**DJANGO FRAMEWORK**

**CRIER UN NOUVEAU PROJET**

* Installer python (exemple python 3.9)
* Crier un environnement **venv dans un répertoire (mon\_repertoire)**

**Python –m venv venv**

* Activer l’environnement **venv**

**source venv/Scripts/activate**

* Installer **Django** dans l’environnement **venv**

**pip install Django**

* Créer le nouveau projet (exemple monprojet) dans (**mon\_repertoire**)

**django-admin startproject monprojet**

* Créer l’application qui sera l’administration django dans (**mon\_repertoire**)

**Django-admin startapp monapp**

**DEMARRER LE PROJET**

* python manage.py runserver

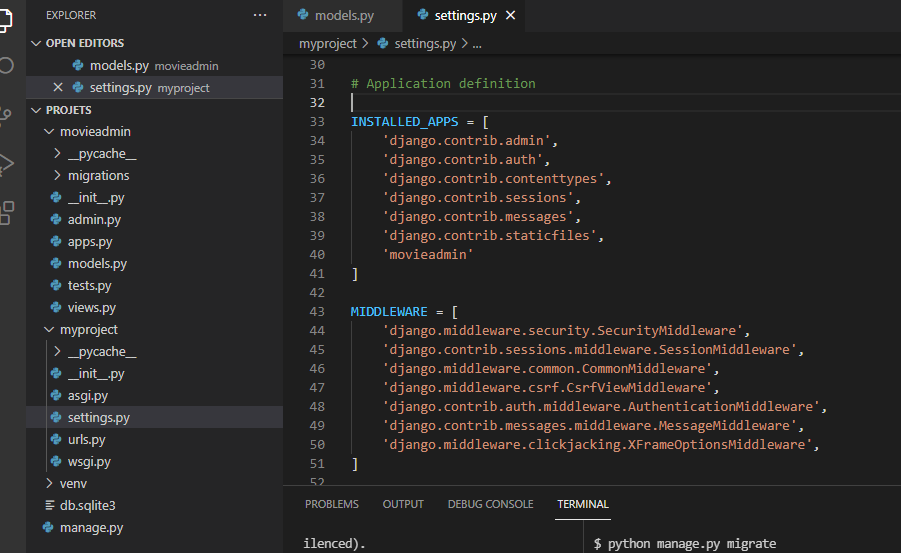
**MIGRATION**

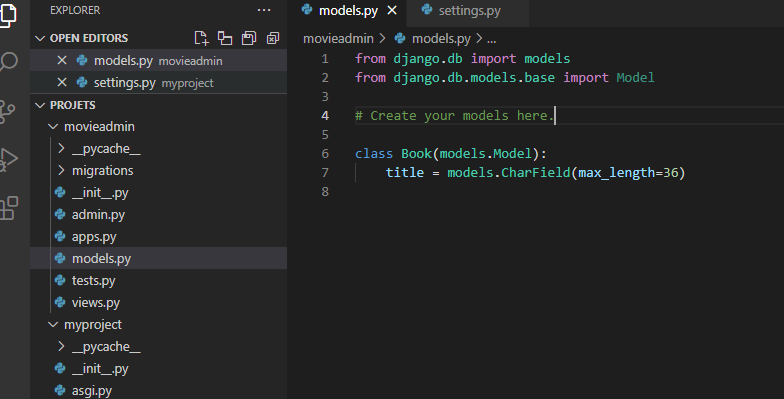
**Lancer sa première migration**

* python manage.py migrate

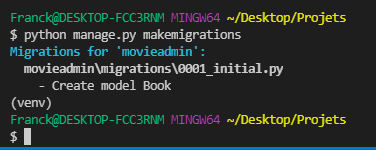
**Créer notre première migration**

Enregistrer le nom de l’application dans le fichier settings.py du répertoire du projet avant de faire une migration dans le modèle (Book) que nous avons défini dans notre class Book.



