

## Projet médiathèque

L'objectif de ce projet est de créer un système fonctionnel utilisable par les bibliothécaires et des membres emprunteurs.

Répertoire Github : [Mauricette-g/project-mediatheque](https://github.com/Mauricette-g/project-mediatheque)

### **Création du modèle :**

Livre, DVD et CD étant des médias, nous créerons une classe parente média puis des sous-classe.

# Classe parente pour les médias

```
class Media(models.Model):
```

```
    nom = models.CharField(max_length=255)
```

```
    disponible = models.BooleanField(default=True)
```

```
    #date_emprunt = models.DateField(null=True, blank=True)
```

```
    #emprunteur = models.ForeignKey('Membre', null=True, blank=True,  
on_delete=models.SET_NULL)
```

```
class Meta:
```

```
    abstract = True
```

# Sous-classes héritant de Media

```
class Livre(Media):
```

```
    auteur = models.CharField(max_length=255)
```

```
class DVD(Media):
```

```
    realisateur = models.CharField(max_length=255)
```

```
class CD(Media):
```

```
    artiste = models.CharField(max_length=255)
```

# Classe pour les jeux de plateau

```
class JeuDePlateau(models.Model):
```

```

name = models.CharField(max_length=255)

createur = models.CharField(max_length=255)


# Classe pour les membres
class Membre(models.Model):

    name = models.CharField(max_length=255)

    bloque = models.BooleanField(default=False)


    def __str__(self):

        return self.name


# Classe pour gérer les emprunts
class Emprunt(models.Model):

    livre = models.ForeignKey('Livre', null=True, blank=True, on_delete=models.CASCADE)
    dvd = models.ForeignKey('DVD', null=True, blank=True, on_delete=models.CASCADE)
    cd = models.ForeignKey('CD', null=True, blank=True, on_delete=models.CASCADE)
    membre = models.ForeignKey('Membre', on_delete=models.CASCADE)
    date_emprunt = models.DateField(auto_now_add=True)
    date_retour = models.DateField()


    def save(self, *args, **kwargs):

        # Vérification des règles métiers avant d'enregistrer

        media = self.livre or self.dvd or self.cd

        if media and not media.disponible:

            raise ValueError("Ce média n'est pas disponible pour l'emprunt.")


        if Emprunt.objects.filter(membre=self.membre, date_retour__gte=self.date_emprunt).count()
>= 3:

            raise ValueError("Le membre a déjà 3 emprunts actifs.")


        super().save(*args, **kwargs)

```

```
def __str__(self):  
    return f"{self.membre.name} - {self.livre or self.dvd or self.cd}"
```

A chaque modification du modèle de données (models.py)

- Il faudra lancer les commandes suivantes pour que ça prenne effet dans la base de données
- `python manage.py makemigrations`
- `python manage.py migrate`

