

Carmelo Catalfamo @mratomicbomb_
Paolo Melchiorre @pauloxnet

Maps *with* GeoDjango PostGIS *and* Leaflet



20tab.com



Paolo Melchiorre @pauloxnet



- Computer Science Engineer
- Python Developer *since 2006*
- PostgreSQL user (*not a DBA*)
- Django Developer *since 2011*
- Remote Worker *since 2015*
- Senior Developer *at 20tab*

Carmelo Catalfamo @MrAtomicBomb_



- JavaScript Developer
- NodeJS developer *since 2014*
- ReactJS developer *since 2015*
- React Native developer
- Django user
- Frontend developer *at 20tab*



www.20tab.com



- Rome based with remote workers
- Meetup and conferences
- Agile and Lean methodologies
- Growth marketing approach
- Software development
- Python, Django, React JS, uWSGI

Goal

*Find a simple way
to integrate a web map
in a Django project.*

A map of Europe with several callout boxes pointing to specific locations. The callouts are: 'Use case' pointing to Marseille, 'Basic map' pointing to Florence, 'PostGIS' pointing to Rome, 'GeoDjango' pointing to Pescara, and 'Leaflet JS' pointing to Messina. The map shows various countries and cities in Europe, including France, Italy, Germany, and others.

Use case

Basic map

PostGIS

GeoDjango

Leaflet JS

Web map



- Map delivered by GIS
- Static and Dynamic
- Interactive and view only
- Raster or Vector tiles
- Spatial databases
- Javascript library



GeoDjango



- `django.contrib.gis`
- Geographic framework
- Spatial Field Types
- Spatial ORM queries
- Admin geometry fields
- Four database backends

PostGIS



- Best GeoDjango backend
- PostgreSQL extension
- Integrated spatial data
- Spatial data types
- Spatial indexing
- Spatial functions



Leaflet



- JavaScript library for maps
- Free Software
- Desktop & Mobile friendly
- Light (< 40 KB of gzip JS)
- Well documented
- Simple, performing, usable

Basic map example

Making queries

```
from django.db import models
```

```
class Blog(models.Model):  
    name = models.CharField(max_length=100)
```

```
class Author(models.Model):  
    name = models.CharField(max_length=200)
```

```
class Entry(models.Model):  
    blog = models.ForeignKey(Blog, on_delete=models.CASCADE)  
    headline = models.CharField(max_length=255)  
    authors = models.ManyToManyField(Author)
```

Settings

```
INSTALLED_APPS = [  
    # ...  
    'django.contrib.gis',  
]  
  
DATABASES = {'default': {  
    'ENGINE': 'django.contrib.gis.db.backends.postgis',  
    # ...  
}}
```

Migrations

```
from django.contrib.postgres import operations
from django.db import migrations

class Migration(migrations.Migration):
    dependencies = [('blog', '0001_initial')]
    operations = [
        operations.CreateExtension('postgis')
    ]
```


Models

```
from django.contrib.gis.db.models import PointField
from django.db import models

class Entry(models.Model):
    # ...
    point = PointField()

    @property
    def lat_lng(self):
        return list(getattr(self.point, 'coords', [])[::-1])
```

