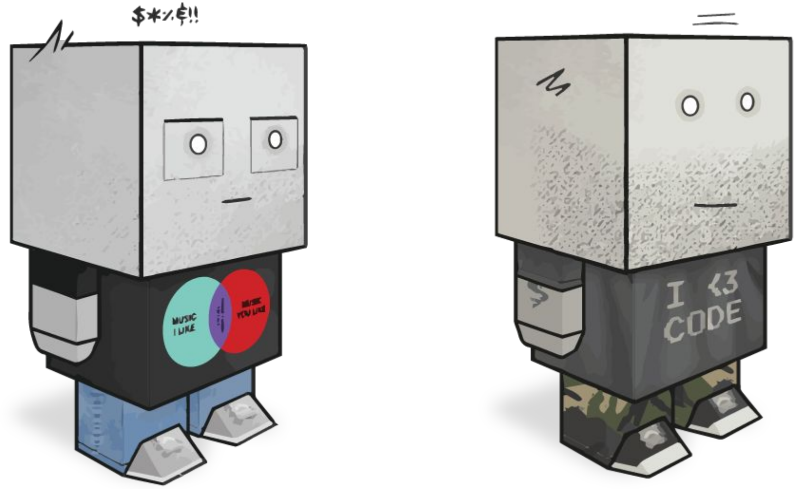
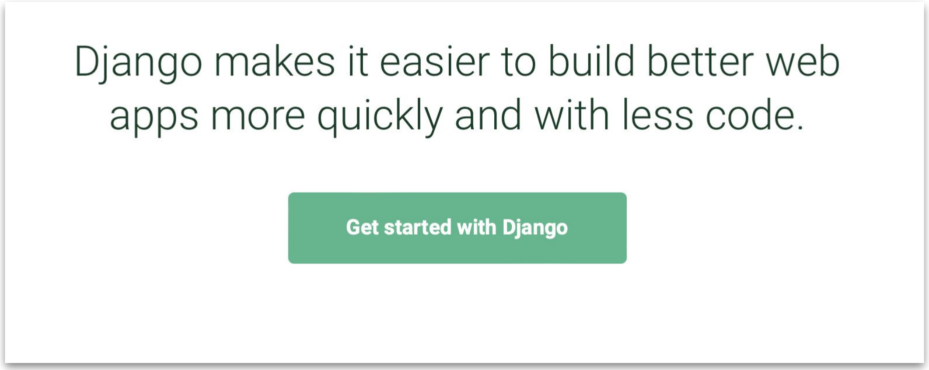
NAUTOGUIDE



Locaria

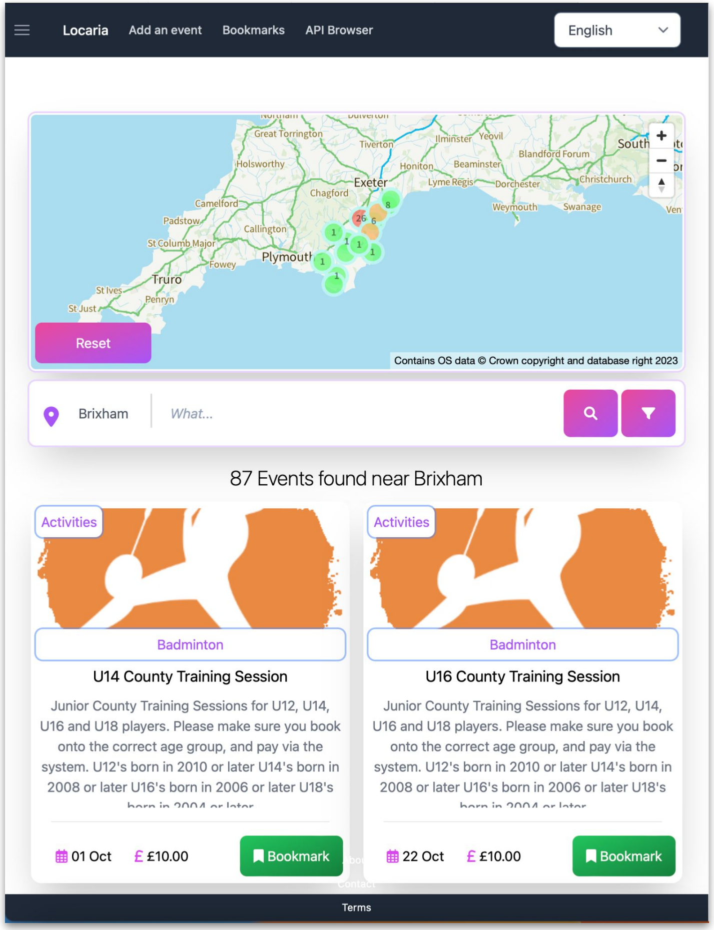
https://explore.locaria.org/



Django makes it easier to build better web apps more quickly and with less code.

Get started with Django

https://www.djangoproject.com



Locaria Add an event Bookmarks API Browser English

Reset

Contains OS data © Crown copyright and database right 2023

Brixham What...

