MAPS WITH DJANGO

PAOLO MELCHIORRE ~ @pauloxnet

+
−
Canada
Hudson Bay
Россия
Ísland
Sverige
Suomi Finland
United Kingdom
Беларусь
Қазақстан
Монгол
United States
Algérie
Niger
Tchad
South Sudan
Soomaaliya
République démocratique du Congo
Angola
Namibia
Mozambique
Perú
Bolivia
Brasil
Paraguay
Argentina
India
Papua
Australia
Drake
Leaflet | © OpenStreetMap contributors

→| Paolo Melchiorre

@pauloxnet

- CTO at **20tab**
- **Remote** worker
- Software **engineer**
- **Python** developer
- **Django** contributor

1.883 m
0 km
27 km
0 km
20 →|

→| Web maps features

- Static or Dynamic
- Interactive or view only
- Raster or Vector tiles
- Spatial databases
- Javascript library

→| Web mapping

"… *process* of using the **maps**

delivered by *Geographic Information Systems* (**GIS**)

on the **Internet** …"

— "Web mapping", *Wikipedia*

django

Requirements

```
$ python3 --version
Python 3.9.0+

$ python3 -m venv ~/.virtualenvs/mymap
$ . ~/.virtualenvs/mymap/bin/activate

$ python3 -m pip install django~=3.1
```

Creating the ‘mymap’ project

```
$ cd ~/projects
$ django-admin startproject mymap
mymap
├── manage.py
└── mymap
    ├── asgi.py
    ├── __init__.py
    ├── settings.py
    ├── urls.py
    └── wsgi.py
```

Creating the ‘markers’ app

```
$ cd mymap
$ django-admin startapp markers
markers
├── admin.py
├── apps.py
├── __init__.py
├── migrations
│   └── __init__.py
├── models.py
├── tests.py
└── views.py
```

Activating the ‘markers’ app

```
# mymap/settings.py

INSTALLED_APPS = [
    "django.contrib.admin",
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",
    "markers",
]
```

Adding a template view

```python
# markers/views.py

from django.views.generic import TemplateView


class MarkersMapView(TemplateView):
    template_name = "map.html"
```

Adding the ‘map’ template

```html
<!-- markers/templates/map.html -->

<!doctype html>
<html lang="en">
<head>
  <title>Markers Map</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
</body>
</html>
```

→| Adding ‘markers’ urls

```
# markers/urls.py

from django.urls import path

from markers.views import MarkersMapView

app_name = "markers"

urlpatterns = [
    path("map/", MarkersMapView.as_view()),
]
```