### **Côté admin :**

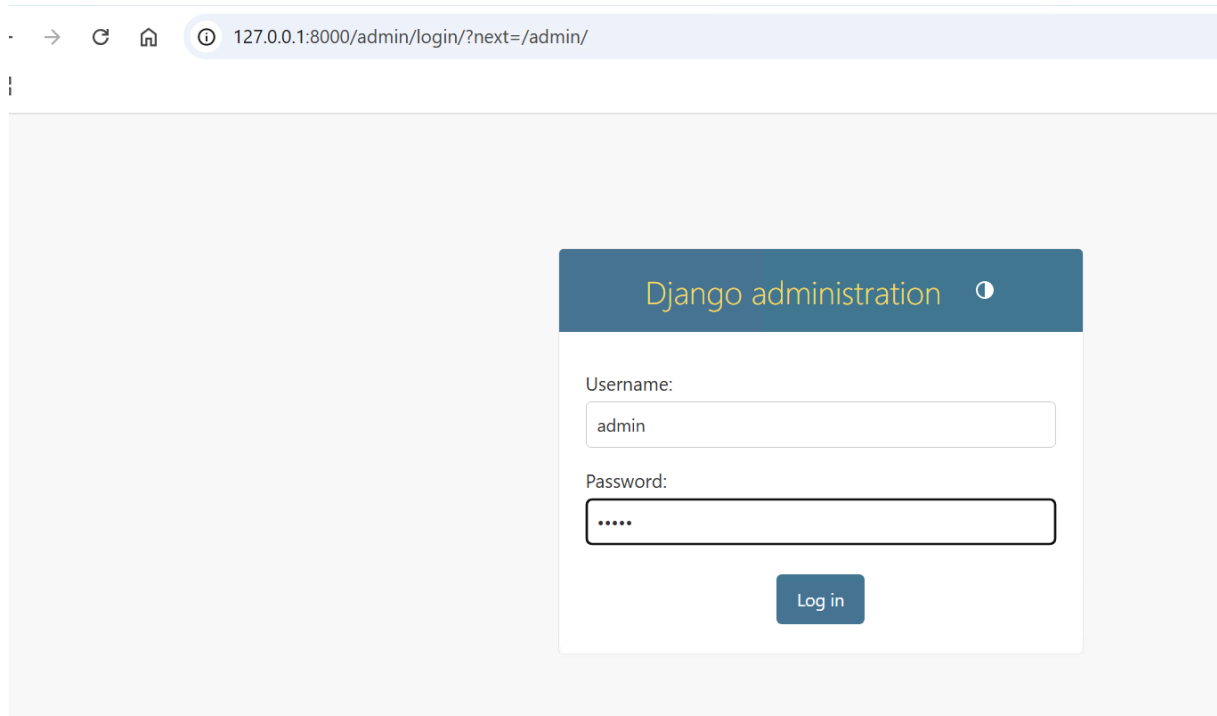
Créer l'utilisateur admin `python manage.py createsuperuser`

Ouvrir la page de développement : `python manage.py runserver`

Ouvrir le portail d'administration

`http://127.0.0.1:8000/admin/`

**user = admin, mot de passe = admin**



→ ↻ 🏠 ⓘ 127.0.0.1:8000/admin/login/?next=/admin/

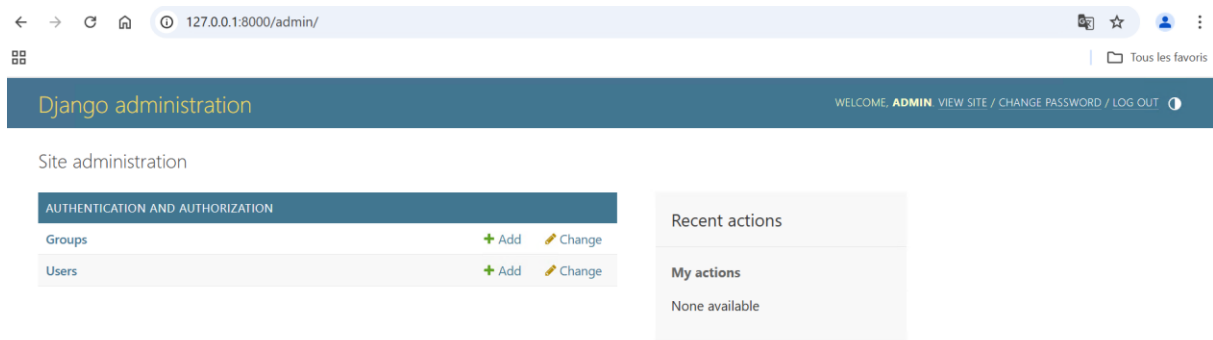
!

Django administration ⓘ

Username:

Password:

Log in



Modifier le fichier admin.py pour la gestion de l'application par l'admin

```
bibliotheque > admin.py
1  from django.contrib import admin
2  from .models import *
3  # Register your models here.
4  admin.site.register(CD)
5  admin.site.register(Livre)
6  admin.site.register(Media)
7  admin.site.register(Membre)
8  admin.site.register(DVD)
9  admin.site.register(JeuDePlateau)
10 admin.site.register(Emprunt)
```

OU

bibliotheque > admin.py

```
1  from django.contrib import admin
2  from .models import CD
3  from .models import Livre
4  from .models import Membre
5  from .models import DVD
6  from .models import Media
7  from .models import JeuDePlateau
8  from .models import Emprunt
9  # Register your models here.
10 admin.site.register(CD)
11 admin.site.register(Livre)
12 admin.site.register(Membre)
13 admin.site.register(DVD)
14 admin.site.register(JeuDePlateau)
15 admin.site.register(Emprunt)
```

