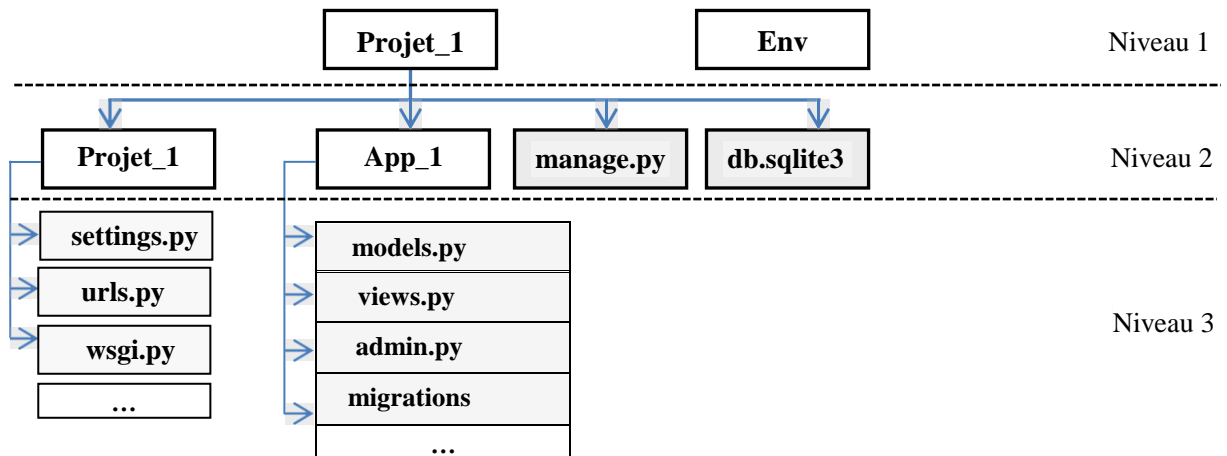




## TP N°02 : L'environnement Django-Python, la première application.

**Objectif :** Le but de ce TP est d'apprendre à créer votre première application Django, définir votre premier modèle et l'afficher dans votre espace d'administration. Les deux fichiers que nous allons manipuler sont : *models.py* et *admin.py*.

**Architecture initiale d'un projet Django avec une seule application:**



- *manage.py* : automatiquement créé dans chaque projet Django pour permettre l'exécution de tâches administratives.
- *db.sqlite3* : un fichier de base de données qui conservera toutes les données que vous allez générer.
- *settings.py* : un fichier de base qui regroupe les règles générales d'un projet Django.
- *urls.py* : regroupe tous les chemins et les routes « urls » utilisés dans notre projet.
- *wsgi.py* : un fichier utilisé pour gérer les réglages de serveur Web pour déployer votre projet (tout à la fin).
- *models.py* : un fichier qui décrit la structure de l'application qui sera créée dans la base de données.
- *views.py* : un ensemble de fonctions/classes Python qui reçoivent une requête Web et renvoient une réponse Web.
- *admin.py* : utilisé pour afficher vos modèles dans le panneau d'administration (configuration de l'espace admin).
- *migrations* : un dossier regroupe des fichiers contenant des instructions modifiant la structure de la base de données.

**Création de votre premier modèle:** Un modèle est une classe qui décrit une structure de données qui sera ensuite stockée dans une BDD. Généralement chaque modèle correspond à une table.

**Exemple :** Considérant l'entité « Product » suivante :

Product
ID : integer Name : char Price : float
Insert() ; Update() ; Delete() ;

models.py

admin.py

```

from django.db import models
# Create your models here.
class Product(models.Model):
    name= models.CharField(max_length=50)
    price= models.FloatField(max_length=10)
  
```

```

from django.contrib import admin
from .models import Product
# Register your models here.
admin.site.register(Product)
  
```

**Ps :** Vous devrez ensuite exécuter les commandes de migrations.

Lorsqu'un modèle est créé, Django crée automatiquement quatre autorisations par défaut pour les actions suivantes :

- **Add** (equiv : **Insert**) : les utilisateurs disposant de cette autorisation peuvent ajouter une instance du modèle.
- **Delete** : les utilisateurs avec cette autorisation peuvent supprimer une instance du modèle.
- **Change** (equiv : **Update**) : les utilisateurs avec cette autorisation peuvent mettre à jour une instance du modèle.
- **View** : les utilisateurs avec cette autorisation peuvent afficher les instances de ce modèle.

➤ Comme vous le voyez, les trois méthodes de notre entité décrite ci-dessus sont créées automatiquement.