

Python Web Applications: Django

Option GIS-Python

hes.
SO
business.

Jean-Paul Calbimonte



School of Management

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

> Before we start ...



install QGIS: useful for visualizing and connecting with PostGIS

<https://www.qgis.org>

> Django

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

What is Django?

<https://www.djangoproject.com>

- high-level Python Web framework
- rapid development and clean, pragmatic design.
- takes care of much of the hassle of Web development
- free and open source.

> Django

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

- Abstraction layer (**models**) for structuring and manipulating the data of the Web application
- Concept of **views** to encapsulate the logic responsible for processing a user's request and for returning the response
- **Template** layer provides a designer-friendly syntax for rendering the information to be presented to the user.
- Rich framework to facilitate the creation of **forms** and the manipulation of form data.
- Automated **admin** interface
- Multiple **protection** tools and mechanisms:
- Robust **internationalization** and localization framework



Django classical installation

> Django installation



```
> conda install -c anaconda django
```

```
> python -m django --version
```

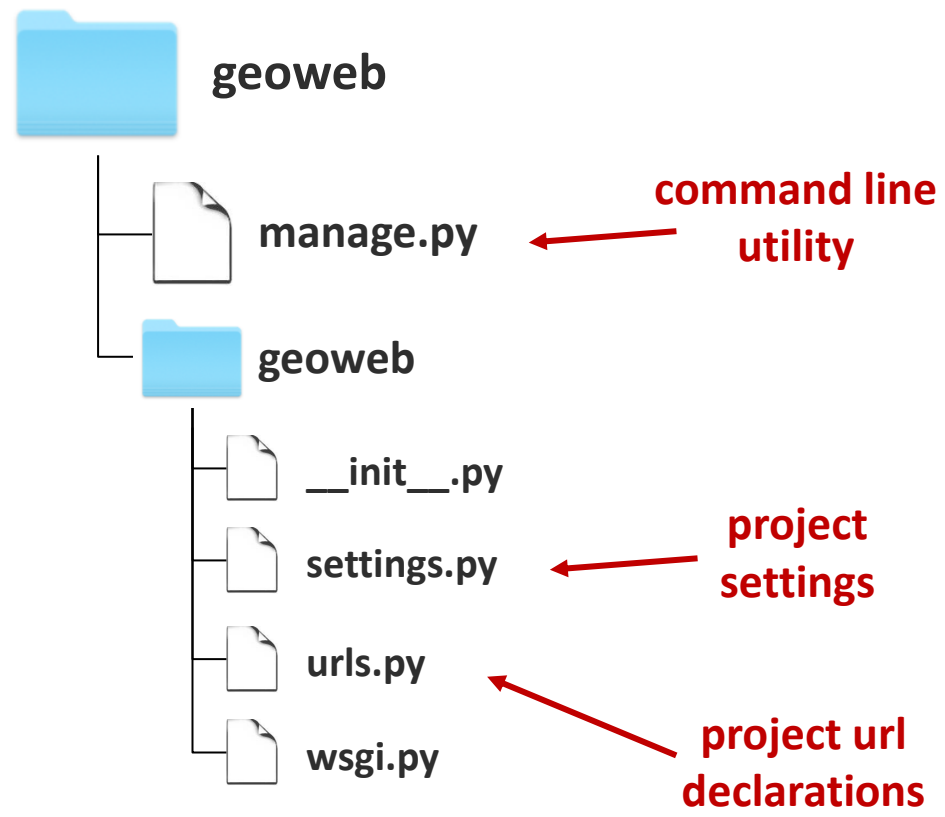
for further info check the docs at:
<https://www.djangoproject.com>

> Create django project

```
> django-admin startproject geoweb
```

project name

generated project structure





Django docker installation

Django with Docker



Dockerfile Python image where we install django

```
FROM python:3
ENV PYTHONUNBUFFERED=1
WORKDIR /code
COPY requirements.txt /code/
RUN pip install -r requirements.txt
RUN apt-get update && \
    apt-get install -y libproj
libfreeexl-dev libgdal-dev gdal
COPY . /code/
```

