Locaria

https://explore.locaria.org/

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

https://www.djangoproject.com

https://www.djangoproject.com
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
```

---

```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 COLLABE 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;
```
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')
Locaria Geocoder

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

OS Open Names

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
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'))
    ]

    def __str__(self):
        return self.name1

- Model fields same as Opennames
- Indexes Created
Loader

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

    def __init__(self, **kwargs):

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

    def download(self, **kwargs):

    def geopackage(self, temp_extract_folder):

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

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

from geocoder import geocoder
point = geocoder('SN3 10G')
```
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
```
Other Components

- 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