**Manuel de mise en place du plateforme Django :**

**EXEMPLE GEODJERBA**

1. **Installation de PostgreSQL :**

La version choisit pour ce projet c’est **PgAdmin4**

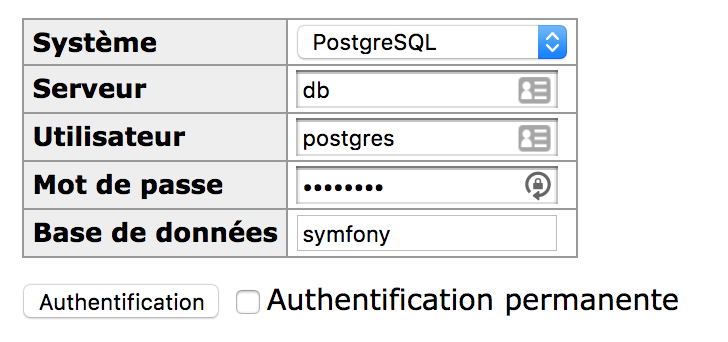


On installe aussi **Postgis** en tant que cartouche spatiale de PostgreSQL



Ensuite, en crée une base de données et on import nos couches vectorielles via Postgis.

Il faut mémoriser les propriétés de la base de données, exemple :



1. **Installation de Geoserver :**

La version choisit pour ce projet c’est [Platform Independent Binary](https://sourceforge.net/projects/geoserver/files/GeoServer/2.20.3/geoserver-2.20.3-bin.zip) 2.20.2

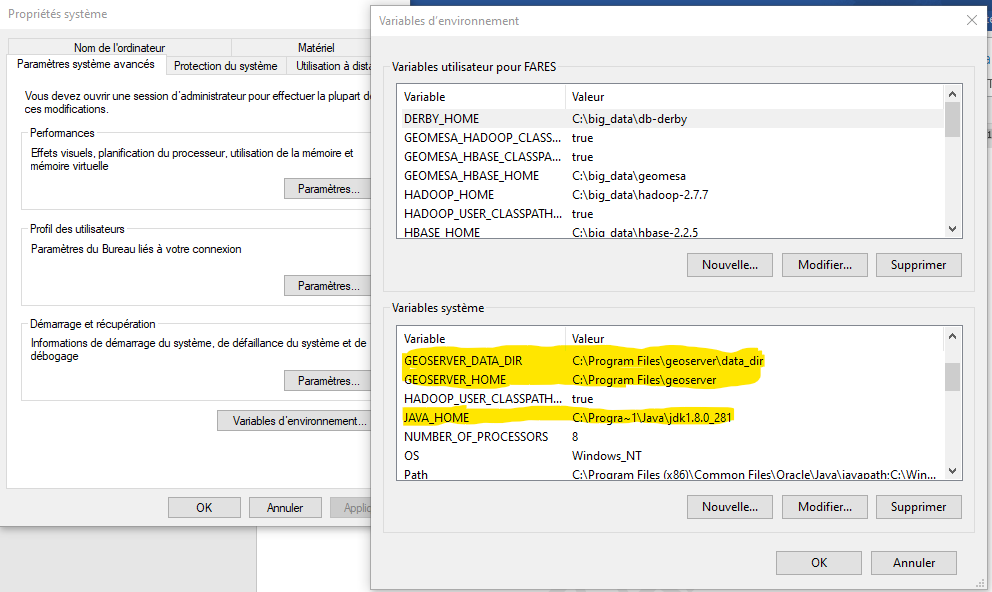


Pour l’installer correctement, il faut suivre les étapes suivantes :

* Télécharger [Platform Independent Binary](https://sourceforge.net/projects/geoserver/files/GeoServer/2.20.3/geoserver-2.20.3-bin.zip) 2.20.2 et fait extraire le dossier

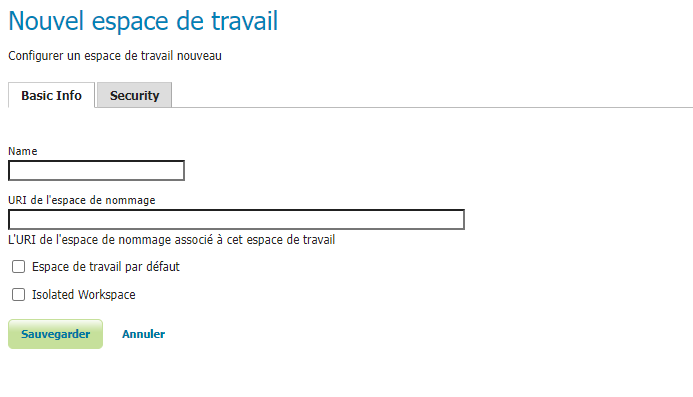
sous **C:/geoserver**

* Installer java JDK 8
* Ajouter GEOSERVER\_HOME, GEOSERVER\_DATA\_DIR, et JAVA\_HOME au variables d’environnement