87 Events found near Brixham

Activities

Badminton

U14 County Training Session

Junior County Training Sessions for U12, U14, U16 and U18 players. Please make sure you book onto the correct age group, and pay via the system. U12's born in 2010 or later U14's born in 2008 or later U16's born in 2006 or later U18's born in 2004 or later

01 Oct £10.00 Bookmark

Activities

Badminton

U16 County Training Session

Junior County Training Sessions for U12, U14, U16 and U18 players. Please make sure you book onto the correct age group, and pay via the system. U12's born in 2010 or later U14's born in 2008 or later U16's born in 2006 or later U18's born in 2004 or later

22 Oct £10.00 Bookmark

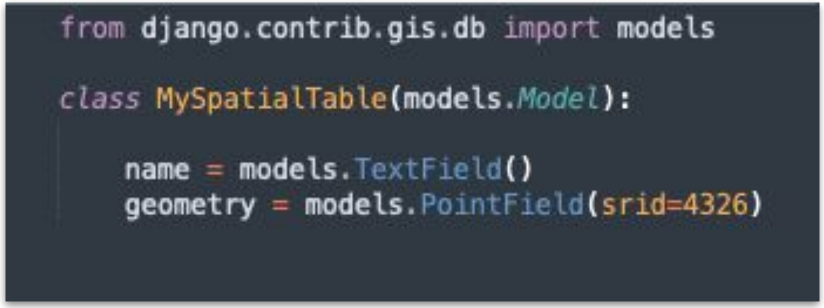
Terms

Why Django?

- Python
- Model/View Separation
- Extended with “Apps”
- Easy to containerise
- Massive community support
- Cool name
- Database abstraction
- Admin out of the box

GeoDjango

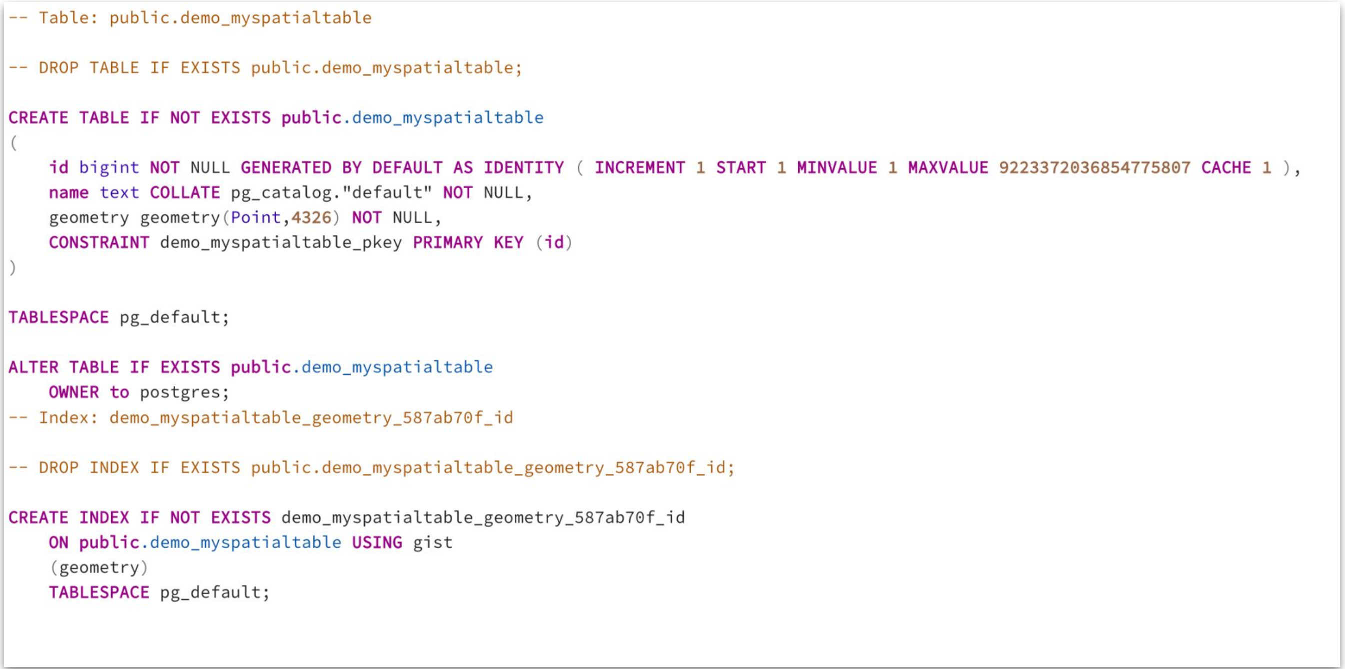
- Model fields for OGC geometries
- Spatial Queries
- Geometry admin and editors