requirements.txt

```
Django>=3.0,<4.0
psycopg2-binary>=2.8
django-leaflet
```

docker-compose.yml

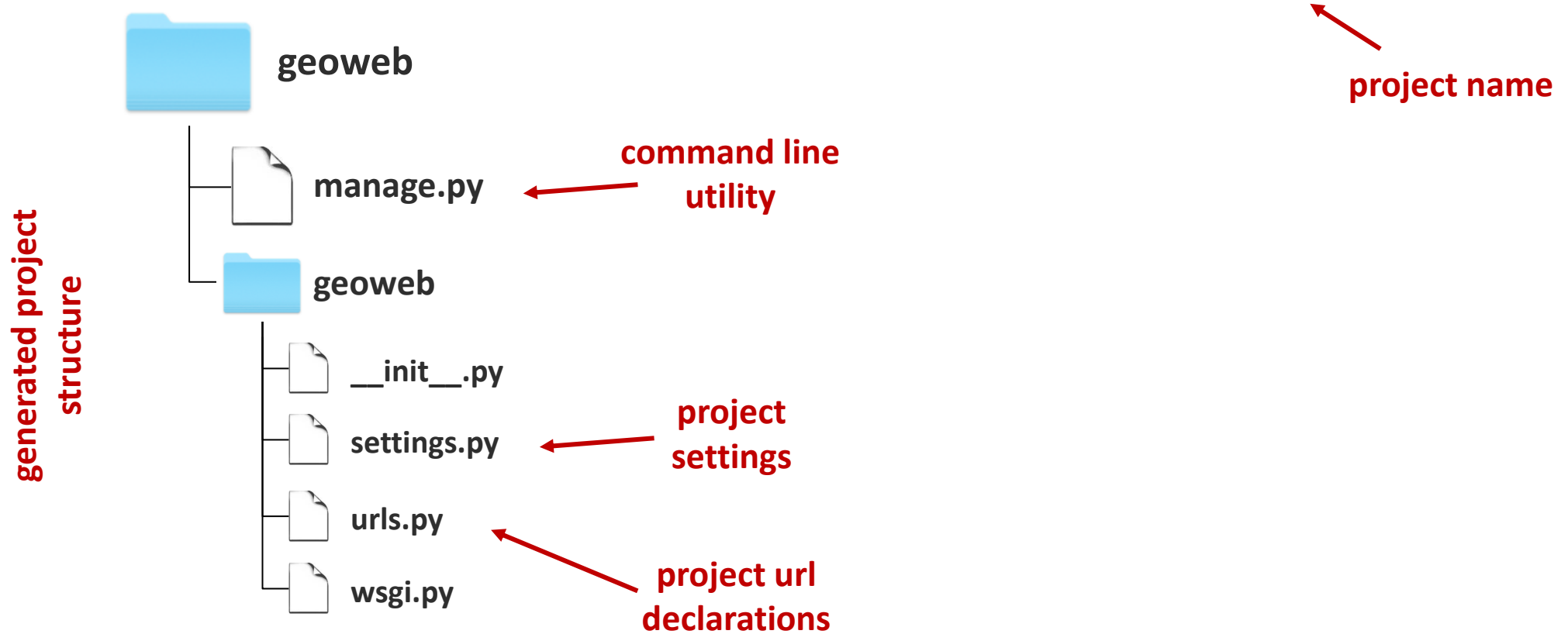
```
version: "3.9"

services:
  db:
    image: postgis/postgis
    volumes:
      - ./data/db:/var/lib/postgresql/data
    environment:
      - POSTGRES_DB=postgres
      - POSTGRES_USER=postgres
      - POSTGRES_PASSWORD=admin
    ports:
      - "5432:5432"
  web:
    build: .
    command: python geoweb/manage.py runserver 0.0.0.0:8000
    volumes:
      - ./code
    ports:
      - "8000:8000"
    depends_on:
      - db
  pgadmin:
    container_name: pgadmin4_container
    image: dpage/pgadmin4
    restart: always
    environment:
      PGADMIN_DEFAULT_EMAIL: admin@admin.com
      PGADMIN_DEFAULT_PASSWORD: root
    ports:
      - "5050:80"
```

> Create django project

Version docker

```
> docker-compose run web django-admin startproject geoweb
```



> settings.py

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

**many applications
per project**

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

**database
configuration**

> Start django server

```
> python manage.py runserver
```

<http://127.0.0.1:8000/>

django

[View release notes for Django 3.2](#)