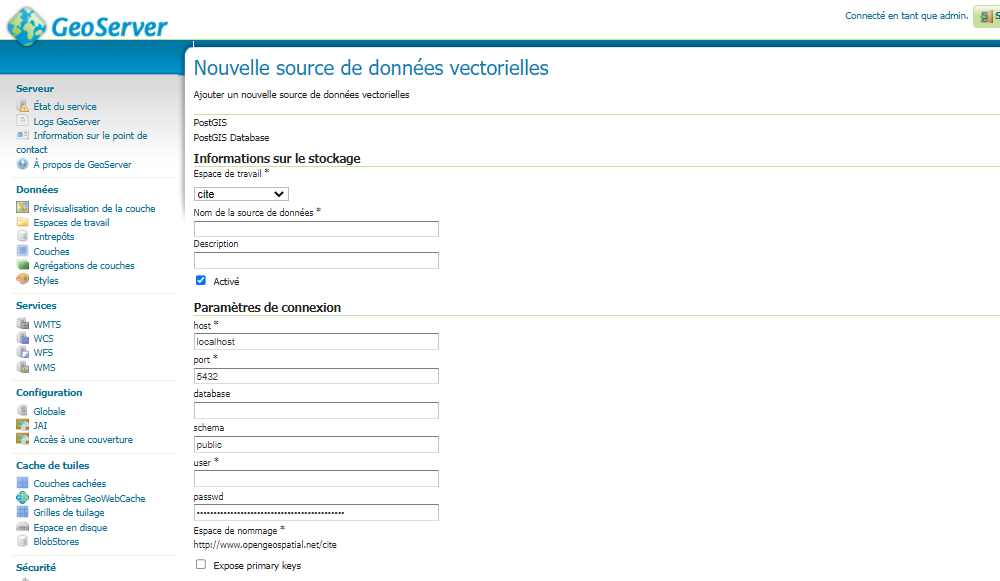


Une fois Geoserver est mise en place , l’ouvre via l’url <http://localhost:8080/geoserver/web/> : login= admin , password = geoserver

En commerce par créer un espace de travail



Puis on crée un entrepôt de travail connecter à postgis

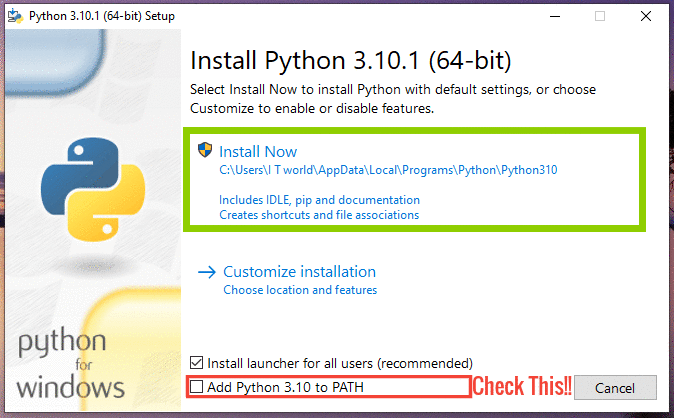


1. **Mise en place du serveur web Django :**
   1. **Installation de python :**

La version installée c’est **python 3.10**.

Télécharger l’exécutable via le l’url <https://www.python.org/downloads/>

NB : assurer que le chemin de python soit ajouté au path dans la configuration de variable d’environnement



* 1. **Création d’un projet Django :**

Il faut assurer que la connexion internet est bonne.

Maintenant, on ouvre l’invite de commande CMD et on tape les commandes suivantes :

cd c:/

git clone <https://github.com/iamtekson/geodjango-app.git>

cd geodjango-app

py –m venv VIRTUAL\_ENV

C:\geodjango-app\VIRTUAL\_ENV\Scripts\activate.bat

pip install pipwin

pipwin refresh

pipwin install gdal

pip install psycopg2-binary

pip install -r requirements.txt

Maintenant, on ouvre le fichier geoProject/settings.py et on modifier les paramètres suivants :

DATABASES = {

'default': {

'ENGINE': 'django.contrib.gis.db.backends.postgis',

'NAME': 'geoapp', #c’est le nom de la base de données

'USER': 'postgres',

'PASSWORD': 'admin',

'HOST': 'localhost',

'PORT': '5432',

}

}

Les paramètres doivent correspondre à la base de données créer sous PostgreSQL.

