

# Mediatheque avec Django (framework python)

## 1. Étude et correctifs du code fourni

Le code fourni avait des classes avec des champs (field) répétitif , car certaines classe avaient des champs identiques, c'est pourquoi j'ai créé une classe global avec ces modèles redondant et que j'ai utilisé l'héritage pour que les modèles puissent les utiliser.

J'ai aussi ajouter des options de champs qui correspondent au paramètres des champs qui seront utilisés dans la base de données mais aussi pour les formulaires.

Pour l'utilisation des menus , j'ai mis en page d'accueil la liste des médias disponibles, accessible à tous , et le reste de l'application accessible uniquement par un utilisateur authentifié pour effectuer tout ce qu'un bibliothécaire est amené à effectuer.

## 2. Mise en place des fonctionnalités demandées

1. Connexion : via un formulaire importer directement par django (AuthenticationForm) avec redirection vers la liste des emprunts si authentifié.

2. Liste des membres : en sélectionnant tous les membres pour ensuite l'envoyer a mon template qui pourra itérer dessus et les afficher.
3. Ajouter un membre : via un formulaire et ensuite l'intégrer à la base de données.
4. Supprimer un membre : avec la fonction delete() fourni par Django pour le supprimer de la bdd.
5. Mettre à jour un membre : Même principe que l'ajout mais on écrase l'ancien nom.
6. Liste des médias : même principe que la liste des membres avec les médias.
7. Ajouter Média : toujours via un formulaire , en obligeant à sélectionner les paramètres pour ensuite l'intégrer à la base de données.
8. Supprimer Média : Toujours avec la méthode delete() mais on vérifie si le média est en cour d'emprunt.
9. Liste emprunt : on sélectionne les emprunts et on passe le paramètre qui correspond à la date d'aujourd'hui pour avoir la date retour prévu.
10. Emprunter un média : on récupère le membre, le type de média , et son nom pour l'attribuer à un membre, on vérifie si

le membre a plus de 3 emprunts, et si le média n'est pas déjà emprunté avec son paramètre "disponible".

11. Retourner un média : j'ai utilisé une méthode de classe , on remet le média disponible et on supprime l'emprunt.

12 . Et pour les membres , la liste média , j'ai sélectionné les médias qui était disponible avec le paramètre filter pour passer à mon template les médias qui était disponible.

### 3. Stratégie de test

J'ai fait un test par fonctionnalité , j'ai créé des user pour tester la connexion , des membres, des médias pour tester les méthodes. J'ai utilisé des import Django comme reverse pour vérifier les redirections grâce au statut HTTP que renvoi certaines pages.

### 4. Base de données avec des données test

Je me suis servi de la base de données sqlite fourni grâce à la migration , et j'ai créé un fichier data comportant du code json pour les données test , avec toutes les instructions dans le fichier README.md