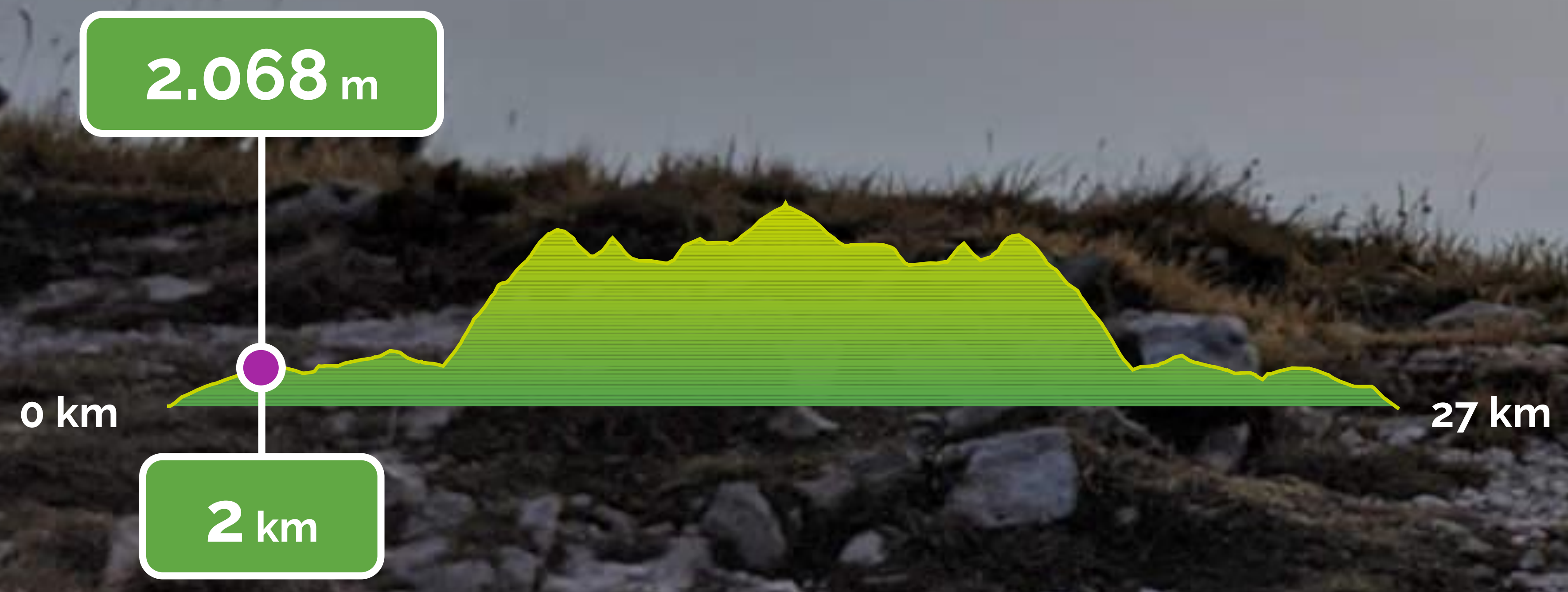
Updating 'mymap' urls

```python
# mymap/urls.py

from django.contrib import admin
from django.urls import include, path

urlpatterns = [
    path("admin/", admin.site.urls),
    path("markers/", include("markers.urls")),
]
```

Rifugio Pomilio 0.40
P
Bivacco Fusco 1.20
Monte Amaro 4.00
Guado di Coccia 8.00
Parco Nazionale della Majella


2.068 m
0 km
27 km
2 km

→| Leaflet

- JavaScript library for maps
- Free Software
- Desktop & Mobile friendly
- Light (~39 KB of gzipped JS)
- Well documented

Updating the ‘map’ template

```
<!-- markers/templates/map.html -->

{% load static %}
<!doctype html>
<html lang="en">
<head>
  <title>Markers Map</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" type="text/css" href="{% static 'map.css' %}">
  <link rel="stylesheet" type="text/css" href="//unpkg.com/leaflet/dist/leaflet.css">
  <script src="//unpkg.com/leaflet/dist/leaflet.js"></script>
</head>
<body>
  <div id="map"></div>
  <script src="{% static 'map.js' %}"></script>
</body>
</html>
```

➔| Adding the 'map' CSS

```
/* markers/static/map.css */

html,
body {
  height: 100%;
  margin: 0;
}
#map {
  height: 100%;
  width: 100%;
}
```

Adding the ‘map’ JavaScript

```
// markers/static/map.js

const copy = '© <a href="https://osm.org/copyright">OpenStreetMap</a> contributors'
const url = 'https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png'
const osm = L.tileLayer(url, { attribution: copy })
const map = L.map('map', { layers: [osm] })
map.fitWorld();
```

