

Prerequisites:

- A stable internet connection.
- Windows 7 64bit SP1 or Higher

Local Installation:

1. The machine must install XAMPP with PHP version 7.1 >=
2. The machine must also install Composer (<https://getcomposer.org/download/>), Please refer to this link about the installation (<http://webdevzoom.com/how-to-install-composer-on-windows/>)
3. In your terminal, type ***composer global require "laravel/installer"***.
4. After installation, place the project at htdocs folder in your XAMPP.
5. Copy the .env.example and rename it .env
6. Change the following .env settings into.

```
16 BROADCAST_DRIVER=pusher
17 CACHE_DRIVER=database
18 SESSION_DRIVER=file
19 SESSION_LIFETIME=120
20 QUEUE_DRIVER=database
21
```

7. Change this credentials by registering your API Keys.

```
DARKSKY_API_KEY= your key
MAIL_ENCRYPTION=null
MAIL_DRIVER=smtp
MAIL_HOST=smtp.sendgrid.net
MAIL_PORT=587
MAIL_USERNAME= your_sendgrid_username
MAIL_PASSWORD= your_sendgrid_password
MAIL_FROM_ADDRESS=your_custom_email_address
PUSHER_APP_ID=your_pusher_app_id
PUSHER_APP_KEY=your_pusher_app_key
PUSHER_APP_SECRET=your_pusher_app_secret
PUSHER_APP_CLUSTER=your_pusher_app_cluster
MAIL_FROM_ADDRESS=developers@pdrmo-bulacan.me
MAIL_FROM_NAME="Bulacan ULAT Dev Team"
```

8. After registering the credentials, create a database in phpMyAdmin, name it 'weathers'.
9. After creating the database, open the terminal in your project

```
D:\xampp\htdocs\to give>
```

And type the following:

- *composer update*
- *php artisan key:generate*
- *php artisan migrate:fresh && php artisan db:seed*
- *php artisan storage:link*

10. To load the frontend components, download Node.js and install it. (Tip: preferably Node.js 64bit for best results)

11. Install ruby. (<https://www.ruby-lang.org/en/>)

12. After installed the node.js and ruby, go to step 9, and type following:

- *npm install -g pngquant-bin*
- *npm install*

13. After the installing of dependencies, goto `node_modules/vue-radial-progress/` and change the `.babelrc` config .

```
{
  "presets": ["env"],
  "plugins": ["transform-runtime"],
  "comments": false
}
```

14. Goto step 9 and type *npm run watch* to compile the codes, *php artisan serve* to boot up the server, and *php artisan queue:listen* to run the queue worker. (Multiple terminals will do.)

15. Success! You can now access the project!

Setting up Darksky API key:

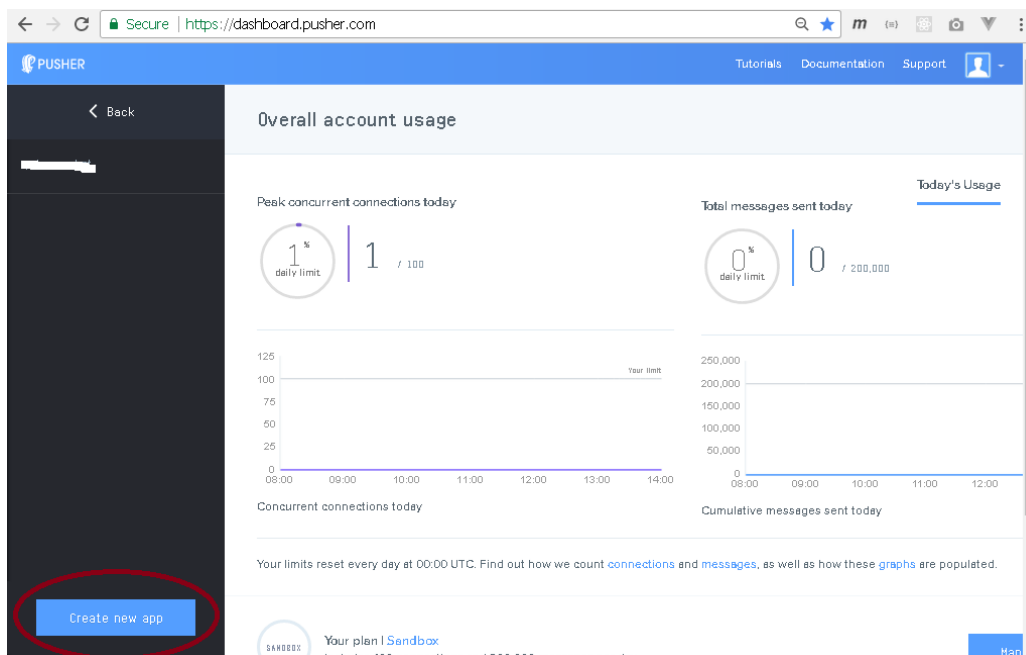
1. Goto Darksky website (<https://darksky.net/dev>) and sign up.
2. After a successful registration, goto (<https://darksky.net/dev/account>) to get your API key.
3. Place the API key in the following files:
 - .env
 - App\Http\Controllers\WeatherNotificationsController.php
 - Resources\assets\js\components\admin\Dashboard.vue, find the apikeys object and darksky and replace it.