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

> Start django server in docker

```
> docker-compose up
```

<http://127.0.0.1:8000/>

django

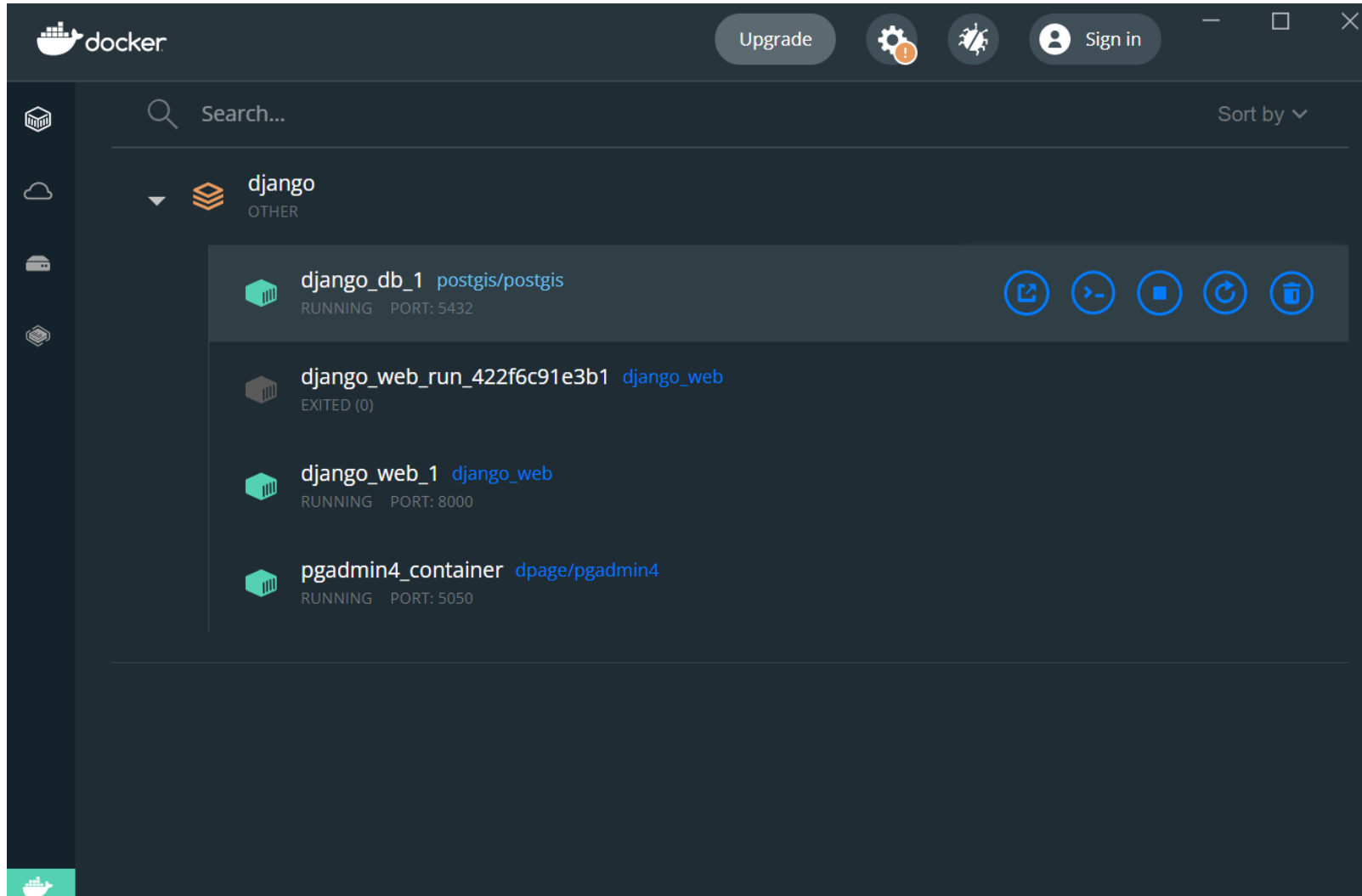
[View release notes for Django 3.2](#)