Admin

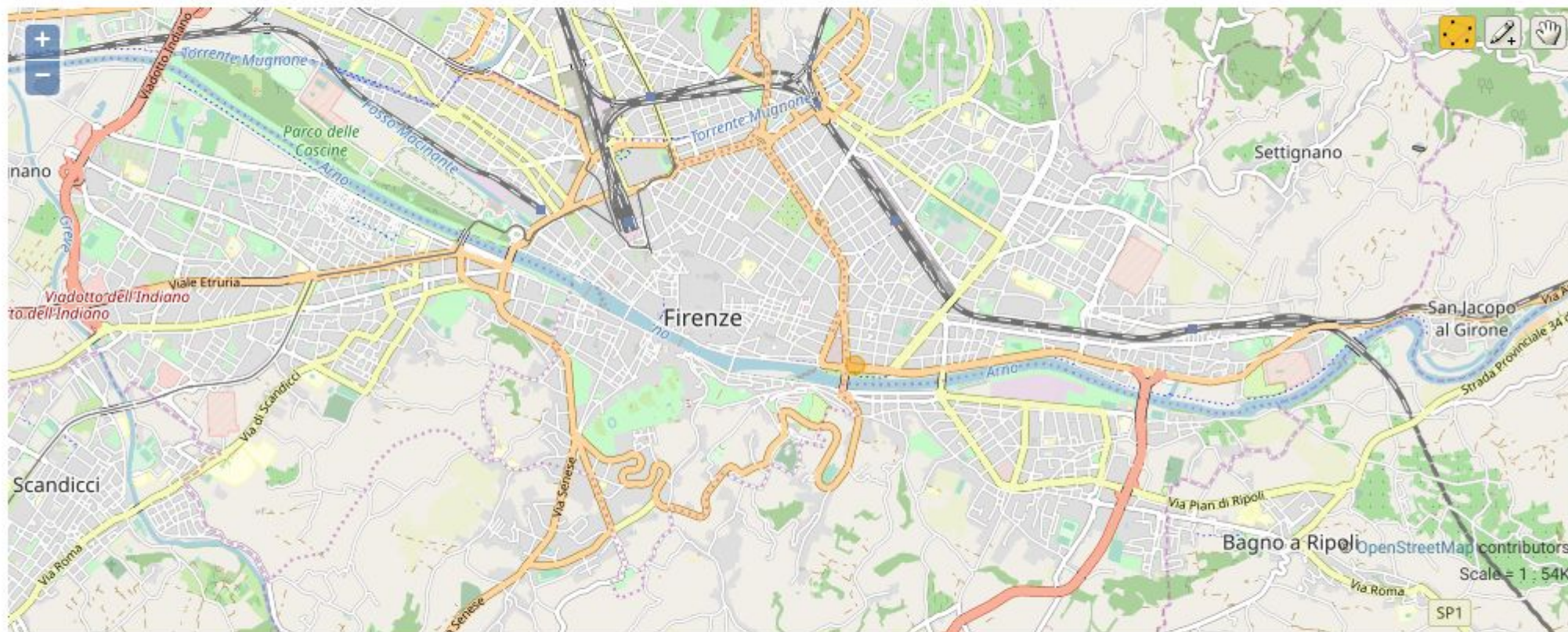
```
from django.contrib import admin
from django.contrib.gis.admin import OSMGeoAdmin
from .models import Entry

@admin.register(Entry)
class EntryAdmin(OSMGeoAdmin):
    default_lon = 1253000
    default_lat = 5430000
    default_zoom = 12
    # ...
```

Change Entry

HISTORY

Point:



Delete all Features

Delete

Save and add another

Save and continue editing

SAVE

Views and urls

```
from django.urls import path
from django.views.generic import ListView
from .models import Entry

class EntryList(ListView):
    queryset = Entry.objects.filter(point__isnull=False)

urlpatterns = [
    path('map/', EntryList.as_view()),
]
```