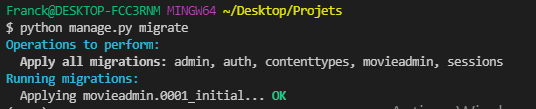


* python manage.py makemigrations



**Enregistrer notre migration dans la base de donnée**

* python manage.py migrate



**Option : Choisir MySQL comme Base de données**

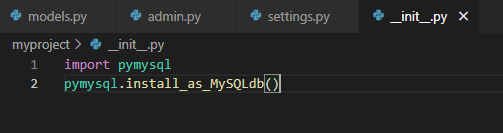
* **Installer le client MySQL de python**

pip install mysqlclient

* Ajouter ces lignes de code dans le fichier \_\_init\_\_.py qui se trouve à la même racine que settings.py pour éviter les erreur

Taper d’abord **pip install pymysql**

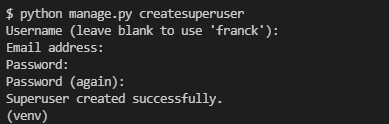
Et ajouter ces lignes



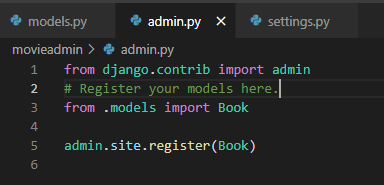
**USER & ADMIN**

**Créer un super user**

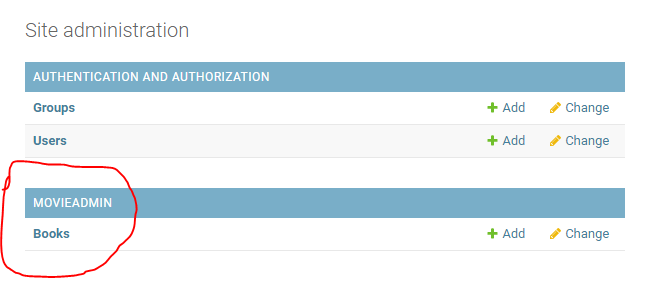
* python manage.py createsuperuser



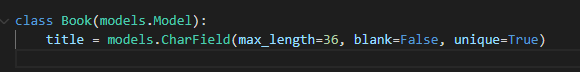
* Enregistrer notre model (Book) dans le fichier admin.py pour le voir dans la page admin



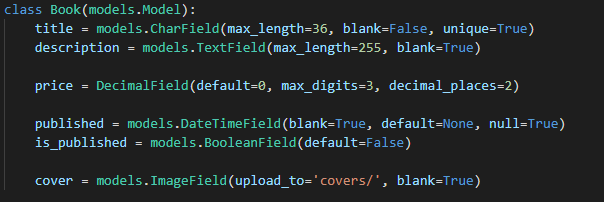
* Puis relancer le serveur : python manage.py runserver



**OPTIONS DE CHAMPS**

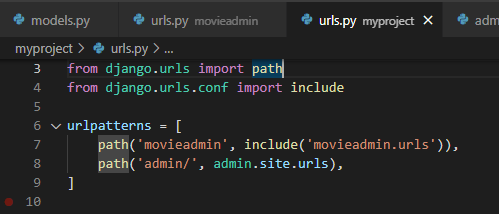


**TYPES DE CHAMPS**

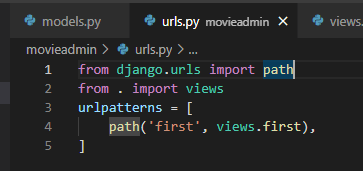


**URLS**

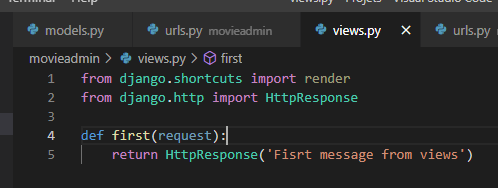
* Configurer notre **Url** **movieadmin** dans le fichier urls.py de notre projet



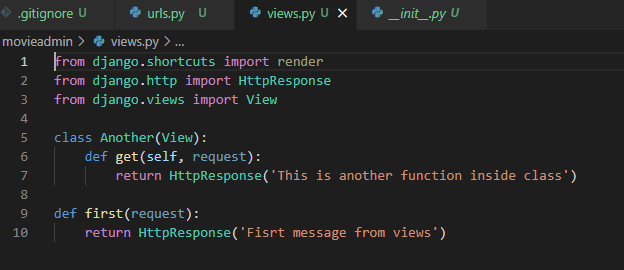
* Créons **l’Url** de notre page first dans notre fichier views de notre application **movieadmin**

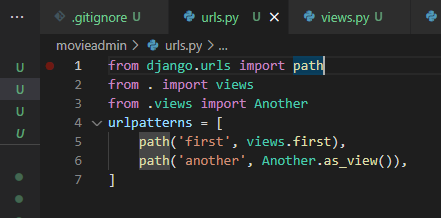


* Créer une vue a partir d’une fonction, La fonction **First** qui retourne le rendu de notre page à partir du fichier views de notre application **movieadmin**



