



GUIDE D'INSTALLATION ET DE MAINTENANCE

Projet GoGoGo

Résumé

Ce document sert de guide pour installer et maintenir le projet GoGoGo.

John GAUDRY, Hugo MRULA, Matthieu THIVARD, Samuel MONTAGNA

S5B1

Table des Matières

Prérequis	3
GNU / Linux.....	3
Technologies utilisées	4
Python	4
Django	4
Redis.....	4
HTMX	4
Installation	5
Python	5
Modules Python	5
Redis.....	5
Configuration / Maintenance	6
Base de données	6
Redis.....	8
Lancer le serveur.....	9
Redis.....	9
Serveur	9

Prérequis

GNU / Linux

Afin de faciliter l'installation, il est préférable de se trouver sur une distribution Linux en raison de Redis qui n'est pas disponibles sur d'autres OS. Les commandes d'installation données seront donc celles requises pour Ubuntu 20.04.

Technologies utilisées

Python

Utilisé comme langage pour le backend.

Django

Utilisé comme Framework pour le backend.

Redis

Utilisé comme base de données RAM pour la gestion des parties, pour une faible latence.

HTMX

Utilitaire de développement pour le frontend qui permet de réduire le JavaScript utilisé.

Installation

Python

Pour lancer le serveur, vous aurez de Python version **3.11** ou supérieur.

Installer python 3.11

```
sudo apt-get install python3.11
```

Installer les packages par défaut

```
sudo apt install python3.11-distutils
```

Installer pip

```
curl -sS https://bootstrap.pypa.io/get-pip.py | python3.11
```

Modules Python

Tout d'abord, merci de vérifier que `django-channels` n'est pas installé, car il rentre en conflit avec le module `channels`.

Placez-vous à la racine de votre projet et installez les dépendances :

```
python3.11 -m pip install -r requirements.txt
```

Redis

Pour jouer une partie, il faut des « rooms » et donc une base de données RAM. Ici nous utiliserons Redis :

```
sudo apt-get install redis
```

Configuration / Maintenance

Le fichier *GoGoGo/settings.py* contient tous les paramètres du serveur. Par conséquent, toutes les bouts de code de cette section sont dans celui-ci.

Voici le lien de la documentation officielle, pour la documentation : <https://docs.djangoproject.com/fr/4.2/>

Email

Le système d'email utilise le protocole SMTP pour fonctionner. Vous pouvez modifier les informations comme le protocole et l'adresse utilisée ici :

```
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_PORT = 587 # Port for secure SMTP (587 for TLS, 465 for SSL)
EMAIL_USE_TLS = True
EMAIL_USE_SSL = False
EMAIL_HOST_USER = 'gogogo.noreply@gmail.com'
EMAIL_HOST_PASSWORD = 'yoyj ekdq kxrn uiwk '
```

La documentation officielle explique très bien comment modifier cette partie :

<https://docs.djangoproject.com/fr/4.2/ref/settings/#email-backend>

Base de données

La base de données par défaut est SQLite. Cependant, si vous souhaitez changer celle-ci, vous pouvez trouver cette partie du fichier.

```
# Database
# https://docs.djangoproject.com/en/4.2/ref/settings/#databases

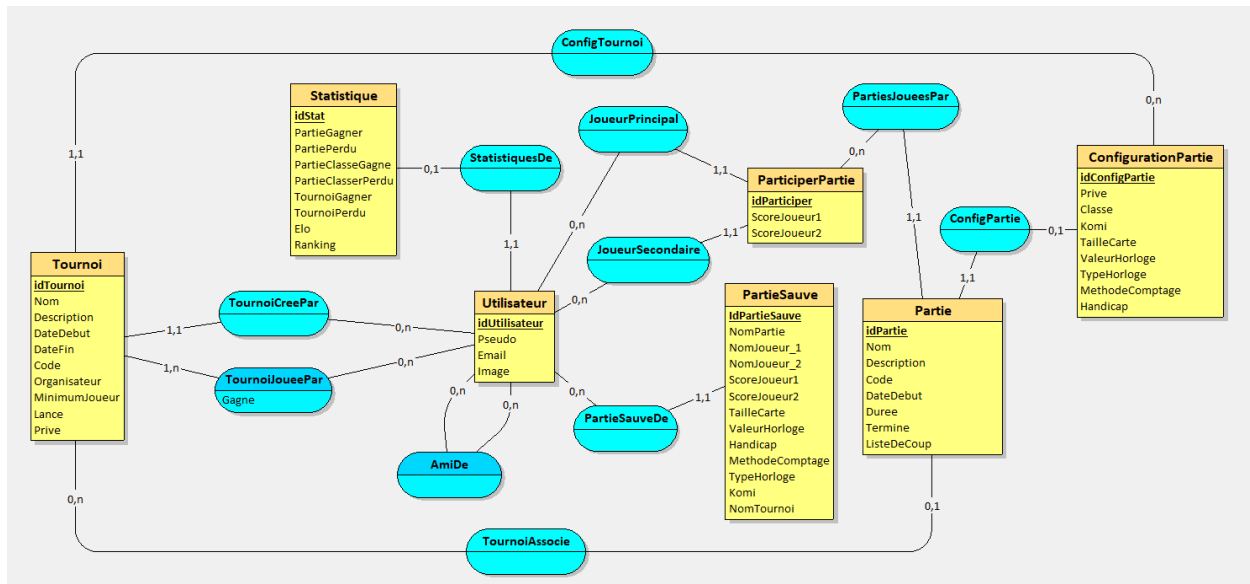
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}
```