The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

> Start django server in docker



> Start django server in docker

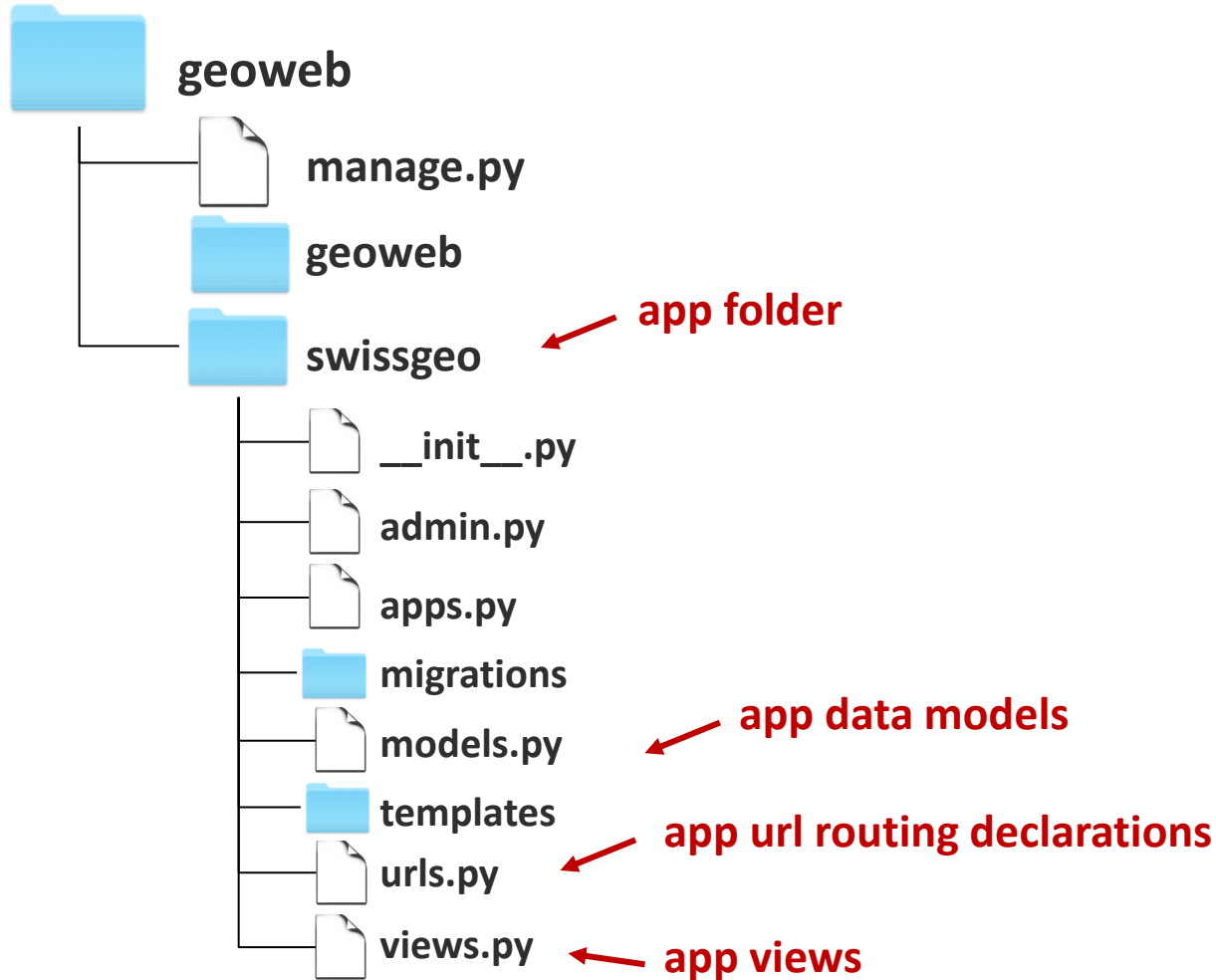
| <input type="checkbox"/> | Name | Image | Status | Port(s) | Last started | Actions |
|--------------------------|---|---------------------------------|---------------|-----------------------------|---------------|---------|
| <input type="checkbox"/> |  django | - | Running (3/5) | | 2 minutes ago | ■ ⋮ 🗑️ |
| <input type="checkbox"/> |  db-1 48975243fe10 📄 | postgis/postgis | Running | 5432:5432 🔗 | 7 minutes ago | ■ ⋮ 🗑️ |
| <input type="checkbox"/> |  django_web_run_dbf6f51f1 e16bdce69daf 📄 | django-web | Exited (1) | | 8 m | ■ ⋮ 🗑️ |
| <input type="checkbox"/> |  django_web_run_b58f5c5f1 6841211af7d8 📄 | django-web | Exited | | 7 m | ■ ⋮ 🗑️ |
| <input type="checkbox"/> |  web-1 d075e6b564fe 📄 | django-web | Running | 8000:8000 🔗 | 2 m | ■ ⋮ 🗑️ |
| <input type="checkbox"/> |  pgadmin4_container 88daec0676ce 📄 | dpage/pgadmin4 | Running | 5050:80 🔗 | 2 m | ■ ⋮ 🗑️ |