Template

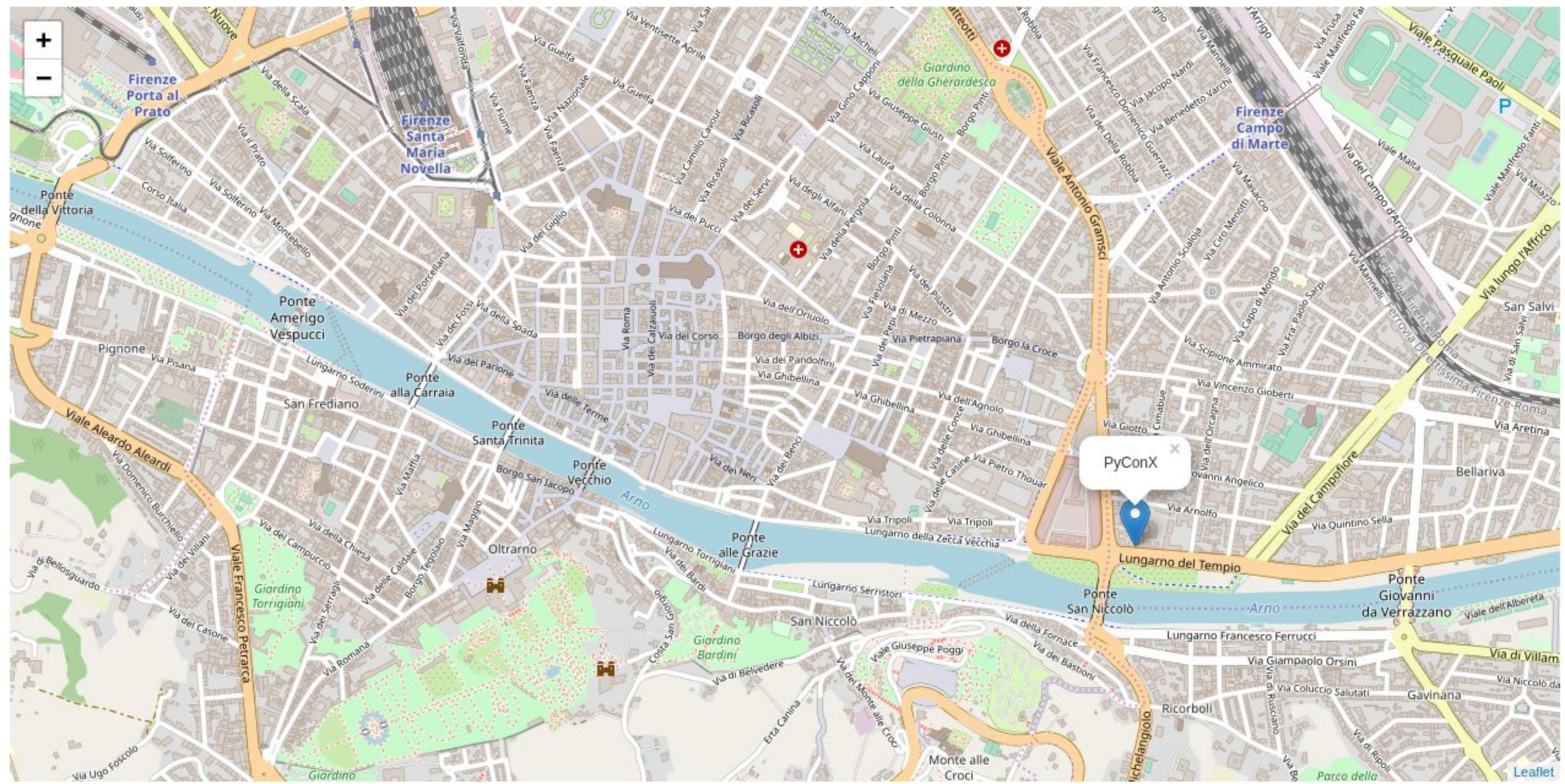
```
<html><head>
  <link rel="stylesheet"
        href="//unpkg.com/leaflet/dist/leaflet.css"/>
  <script src="//unpkg.com/leaflet/dist/leaflet.js"></script>
</head>

<body><h1>PyConX Venues</h1>
  <div id="m" style="width: 1920px; height: 1080px;"></div>
  <!-- add javascript here -->
</body></html>
```

Javascript

```
<script type="text/javascript">  
    var m = L.map('m').setView([43.77, 11.26], 15); # Florence  
    L.tileLayer('://{s}.tile.osm.org/{z}/{x}/{y}.png').addTo(m);  
  
    {% for e in object_list %}  
        L.marker({{e.lat_lng}}).addTo(m);  
    {% endfor %}  
  
</script>
```


