

Geo Web applications: GeoDjango

Option GIS-Python

hes.
SO
business.

Jean-Paul Calbimonte



School of Management

Bachelor of Science HES-SO (BSc) in Business
Information Technology

> GeoDjango

The Django logo, consisting of the word "django" in a white, lowercase, sans-serif font, centered within a dark green rectangular background.

What is GeoDjango?

<https://docs.djangoproject.com/en/3.0/ref/contrib/gis/>

- Contrib module for Django
- Geographic Web framework for Django
- Simple as possible to create geographic Web applications
- Django model fields for OGC geometries and raster data.
- Extensions to Django's ORM for querying and manipulating spatial data.
- Loosely-coupled, high-level Python interfaces for GIS geometry and raster operations
- Data manipulation in different formats.
- Editing geometry fields from the admin.

> settings.py

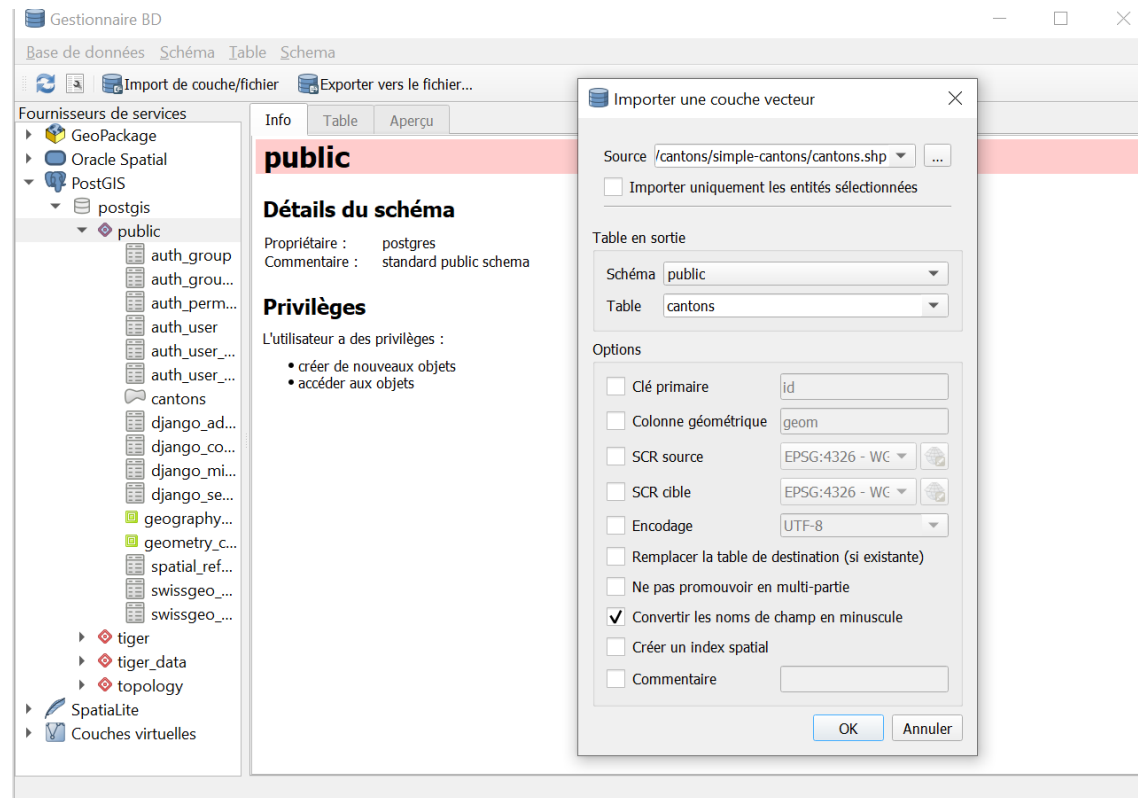
```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'django.contrib.gis',  
]
```

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.contrib.gis.db.backends.postgis',  
        'NAME': 'postgres',  
        'USER': 'postgres',  
        'PASSWORD': 'admin',  
        'HOST': 'db',  
        'PORT': '5432',    }  
}
```

> Import cantons shapefile with QGIS

Get the file at: data/cantons/simple-cantons/cantons.shp

Go to QGIS>Base de données>Gestionnaire BD



> models.py

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