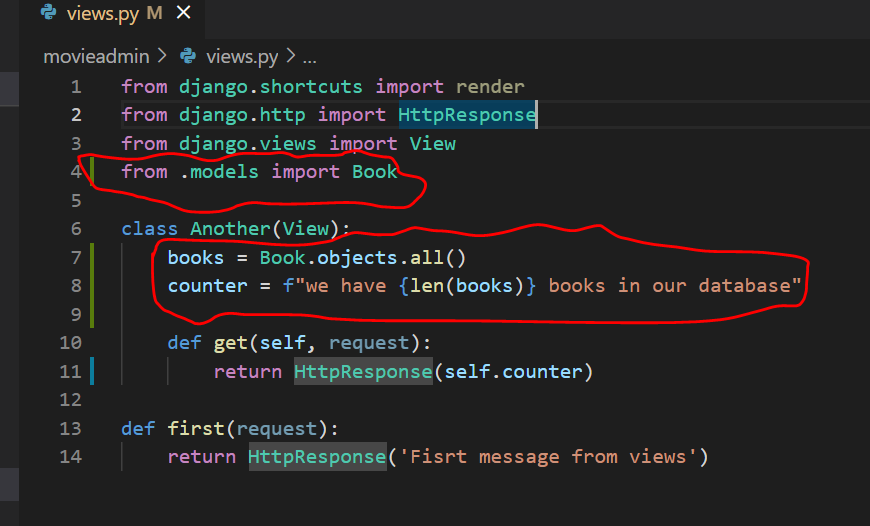
* Créer une vue a partir d’une classe

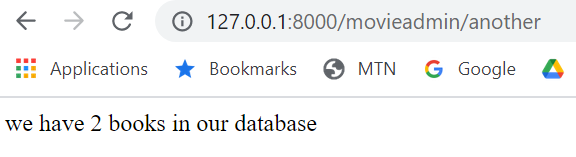




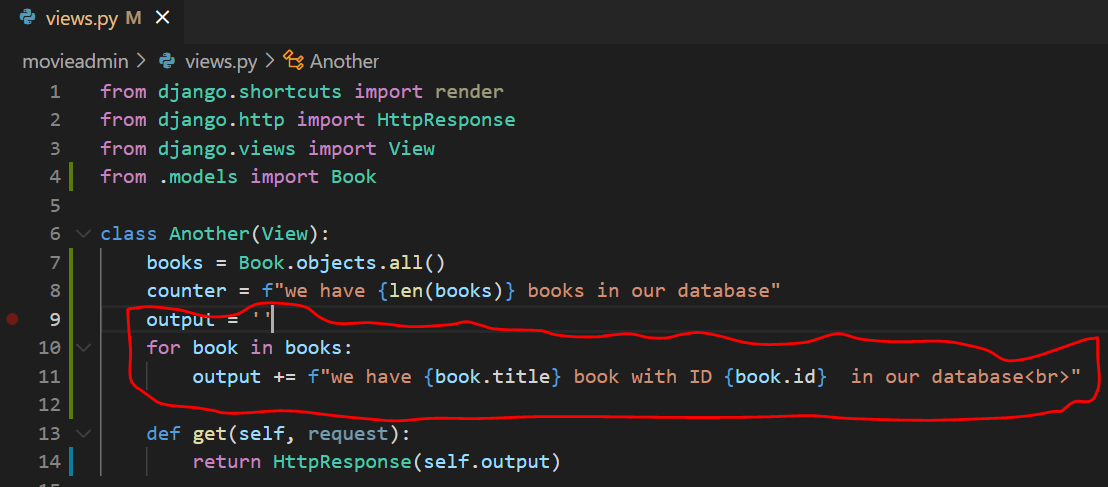
**RETOURNER DES DONNEES DEPUIS LA BASE DE DONNEES DANS NOTRE VUE**

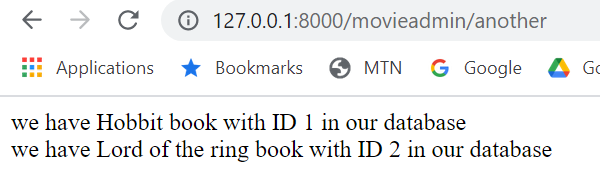
* **Compter le nombre d’occurrence de notre model Book**