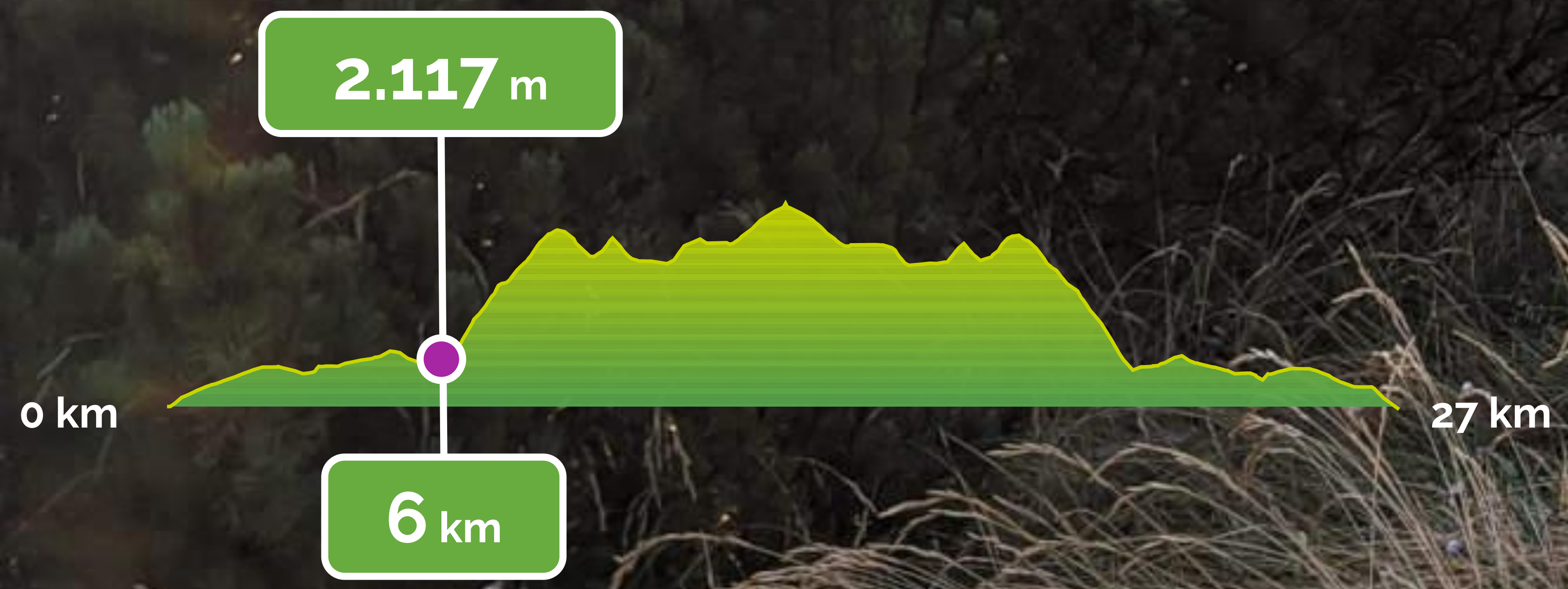
Show the empty web map

```
$ python3 manage.py runserver

$ python3 -m webbrowser -t localhost:8000/markers/map
```

+
−
Canada
United States
México
Colombia
Perú
Brasil
Bolivia
Paraguay
Argentina
Nunaat
Ísland
United Kingdom
France
España
Deutschland
Italia
Sverige
Suomi / Finland
Беларусь
Україна
România
Ελλάς
Türkiye
Россия
Қазақстан
O'zbekiston
Türkmenistan
Монгол улс
中国
日本
India
Việt Nam
Philippines
Malaysia
Indonesia
Papua Niugini
Australia
New Zealand / Aotearoa
Mali
Niger
Nigeria
Tchad تشاد
South Sudan
Soomaaliya الصومال
Kenya
Tanzania
République démocratique du Congo
Angola
Namibia
Moçambique
South Africa
Leaflet | © OpenStreetMap contributors

Bivacco Fusco 0.40
Monte Amaro 3.20
Guado di Coccia 7.20
Rifugio Pomilio 1.20
Monte Focalone 1.10
Parco Nazionale della Majella
ATTENZIONE!