```
class Canton(models.Model):  
    id=models.PositiveIntegerField(primary_key=True)  
    name=models.CharField(max_length=200)  
    geom=models.MultiPolygonField(srid=4326,null=True)  
  
    class Meta:  
        db_table = "cantons"  
  
    def __str__(self):  
        return self.name
```

```
> python manage.py makemigrations swissgeo
```

```
> python manage.py migrate --fake
```

The screenshot shows the Django Admin interface for the 'cantons' model. The 'Columns (11)' section is expanded, listing the following fields: gid, id_0, iso, country, id_1, name, type_1, engtype_1, nl_name_1, varname_1, and geom. The 'nl_name_1' field is currently selected and highlighted.

Column
gid
id_0
iso
country
id_1
name
type_1
engtype_1
nl_name_1
varname_1
geom

> Django admin

http://127.0.0.1:8000/admin

Django administration

WELCOME, ADMIN. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups

[+ Add](#) [✎ Change](#)

Users

[+ Add](#) [✎ Change](#)

Django administration

WELCOME, ADMIN. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Authentication and Authorization > Users > admin

Change user

HISTORY

Username:

admin

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Password:

algorithm: pbkdf2_sha256 iterations: 120000 salt: qfm2IB***** hash: crN1eL*****

Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using [this form](#).

Personal info

First name:

Last name:

Email address:

admin@hevs.ch

> admin.py

```
from django.contrib.gis import admin

from .models import City, Hospital, Canton
admin.site.register(City)
admin.site.register(Hospital)
admin.site.register(Canton, admin.ModelAdmin)
```

Django administration

WELCOME, ADMIN. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

[Home](#) > [Swissgeo](#) > [Cantons](#)

Select canton to change

ADD CANTON +

Action:

Go

0 of 51 selected

CANTON

Appenzell Innerrhoden

Appenzell Innerrhoden

Fribourg

Django administration

WELCOME, ADMIN. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

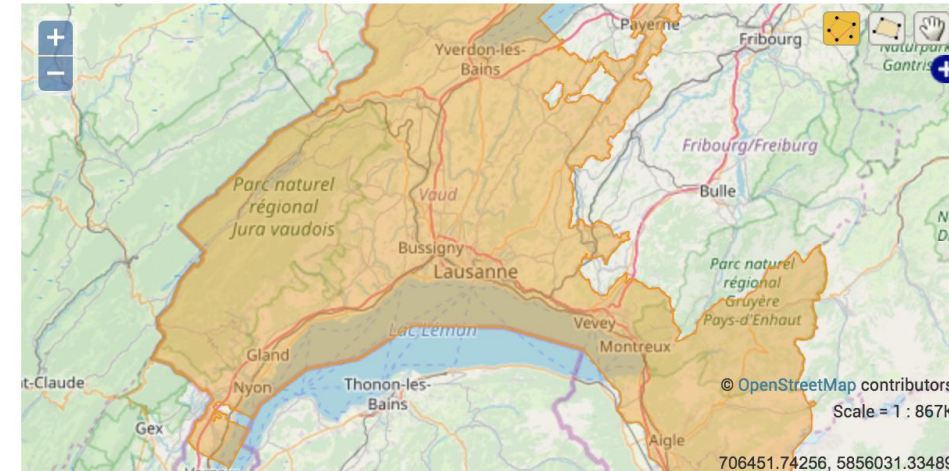
[Home](#) > [Swissgeo](#) > [Cantons](#) > [Vaud](#)

Change canton

Name:

Vaud

Geom:



Delete all Features

Delete

Save and add another

Save and continue editing

> Requests with Geo Models

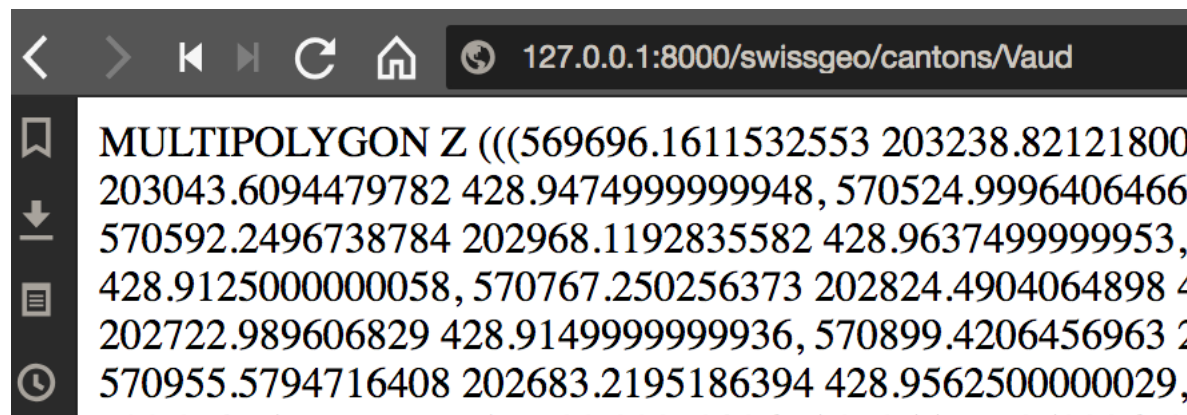
<http://127.0.0.1:8000/swissgeo/cantons/Vaud>

urls.py