Setting up the World Weather Online API key:

1. Goto (<https://developer.worldweatheronline.com/api/>) and register an account.
2. After a successful regtraton, goto (<https://developer.worldweatheronline.com/my/>) to get the api key.
3. Place the API key in the following files:
 - Resources\assets\js\components\admin\Tides.vue, find the apikeys object and wwo and replace it.
 - Resources\assets\js\components\admin\notifications\TidesNotifications.vue, find the apikeys object and wwo and replace it.
 - Resources\assets\js\app.js, find the apikeys object and wwo and replace it.

Setting up Pusher API Key

1. Goto Pusher (<https://pusher.com/>) and register for an account. After a successful registration, click your apps and create new app.



2. Refer to this example on configuring the app. And click create my app.

The screenshot shows the 'Create your app' form in the Pusher dashboard. The form is titled 'Create your app' and includes a close button (X) in the top right corner. Below the title, it says 'Create an app to generate your unique credentials. You can create as many apps as you need.'

The form is divided into several sections:

- 1 Name your app**: A text input field containing 'your_app_name'.
- 2 Select a cluster**: A dropdown menu showing 'ap1 (Asia Pacific (Singapore))'.
- ☐ Create apps for multiple environments?
- Customize your getting started guide (optional).**
- 3 What's your front-end tech?**: A row of icons for various front-end technologies: Android, Angular, Apple, React Native, JavaScript, Vue.js, and Svelte.
- 4 What's your back-end tech?**: A row of icons for various back-end technologies: Node.js, Django, Ruby on Rails, PHP, Python, Java, and others.
- 5 Tell us about your app (optional)**: A text input field with the placeholder text 'I am building...'. Below this field is a blue button labeled 'Create my app'.

3. Click the Appkeys Tab and get your credentials.

The screenshot shows the Pusher dashboard for an app named 'your_app_name' in the 'API (Asia Pacific (Singapore))' cluster. The 'App Keys' tab is selected and circled in red. The dashboard includes a sidebar with navigation links: Dashboard, Your apps, Account stats, and Billing. The main content area is divided into two columns:

- 1 Add this to your client**: A section with links for JavaScript, iOS (Obj-C), iOS (Swift), Android (Java), and React Native. Below these links is a green box with a checkmark and the text 'This page includes the JavaScript below. Connection state: Connected'. Below this is a code block for an HTML document with the following content:

```
<!DOCTYPE html>
<html>
<head>
<title>Pusher Test</title>
<script src="https://js.pusher.com/4.1/pusher.min.js">
```
- 2 Add this to your server**: A section with links for Curl, Ruby, Rails, Node.js, ASP.NET MVC, Python, PHP, Go, and Java. Below these links is a section titled 'Install via RubyGems:' with a code block containing the command `$ gem install pusher` and a 'Copy' button. Below this is a section titled 'Import the Pusher package and trigger your first event:' with a code block containing the command `require 'pusher'`.

4. Success you have your own api key!

your_app_name API (ASIA PACIFIC (SINGAPORE))

Overview Getting Started **App Keys** Stats Debug Console Error Logs Webhooks Push Notifications **BETA** Collaborators

If a token becomes compromised, you can create a new key/secret pair. You will be able to delete the old token when you've updated your app.

Date created	Credentials
Created 3 minutes ago	<pre>app_id = [REDACTED] key = [REDACTED] secret = [REDACTED] cluster = [REDACTED]</pre>

5. Change the project credentials in the following:

- .env
- Resources\assets\js\app.js, find the window.Echo object and replace it.
- Resources\assets\js\rescue.js, find the window.Echo object and replace it.
- Resources\assets\js\console.js, find the window.Echo object and replace it.

