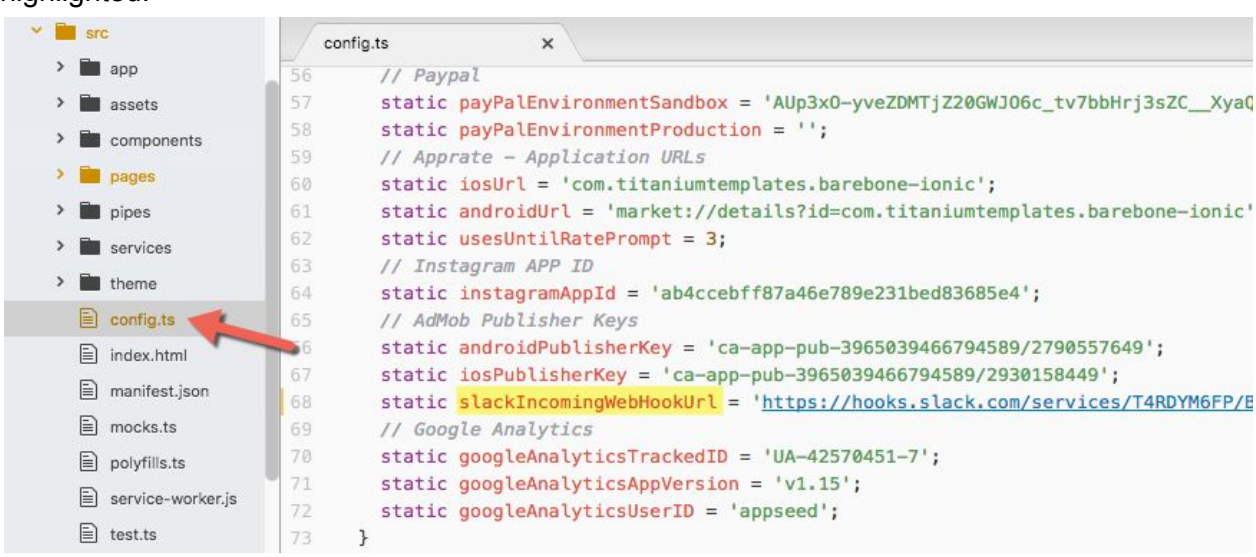
Firstly, you should set up an incoming webhook integration in your Slack team. Visit https://my.slack.com/services/new/incoming-webhook/, select the channel into where the app will post the messages and hit “Add Incoming Webhooks integration”.



On the next screen, make a note of the webhook URL.



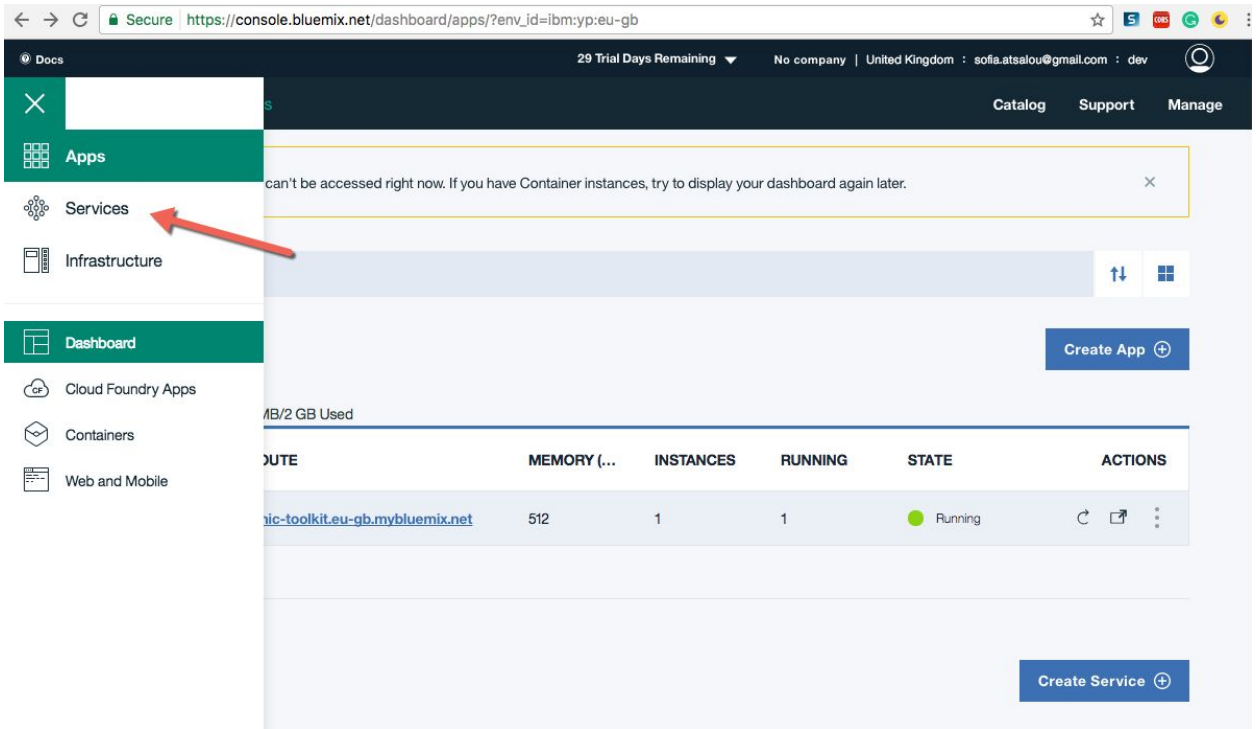
Finally, open the config.ts file in the src folder and set the `slackIncomingWebHookUrl` as highlighted:



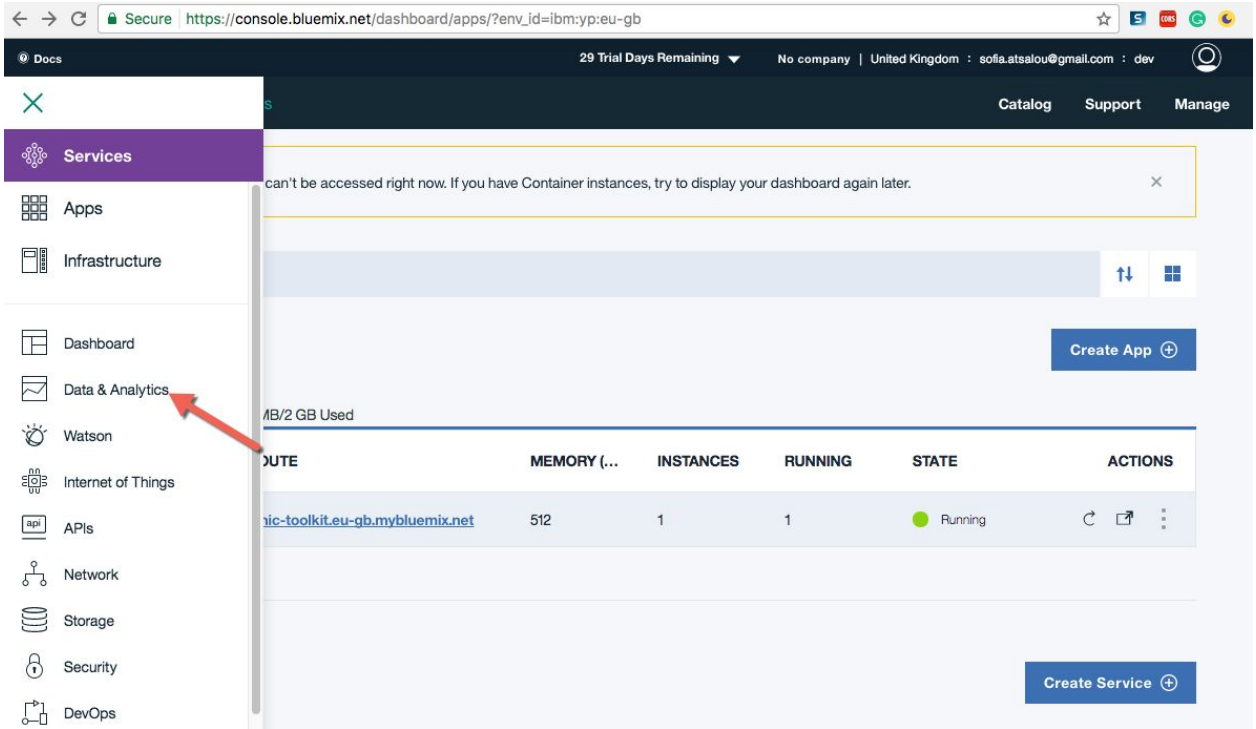
PouchDB and Cloudant

The app uses PouchDB to store data locally and it synchronizes the data with a remote database hosted on Bluemix Cloudant when there is a network connection.