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

class MySpatialTable(models.Model):

    name = models.TextField()
    geometry = models.PointField(srid=4326)
```

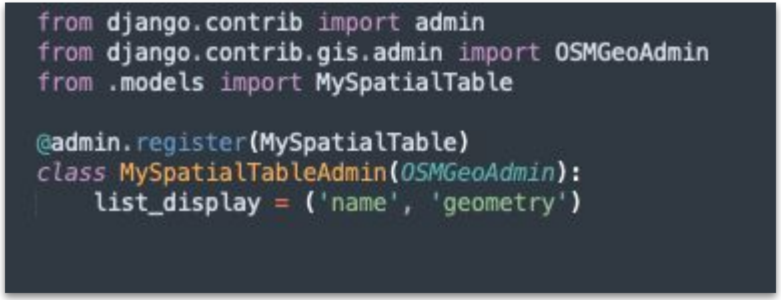
```
python manage.py makemigrations

python manage.py migrate
```

No SQL

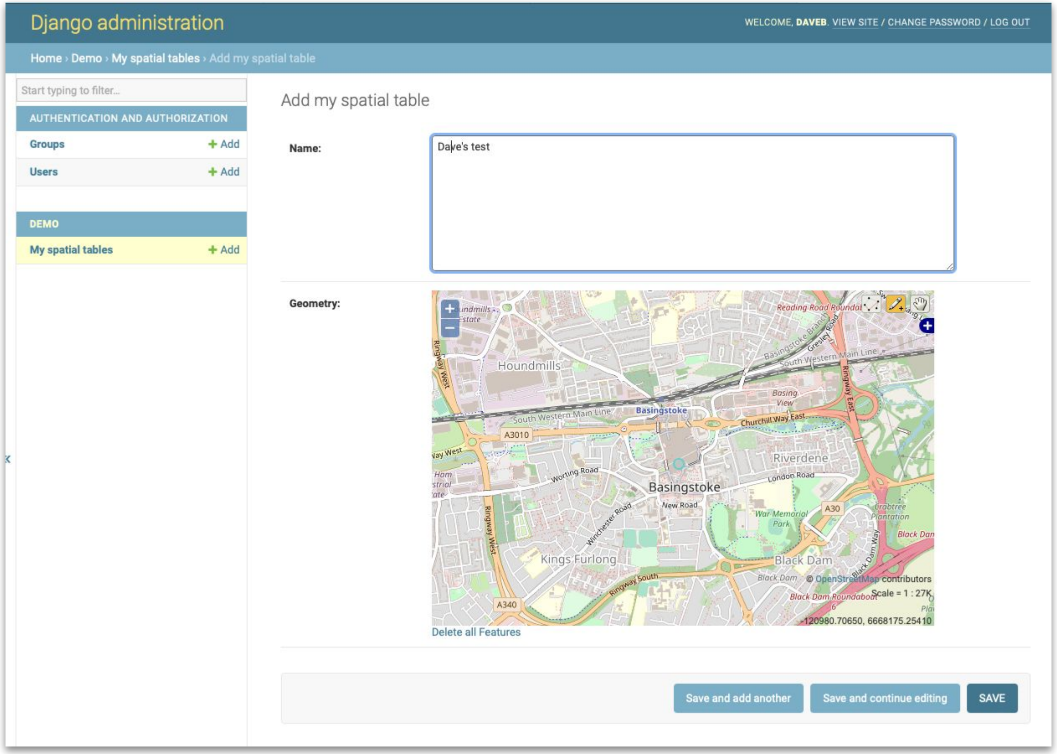


```sql
-- Table: public.demo_myspatialtable

-- DROP TABLE IF EXISTS public.demo_myspatialtable;

CREATE TABLE IF NOT EXISTS public.demo_myspatialtable
(
    id bigint NOT NULL GENERATED BY DEFAULT AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 9223372036854775807 CACHE 1 ),
    name text COLLATE pg_catalog."default" NOT NULL,
    geometry geometry(Point,4326) NOT NULL,
    CONSTRAINT demo_myspatialtable_pkey PRIMARY KEY (id)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.demo_myspatialtable
    OWNER to postgres;
-- Index: demo_myspatialtable_geometry_587ab70f_id

-- DROP INDEX IF EXISTS public.demo_myspatialtable_geometry_587ab70f_id;

CREATE INDEX IF NOT EXISTS demo_myspatialtable_geometry_587ab70f_id
    ON public.demo_myspatialtable USING gist
    (geometry)
    TABLESPACE pg_default;
```

Easy Geo-Admin



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

@admin.register(MySpatialTable)
class MySpatialTableAdmin(OSMGeoAdmin):
    list_display = ('name', 'geometry')
```