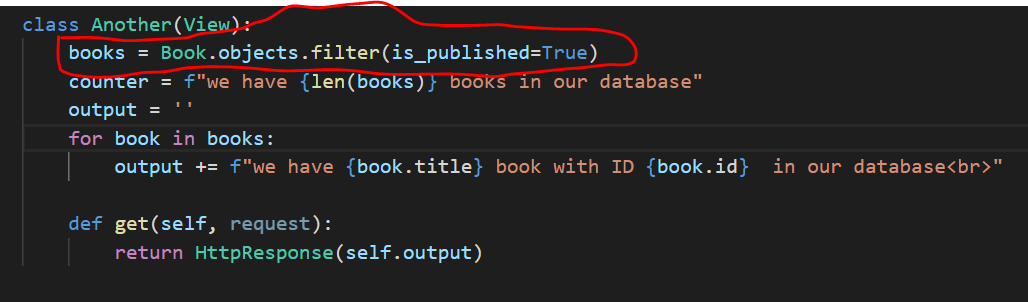


* Afficher les lignes de nos occurrences

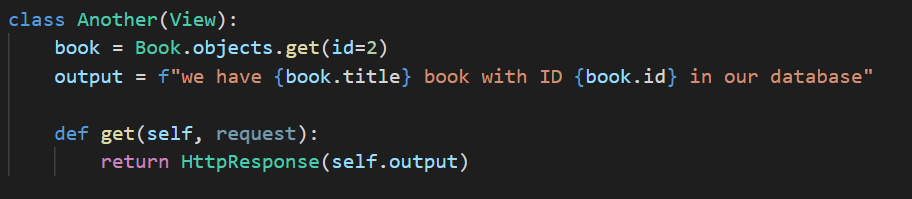




* Filtrer (par exemple filtrer tous les articles dont **is\_published = True**)

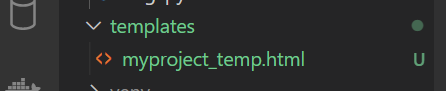


* Récupérer un enregistrement par son ID

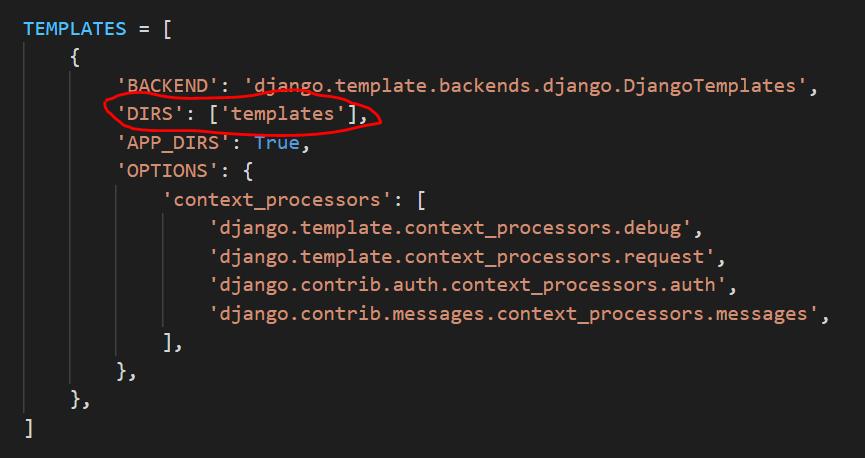


**TEMPLATES**

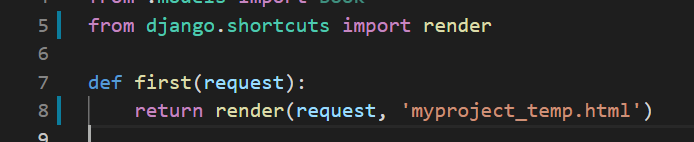
* Créer un répertoire **templates** à la racine de notre projet et créer notre fichier html



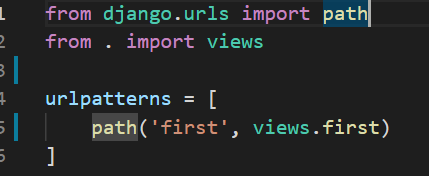
* Ajouter le nom du répertoire dans le dictionnaire de notre liste **TEMPLATE** du fichier settings