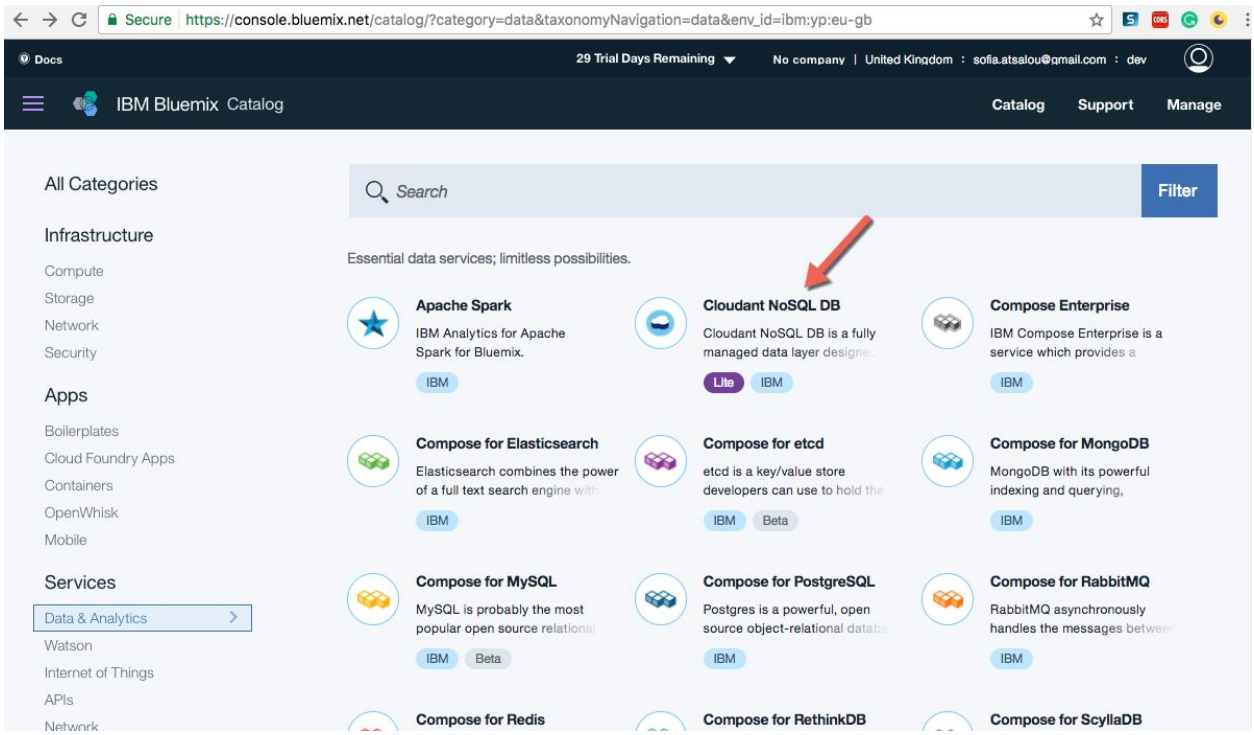
To configure the app with your own remote database, visit https://console.bluemix.net/ and create an account. Then, hit the menu icon at the top left side and select “Services”.



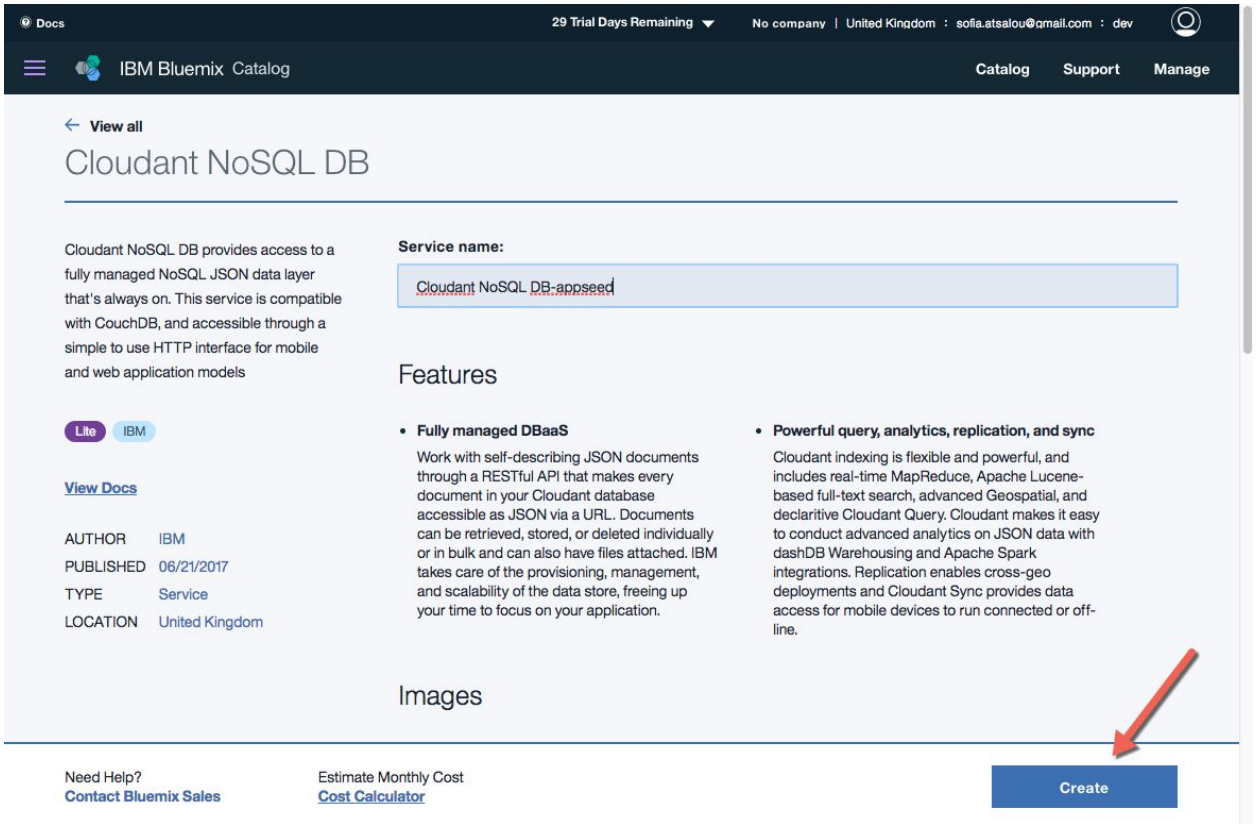
Then, select “Data & Analytics” from the menu.



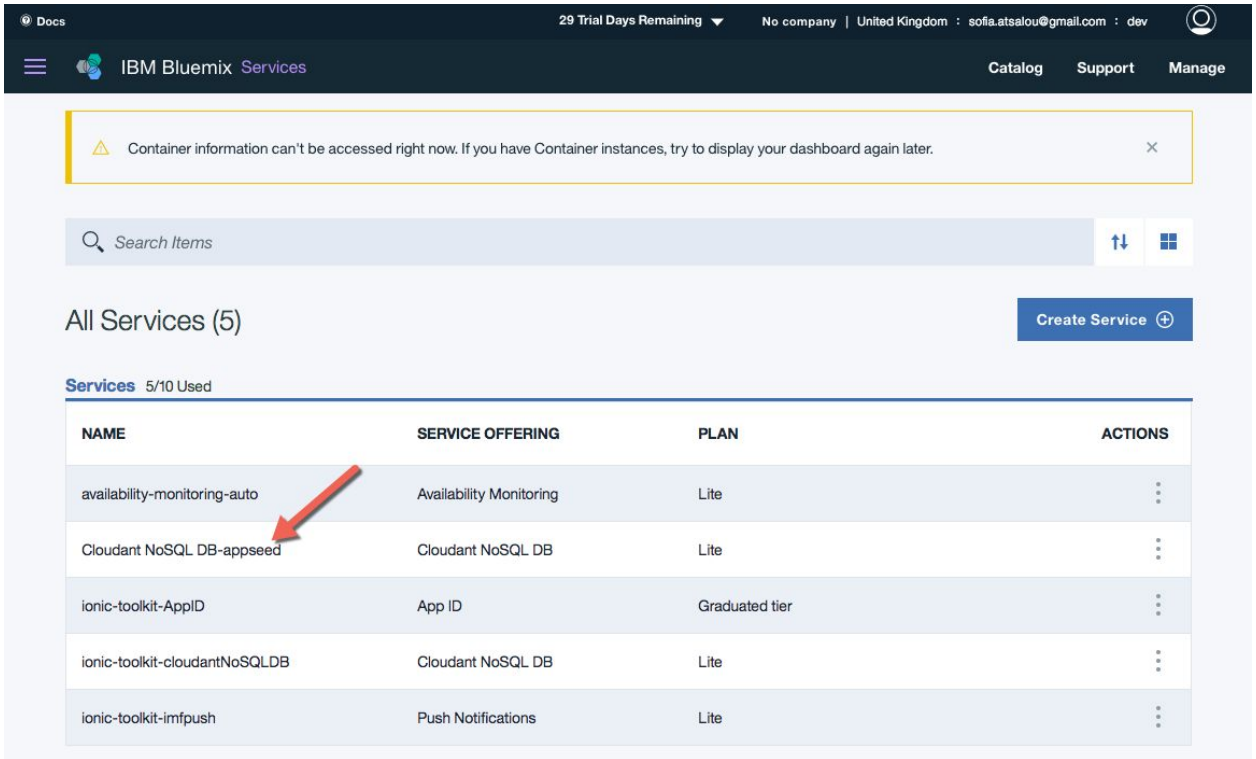
Click the “Create Data & Analytics service” button and, then, select “Cloudant NoSQL DB”.