PyConX Venue



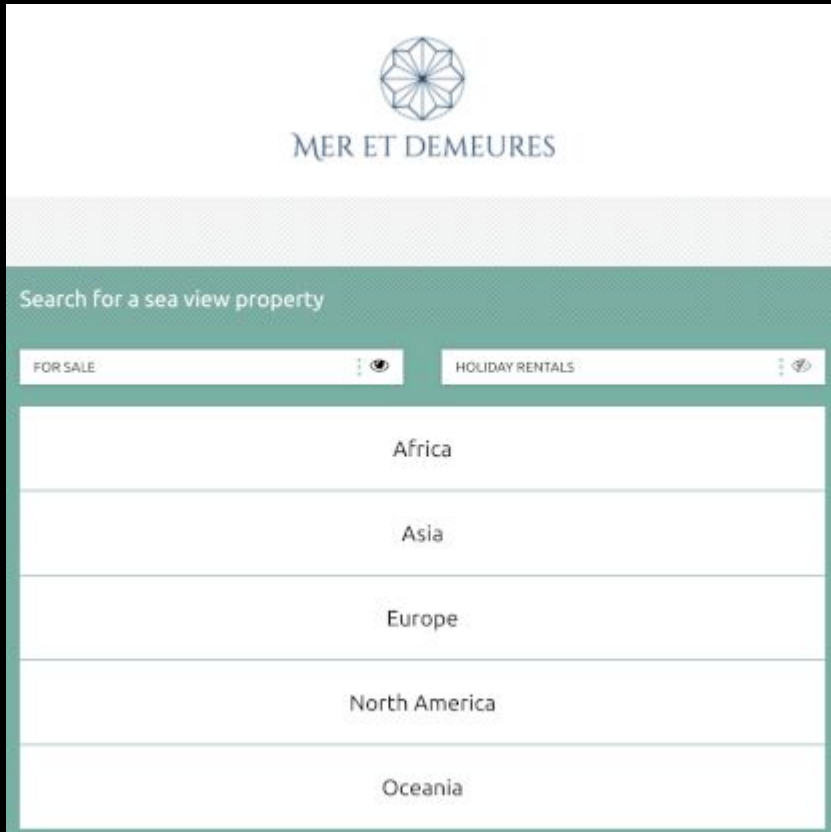
Use case



MER ET DEMEURES

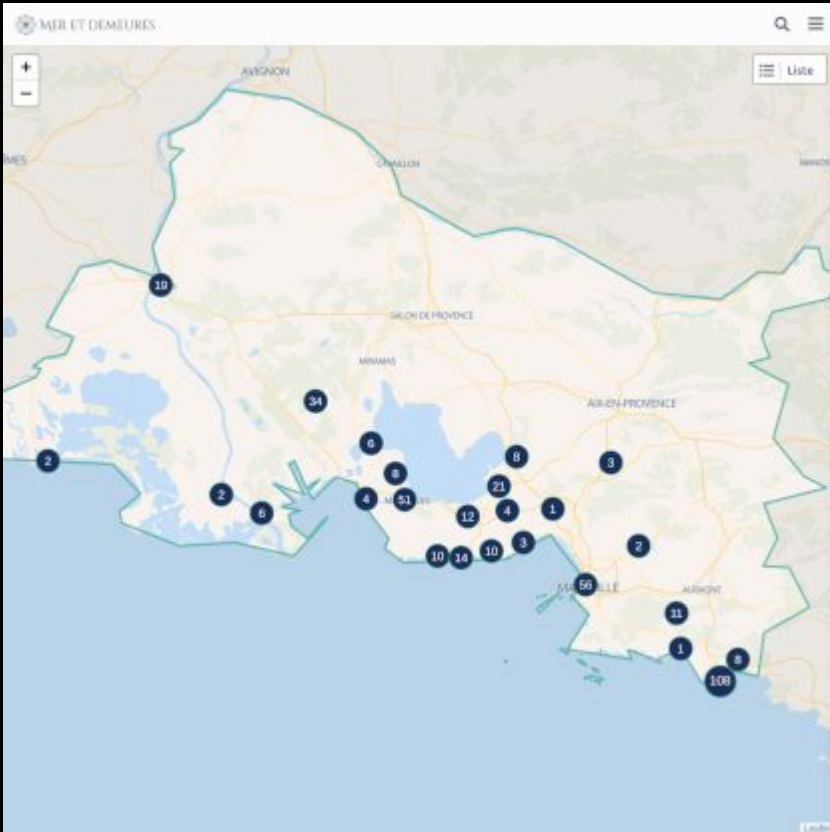
- Coastal properties
- Active since 2014
- 8 Languages
- ~ 100k active advertisements
- ~ 40 Countries
- 6 Continents

