

# TP3 Python Django – Modèles et base de données

Librement tiré de la documentation officielle de Django version 3.1 :

<https://docs.djangoproject.com/fr/3.1/>

L3 - IFNTI Sokodé

19 Novembre 2020

Dans ce TP, nous allons configurer la base de données et créer les premiers modèles de notre application Notes.

## 1 Mise en route

1. Installez-vous confortablement, et dirigez-vous vers le dossier où vous aviez déjà travaillé lors du TP précédent. Rappelez la commande pour activer l'environnement virtuel, activez-le et donnez les versions de Python et de Django utilisées.
2. Rappelez l'arborescence de votre projet. Où se trouve l'application Notes ? Pourquoi n'est-elle pas dans le dossier `ifnti_13` ? Qu'est-ce que cela changerait si elle y était ?

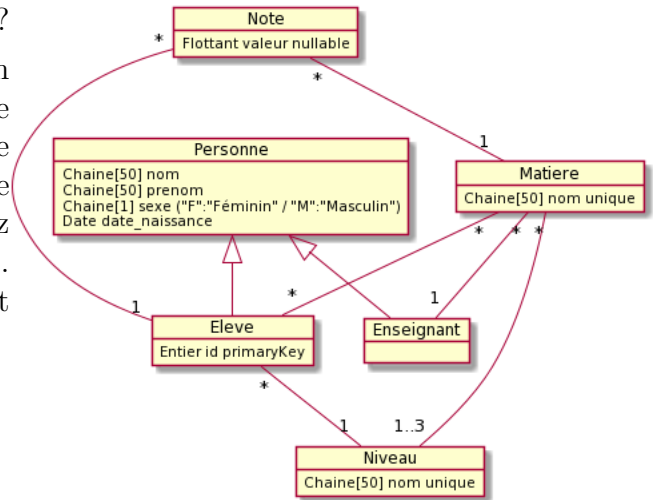
## 2 Configuration de la base de données

1. Ouvrez le fichier `ifnti_13/settings.py`. Trouvez dans ce fichier l'endroit où est configurée la base de données. Quelle moteur de base de données est utilisé par défaut ? Quelles sont les caractéristiques des bases de données de ce type ? L'utiliseriez-vous en production et pourquoi ? Dans le cadre de ce cours, nous garderons ce choix par défaut, car nous n'aurons pas de modèle de données très complexes. Renseignez-vous tout de même sur comment mettre en place d'autres moteurs comme MySQL, PostgreSQL et Oracle.
2. Dans le même fichier, trouvez la constante `TIME_ZONE` et définissez-la selon le fuseau horaire de Lomé (Sur Moodle se trouve la liste des fuseaux horaires avec un format accepté par Django). Quelle valeur avez-vous rentrée ?
3. Toujours dans `settings.py`, trouvez la constante `INSTALLED_APPS`. Quel est le type de cette constante ? Que contient-elle pour le moment ? Nous n'allons pas rentrer dans les détails de chacune des valeurs qu'elle contient. Ce sont des applications contenues dans Django par défaut, nous allons nous intéresser à certaines d'entre elles plus tard dans le semestre.
4. Vous pouvez enregistrer et fermer le fichier `settings.py`. Rappelez la commande pour lancer le serveur et lancez-le. Nous allons maintenant nous intéresser à ce qui s'écrit en rouge lorsque nous lançons le serveur. Vous pouvez remarquer quelques similarités avec ce que contient `INSTALLED_APPS`. Qu'est-ce que le texte en rouge signifie ? Que faire pour régler le problème ? Faites-le. Vous devriez voir une suite de lignes terminées par un "OK" écrit en vert. Si vous relancez le serveur, vous verrez que le texte en rouge n'apparaît plus.

### 3 Création des modèles

Maintenant que notre site fonctionne et que la base de données est créée, nous allons pouvoir commencer de manière plus concrète la construction de notre application Notes. Nous allons tout d'abord créer les modèles Niveau, Élève, Matière et Note.

1. Qu'est-ce qu'un modèle ?
2. Ouvrez le fichier `notes/models.py`. Que contient-il ? Que signifie ce qu'il contient ?
3. À l'aide de la documentation Django sur les instances de modèle ([ref/models/instances.html](http://ref/models/instances.html)) et de celle sur les champs de modèle ([ref/models/fields.html](http://ref/models/fields.html)), construisez les 4 classes du diagramme ci-contre. Rendez le modèle **Personne** abstrait et faites-bien attention à tous les détails :



- (a) Un seul champ est **nullable**.
- (b) 2 champs sont **unique**.
- (c) Un champ est **primaryKey**.
- (d) Le champ **sexe** de **Personne** est "F" ("Féminin") ou "M" ("Masculin").

Qu'avez-vous écrit ? Recopiez le contenu du fichier `models.py`.

Rappel relations :      `**` = ManyToMany      `*-1` = ManyToOne (indice : clé étrangère)

### 4 Activation des modèles

Les modèles précédemment créés ne sont pas encore utilisables. Pour pouvoir les utiliser, il faut les activer, c'est-à-dire créer les tables correspondantes en base de données. Heureusement, Django sait faire ça tout seul. Seulement il faut lui dire que l'on vient de créer les modèles. Et avant cela, il faut aussi lui dire que nous avons créé l'application `notes`. Car le fait d'y avoir accès via l'URL ne suffit pas ! Pour l'instant, Django sait juste qu'un chemin a été ajouté à l'URL.

1. Ré-ouvrez le fichier `ifnti_13/settings.py` et retrouvez la constante `INSTALLED_APPS`. Ajoutez `"notes.apps.NotesConfig"` à la liste des applications installées. Que contient maintenant la constante `INSTALLED_APPS` ? Recopiez son contenu. Qu'est-ce que `NotesConfig` ? Où se trouve-t-il ?
2. Maintenant, Django sait qu'il doit inclure l'application `notes`. Nous pouvons lui dire que les modèles ont été créés. Lancez la commande permettant de construire les migrations. Quelle est cette commande et que fait-elle ? Écrivez ce que vous voyez dans le terminal et expliquez. Si vous rencontrez une erreur sur les modèles ou les champs, c'est sûrement que vous les aviez mal créés. Corrigez ces erreurs dans le fichier correspondant.
3. Lancez la commande : `[alias] manage.py sqlmigrate notes 0001`  
Que voyez-vous ? C'est ce que Django fait grâce aux modèles que nous lui avons donné.
4. Relancez le serveur. Du texte en rouge apparaît de nouveau. Faites en sorte qu'il n'apparaisse plus. Quelle commande avez-vous lancé ? Que signifie le texte qui s'affiche dans le terminal ?

Dans le prochain TP, nous nous intéresserons au site d'administration de Django automatiquement créé et déjà accessible. Comme avant goût, lancez le serveur et dirigez-vous à l'adresse :

`http://127.0.0.1:8000/admin`      N'oubliez pas de désactiver l'environnement virtuel.