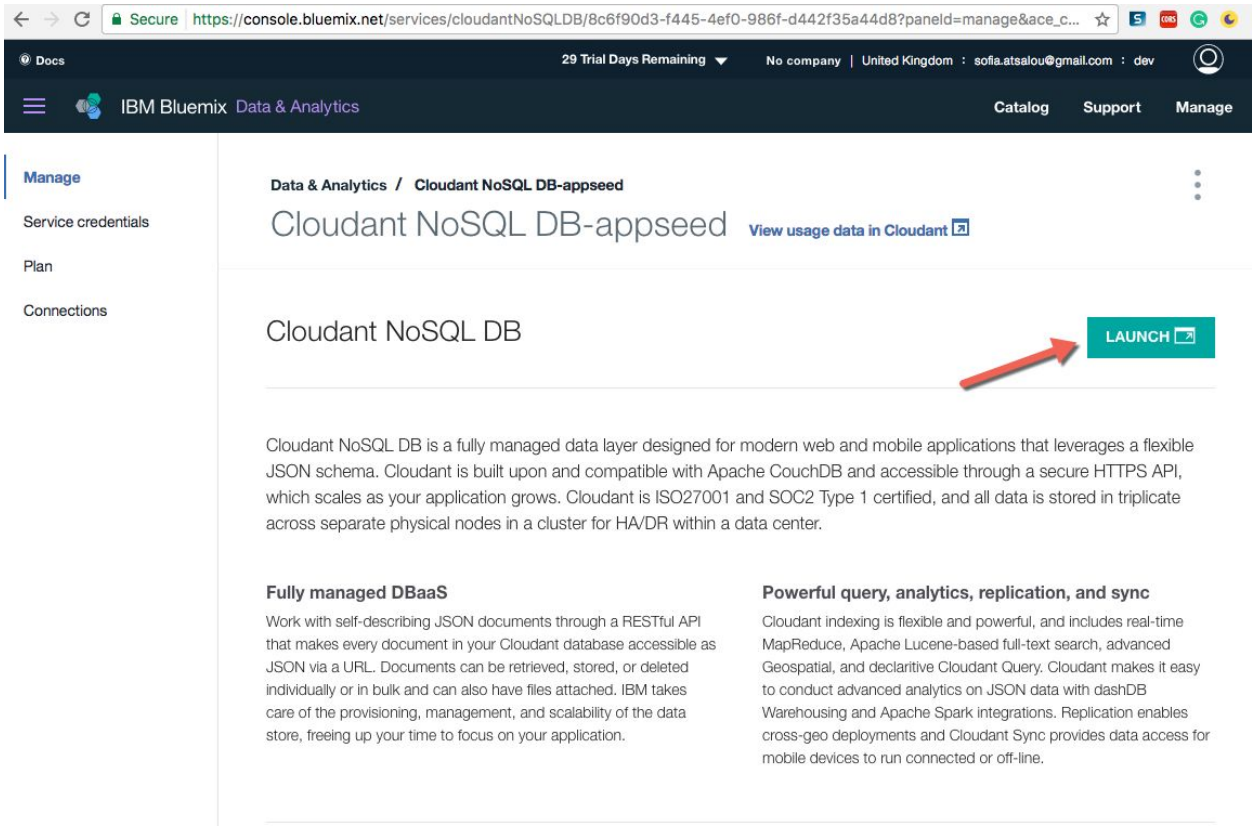
On the next screen, give your service the name you prefer, choose your pricing plan and hit “Create”.



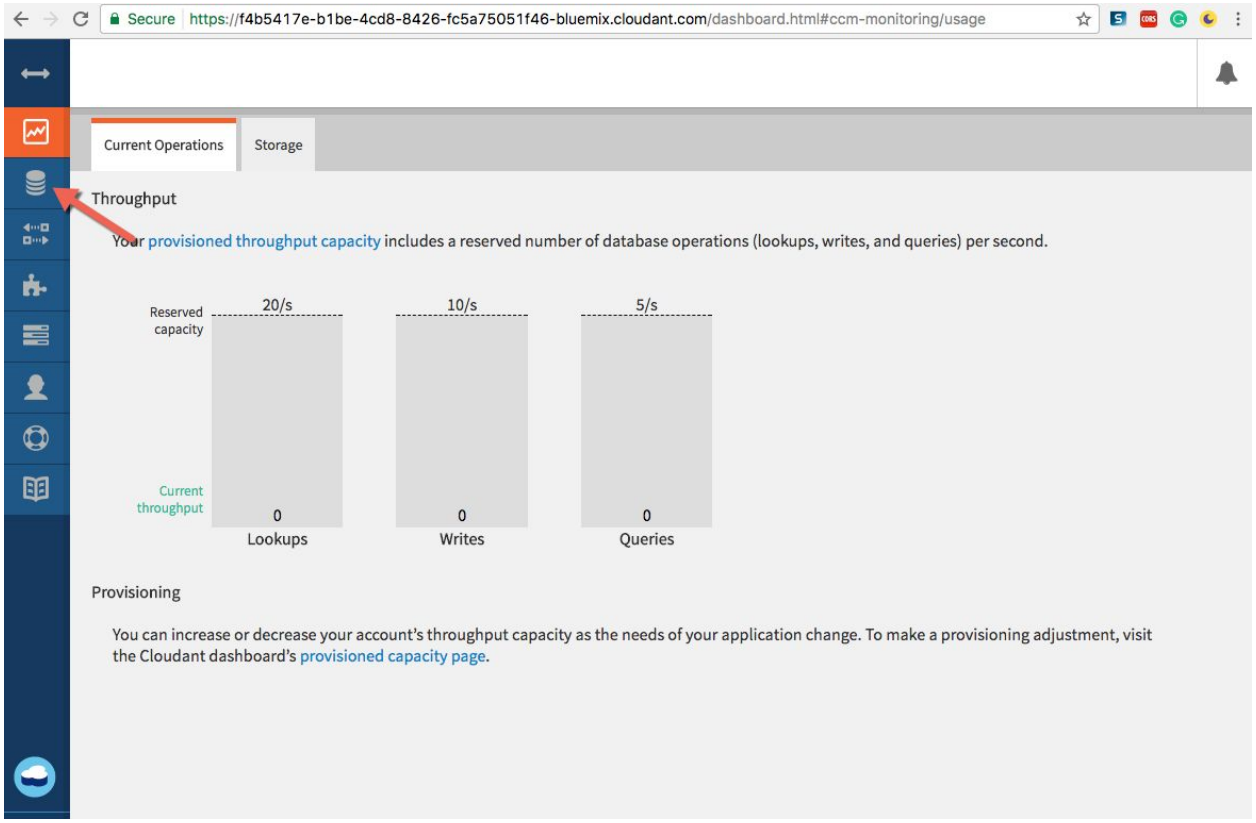
After your service is created, you will be redirected to a screen showing all your services. Select the service you have just created.



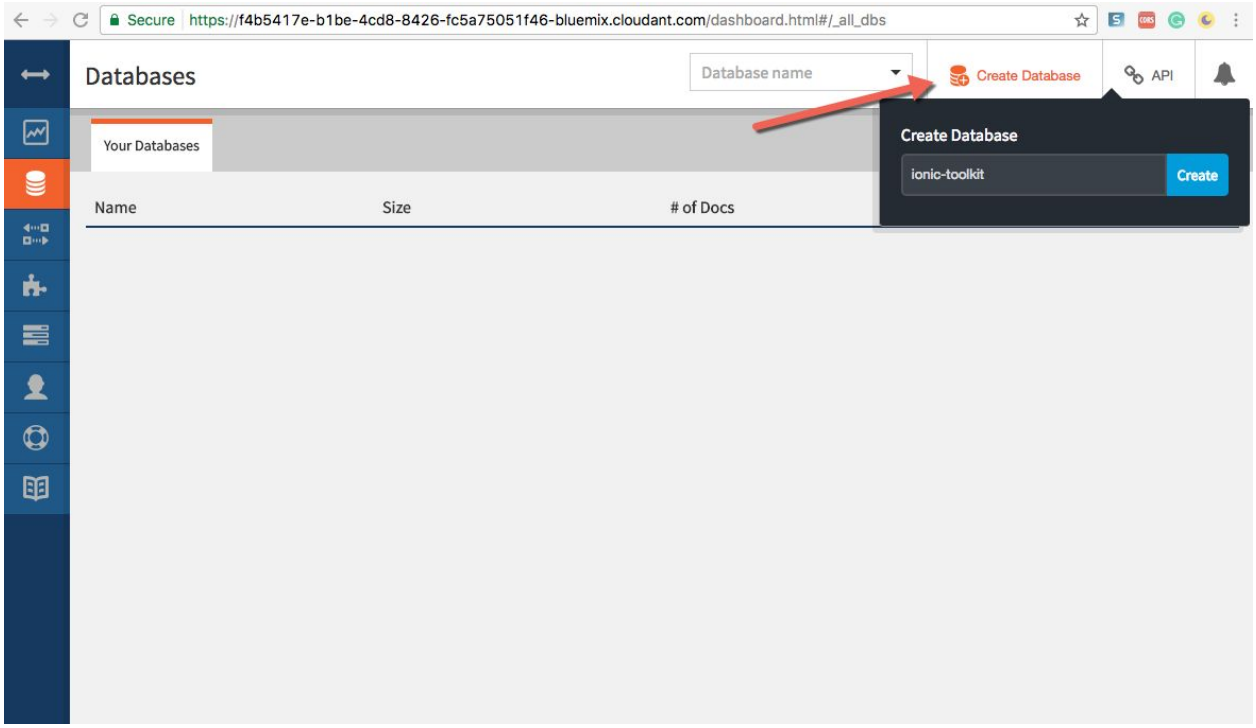
Then, hit the “Launch” button.



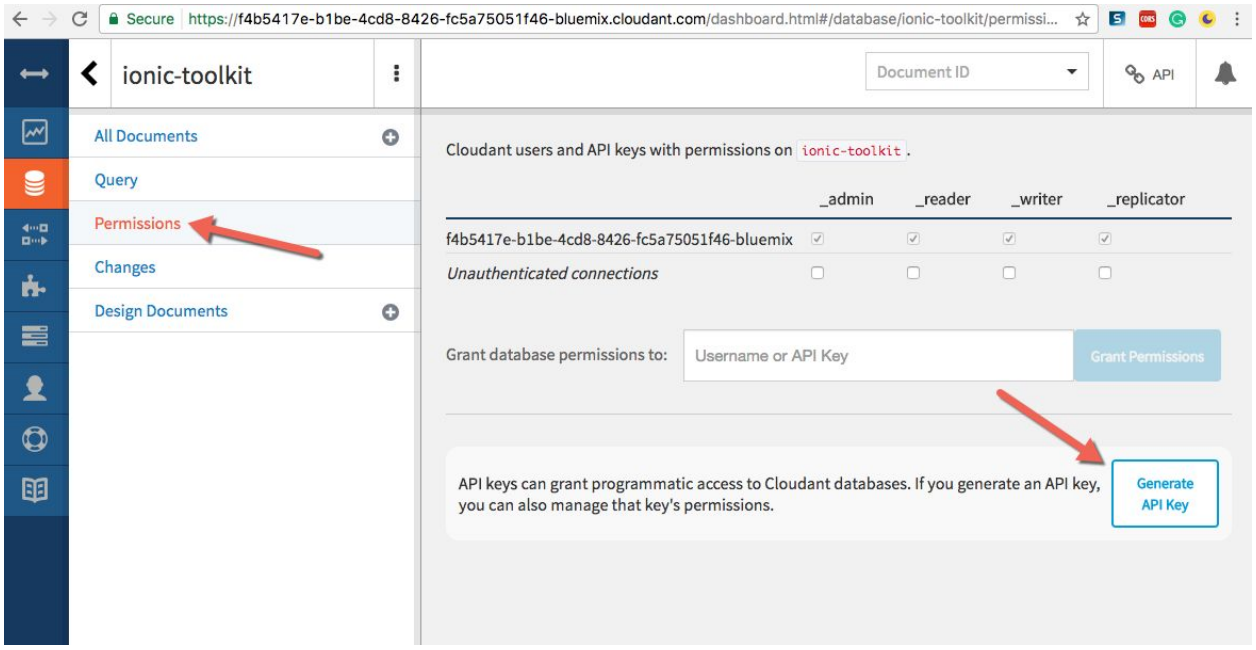
Select the second button from the side menu to view your databases.