```
urlpatterns = [
    path('city/', views.cities, name='cities'),
    path('city/<int:city_id>', views.city, name='city'),
    path('cantons/<str:canton_name>', views.canton, name='canton'),
```

views.py

```
def canton(request, canton_name):
    cantons=Canton.objects.filter(name=canton_name)
    return HttpResponse(cantons[0].geom.wkt)
```



> Templates with Geo Models

views.py

<http://127.0.0.1:8000/swissgeo/cantons/Fribourg>

```
def canton(request, canton_name):  
    cantons=Canton.objects.filter(name=canton_name)  
    return render(request, 'swissgeo/canton.html',  
                  {'cantonobj':cantons[0]})
```

swissgeo/canton.html

```
<ul class="list-group">  
  <li class="list-group-item d-flex justify-content-between align-items-center">  
    <h2>{{cantonobj.name}}</h2>  
    <span class="badge badge-primary badge-pill">Id: {{cantonobj.id}}</span>  
    <span class="badge badge-primary badge-pill">Area: {{cantonobj.geom.area }}</span>  
    <span class="badge badge-primary badge-pill"># Polygons: {{cantonobj.geom.num_geom}}</span>  
  </li>  
</ul>
```



> Returning GeoJSON

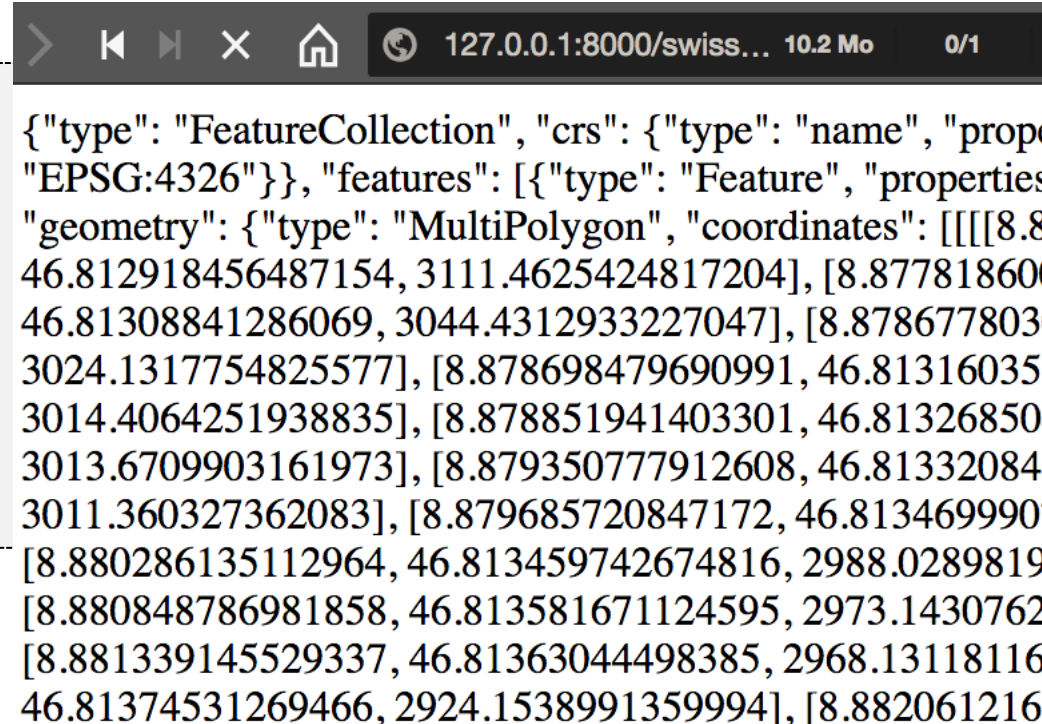
<http://127.0.0.1:8000/swissgeo/cantons.json>

urls.py