Django administration

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

Home › Demo › My spatial tables › Add my spatial table

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

DEMO

My spatial tables + Add

Add my spatial table

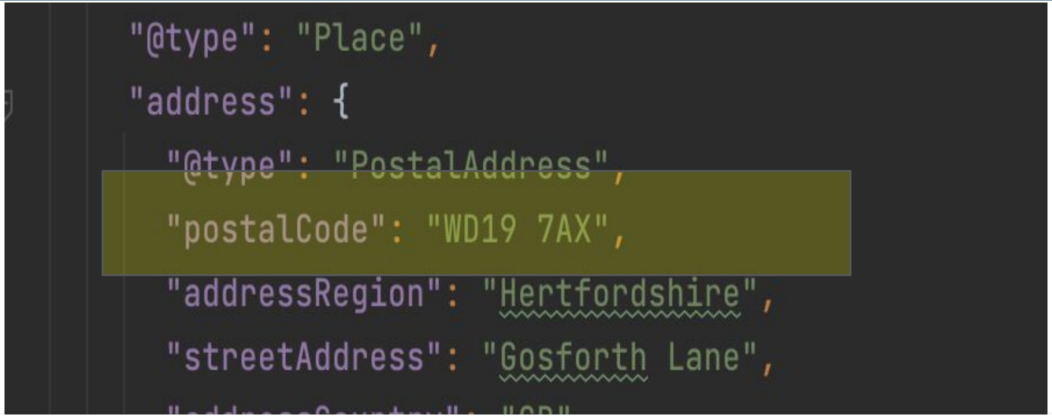
Name: Dave's test

Geometry:

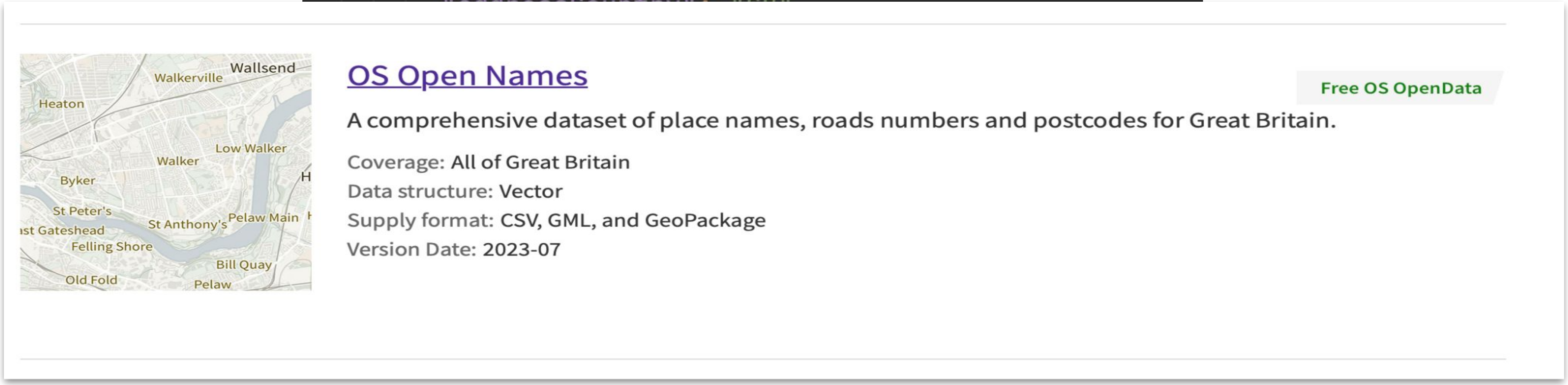
Delete all Features

Save and add another | Save and continue editing | SAVE

Locaria Geocoder



```
"@type": "Place",
"address": {
  "@type": "PostalAddress",
  "postalCode": "WD19 7AX",
  "addressRegion": "Hertfordshire",
  "streetAddress": "Gosforth Lane",
```



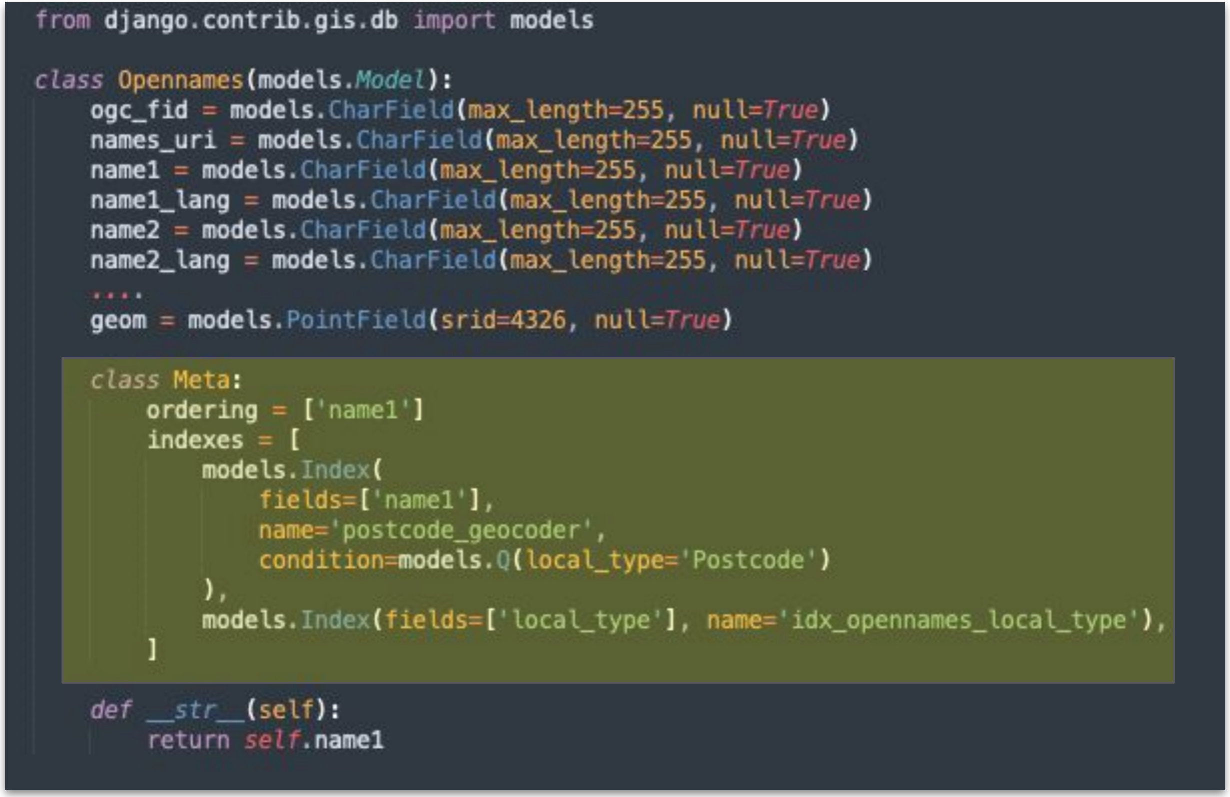
[OS Open Names](#)