Version 1.0



- Django 1.6
- Python 2.7
- PostgreSQL 9.3
- Text Spatial Fields
- Leaflet 1.0
- Static/View-only map

Version 2.0



- Django 2.2 / GeoDjango
- Python 3.6
- PostgreSQL 10
- PostGIS 2.4 / Spatial data
- Leaflet 1.4
- Dynamic/Interactive map

Models

```
from django.db import models
from django.contrib.gis.db.models import (
    MultiPolygonField, PointField
)

class City(models.Model):
    borders = MultiPolygonField()

class Ad(models.Model):
    city = models.ForeignKey(City, on_delete=models.CASCADE)
    location = PointField()
```

Setup

```
$ pip install djangorestframework      # RESTful API
$ pip install djangorestframework-gis  # Geographic add-on
$ pip install django-filter            # Filtering support
```

```
INSTALLED_APPS = [
    # ...
    'django.contrib.gis',
    'rest_framework',
    'rest_framework_gis',
    'django_filters',
]
```

Serializer

```
from rest_framework_gis.serializers import (  
    GeoFeatureModelSerializer  
)  
from .models import Ad  
  
class AdSerializer(GeoFeatureModelSerializer):  
    class Meta:  
        model = Ad  
        geo_field = 'location'  
        fields = ('id',)
```

Views

```
from rest_framework.viewsets import ReadOnlyModelViewSet
from rest_framework_gis.filters import InBoundingBoxFilter
from .models import Ad
from .serializers import AdSerializer

class AdViewSet(ReadOnlyModelViewSet):
    bbox_filter_field = 'location'
    filter_backends = (InBoundingBoxFilter,)
    queryset = Ad.objects.filter(location__isnull=False)
    serializer_class = AdSerializer
```


Urls

```
from rest_framework.routers import DefaultRouter
from .views import AdViewSet
```

```
router = DefaultRouter()
router.register(r'markers', AdViewSet, basename='marker')
urlpatterns = router.urls
```

GeoJSON

```
{ "type": "FeatureCollection", "features": [{  
  "id": 1,  
  "type": "Feature",  
  "geometry": {  
    "type": "Point",  
    "coordinates": [11.255814, 43.769562]  
  },  
  "properties": {}  
}] }
```

Ajax / Leaflet

```
<script type="text/javascript">
  var m = L.map('m').setView([43.77, 11.26], 15)
  L.tileLayer('//{s}.tile.osm.org/{z}/{x}/{y}.png').addTo(m)

  fetch('/markers')
    .then(function (results) {
      L.geoJSON(results).addTo(m)
    })
</script>
```

React Leaflet

```
import React from 'react'
import { Map, TileLayer, GeoJSON } from 'react-leaflet'

export default class Map extends Component {
  state = { geoJson: {} }

  onMove = () => {
    fetch('/markers')
      .then(geoJson => this.setState({ geoJson }))
  }

  // render () {...}
}
```


React Render method

```
render () {  
  return (  
    <Map center={c} zoom={z} onMoveend={this.onMove}>  
      <TileLayer url="//{s}.tile.osm.org/{z}/{x}/{y}.png"/>  
      <GeoJSON data={this.state.geoJson} />  
    </Map>  
  )  
}
```