- 👁️ View details
- 🛡️ View image packages and CVEs
- 📄 Copy docker run
- 🖥️ Open in terminal
- 📁 View files
- ⏸️ Pause
- 🔄 Restart
- 🔗 Open with browser

> Create django app

app name

```
> python manage.py startapp swissgeo
```



> apps.py

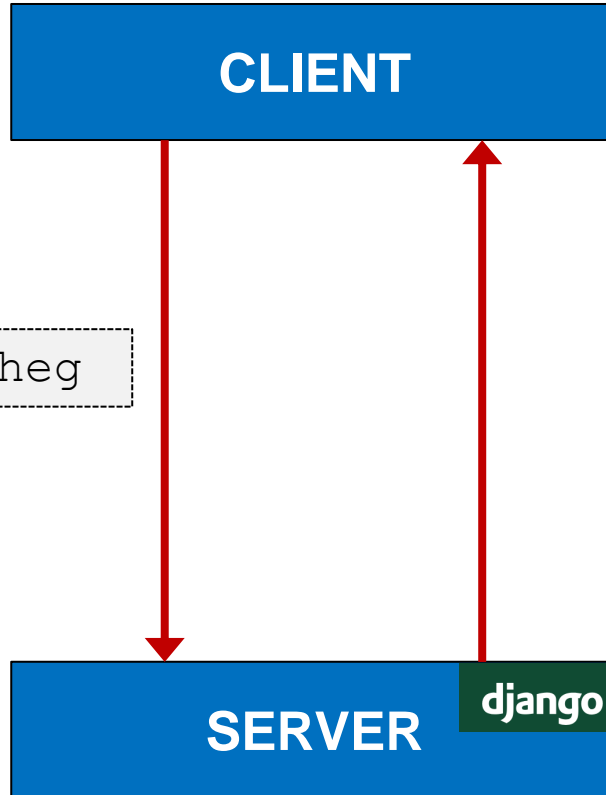
```
class SwissgeoConfig(AppConfig):  
    name = 'swissgeo'
```

app config class

Settings.py

```
INSTALLED_APPS = [  
    'swissgeo.apps.SwissgeoConfig',  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',
```

> HTTP requests & reponses



HTTP Request

```
GET http://www.hevs.ch/heg
```

```
GET /heg HTTP/1.1
Host: www.hevs.ch
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 ...
Accept: text/html,
Accept-Encoding: gzip, deflate
Accept-Language: fr-CH;q=0.8
Cookie: SERVERID=consult8-6
```

HTTP Response

```
HTTP/1.1 302 Found
Server: Apache
Cache-Control: no-cache
Content-Encoding: gzip
Vary: Accept-Encoding
Location: http://www.hevs.ch/fr/hautes-ecoles/
        haute-ecole-de-gestion
Content-Type: text/html; charset=UTF-8
Content-Length: 240
Accept-Ranges: bytes
Date: Thu, 28 Mar 2019 19:44:36 GMT
```

> Views & Urls: HTTP

HTTP Request GET http://127.0.0.1:8000/swissgeo/

geoweb/urls.py

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

urlpatterns = [
    path('swissgeo/', include('swissgeo.urls')),
    path('admin/', admin.site.urls),
]
```

match path

swissgeo/urls.py