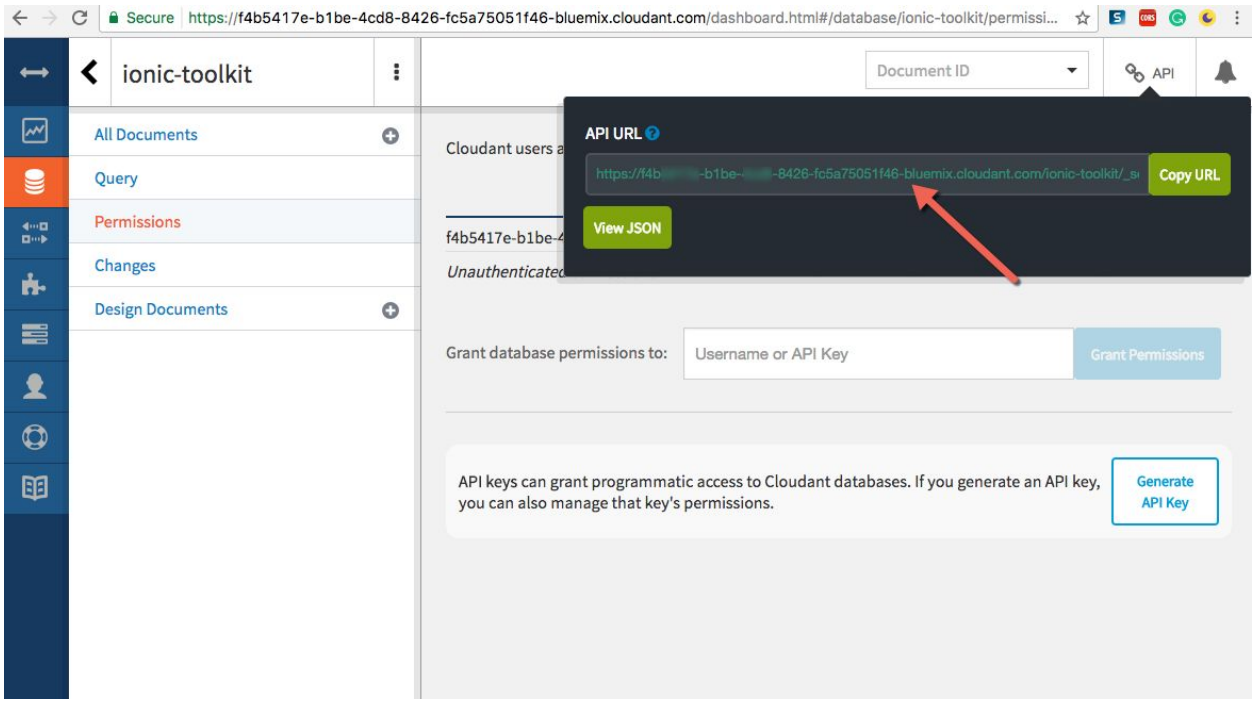
Hit the "Create Database" button and give it a name.



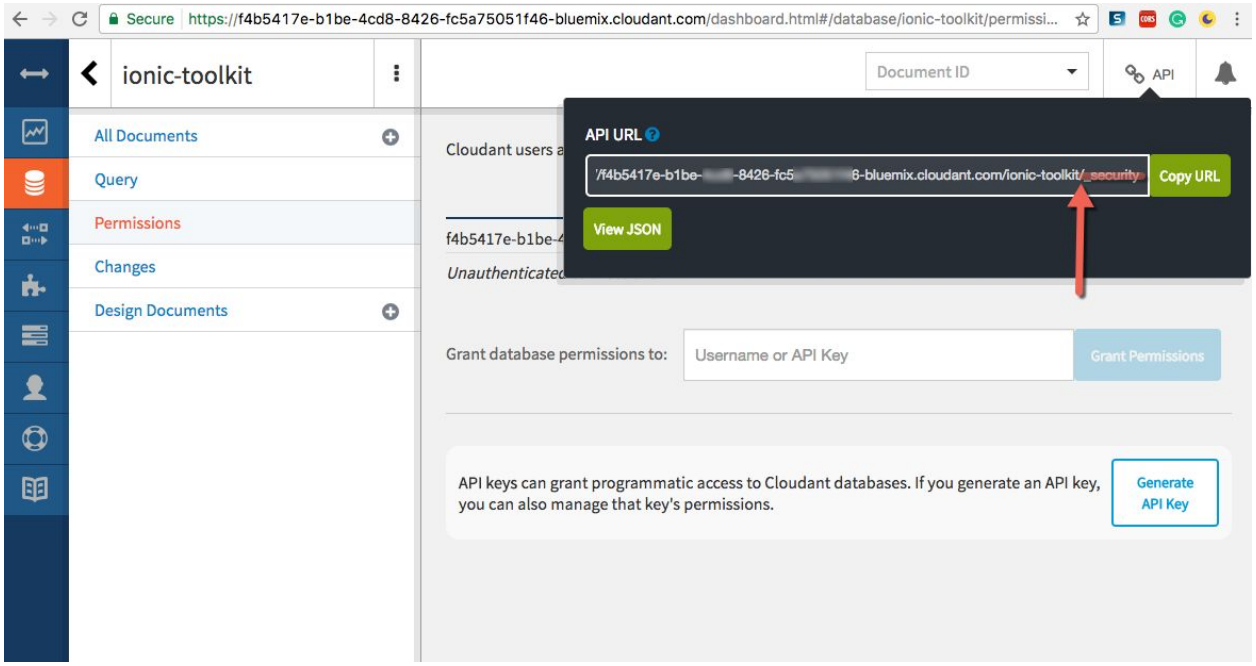
You will also need to generate an API key for your remote database. To do that, go to the "Permissions" section and click on the related button.



Make a note of the generated username and password. Additionally, make a note of your API url by hitting the "API" button.



Open the `config.ts` file under the `src/` folder. Set the key, password and the name of the database you created. Also, set the URL to the remote database by using the API URL you noted earlier after removing whatever exists after your database name.

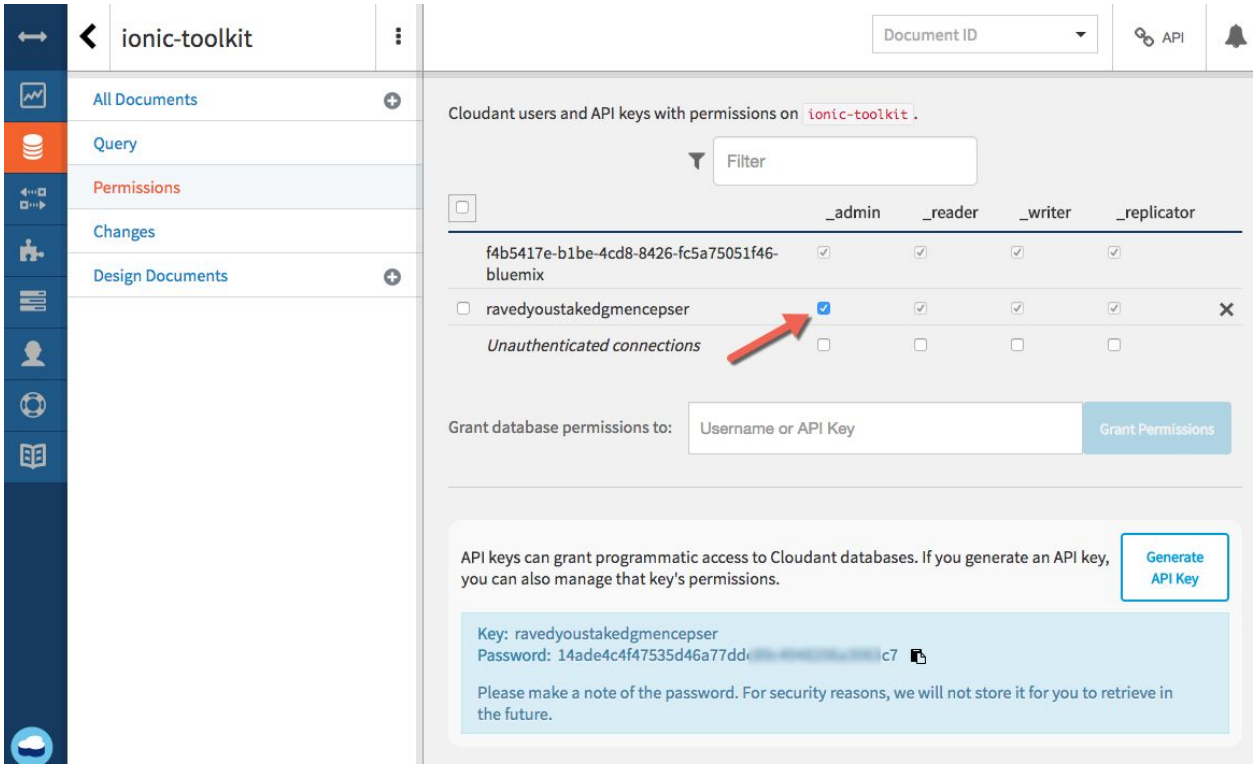


Finally, your `config.ts` file should look similar to what is shown below:

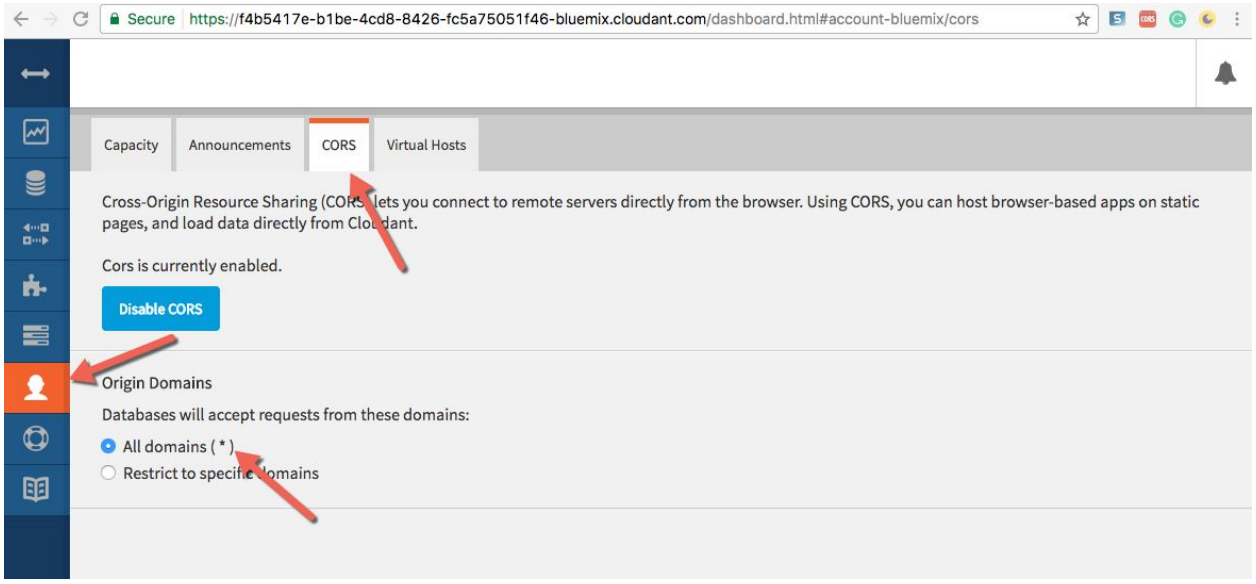
```
config.ts

// CouchDB/PouchDB/Cloudand
// Enable CORS - https://cl.ly/3U2s0r3u3e0J
// key/password - generate here https://cl.ly/2y0v0Y3P1E1k
static couchdb = {
  key: 'frormelfulessetoureanyue',
  password: '50dd799df0494c7feb75bd184b2f4149725892d7',
  local: 'ionic-toolkit-db',
  remote: 'https://6e299356-e39c-4549-9dd5-7161481481f8-bluemix.cloudant.com/ionic-toolkit'
};
```

Also, do not forget to give your API key *admin* permissions.



Finally, you need to enable CORS by going to “Account”, selecting “CORS” tab and choosing the “All Domains(*)” option.

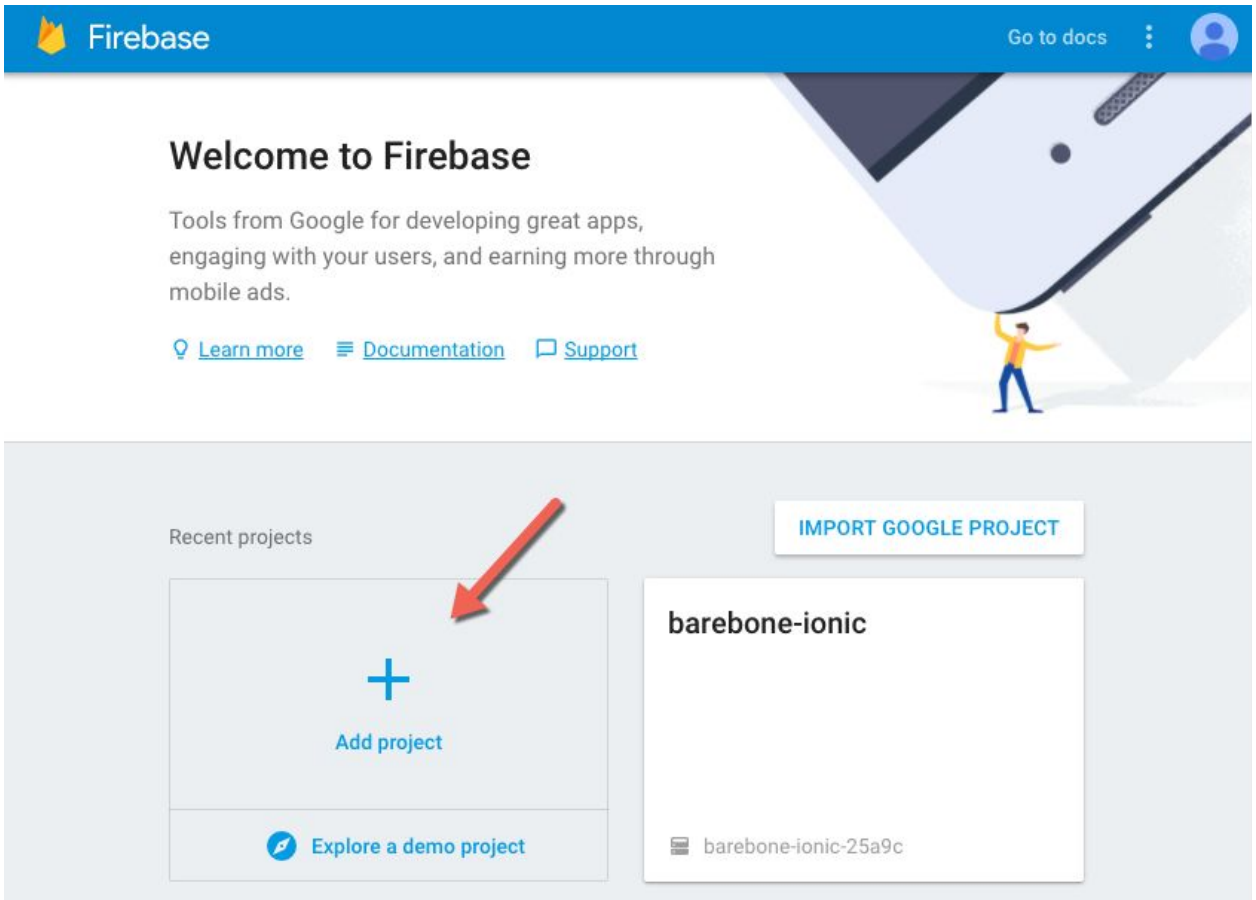


Firebase

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

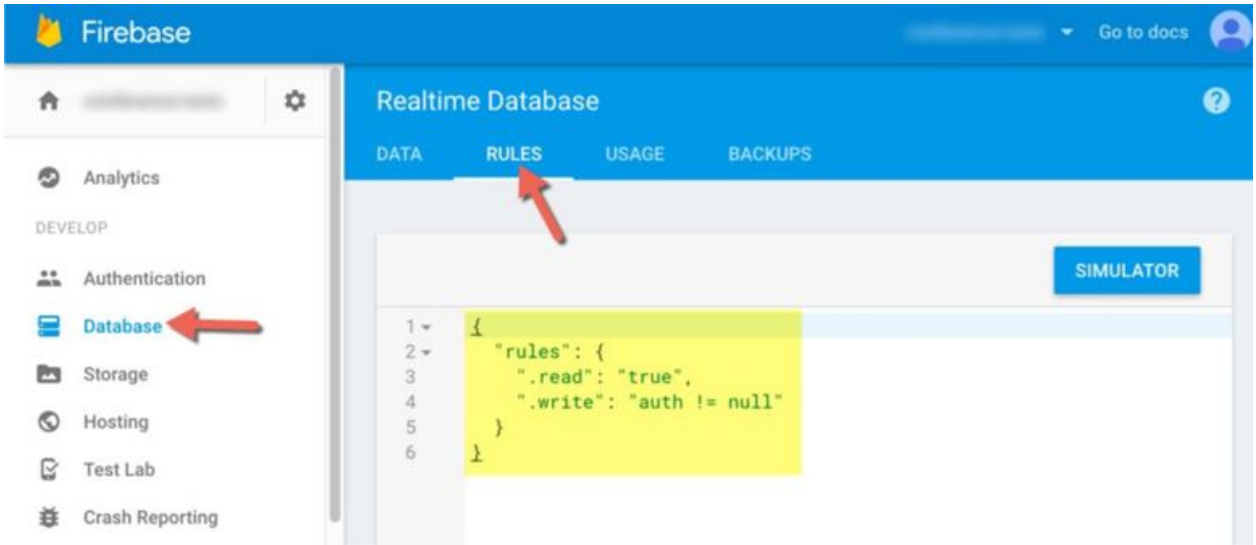
Firstly, visit https://firebase.google.com/ to create a free Firebase account and create a new app.