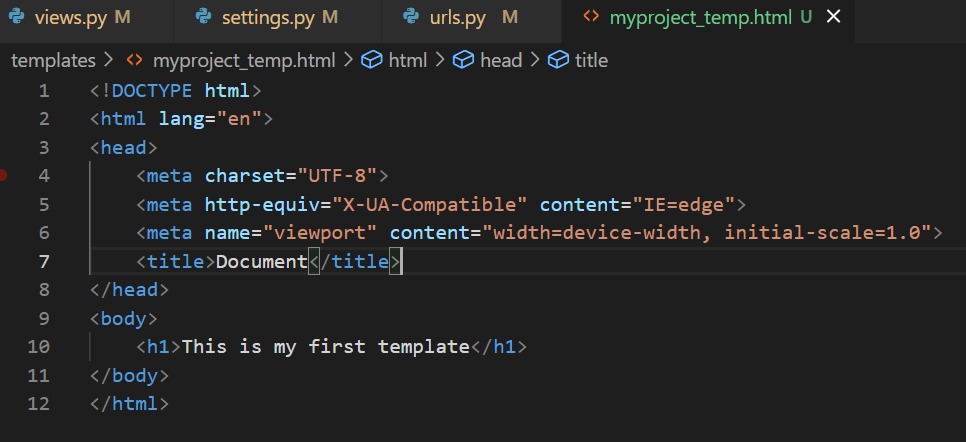
* Dans notre fonction first prend en paramètre request et myproject\_temp.html



* Indiquer l’Url



* Le fichier myproject\_temp.html

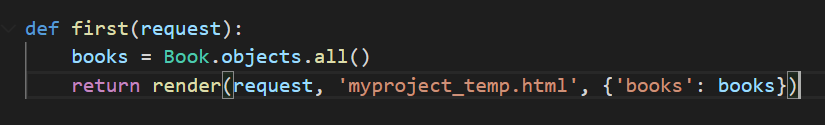


* Le rendu de la page

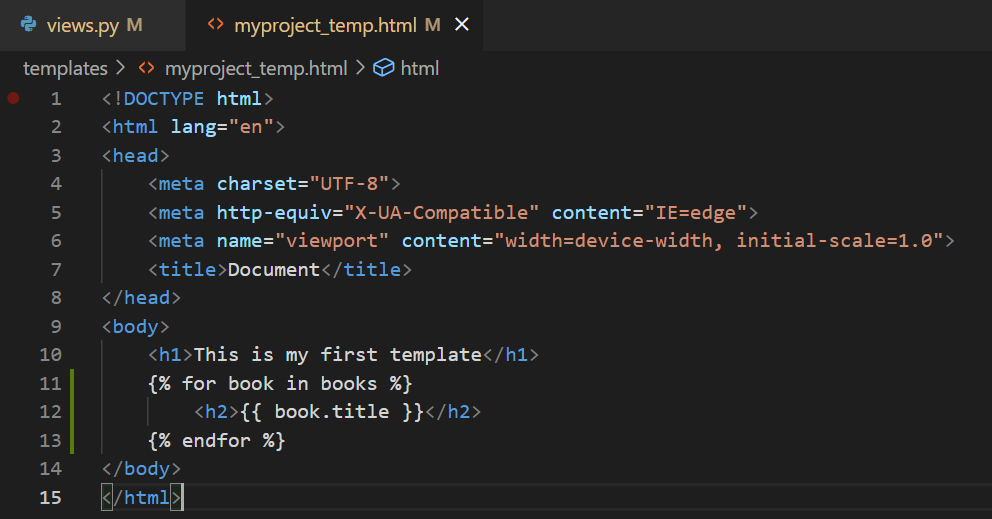


**SELECTION DYNAMIQUE DES DONNEES DE NOTRE BD**

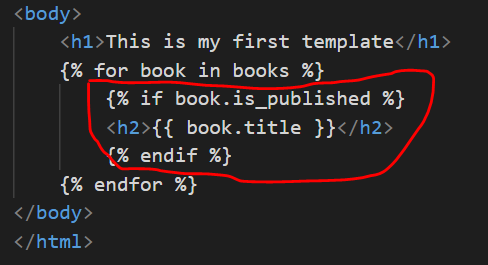
* Dans le fichier views.py de **movieadmin**