```
urlpatterns = [  
    path('cantons/<str:canton_name>', views.canton, name='canton'),  
    path('cantons.json', views.cantonsjson, name='cantonsjson'),
```

views.py

```
from django.core.serializers import serialize  
  
def cantonsjson(request):  
    cantons=Canton.objects.all()  
    ser=serialize('geojson',cantons,  
                 geometry_field='geom',  
                 fields=('name',))  
    return HttpResponse(ser)
```



```
{  
  "type": "FeatureCollection",  
  "crs": {"type": "name", "properties": {"name": "EPSG:4326"}},  
  "features": [{"type": "Feature", "properties": {"name": "Canton of Appenzle Aargau"},  
    "geometry": {"type": "MultiPolygon", "coordinates": [[[[[8.8778186046812918456487154, 3111.4625424817204], [8.877818604681308841286069, 3044.4312933227047], [8.87867780330241317754825577], [8.878698479690991, 46.8131603530144064251938835], [8.878851941403301, 46.8132685030136709903161973], [8.879350777912608, 46.813320843011360327362083], [8.879685720847172, 46.8134699908.880286135112964, 46.813459742674816, 2988.02898198.880848786981858, 46.813581671124595, 2973.14307628.881339145529337, 46.81363044498385, 2968.1311811646.81374531269466, 2924.1538991359994], [8.882061216
```

> Leaflet in GeoDjango

<https://django-leaflet.readthedocs.io>
settings.py

```
INSTALLED_APPS = [  
    'swissgeo.apps.SwissgeoConfig',  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'django.contrib.gis',  
    'leaflet',  
]
```

```
LEAFLET_CONFIG = {  
    'MIN_ZOOM': 3,  
    'MAX_ZOOM': 15,  
    'DEFAULT_ZOOM': 7,  
    'DEFAULT_CENTER': (46.7, 8),  
    'TILES': [ ('OSM',  
    '//tile.openstreetmap.org/{z}/{x}/{y}.png',  
    {'attribution': '© <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a>  
    contributors', 'referrerPolicy': 'strict-origin'}) ]  
}
```

> Leaflet in django templates

urls.py

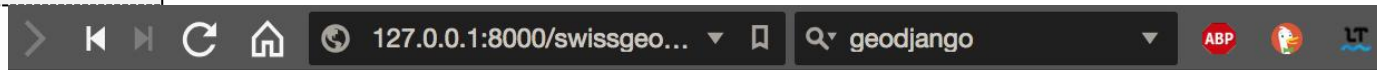
```
urlpatterns = [  
    path('cantons', views.cantons, name='cantons'),
```

views.py