2.117 m
0 km
27 km
6 km

GeoDjango

django.contrib.gis *(v1.0 ~2008)*

- Fields, backends, queries, admin, …
- Spatialite backend *(v1.1 ~2009)*
- Multiple backends *(v1.2 ~2010)*
- OpenLayers-based widgets *(v1.6 ~2013)*
- GeoJSON serializer *(v1.8 ~2015)*
- GeoIP2 Geolocation *(v1.9 ~2016)*

GDAL

- OSGeo library
- Free Software
- Read/Write geospatial data
- Raster/Vector formats
- Command line interface

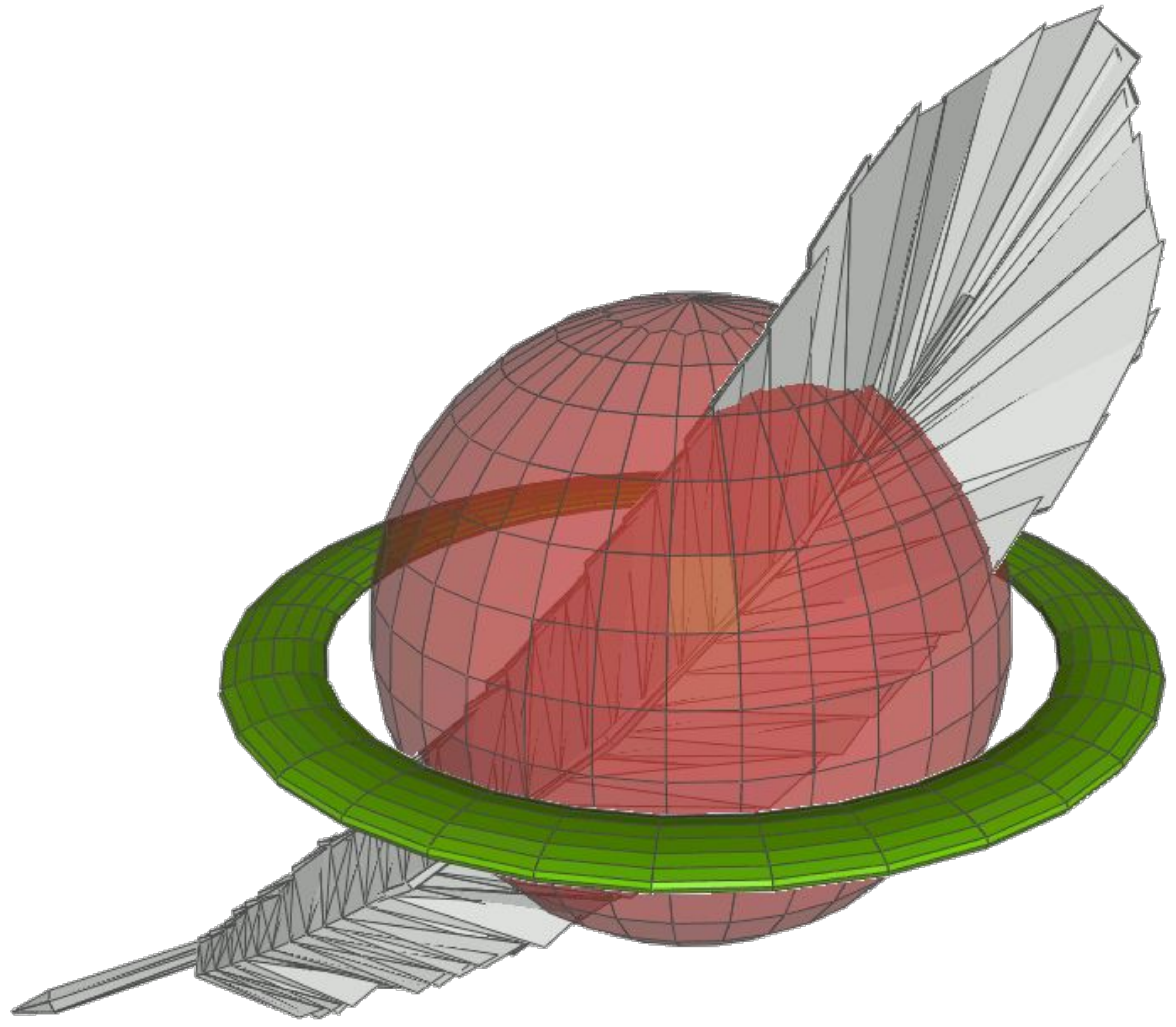
Installing GDAL

```
# apt-get install gdal-bin
#
# -- Read the docs for other operating systems.
```