* Notre template **myproject\_temp.html**

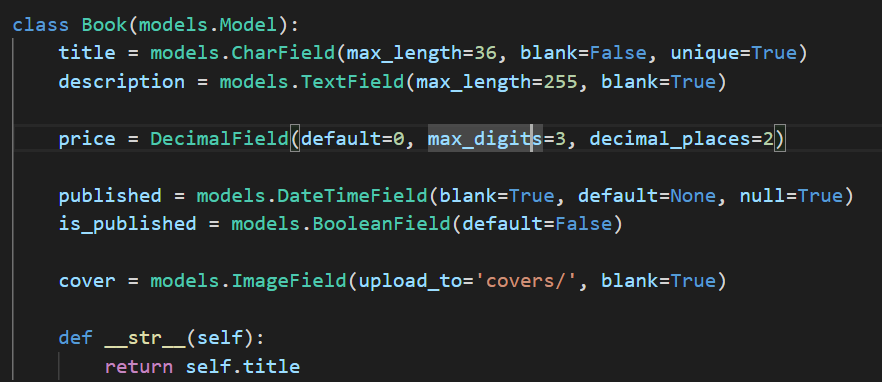


* Filtrage

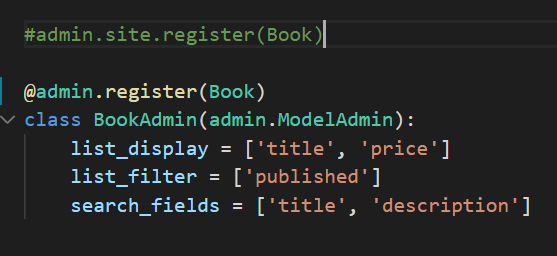


**ADMIN CUSTOMIZATION**

* Afficher les titres des objet dans la page d’administration. Ajouter la fonction \_\_str\_\_ dans models.py

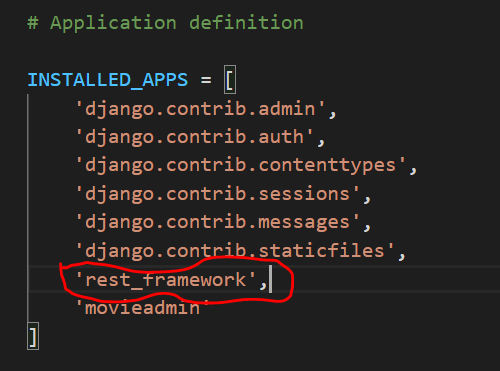


* Modifier le fichier admin.py



**DJANGO REST**

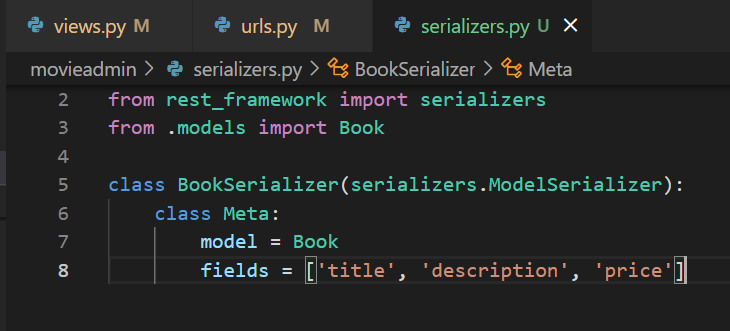
* pip install djangorestfraamework
* Ajouter rest\_framework dans notre fichier settings.py



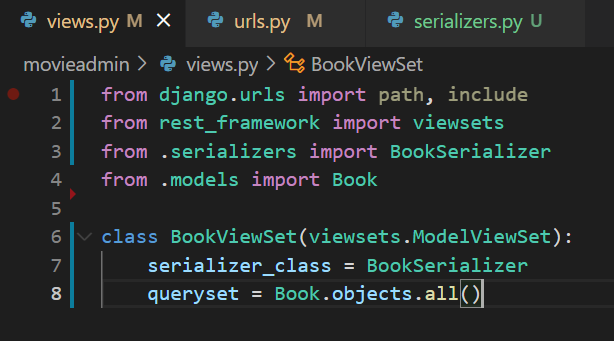
* python manage.py migrate

**SERIALIZERS**

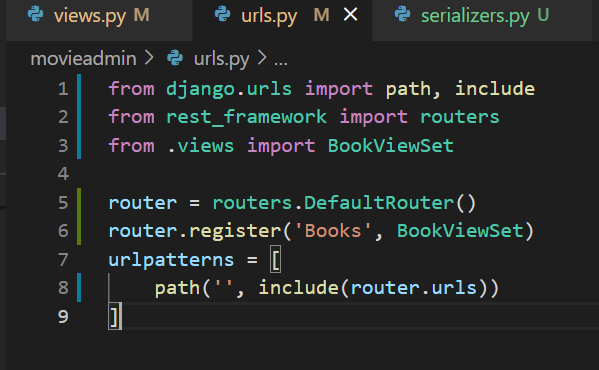
* Créer le fichier serializers.py à la racine de notre application **movieadmin**