Sale

France

Type de Bien

Plus de Critères

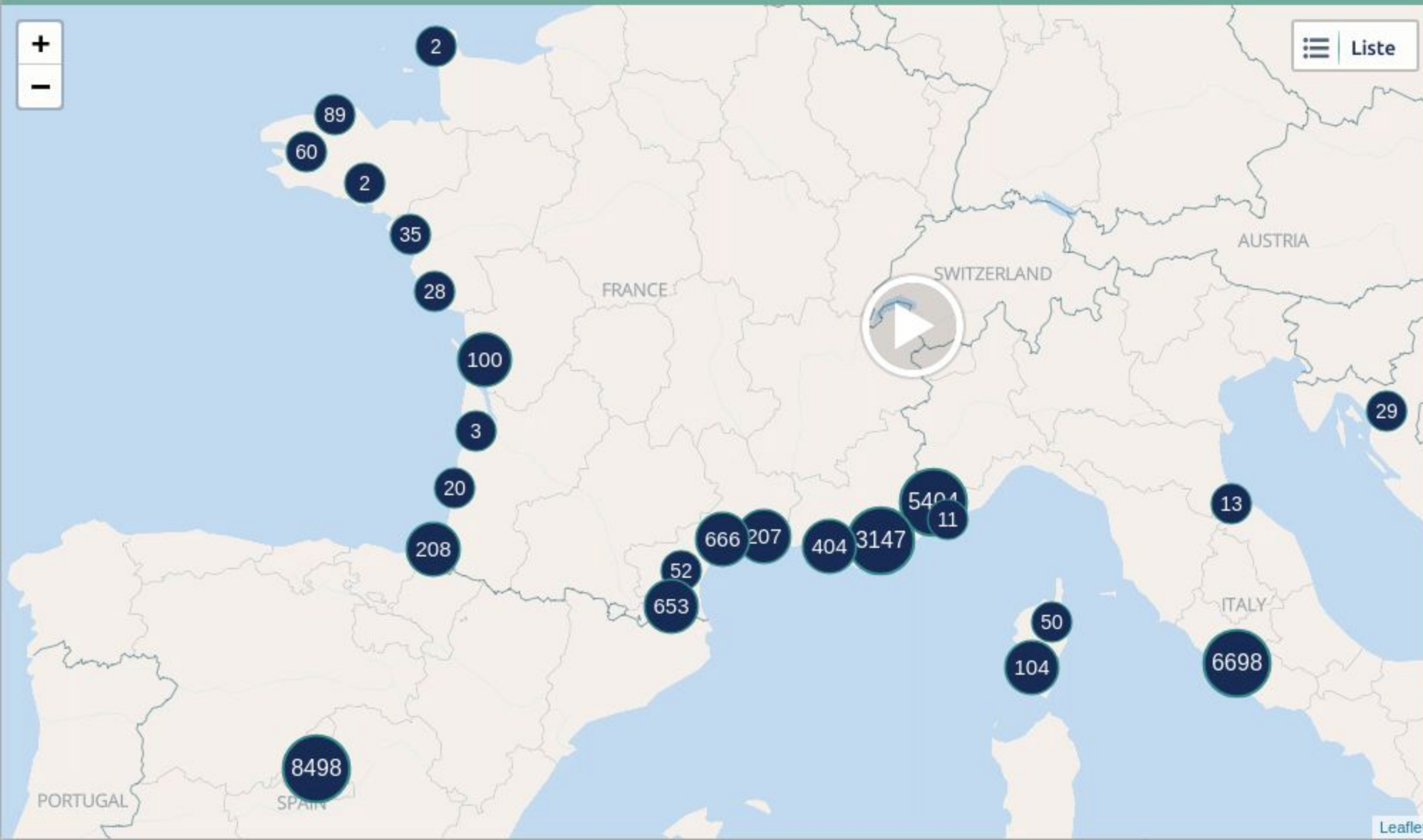


11235 résultats pour:
Sale properties - France

Trier par...



Liste



€520,000

Apartment, Royan

2 1 70 m²



€290,000

Apartment, Cannes

35 m²



villa for sale in le-lavandou var

Conclusion



- Out-of-the-box features
- Spatial & Relational queries
- Django/PostgreSQL
- Backend clusterization
- Administrative levels
- Dynamic spatial entity



Resources



- docs.djangoproject.com/en/
- github.com/django/django
- postgis.net/docs/
- github.com/postgis/postgis
- leafletjs.com/reference.html
- github.com/leaflet/leaflet

Acknowledgments



Mer et Demeures
meretdemeures.com



Twentytab
20tab.com/lavora-con-noi

Info



- www.paulox.net/talks
- Slides
- Code samples
- Resource URLs
- Questions and comments
- License (CC BY-SA)

Thank you!

Carmelo Catalfamo
carmelocatalfamo.it
@mratomicbomb_

Paolo Melchiorre
paulox.net
@pauloxnet



20tab.com

paulox.net