Activating GeoDjango

```
# mymap/settings.py

INSTALLED_APPS = [
    "django.contrib.admin",
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",
    "django.contrib.gis",
    "markers",
]
```



→| SpatiaLite

- **SQLite** spatial extension
- Vector geodatabase functions
- Free Software
- Simple architecture
- Single file

Installing SpatiaLite

```
# apt-get install libsqlite3-mod-spatialite
#
# -- Read the docs for other operating systems.
```

→| Activating SpatiaLite

```
# mymap/settings.py

DATABASES = {
    "default": {
        "ENGINE": "django.contrib.gis.db.backends.spatialite",
        "NAME": BASE_DIR / "db.sqlite3",
    }
}
```

Adding the Marker model

```python
# markers/models.py

from django.contrib.gis.db import models


class Marker(models.Model):
    name = models.CharField(max_length=255)
    location = models.PointField()

    def __str__(self):
        return self.name
```

Adding the Marker admin

```
# markers/admin.py

from django.contrib.gis import admin

from markers.models import Marker


@admin.register(Marker)
class MarkerAdmin(admin.OSMGeoAdmin):
    list_display = ("name", "location")
```

Adding some markers

```
$ python3 manage.py makemigrations
$ python3 manage.py migrate

$ python3 manage.py createsuperuser

$ python3 manage.py runserver

$ python3 -m webbrowser -t localhost:8000/admin
```

