Jorl17 Correct documentation and add missing Dockerfile dependencies

6d59908 on 14 Sep 2017

1 contributor

Hosting your own Open-Elevation instance

You can freely host your own instance of Open-Elevation. There are two main options: Docker or native. We recommend using docker to ensure that your environment matches the development environment

Clone the repository

First things first, clone the repository and `cd` onto its directory

```
git clone http://github.com/Jorl17/open-elevation
cd open-elevation
```

Using Docker

An image of Open-Elevation is available at DockerHub. You can use this image as the basis for your Open-Elevation installation.

The Docker image roots itself at `/code/` and expects that all GeoTIFF datafiles be located at `/code/data/`, which you should mount using a volume.

Prerequisites: Getting the dataset

Open-Elevation doesn't come with any data of its own, but it offers a set of scripts to get the whole SRTM 250m dataset.

Whole World

If you wish to host the whole world, just run

```
mkdir data # Create the target folder for the dataset
docker run -t -i -v $(pwd)/data:/code/data openelevation/open-elevation
/code/create-dataset.sh
```

The above command should have downloaded the entire SRTM dataset and split it into multiple smaller files in the `data` directory. **Be aware that this directory may be over 20 GB in size after the process is completed!**

Custom Data

If you don't want to use the whole world, you can provide your own dataset in GeoTIFF format, compatible with the SRTM dataset. Simply drop the files for the regions you desire in the `data` directory. You are advised to split these files in smaller chunks so as to make Open-Elevation less memory-hungry (the largest file has to fit in memory). The `create-tiles.sh` is capable of doing this, and you can see it working in `create-dataset.sh`. Since you are using docker, you should always run the commands within the container. For example:

```
docker run -t -i -v $(pwd)/data:/code/data openelevation/open-elevation
/code/create-tiles.sh  /code/data/SRTM_NE_250m.tif 10 10
```

The above example command splits `SRTM_NE_250m.tif` into 10 by 10 files inside the `/code/data` directory, which is mapped to `$(pwd)/data`.

Running the Server

Now that you've got your data, you're ready to run Open-Elevation! Simply run

```
docker run -t -i -v $(pwd)/data:/code/data -p 8080:8080
openelevation/open-elevation
```

This command:

1. Maps `$(pwd)/data` (your data directory) to `/code/data` within the container
2. Exposes port 8080 to forward to the container's port 8080
3. Runs the default command, which is the server at port 8080

You should now be able to go to `https://localhost:8080` for all your open-route needs.

Without Docker

Installing Dependencies

In order for Open-Elevation to work, you need `GDAL` and `libspatialindex`. For the full process to work you also need a version of `unrar`.

The setup for `gdal` depends on the distro and may even change among distro versions, thus being outside the scope of this documentation. Please follow the documentation found in GDAL's homepage.

The following are instructions for Ubuntu/Debian compatible distros, and similar ones might apply to your particular setup. Again, make sure you also install GDAL. Please consider using a `virtualenv` instead of using the default python installation for the following commands.

```
apt-get update -y
apt-get install -y libspatialindex-dev unrar-free bc
pip install -r requirements.txt
```

If all goes well, you now have the required dependencies to run Open-Elevation.

Prerequisites: Getting the dataset

Open-Elevation doesn't come with any data of its own, but it offers a set of scripts to get the whole SRTM 250m dataset.

Whole World

If you wish to host the whole world, just run

```
./create-dataset.sh
```

Assuming you have `wget` and `unrar` installed, the above command should have downloaded the entire SRTM dataset and split it into multiple smaller files in a `data` directory. **Be aware that this directory may be over 20 GB in size after the process is completed!**

Custom Data

If you don't want to use the whole world, you can provide your own dataset in GeoTIFF format, compatible with the SRTM dataset. Simply drop the files for the regions you desire in the `data` directory. You are advised to split these files in smaller chunks so as to make Open-Elevation less memory-hungry (the largest file has to fit in memory). The `create-tiles.sh` is capable of doing this, and you can see it working in `create-dataset.sh`.

After this process, you should now have your dataset in the `tiles` subdirectory. You're good to go!

Running the server

Now that you've got your data, you're ready to run Open-Elevation! Simply run

```
python server.py
```

And you should be good to go!

Problems

Have you found any problems? Open an issue or submit your own pull request!