Setting up Sendgrid API Key.

1. Register an account at Sendgrid(<https://sendgrid.com/>) , after a successful registration, Place you credentials at .env

Setting up Google Maps Api Key.

1. Register an account in Google. (<https://mail.google.com/>) , after a successful registration. Go to (<https://developers.google.com/maps/documentation/javascript/tutorial>) and click get a key.

Web > Maps JavaScript API

GET A KEY VIEW PRICING AND PLANS

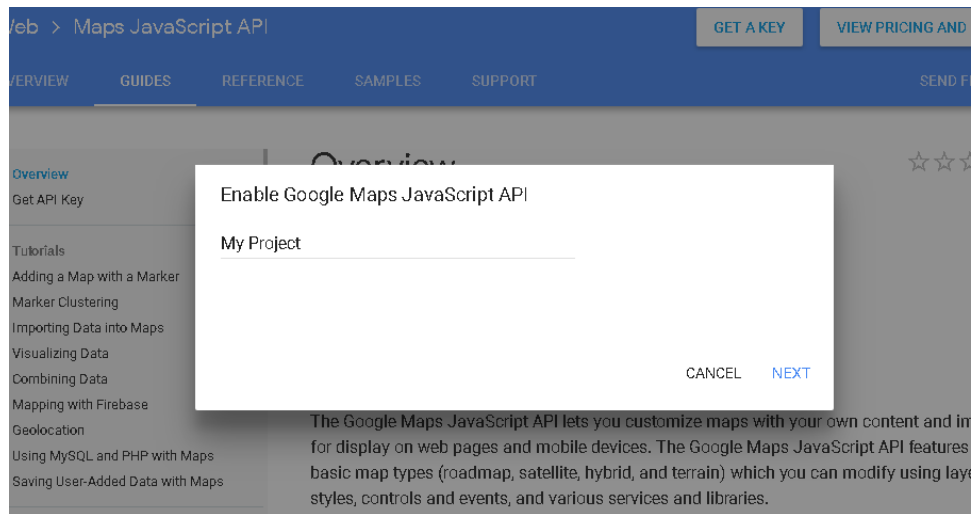
OVERVIEW GUIDES REFERENCE SAMPLES SUPPORT SEND FEEDBACK

Overview

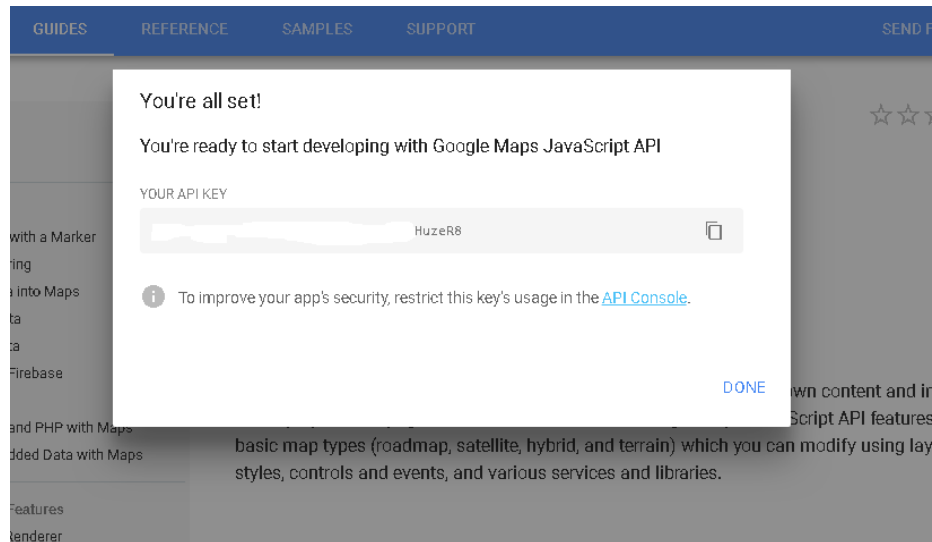
Overview
Get API Key
Tutorials

Contents Audience

2. Change my project in to your project name and click next.



3. Copy the api key and replace the credentials in the following:



- Resources\assets\js\app.js, find the `Vue.use(VueGoogleMaps,object` and replace it.
- Resources\assets\js\components\admin\Dashboard.vue, find the `apikeys` object and google and replace it.
- Resources\assets\js\rescue.js, find the `Vue.use(VueGoogleMaps,object` and replace it.