Django administration

WELCOME, **PAULOX**. VIEW SITE / CHANGE PASSWORD / LOG OUT

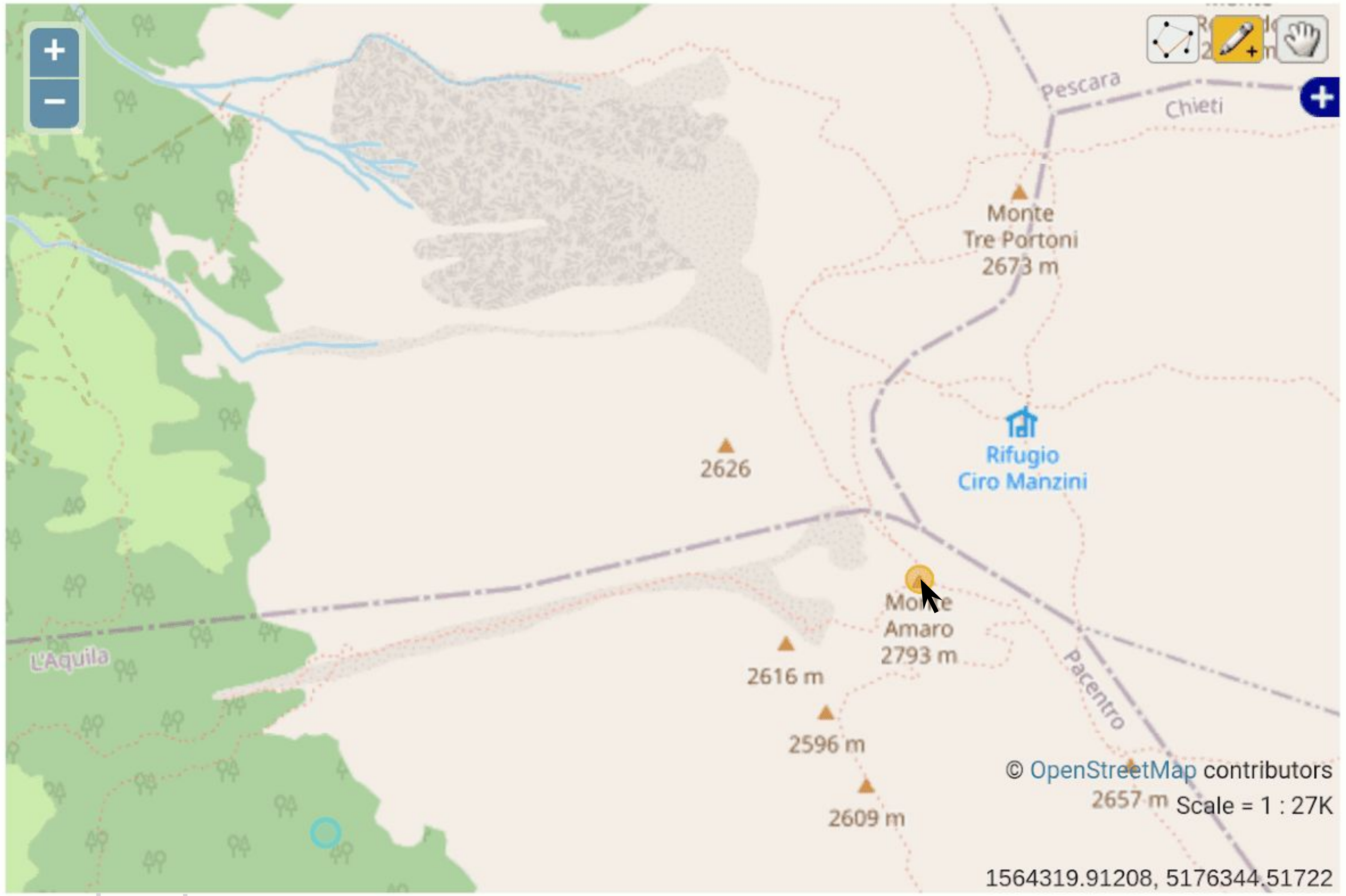
Home › Markers › Markers › Add marker

Add marker

Save and add another | Save and continue editing | SAVE

Name: Monte Amaro 2793m 🇮🇹

Location:



Updating the view

```python
# markers/views.py

import json
from django.core.serializers import serialize
from django.views.generic.base import TemplateView

from markers.models import Marker


class MarkersMapView(TemplateView):
    template_name = "map.html"

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        markers = Marker.objects.all()
        context["markers"] = json.loads(serialize("geojson", markers))
        return context
```

Inserting markers in the template

```
<!-- markers/templates/map.html -->
{% load static %}
<!doctype html>
<html lang="en">
<head>
  <title>Markers Map</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" type="text/css" href="{% static 'map.css' %}">
  <link rel="stylesheet" type="text/css" href="//unpkg.com/leaflet/dist/leaflet.css">
  <script src="//unpkg.com/leaflet/dist/leaflet.js"></script>
</head>
<body>
  {{ markers|json_script:"markers-data" }}
  <div id="map"></div>
  <script src="{% static 'map.js' %}"></script>
</body>
</html>
```

Generated GeoJSON

```
<script id="markers-data" type="application/json">
  {
    "type": "FeatureCollection",
    "crs": { "type": "name", "properties": { "name": "EPSG:4326" } },
    "features": [
      {
        "type": "Feature",
        "properties": { "name": "Monte Amaro 2793m", "pk": "1" },
        "geometry": {
          "type": "Point",
          "coordinates": [14.08591836494682, 42.08632592463349]
        }
      }
    ]
  }
</script>
```

Rendering all markers in the map

```
// markers/static/map.js

const copy = '© <a href="https://osm.org/copyright">OpenStreetMap</a> contributors'
const url = 'https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png'
const osm = L.tileLayer(url, { attribution: copy })
const map = L.map('map', { layers: [osm], minZoom: 5 })
const markers = JSON.parse(document.getElementById('markers-data').textContent)
const features = L.geoJSON(markers).bindPopup(layer => layer.feature.properties.name)
map.addLayer(features).fitBounds(feature.getBounds())
```