Free OS OpenData

A comprehensive dataset of place names, roads numbers and postcodes for Great Britain.

Coverage: All of Great Britain
Data structure: Vector
Supply format: CSV, GML, and GeoPackage
Version Date: 2023-07

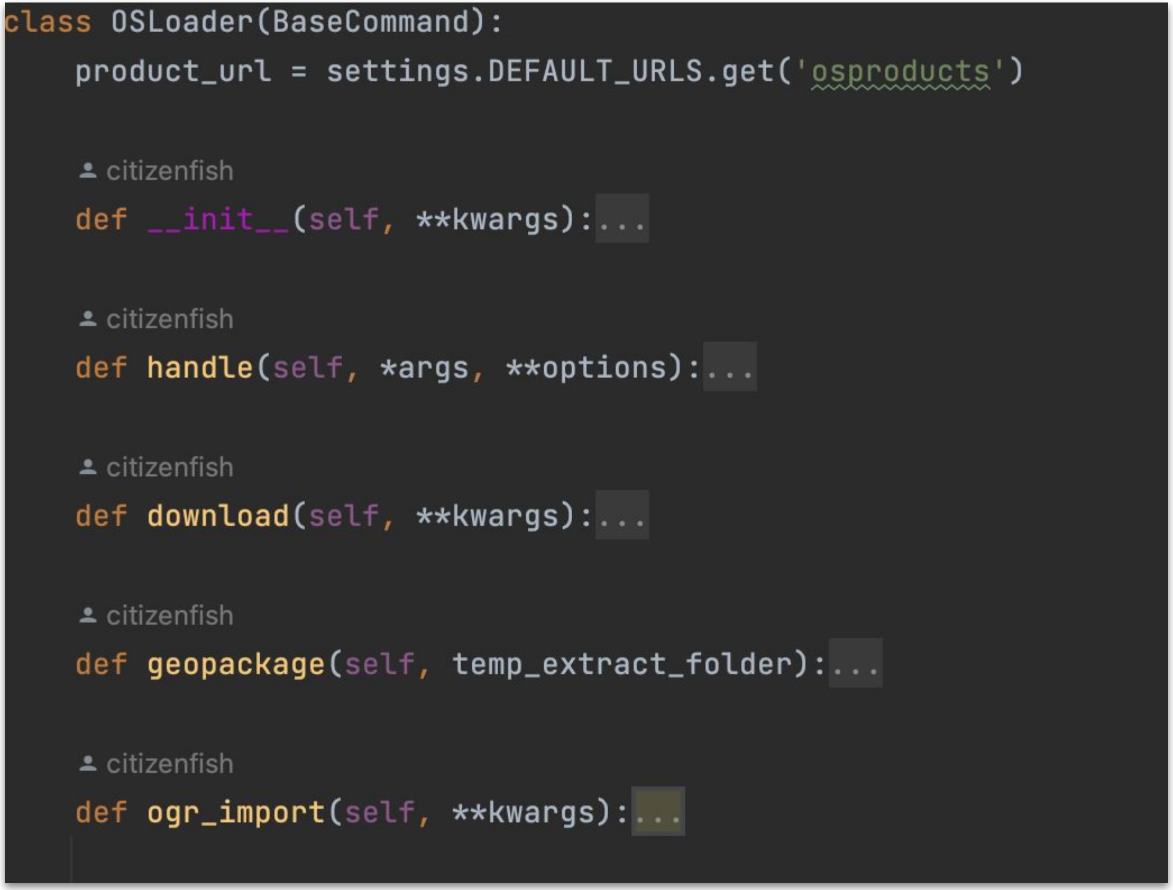
Model



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

class Opennames(models.Model):
    ogc_fid = models.CharField(max_length=255, null=True)
    names_uri = models.CharField(max_length=255, null=True)
    name1 = models.CharField(max_length=255, null=True)
    name1_lang = models.CharField(max_length=255, null=True)
    name2 = models.CharField(max_length=255, null=True)
    name2_lang = models.CharField(max_length=255, null=True)
    ....
    geom = models.PointField(srid=4326, null=True)