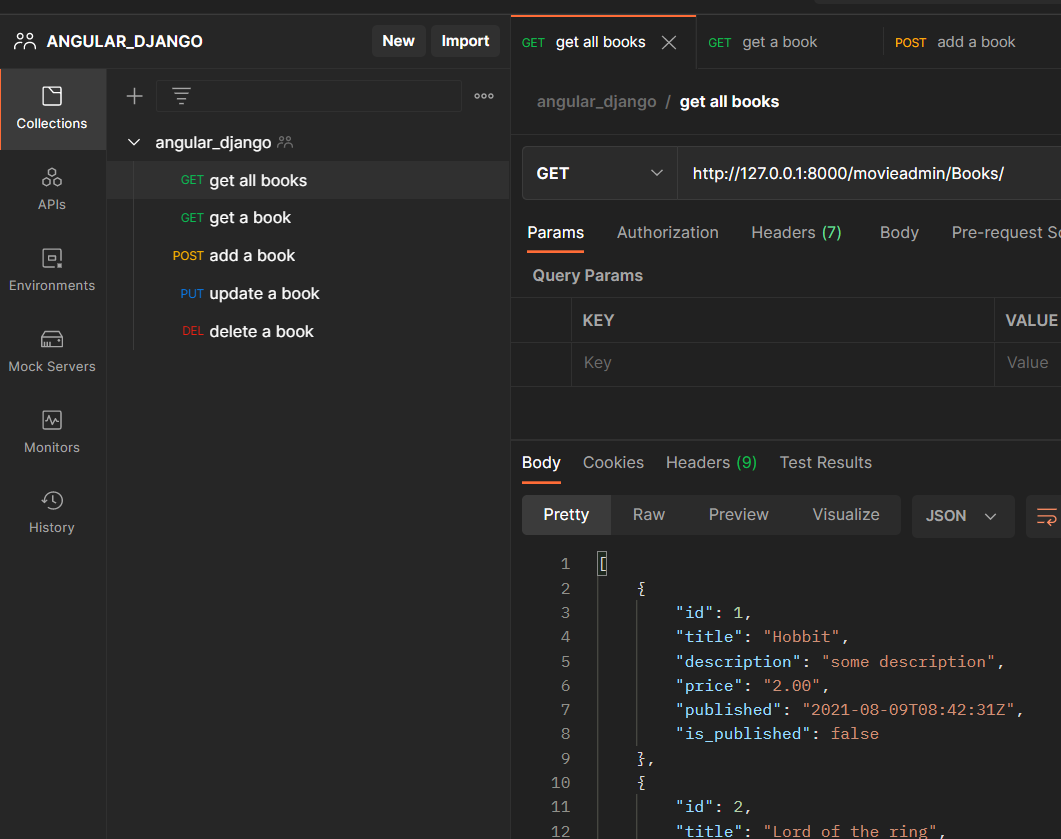
* Dans views.py



* Dans urls.py

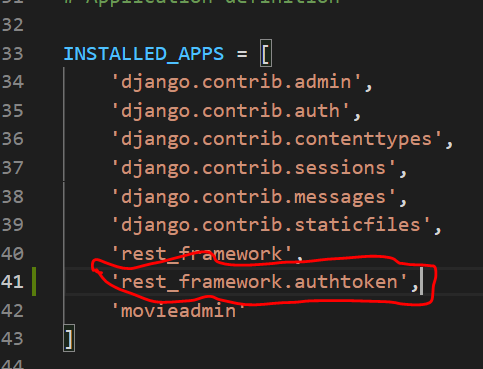


**TEST AVEC POSTMAN**



**AUTH TOKEN**

* Dans notre fichier settings.py ajouter **rest\_framework.authtoken**



* **Faire une migration**

python manage.py migrate