# Oplao Weather Map Project

## Preface

The project consists of the following three parts

1. Frontend (web client) based on Angular technology.
2. Backend based on Python technology
3. Content generator based on Python technology

Each of the parts works as a detached application.

The frontend requests some information and content from the backend.

The backend provides the information via an entry point

The content generator generates images and data (json files) and puts ones in the place where the frontend fetches them from.

The project resides on <https://bitbucket.org/Oplao/weathermap/src/master/>.

Please clone it locally.

## Frontend

The frontend is developed with Angular 4 technology.

### To build a production distributive, please follow

The source code is located in

<cloned repo>/frontend/

1. Install Node js and npm using the following link <https://nodejs.org/en/download/>
2. Install Angular CLI globally using the following command

npm install –g @angular/cli

1. Go to the folder <cloned repo>/frontend/
2. Build the application by the command

ng build –prod –aot=false

1. The distributable files are in folder dist

Note. Before building you need to set up the url to the backend entry point. Please go to frontend/src/app/utils/develop.utils.ts and update variable HOST. For example,

*export class DevelopUtil {*

*private static HOST = 'http://35.173.37.26:8888/';*

### Development on Windows 7/10

Please follow

1. Download and install Visual Studio Code using the following link <https://code.visualstudio.com/>
2. After installing Visual Studio Code, open the IDE and choose the Extensions menu option located on the left side of the window. Search and choose TSLint for Visual Studio Code >= 1.0.24 (egamma). The extension by default enables standards for code formatting, naming conventions and more for typescript development.
3. To enable debugging within Visual Studio Code, install the Debugger for Chrome >= 3.5.0 (Microsoft) extension. Remember to close and open Visual Studio Code after installation. Replace the contents of the launch.json file with the snippet below.

|  |
| --- |
| {  "version": "0.2.0", "configurations": [ { "name": "Launch Chrome against localhost", "type": "chrome", "request": "launch", "url": "http://localhost:4200", "webRoot": "${workspaceRoot}" } ] } |

1. To start debugging, navigate to <cloned repo>/frontend/ and run the following command

npm start

Open the debug perspective in Visual Studio Code and press the green arrow located in the top left hand corner of the window. The debugger will attach itself to the running process and you should now be able to start debugging your Angular application.

## Backend

The backend is a python script being launched as a REST service

The source code is located in

<cloned repo>/api/

To run the backend you need

1. Python 2.7 <https://www.python.org/downloads/>
2. Utility pip if you run the system on Linux <https://pip.pypa.io/en/latest/installing/>

Note. pip is already installed if you are using Python 2 >=2.7.9 downloaded from [python.org](https://www.python.org/)

After installation of Python, you should add the following libraries:

1. Flask

Please run

python.exe -m pip install flask (on Windows) or pip install flask (on Linux)

1. Flask\_cors

Please run

python.exe -m pip install flask-cors (on Windows) or pip install flask-cors (on Linux)

1. Flask MySQL connector

Please run

python.exe -m pip install flask-mysql (on Windows) or pip install flask-mysql (on Linux)

Then you should set up the properties to connect to the database. Please go to “api\DB.py” and change the parameters according to your environment

MYSQL\_USER = <your user>  
MYSQL\_PASSWORD = <your password>  
MYSQL\_DB = <your database>  
MYSQL\_HOST = <your host>

Default port is 3306

Weather.py is the start script.

## Generator

The generator is a python script being launched as a standalone application

The source code is located in

<cloned repo>/generator/

1. To run the backend you need
   1. Python 2.7

<https://www.python.org/downloads/>

* 1. Library PIL

<http://www.pythonware.com/products/pil/> (for windows the libs are of 32 bit architecture)

*Note. Both Python and PIL should be of 64 bit architecture.*

*There is 64 bit PIL for Windows here*

[*https://github.com/lightkeeper/lswindows-lib/blob/master/amd64/python/PIL-1.1.7.win-amd64-py2.7.exe*](https://github.com/lightkeeper/lswindows-lib/blob/master/amd64/python/PIL-1.1.7.win-amd64-py2.7.exe)

*Note. However If you are installing PIL 32x on Windows, before you should do the following in your windows registry*

*Copy all the registry keys from:*

*HKEY\_LOCAL\_MACHINE\SOFTWARE\Python*

*To*

HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Python

* 1. MySQL connectors

<https://dev.mysql.com/downloads/connector/python/>

* 1. MySQL database of version 5.6

<https://dev.mysql.com/downloads/mysql/5.6.html#downloads>

1. Setting up environment
   1. Create the following table in a MySQL database

CREATE TABLE `images` (

`path` varchar(64) DEFAULT NULL,

`date` varchar(20) DEFAULT NULL,

`type` varchar(20) DEFAULT NULL

)

* 1. Change the database properties according to your settings in “generator/DB.py”

MYSQL\_USER = <your user>  
MYSQL\_PASSWORD = <your password>  
MYSQL\_DB = <your database>  
MYSQL\_HOST = <your host>

Default port is 3306

* 1. Create the following folders

generator/data

generator/images

generator/temp

generator/jsons/parser

* 1. Create the following empty file

generator/data/wwo-data.csv.gz

* 1. Change the following paths in “generator/main.py” according to your environment

filename = '/home/ubuntu/maps\_app/generator/data/weather.csv'  
zip\_filename = '/home/ubuntu/maps\_app/generator/data/wwo-data.csv.gz'

1. Run Generator

Run the generator by launching

generator/main.py

## Python Development

1. You can use IDE IDEA PyCharm for developing python scripts

<https://www.jetbrains.com/pycharm/>

1. On development stage you can publish the generated content with the script api/content\_access.py

Please uncomment the following lines on the page api/weather.py in order to make the content accessible via the backend endpoint

#for development only  
from content\_access import get\_image  
from content\_access import get\_wind  
app.add\_url\_rule( '/images/<image>', 'get\_image', get\_image, methods=['GET'])  
app.add\_url\_rule( '/images/wind/<wind>', 'get\_wind', get\_wind, methods=['GET'])  
#end of for development