Set database permissions

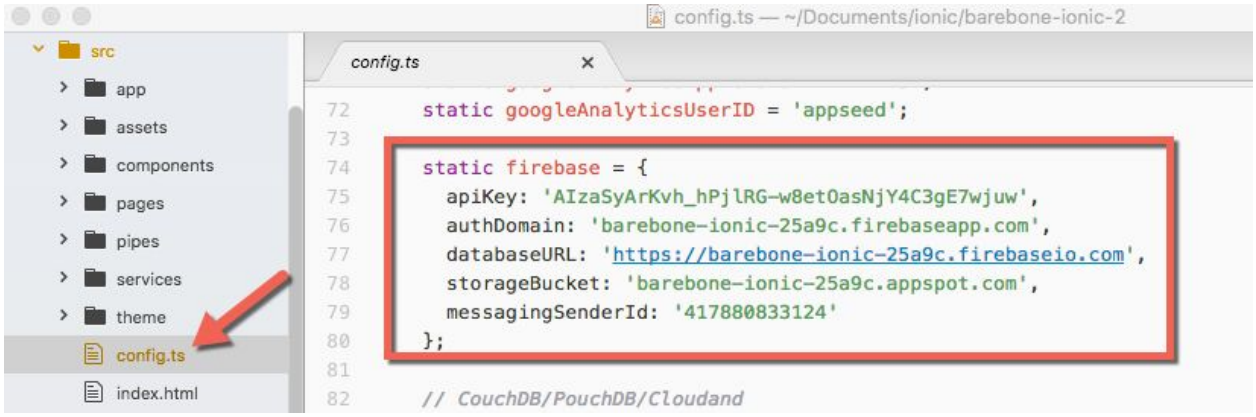
In order to allow the app to read all the data you have created and write/update data in the database, you should set the permissions by navigating to the “Rules” tab and pasting the following rules:

```
{
    "rules": {
        ".read": "true",
        ".write": "true"
     }
}
```

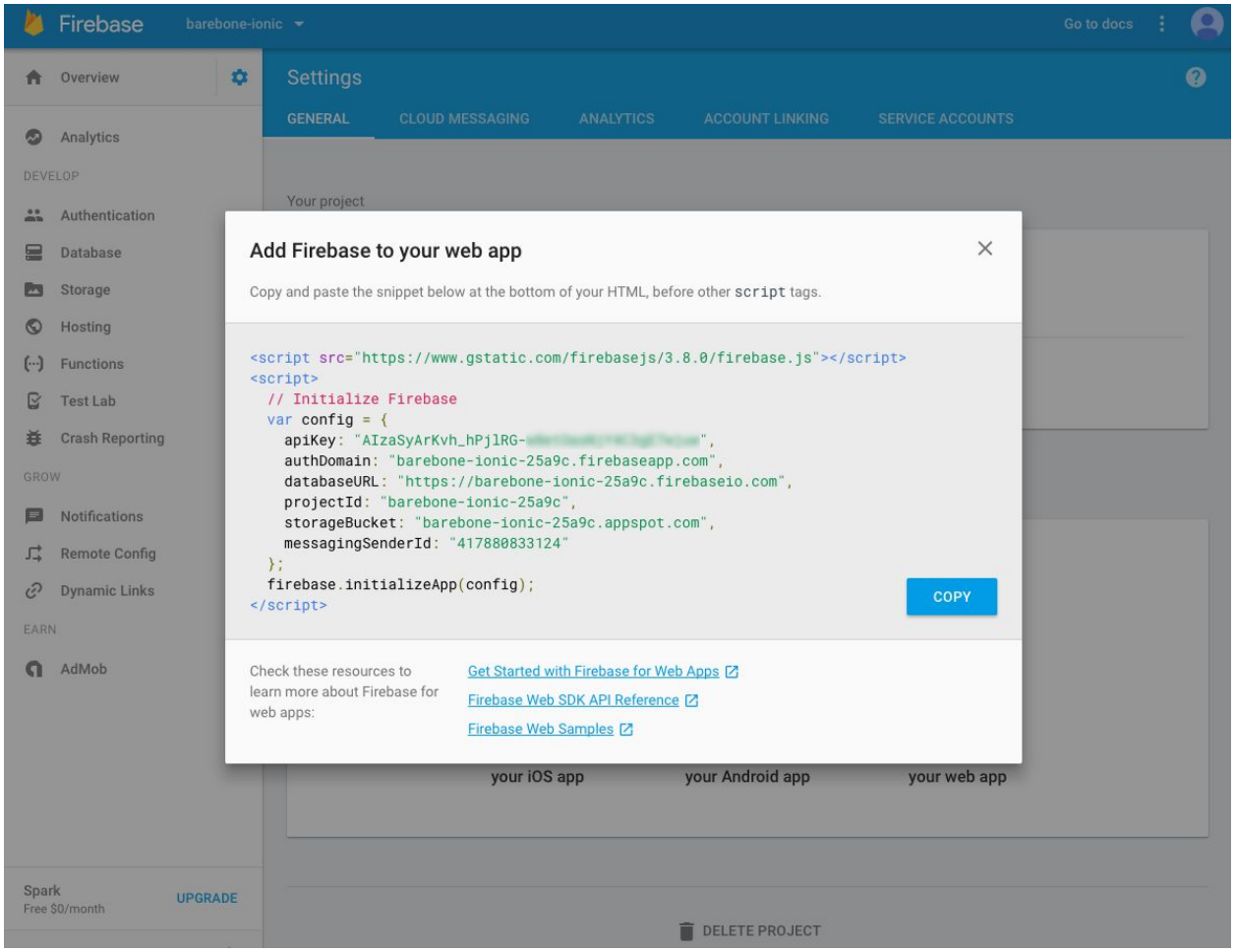


Configuration in app

To set the Firebase project the app will work with, you should set the `firebase` variable in `config.ts` located in the `src` folder:

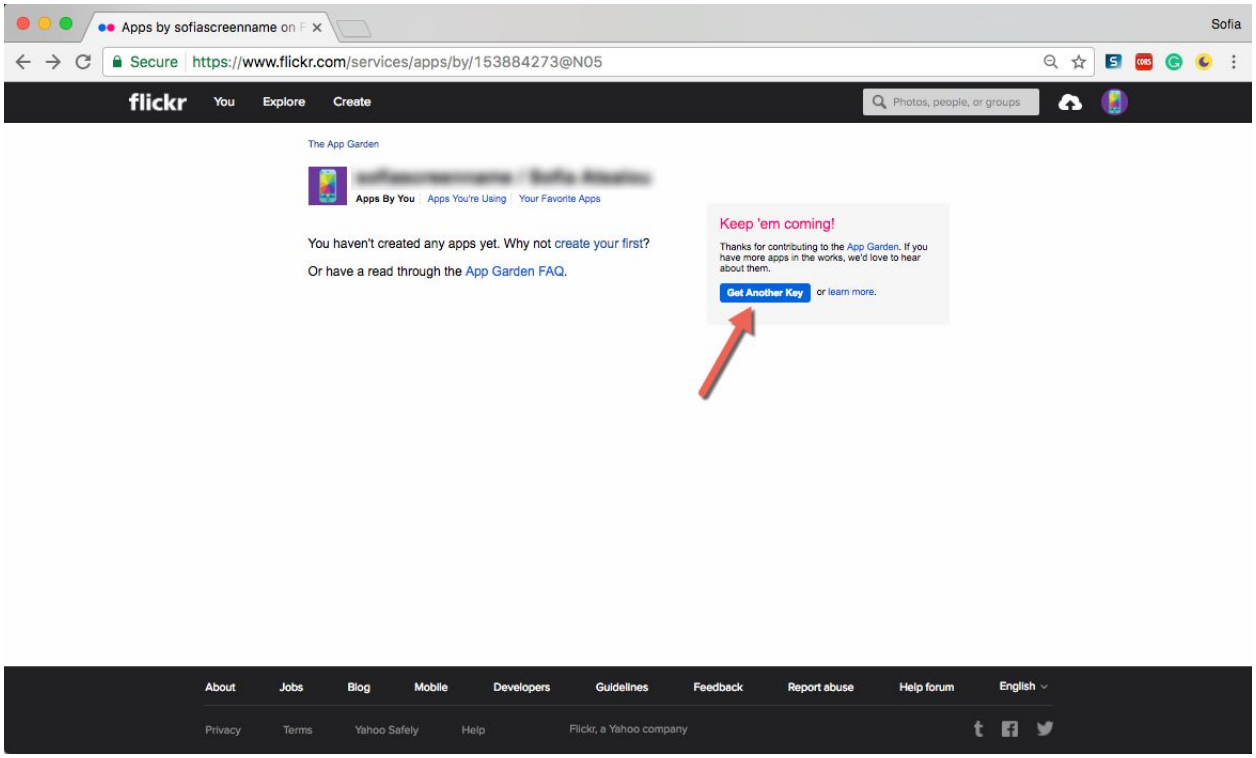


To find these information of your Firebase project, hit the settings icon on the side menu and then, “Project settings” from the expanded menu. Finally, click on “Add Firebase to your web app”.