La documentation officielle explique très bien comment modifier cette partie :

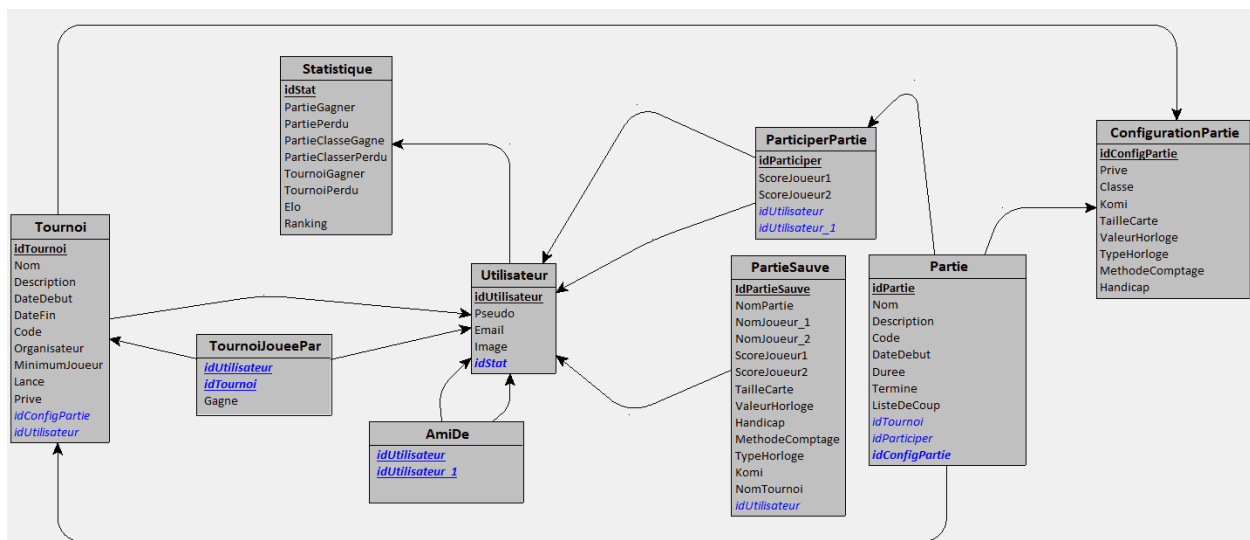
<https://docs.djangoproject.com/fr/4.2/ref/settings/#databases>

Voici le MCD et le MLD de l'application pour faciliter la modification de la base de données.

MCD



MLD



Redis

La base de données Redis utilisée est celle présente en local. Cependant, si vous souhaitez qu'elle soit distante, vous pouvez modifier cette partie du fichier.

```
CHANNEL_LAYERS = {  
    "default": {  
        "BACKEND": "channels_redis.core.RedisChannelLayer",  
        "CONFIG": {  
            "hosts": [("127.0.0.1", 6379)],  
        },  
    },  
}
```

La documentation officielle explique très bien comment modifier cette partie :

https://channels.readthedocs.io/en/stable/topics/channel_layers.html

Lancer le serveur

Pour lancer le serveur, nous devons démarrer certains services. Placez vous à la racine du projet et exécutez les commandes suivantes.

Redis

```
sudo service redis-server start
```

Serveur

```
python3.11 manage.py runserver
```