On retourne maintenant au cmd est on tape :

python manage.py makemigrations

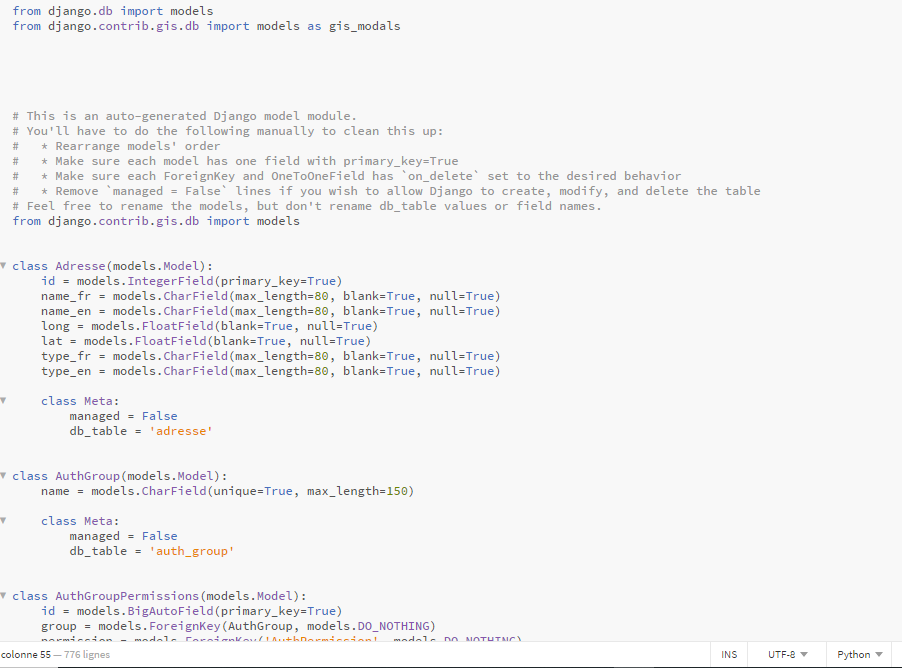
python manage.py migrate

python manage.py collectstatic

* 1. **Importation des données de PostgreSQL vers Django :**

cd C:\geodjango-app\geoApp

python manage.py inspectdb > models.py



Maintenant, on ouvre le fichier geoApp/admin.py et on ajoute les lignes suivantes :

from django.contrib import admin

from leaflet.admin import LeafletGeoAdmin

from .models import Banque

from .models import BoxChange

.

.

.

. #on import tous les modèles déclarés

admin.site.register(BoxChange, LeafletGeoAdmin)

admin.site.register(CentresCommercials, LeafletGeoAdmin)

admin.site.register(Cofe, LeafletGeoAdmin)

.

.

.

On modifie geoProject/settings.py :

On ajoute ces lignes :

STATIC\_URL = '/static/'

STATIC\_ROOT = os.path.join(BASE\_DIR, 'static')

STATICFILES\_DIRS = [os.path.join(BASE\_DIR, 'geoProject/static/')]

Et on modifie la configuration de leaflet :

LEAFLET\_CONFIG = {

'DEFAULT\_CENTER': (33.80985,10.990),

'DEFAULT\_ZOOM': 16,

'MIN\_ZOOM': 3,

'MAX\_ZOOM': 20,

'SCALE': 'both',

'ATTRIBUTION\_PRIFIX': 'tekson',

'RESET\_VIEW': False,

'TILES': [

('Dark Map', 'https://{s}.basemaps.cartocdn.com/light\_all/{z}/{x}/{y}{r}.png', {

'attribution': '&copy; CNCT 2022 ,Contributors : &copy; <a href="https://carto.com/attributions">CARTO</a>',

'subdomains': 'abcd',

'maxZoom': 19

})],

}

* 1. **Installation de plugin django-admin-interface :**

pip install django-admin-interface

Maintenant on modifie geoProject/settings.py :

INSTALLED\_APPS = (

#...

"admin\_interface",

"colorfield",

#...

"django.contrib.admin",

#...

)

X\_FRAME\_OPTIONS = "SAMEORIGIN"

SILENCED\_SYSTEM\_CHECKS = ["security.W019"]

Puis on exécute les commandes suivantes :

python manage.py migrate

python manage.py collectstatic

* 1. **Lancement de l’application :**

On ouvre l’invite de commandes et on tape :

ipconfig

Cette commande va renvoyer l’adresse IPv4 de la machine exemple **192.172.5.77**

Mainetenant on ouvre les fichier html de templates et on remplace tous les adresses IP (localhost ou autres) par l’adresse **192.172.5.77**

Maintenant on modifie geoProject/settings.py :

* Dans ALLOWED\_HOSTS , on ajouter l’adresse **192.172.5.77**

On lance l’invite de commandes CMD :

cd c:/geodjango-app

C:\geodjango-app\VIRTUAL\_ENV\Scripts\activate.bat

python manage.py runserver **192.172.5.77**:8000

A ce stade, on va avoir une interface prête à être exploitable pour faire la gestion, manipulation et mis à jour de la base de données.