```
from django.urls import path
from . import views

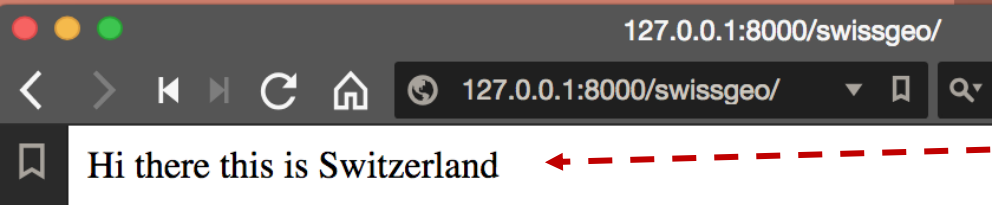
urlpatterns = [
    path('', views.index, name='index'),
]
```

empty path

views.py

```
from django.http import HttpResponse

def index(request):
    -- return HttpResponse("Hi there this is Switzerland")
```



> settings.py

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

**database
connection
settings**

- ▼ Servers (1)
 - ▼ pgapp
 - ▼ Databases (2)
 - > postgres
 - ▼ testGIS
 - > Casts
 - > Catalogs
 - > Event Triggers
 - ▼ Extensions (3)

> models.py

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

```
class City(models.Model):  
    city_name = models.CharField(max_length=100)
```

map to column

map to datatype

model classes map
to db tables

```
class Hospital(models.Model):  
    hospital_name = models.CharField(max_length=100)  
    pub_date = models.DateTimeField('date published')  
    city = models.ForeignKey(City, on_delete=models.CASCADE)
```

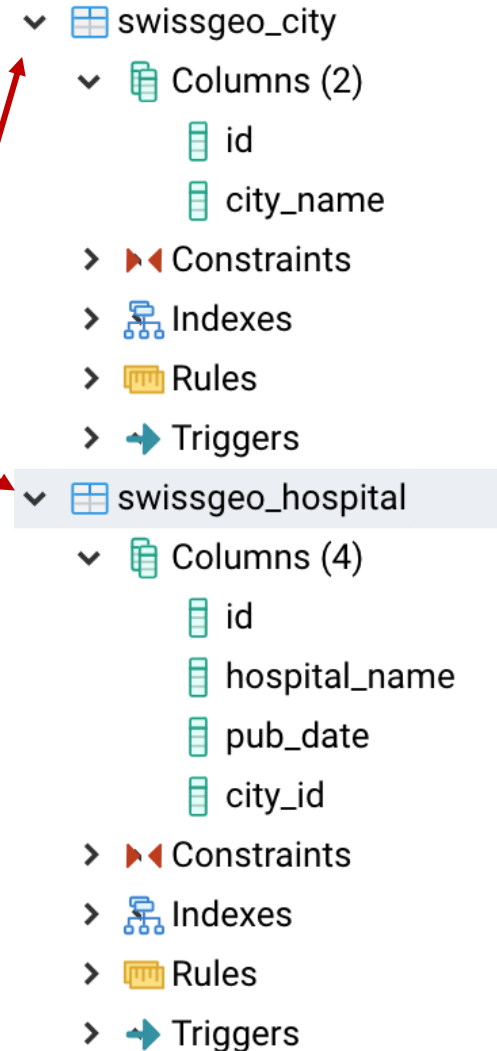
foreign key object

> Make migrations

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

```
> python manage.py migrate
```

Django generates SQL to create/modify tables



> Manipulating data through the API

```
> python manage.py shell
```

```
from swissgeo.models import City
```

```
c = City(city_name="Sierre")
c.save() create a city
```

```
c.id
1
c.city_name
'Sierre'
```

```
City.objects.all()
<QuerySet [<City: Sierre>, <City: Sion>, <City: Martigny>]> query all cities
```

```
City.objects.filter(id=1)
<QuerySet [<City: Sierre>]> filter
```

```
City.objects.filter(city_name__startswith='S')
<QuerySet [<City: Sierre>, <City: Sion>]>
```

public.swissgeo_city/testGIS/postgres@pgapp

Query Editor Query History

```
1 SELECT * FROM public.swissgeo_city
2 ORDER BY ID DESC LIMIT 100
3
```

Data Output Explain Messages Notifications

| | id [PK] integer | city_name character varying (100) |
|---|--------------------|--------------------------------------|
| 1 | 3 | Martigny |
| 2 | 2 | Sion |
| 3 | 1 | Sierre |

> models.py