    class Meta:
        ordering = ['name1']
        indexes = [
            models.Index(
                fields=['name1'],
                name='postcode_geocoder',
                condition=models.Q(local_type='Postcode')
            ),
            models.Index(fields=['local_type'], name='idx_opennames_local_type'),
        ]

    def __str__(self):
        return self.name1
```

- Model fields same as Opennames
- Indexes Created

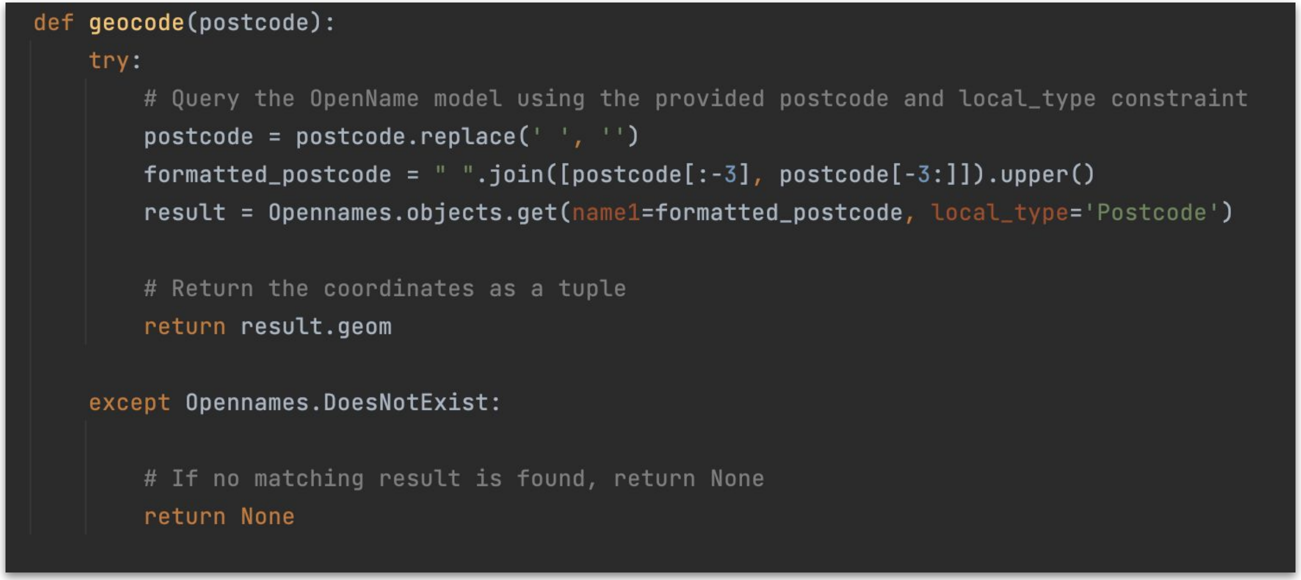
Loader



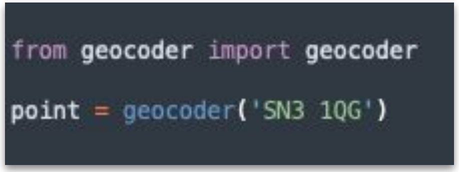
```
class OSLoader(BaseCommand):
    product_url = settings.DEFAULT_URLS.get('osproducts')

    citizenfish
    def __init__(self, **kwargs):...

    citizenfish
    def handle(self, *args, **options):...

    citizenfish
    def download(self, **kwargs):...

    citizenfish
    def geopackage(self, temp_extract_folder):...

    citizenfish
    def ogr_import(self, **kwargs):...
```

- Commands use Django framework
- Use OS API to download data in Geopackge
- OGR2OGR for import

Postcode Geocoder



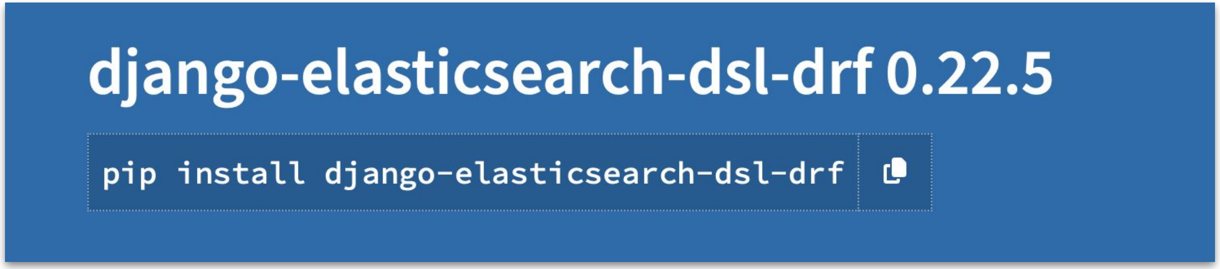
```python
def geocode(postcode):
    try:
        # Query the OpenName model using the provided postcode and local_type constraint
        postcode = postcode.replace(' ', '')
        formatted_postcode = " ".join([postcode[:-3], postcode[-3:]]).upper()
        result = Opennames.objects.get(name1=formatted_postcode, local_type='Postcode')