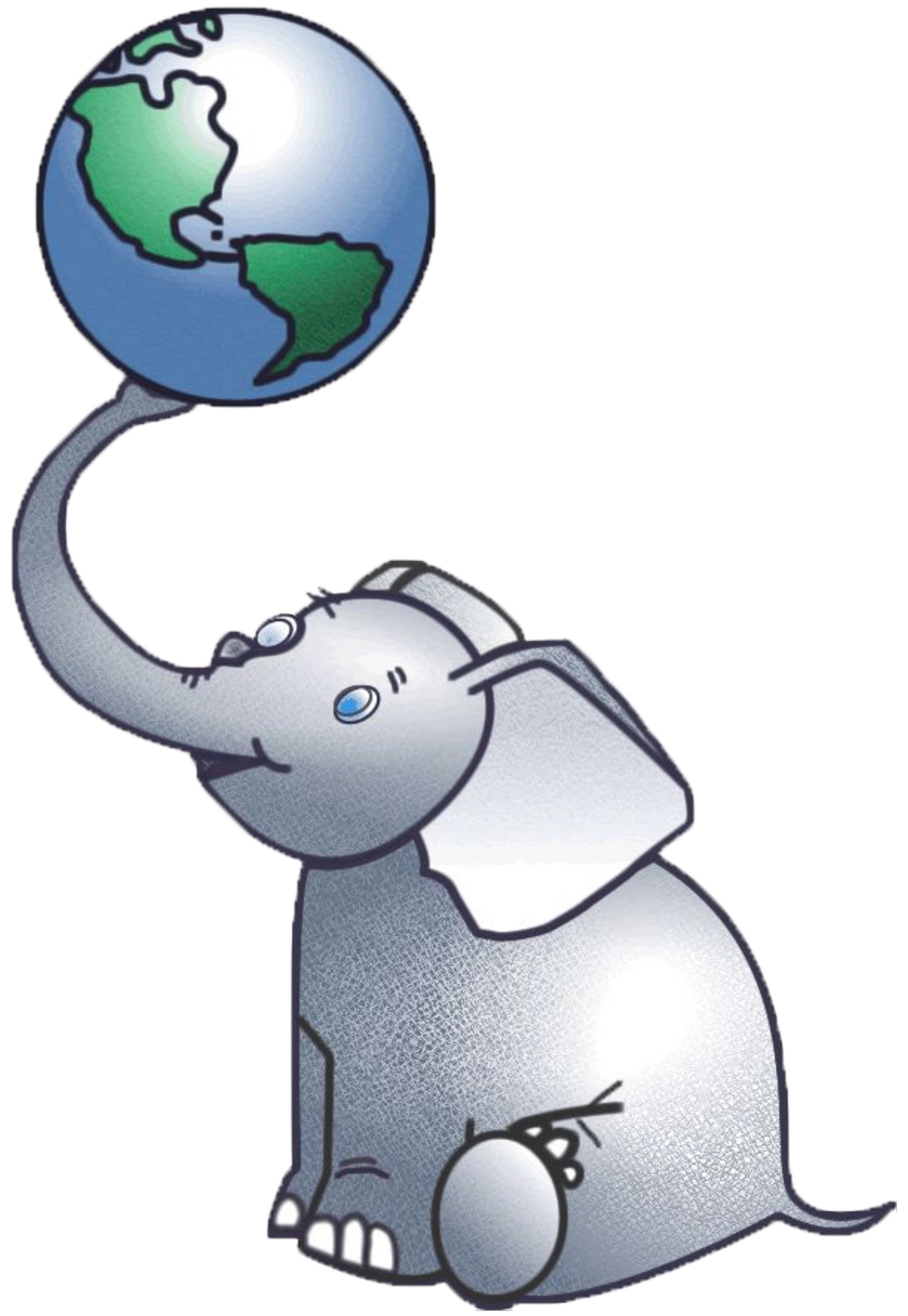
Show the populated web map

```
$ python3 manage.py runserver

$ python3 -m webbrowser -t localhost:8000/markers/map
```

Monte Amaro 2793m 🇮🇹
Leaflet | © OpenStreetMap contributors

Monte Focalone 0.30
Monte Amaro 2.40
Guado di Coccia 6.40
P
Bivacco Fusco 0.05
P
G9
G7 - Anfiteatro delle Murelle 0.30
2.450 m
0 km
27 km
7 km