```
class City(models.Model):  
    city_name = models.CharField(max_length=100)  
  
    class Meta:  
        verbose_name_plural = "cities"  
  
    def __str__(self):  
        return self.city_name
```


> Django admin

```
> python manage.py createsuperuser  
Username: admin  
Email address: ...
```

create admin user

**django user/group/permission
tables**

- Tables (15)
 - auth_group
 - auth_group_permissions
 - auth_permission
 - auth_user
 - Columns (11)
 - id
 - password
 - last_login
 - is_superuser
 - username
 - first_name
 - last_name
 - email
 - is_staff
 - is_active
 - date_joined
 - Constraints
 - Indexes
 - Rules
 - Triggers
 - auth_user_groups
 - auth_user_user_permissions

- django_admin_log
- django_content_type
- django_migrations
- django_session
 - Columns (3)
 - Constraints
 - Indexes
 - Rules
 - Triggers

**django
admin/log/migration
tables**

> 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 import admin
from .models import City
admin.site.register(City)
```

also manage app models

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

Home > Swissgeo > Cities

Select city to change

ADD CITY +

Action:

0 of 3 selected

- CITY
- Martigny
- Sion
- Sierre

3 cities

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

Home > Swissgeo > Cities > Martigny

Change city

HISTORY

City name:

SAVE

Save and add another

Save and continue editing

> Views using Models

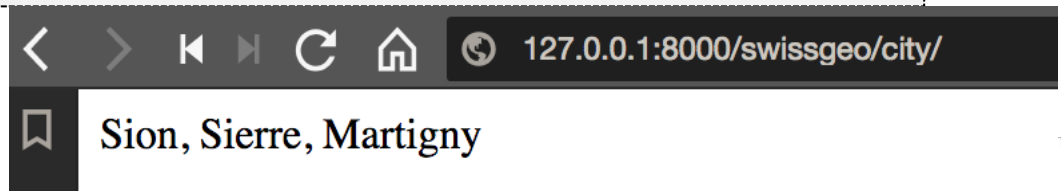
`http://127.0.0.1:8000/swissgeo/city/`

urls.py

```
urlpatterns = [  
    path('', views.index, name='index'),  
    path('city/', views.cities, name='cities'),  
]
```

views.py

```
from .models import City  
  
def cities(request):  
    top_cities=City.objects.order_by('-city_name')[:3]  
    output = ', '.join([c.city_name for c in top_cities])  
    return HttpResponse(output)
```



> Django Templates

views.py

```
from django.template import loader

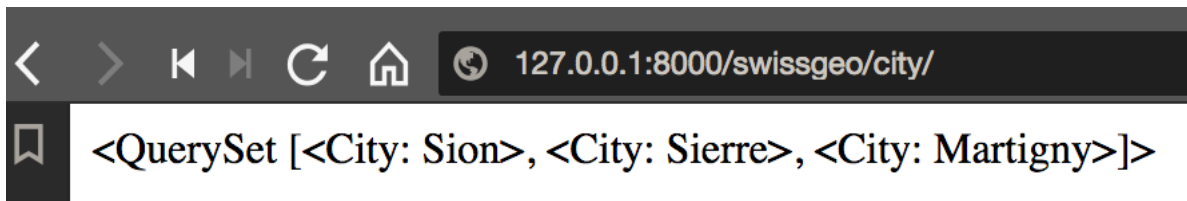
def cities(request):
    top_cities=City.objects.order_by('-city_name')[:3]
    template= loader.get_template('swissgeo/cities.html')
    context = { 'top_cities':top_cities, }
    return HttpResponse(template.render(context,request))
```

← query 3 cities

← pass objects to template

templates/swissgeo/cities.html

```
{{top_cities}}
```



> Django Templates

cities.html

```
{{ top_cities }}
```



```
<ul>
  {% for city in top_cities %}
    <li>
      <a href="/swissgeo/city/{{city.id}}">
        {{city.city_name}}
      </a>
    </li>
  {% endfor %}
</ul>
```



> Views: shortcuts

```
def cities(request):  
    top_cities=City.objects.order_by('-city_name')[:3]  
    template= loader.get_template('swissgeo/cities.html')  
    context = { 'top_cities':top_cities, }  
    return HttpResponse(template.render(context,request))
```



```
from django.shortcuts import render  
  
def cities(request):  
    top_cities=City.objects.order_by('-city_name')[:3]  
    context = { 'top_cities':top_cities, }  
    return render(request, 'swissgeo/cities.html', context)
```