Setting up SMS Gateway ME

1. Register at (<https://msgateway.me/>) to get an account.
2. Download the SMS Gateway ME App via Google Play and install.
3. After installing the application in your mobile phone, login your credentials in the application.

Note:

- It will requires an internet connection to use the application.

4. After login your account in the app, get the device id of your app.
5. Copy your email, password and device id and replace in App\Jobs\SendSMSJob.php

Note:

- Make sure that you have sufficient load and the server is online in able to send a message.
- Make sure that your worker is running at the terminal.

Deploying the application in VPS.

1. Get your own VPS in able to deploy the project. You may checkout in this link to decide on your VPS. (<https://www.whoishostingthis.com/compare/laravel/>)

Note:

- For best results, you may choose Ubuntu as your operating system, there a lot of references in deploying a Laravel application using this operating system.
2. Buy also a domain in your server.
 3. Configure your domain according to its settings.
 4. After configure your domain, refer to this link (<https://websiteforstudents.com/install-laravel-on-ubuntu-17-04-17-10-with-apache2-mariadb-and-php-support/>) in deploying the application except
 - Step 3: Type `sudo apt install php libapache2-mod-php libapache2-mod-php php-common php-mbstring php-xmlrpc php-soap php-gd php-xml php-mysql php-cli php-mcrypt php-zip php-curl`. Then continue the step 3.
 - Step 4: After `cd /var/www/html/`, clone the repository at github by typing `git clone some_repo@github.com.git`. Then type `sudo composer global require "laravel/installer"`.
 - Before Step 5: Create a database in MariaDB by typing `mysql -u root -p mysql` or `mysql -u root -p`.

```
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 12
Server version: 5.5.40-MariaDB-log FreeBSD Ports

Copyright (c) 2000, 2014, Oracle, Monty Program Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input

MariaDB [mysql]>
```

- After login, type `CREATE DATABASE weathers;`.
- After the creating the database, create a user in MariaDB, by typing `CREATE USER 'pdrmo-bulacan'@'localhost' IDENTIFIED BY '';`.
- Type also `GRANT ALL PRIVILEGES ON weathers.* To 'pdrmo-bulacan'@'localhost' IDENTIFIED BY '';`
- Lastly type `FLUSH PRIVILEGES;` and type `exit`.
- Refer to the Local Installation Guide, Steps 5 to 13 in installing Laravel.

- Then type the following
 - ***sudo chown -R www-data:www-data /var/www/html/your_project_name/***
 - ***sudo chmod -R 755 /var/www/html/your_project_name/***
 - Note: Check if the directories ***rescuers*** and ***reports*** are in ***storage/app/public***, if not, create a directory named ***rescuers*** and ***reports*** inside of ***storage/app/public***.
 - ***sudo chmod -R 777 /var/www/html/your_project_name/storage/***
 - ***sudo chmod -R 777 /var/www/html/your_project_name/storage/app/public/rescuers***
 - ***sudo chmod -R 777 /var/www/html/your_project_name/storage/app/public/reports***
 - Continue the steps 5 in the link up to the last step.
- 5. In able to use the Geolocation function of the application, install a SSL Certificate in the server.
- 6. Install your queue worker in your server by typing ***sudo apt-get install supervisor***.
 - After installing supervisor on our server we need to go to conf.d directory, type ***cd /etc/supervisor/conf.d***
 - Now in this directory lets create a file say queue-worker.conf

```
sudo vim /etc/supervisor/conf.d/queue-worker.conf

[program:queue-worker]
process_name=%(program_name)s_%(process_num)02d
command=php /var/www/html/your_project/artisan queue:listen
autostart=true
autorestart=true
user=root
numprocs=8
redirect_stderr=true
stdout_logfile=/var/www/html/worker.log
```

- Start reading the config file by typing ***sudo supervisorctl reread***.
 - Restart the supervisor by typing ***sudo supervisorctl update***.
 - Start the worker by typing ***sudo supervisorctl start queue-worker:****
7. The Default user was, ***pdrmmobulacan@gmail.com*** and the password was ***sample123***.

Prepared by:

Rannie Ollit
Software Developer
einnar82@gmail.com