→| PostGIS

- **PostgreSQL** extension
- Best* GeoDjango backend
- Spatial data types
- Spatial indexing
- Spatial functions

Installing PostgreSQL C client library

```
# apt-get install libpq5
#
# -- Read the docs for other operating systems.
```

Activating PostGIS

```
# mymap/settings.py

DATABASES = {
    "default": {
        "ENGINE": "django.contrib.gis.db.backends.postgis",
        "HOST": "database",
        "NAME": "mymap",
        "PASSWORD": "password",
        "PORT": 5432,
        "USER": "postgres",
    }
}
```

Requirements

```
# requirements.txt

django~=3.1.0
psycopg2~=2.8.0
djangorestframework~=3.12.0
djangorestframework-gis~=0.17
django-filter~=2.4.0
```

Installing requirements

```
$ python3 -m pip install -r requirements.txt
```

Activating Django REST Framework

```
# mymap/settings.py

INSTALLED_APPS = [
    "django.contrib.admin",
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",
    "django.contrib.gis",
    "rest_framework",
    "rest_framework_gis",
    "markers",
]
```

Adding the Marker serializer

```python
# markers/serializers.py

from rest_framework_gis import serializers

from markers.models import Marker


class MarkerSerializer(serializers.GeoFeatureModelSerializer):

    class Meta:
        fields = ("id", "name")
        geo_field = "location"
        model = Marker
```

Adding the Marker viewset

```python
# markers/api_views.py

from rest_framework import viewsets
from rest_framework_gis import filters

from markers.models import Marker
from markers.serializers import MarkerSerializer


class MarkerViewSet(viewsets.ReadOnlyModelViewSet):
    bbox_filter_field = "location"
    filter_backends = (filters.InBBoxFilter,)
    queryset = Marker.objects.all()
    serializer_class = MarkerSerializer
```

Adding API 'markers' urls

```python
# markers/api_urls.py

from rest_framework import viewsets
from rest_framework import routers

from markers.api_views import MarkerViewSet

router = routers.DefaultRouter()
router.register(r"markers", MarkerViewSet)

urlpatterns = router.urls
```

Updating ‘mymap’ urls

```
# mymap/urls.py

from django.contrib import admin
from django.urls import include, path

urlpatterns = [
    path("admin/", admin.site.urls),
    path("api/", include("markers.api_urls")),
    path("markers/", include("markers.urls")),
]
```

→| Trying to locate the user

```
// markers/static/map.js

const copy = '© <a href="https://osm.org/copyright">OpenStreetMap</a> contributors'
const url = 'https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png'
const osm = L.tileLayer(url, { attribution: copy })
const map = L.map('map', { layers: [osm], minZoom: 5 })
map.locate()
  .on('locationfound', e => map.setView(e.latlng, 8))
  .on('locationerror', () => map.setView([0, 0], 5))
// …
```

Rendering markers incrementally

```
// markers/static/map.js

async function load_markers() {
  const markers_url = `/api/markers/?in_bbox=${map.getBounds().toBBoxString()}`
  const response = await fetch(markers_url)
  const geojson = await response.json()
  return geojson
}

async function render_markers() {
  const markers = await load_markers()
  L.geoJSON(markers).bindPopup(layer => layer.feature.properties.name).addTo(map)
}

map.on('moveend', render_markers)
```

Leaflet | © OpenStreetMap contributors

2.793 m
0 km
27 km
14 km

→| What's next

- Markers customization
- Relational filtering
- Clustering frontend/backend
- Geocoding services
- ...

Tips

- docs in djangoproject.com
- details in postgis.net
- source code in github.com
- questions in gis.stackexchange.com

→| License

CC BY-SA 4.0

20tab.com

info@20tab.com

20tab

20tab

@20tab

paulox.net

paolo@melchiorre.org

pauloxnet

paolomelchiorre

@pauloxnet