        # Return the coordinates as a tuple
        return result.geom

    except Opennames.DoesNotExist:

        # If no matching result is found, return None
        return None
```



```python
from geocoder import geocoder

point = geocoder('SN3 1QG')
```

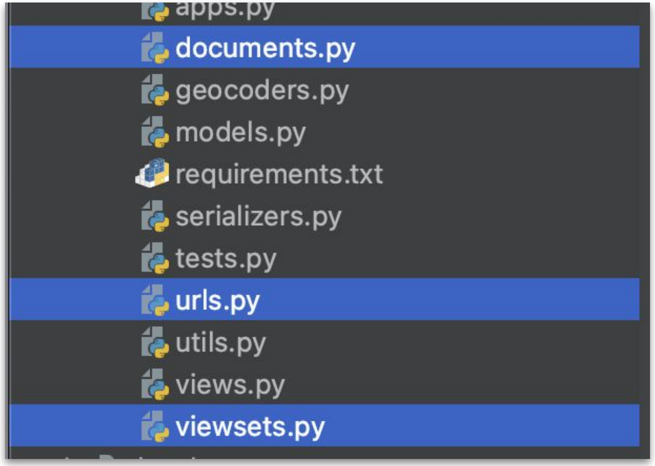
API Views

- Free text search
- Nearest
- Within
- By Type
- In a specific format
- With no usage restrictions

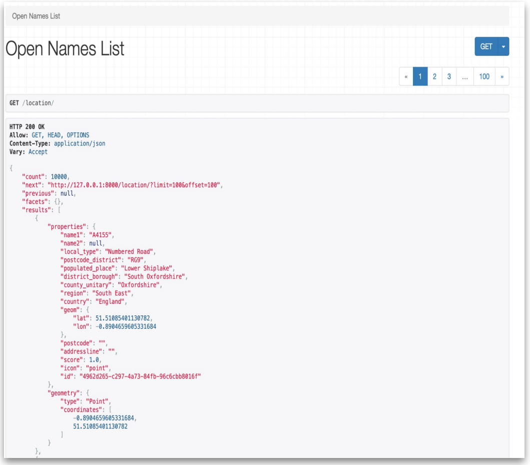


django-elasticsearch-dsl-drf 0.22.5

```
pip install django-elasticsearch-dsl-drf
```



```
apps.py
documents.py
geocoders.py
models.py
requirements.txt
serializers.py
tests.py
urls.py
utils.py
views.py
viewsets.py
```



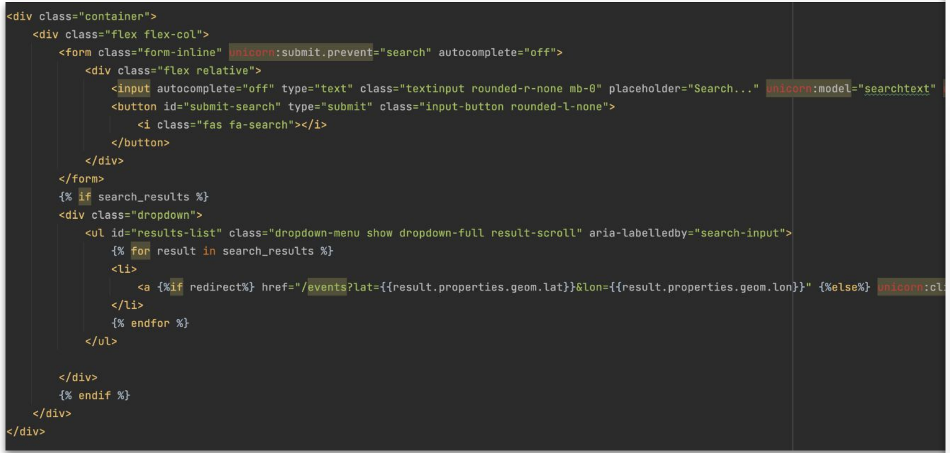
Open Names List

Open Names List

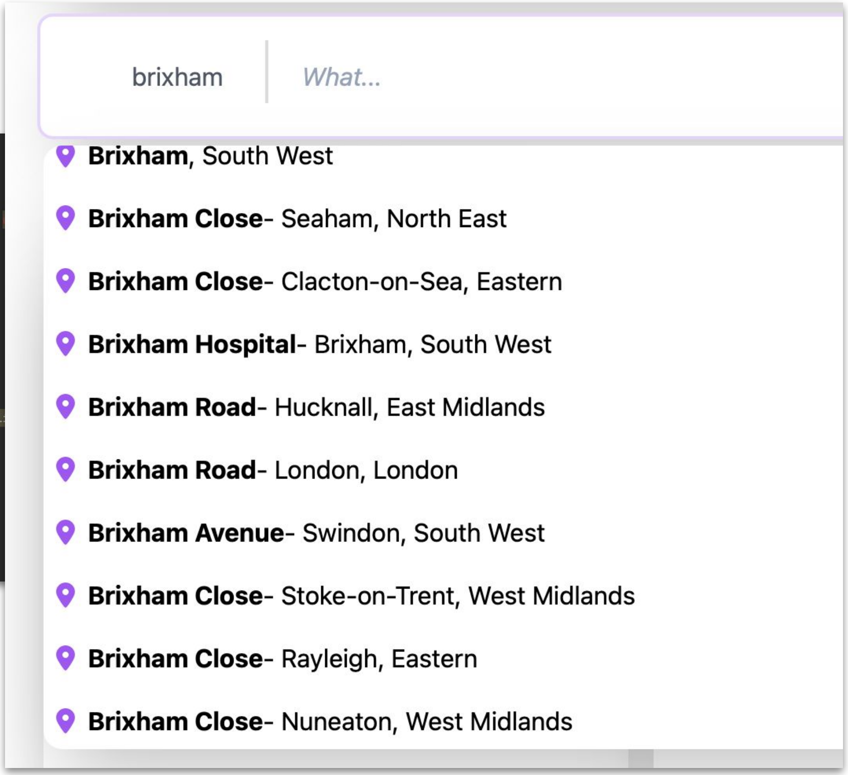
```
GET /location/
```