```
def cantons(request):  
    context = {  
    }  
    return render(request,  
        'swissgeo/cantons.html', context)
```

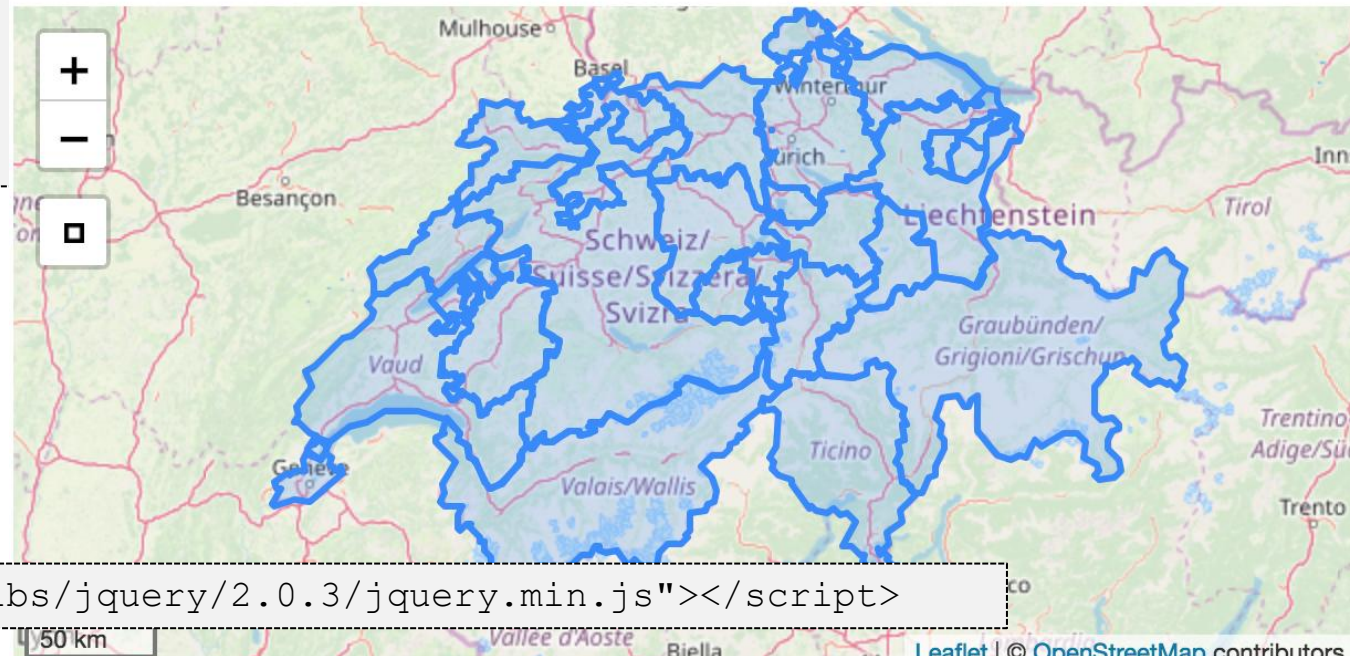
cantons.html

```
{% load leaflet_tags %}  
<head>  
{% leaflet_js %}  
{% leaflet_css %}  
</head>  
  
<body>  
{% leaflet_map "cantons" %}  
</body>
```



> Leaflet in django templates

```
<body>
  {% leaflet_map "cantons" callback="main_map_init" %}
  <script type="text/javascript">
    function main_map_init (map, options) {
      var dataurl = '{% url "cantonsjson" %}'
      $.getJSON(dataurl, function (data) {
        L.geoJson(data).addTo(map); });
    }
  </script>
</body>
```



```
<script src="//cdnjs.cloudflare.com/ajax/libs/jquery/2.0.3/jquery.min.js"></script>
```

> Leaflet in django templates

```
function main_map_init (map, options) {  
  var dataurl = '{% url "cantonsjson" %}'  
  $.getJSON(dataurl, function (data) {  
    L.geoJson(data, {onEachFeature: onEachFeature}).addTo(map);  
  });  
  
  function onEachFeature(feature, layer) {  
    if (feature.properties &&  
        feature.properties.name) {  
      layer.bindPopup(feature.properties.name);  
    }  
  }  
}
```





Site administration | Django site admin

127.0.0.1:8000/admin/

Django administration

WELCOME, ADMIN. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Site administration

AUTHENTICATION AND AUTHORIZATION	
Groups	+ Add ✎ Change
Users	+ Add ✎ Change

SWISSGEO	
Cantons	+ Add ✎ Change
Cities	+ Add ✎ Change
Hospitals	+ Add ✎ Change

Recent actions

My actions

- + Brig
City
- + Martigny
City
- + Sion
City
- ✎ rolo
User
- + rolo
User

Thanks!
Questions?