> Display city

```
urlpatterns = [  
    path('', views.index, name='index'),  
    path('city/', views.cities, name='cities'),  
    path('city/<int:city_id>', views.city, name='city'), ]
```

```
def city(request, city_id):  
    city=City.objects.get(pk=city_id)  
    return render(request, 'swissgeo/city.html', {'city':city})
```

templates/swissgeo/city.html

```
{{city}}
```


> Exceptions

```
urlpatterns = [
    path('', views.index, name='index'),
    path('city/', views.cities, name='cities'),
    path('city/<int:city_id>', views.city, name='city'), ]
```

```
from django.http import Http404

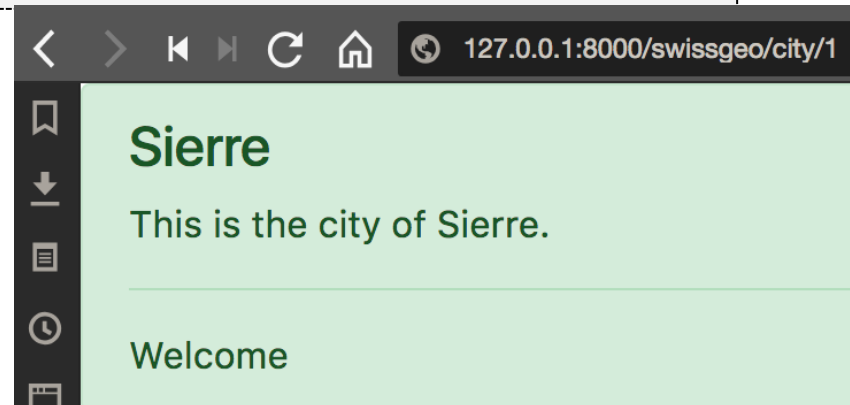
def city(request, city_id):
    try:
        city=City.objects.get(pk=city_id)
    except City.DoesNotExist:
        raise Http404("City not found!!")
    return render(request, 'swissgeo/city.html', {'city':city})
```

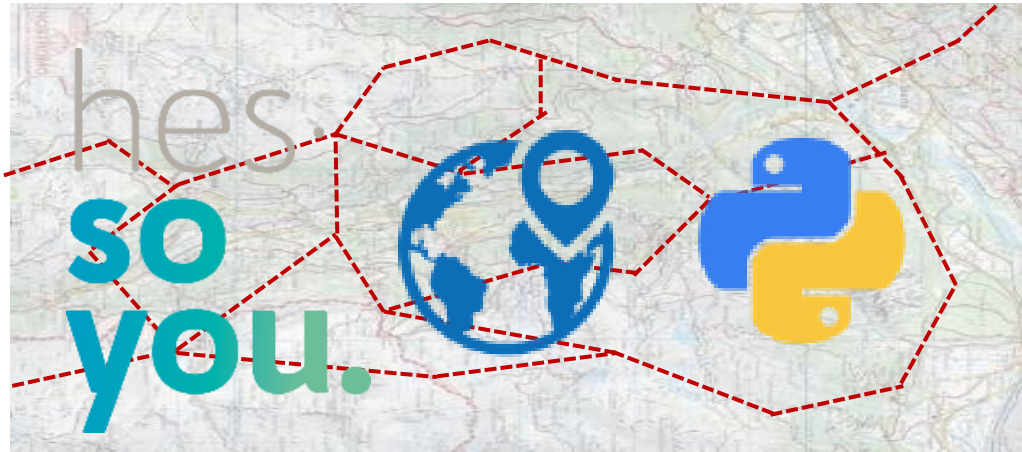


> Using styles

```
<head>
  <link rel="stylesheet"
    href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
    integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
    crossorigin="anonymous">
</head>
```

```
<div class="alert alert-success" role="alert">
  <h4 class="alert-heading">{{city}}</h4>
  <p>This is the city of {{city}}.</p> <hr>
  <p class="mb-0">Welcome</p>
</div>
```





School of Management
Route de la Plaine 2
3960 Sierre

hevs.ch/heg



Thank you for your attention.

swissuniversities