```
HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "count": 10000,
    "next": "http://127.0.0.1:8000/location/?limit=100&offset=100",
    "previous": null,
    "facets": {},
    "results": [
        {
            "properties": {
                "name1": "A4155",
                "name2": null,
                "local_type": "Numbered Road",
                "postcode_district": "RG9",
                "populated_place": "Lower Shiplake",
                "district_borough": "South Oxfordshire",
                "county_unitary": "Oxfordshire",
                "region": "South East",
                "country": "England",
                "geom": {
                    "lat": 51.51085401130782,
                    "lon": -0.8904659605331684
                },
                "postcode": "",
                "addressline": "",
                "score": 1.0,
                "icon": "point",
                "id": "4962d265-c297-4a73-84fb-96c6cbb8016f"
            },
            "geometry": {
                "type": "Point",
                "coordinates": [
                    -0.8904659605331684,
                    51.51085401130782
                ]
            }
        },
```

Other Components

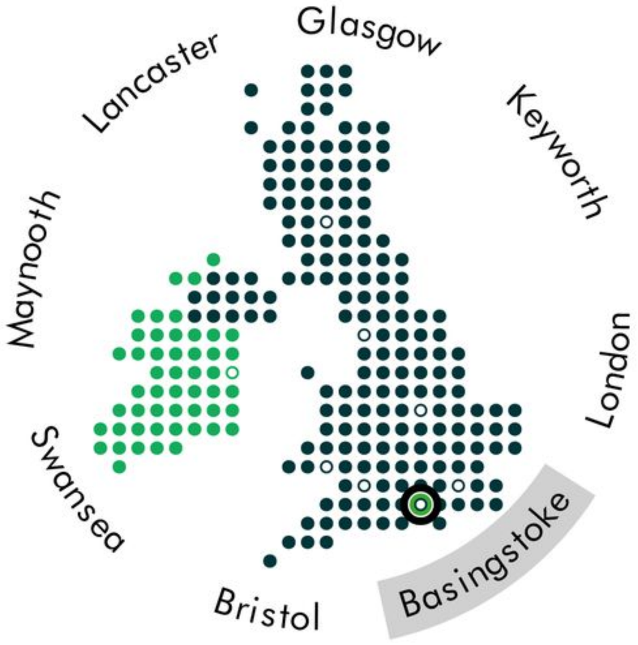


```
<div class="container">
    <div class="flex flex-col">
        <form class="form-inline" unicorn:submit.prevent="search" autocomplete="off">
            <div class="flex relative">
                <input autocomplete="off" type="text" class="textinput rounded-r-none mb-0" placeholder="Search..." unicorn:model="searchtext"
                <button id="submit-search" type="submit" class="input-button rounded-l-none">
                    <i class="fas fa-search"></i>
                </button>
            </div>
        </form>
        {% if search_results %}
        <div class="dropdown">
            <ul id="results-list" class="dropdown-menu show dropdown-full result-scroll" aria-labelledby="search-input">
                {% for result in search_results %}
                <li>
                    <a {%if redirect%} href="/events?lat={{result.properties.geom.lat}}&lon={{result.properties.geom.lon}}" {%else%} unicorn:cl
                </li>
                {% endfor %}
            </ul>

        </div>
        {% endif %}
    </div>
</div>
```



brixham | What...

- **Brixham**, South West
- **Brixham Close**- Seaham, North East
- **Brixham Close**- Clacton-on-Sea, Eastern
- **Brixham Hospital**- Brixham, South West
- **Brixham Road**- Hucknall, East Midlands
- **Brixham Road**- London, London
- **Brixham Avenue**- Swindon, South West
- **Brixham Close**- Stoke-on-Trent, West Midlands
- **Brixham Close**- Rayleigh, Eastern
- **Brixham Close**- Nuneaton, West Midlands



https://github.com/nautoguide/foss4gUK2023