← → ↺ 🏠 127.0.0.1:8000/admin/ 🖨️ ☆ 👤 ⋮  
🗄️ Tous les favoris

## Django administration

WELCOME, **ADMIN**. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

### Site administration

#### AUTHENTICATION AND AUTHORIZATION

Groups	<a href="#">+ Add</a>	<a href="#">Change</a>
Users	<a href="#">+ Add</a>	<a href="#">Change</a>

#### BIBLIOTHEQUE

Cds	<a href="#">+ Add</a>	<a href="#">Change</a>
Dvds	<a href="#">+ Add</a>	<a href="#">Change</a>
Emprunts	<a href="#">+ Add</a>	<a href="#">Change</a>
Jeu de plateaus	<a href="#">+ Add</a>	<a href="#">Change</a>
Livres	<a href="#">+ Add</a>	<a href="#">Change</a>
Membres	<a href="#">+ Add</a>	<a href="#">Change</a>

#### Recent actions

##### My actions

None available

← → ↺ 🏠 127.0.0.1:8000/admin/bibliotheque/membre/ 📄 ☆ 👤 ⋮

📄 Tous les favoris

Django administration WELCOME, **ADMIN** VIEW SITE / CHANGE PASSWORD / LOG OUT ⓘ

Home » Bibliotheque » Membres

Select membre to change ADD MEMBRE +

0 membres

---

← → ↺ 🏠 127.0.0.1:8000/admin/bibliotheque/membre/add/ 📄 ☆ 👤 ⋮

📄 Tous les favoris

Django administration WELCOME, **ADMIN** VIEW SITE / CHANGE PASSWORD / LOG OUT ⓘ

Home » Bibliotheque » Membres » Add membre

Add membre

Name: Irène Mauricette Mendy

☐ Bloque

»

SAVE Save and add another Save and continue editing

---

← → ↺ 🏠 127.0.0.1:8000/admin/bibliotheque/membre/ 📄 ☆ 👤 ⋮

📄 Tous les favoris

Django administration WELCOME, **ADMIN** VIEW SITE / CHANGE PASSWORD / LOG OUT ⓘ

Home » Bibliotheque » Membres

✔ The membre "Irène Mauricette Mendy" was added successfully.

Select membre to change ADD MEMBRE +

Action: ----- Go 0 of 1 selected

☐ MEMBRE

☐ Irène Mauricette Mendy

»

1 membre

## Côté utilisateur :

Ajouter dans les vues les paths

```

bibliotheque > views.py > index
1  from django.http import HttpResponseRedirect
2
3
4  # Create your views here.
5  def index(request):
6      return HttpResponseRedirect("Bonjour sur l'application de la bibliothèque.")
7
8  def detail(request, auteur):
9      return HttpResponseRedirect("Voici le livre de %s." % auteur)
10
11
12  def results(request, auteur):
13      response = "Voici le livre de l'auteur %s."
14      return HttpResponseRedirect(response % auteur)
15
16
17

```

Ajouter les urls correspondants

```

bibliotheque > urls.py > ...
1  from django.urls import path
2
3  from . import views
4
5  urlpatterns = [
6      # ex: /polls/
7      path("", views.index, name="index"),
8      # ex: /polls/5/
9      path("<str:auteur>/", views.detail, name="detail"),
10     # ex: /polls/5/results/
11     path("<str:auteur>/results/", views.results, name="results")
12 ]

```

Dans la partie admin ajouter un livre ou un media de test :

127.0.0.1:8000/admin/bibliotheque/livre/1/change/

Django administration

WELCOME ADMIN VIEW SITE / CHANGE PASSWORD / LOG OUT

Home » Bibliotheque » Livres » Livre object (1)

Change livre

Livre object (1)

Nom: Test

☒ Disponible

Auteur: Irene

SAVE Save and add another Save and continue editing Delete

Tester en jouter dans l'url le nom de l'auteur

localhost:8000/bibliotheque/Irene/

Voici le livre de Irene.