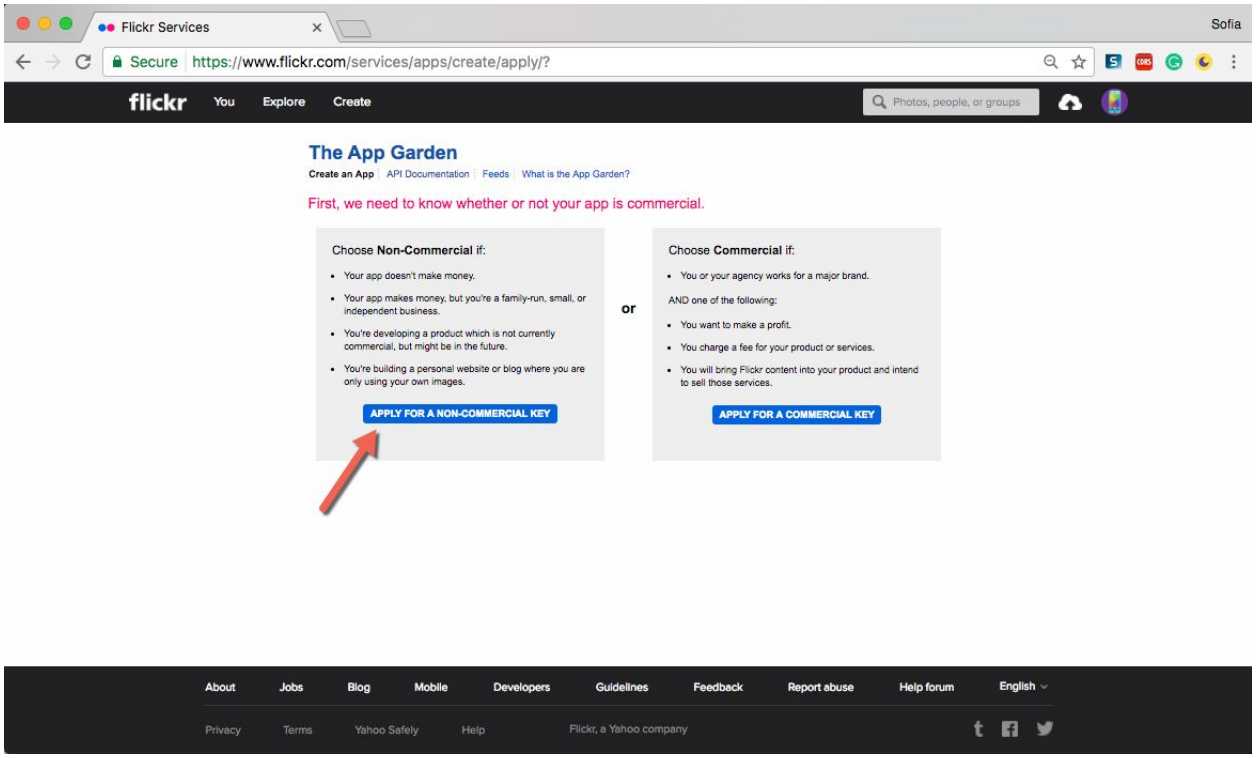


Flickr

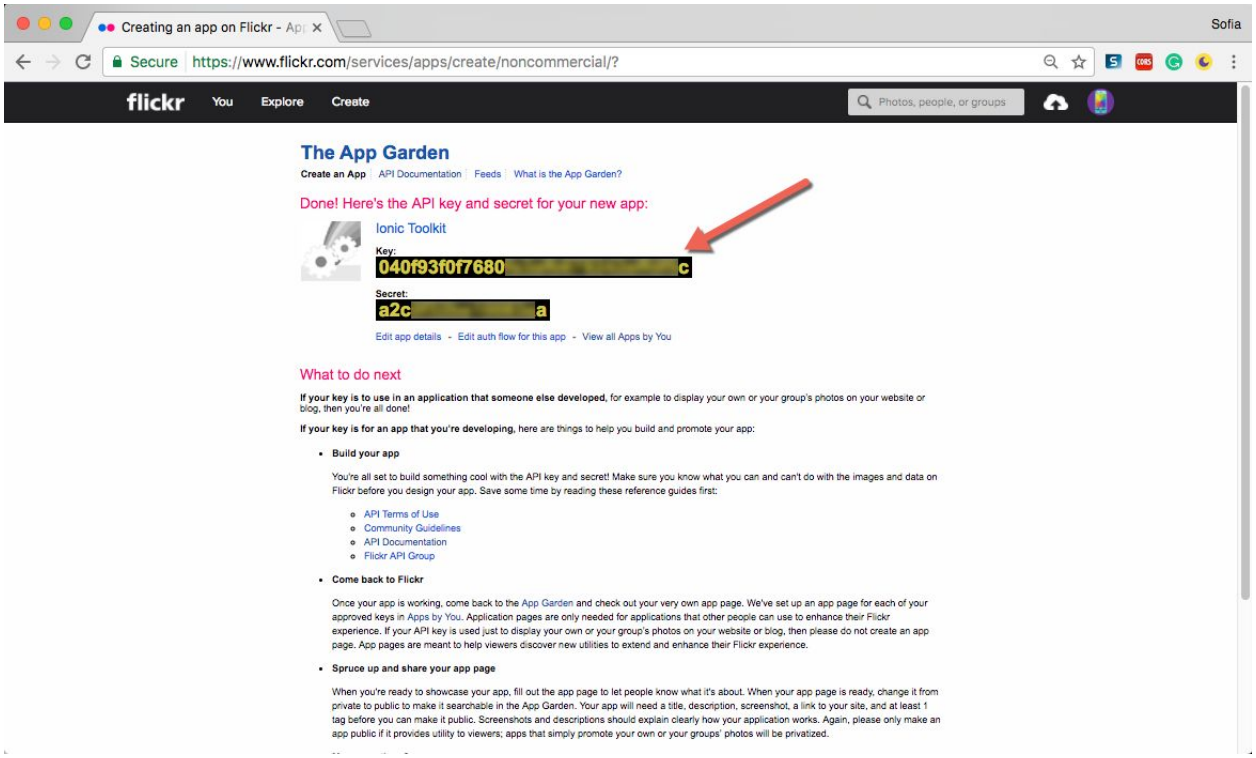
Firstly, you will need to create a Flickr API key. For this, you need to have a Yahoo account. To sign in to your Yahoo account or create a new one, visit https://www.yahoo.com/. Then, select “API Keys” at https://www.flickr.com/services/api/. Hit the “Get ANother Key” button:



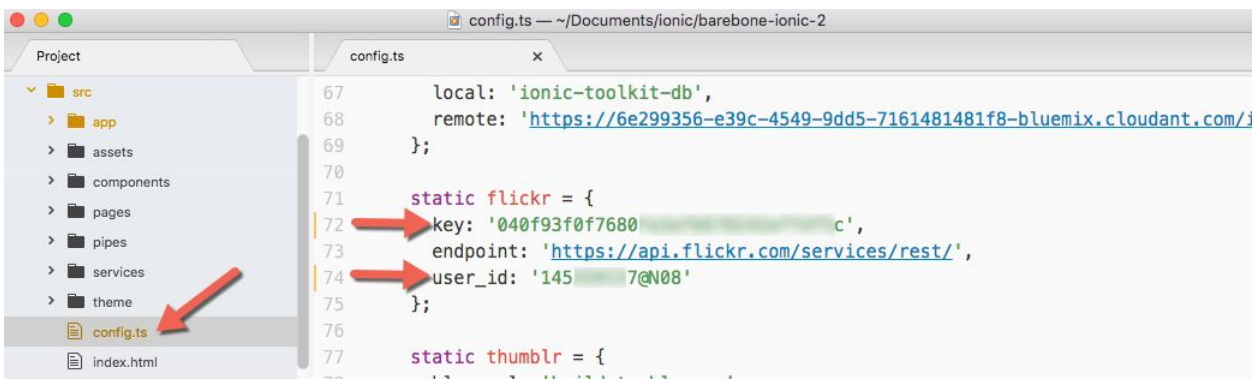
You can choose to request a commercial or a non-commercial key upon your requirements. In this guide, we are going to choose the non-commercial key:



Next, fill in the form and press “Submit”. Make a note of the key that will be shown:



Finally, open the `config.ts` file located in the `src` folder and set the **key** and **user_id** variables.



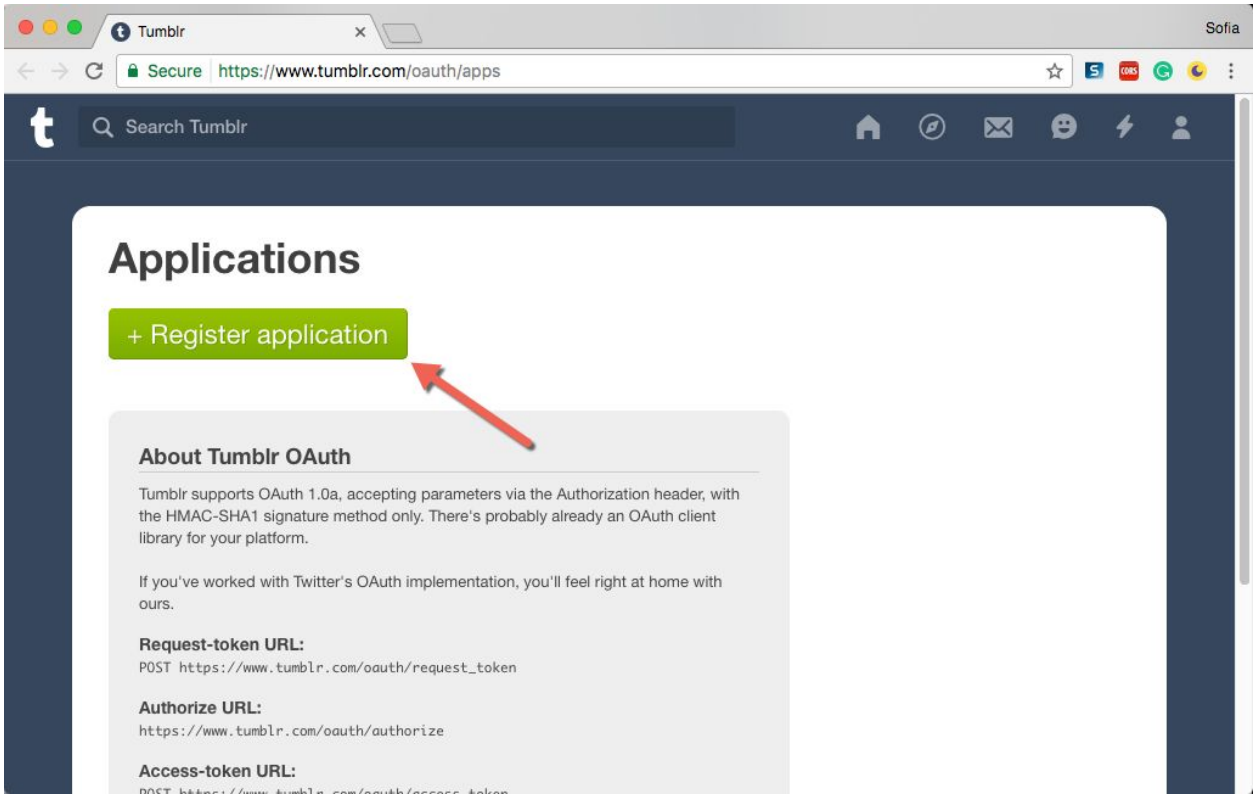
The **key** is your API key and the **user_id** is the user whose photos are fetched into the app.

Note: You can find the user_id by the URL pointing to the user’s photos. For example, the user_id extracted from https://www.flickr.com/photos/126545133@N08/ would be *126545133@N08*.

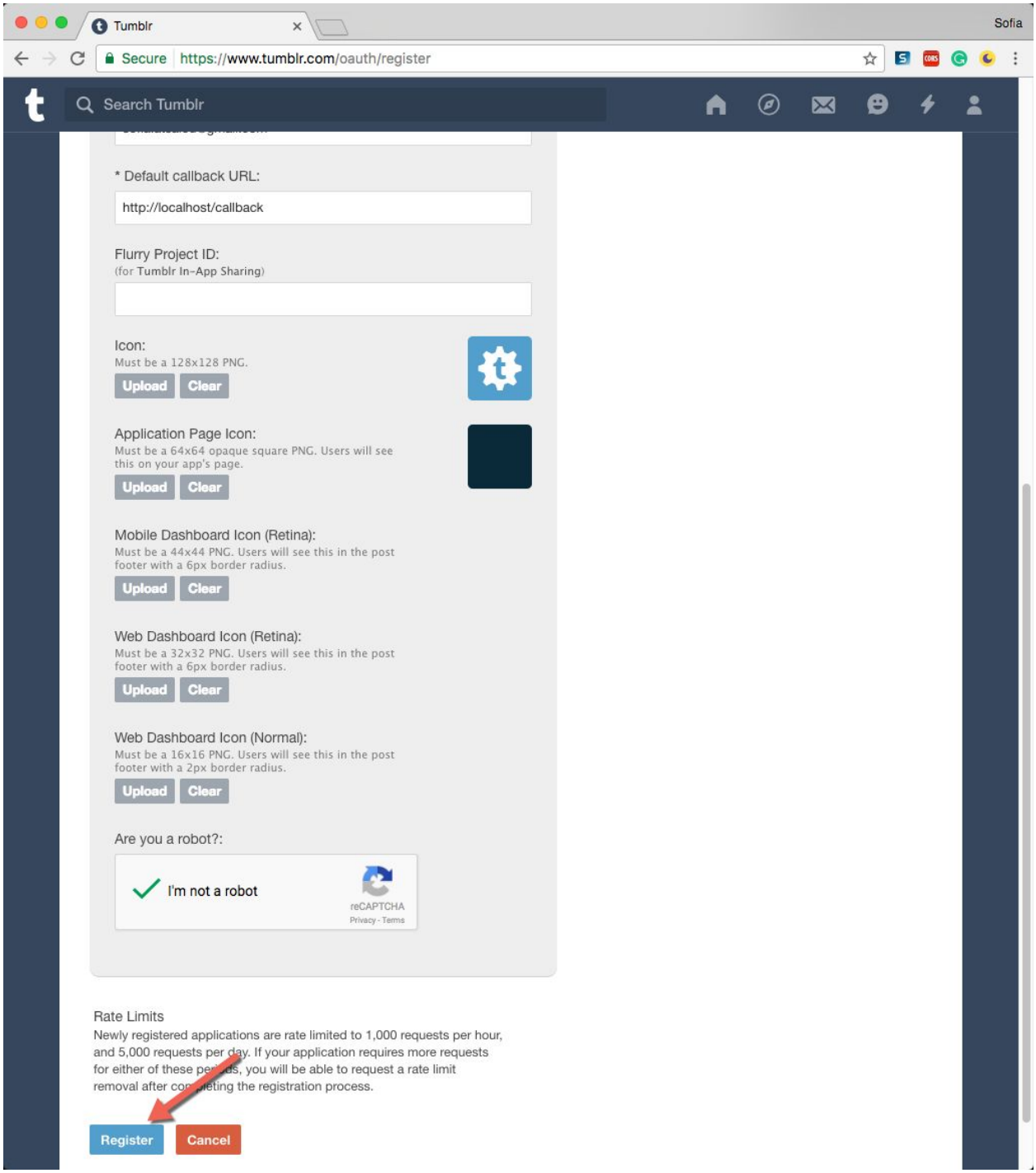
Tumblr

This feature fetches the posts of a specific Tumblr blog. You will need a Tumblr account in order to register an application and get an API key. Login to your Tumblr account at

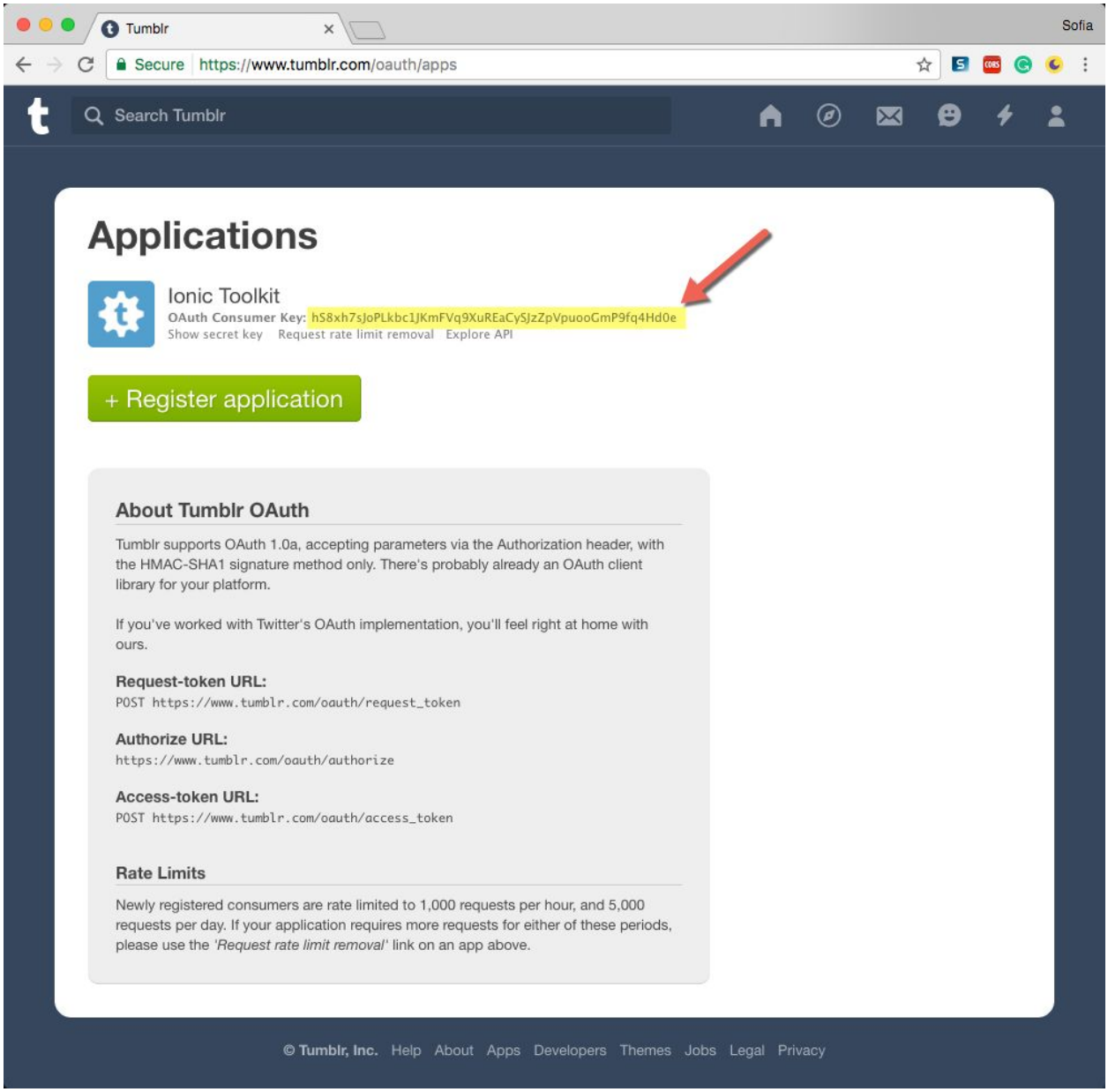
https://www.tumblr.com/login or create an account at https://www.tumblr.com/register. Then, register your application by visiting https://www.tumblr.com/oauth/apps and hitting the “Register application” button:



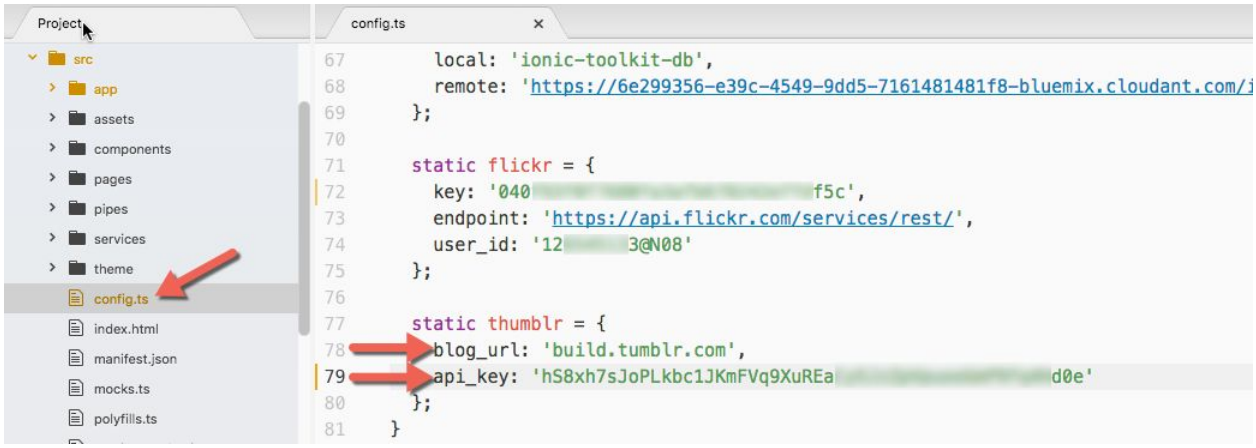
Fill in the registration form and press “Register”:



Now, you are ready to make a note of your OAuth Consumer Key and use it in the app.



Open the `config.ts` file located in the `src` folder and set the **blog_url** and **api_key** variables as shown:



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.
- AppSeed Support Centre
  Submit your question to our Support Centre mentioning the product name.
- Quick Start Guide
  The online version of this document.
- appseed.io
  Our official website.

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 us to our Support Centre.

--- The Appseed team. appseed.io