

Création du logiciel de gestion de médiathèque

1. Introduction

La demande est la création d'une application de médiathèque grâce à Django. Cette dernière va avoir deux interfaces :

- Utilisateur : interface de consultation des médias
- Personnel : interface sécurisée par une authentification permettant la gestion des données.

2. Instructions d'installation et d'exécution

- Ouvrir le dossier mediatheque sur un IDE
- Créer et activer un environnement virtuel :
 - Windows :
 - `py -m venv media-env`
 - `media-env\Scripts\activate`
 - Linux/Mac :
 - `python -m venv media-env`
 - `source media-env/bin/activate`

Si l'environnement est bien activé, (media-env) devrait alors apparaître au début de votre ligne de commande.

- Installer les dépendances
 - `pip install -r requirements.txt`
- Lancer le serveur :
 - Windows : `py manage.py runserver`
 - Mac/Linux : `python manage.py runserver`
- Pour se connecter à l'interface "staff" utiliser l'identifiant **DjangoCEF** et le mot de passe **DjangoCEF** (Pour l'exercice, des logs simples sont utilisés afin de faciliter les choses)

3. Etude du code fourni

Pour ce devoir, une partie du code des classes models est fourni, seulement, ce dernier se veut incomplet et mal optimisé. Il a fallu donc retravailler tout ça. En analysant le code fourni, on remarque rapidement que le système d'héritage n'est pas utilisé, il y a des redondances dans les class ("name", "dateEmprunt", "disponible" etc...).

Il a donc fallu reprendre les éléments de code fourni afin d'en faire un code propre, lisible et optimisé pour avoir une bonne base pour le [models.py](#) de notre application.

A la suite des différents correctifs apportés, le code est agrémenté afin d'intégrer les différentes contraintes demandées. (Emprunteur bloqué après 3 emprunts en cours ou alors après une semaine d'emprunt, que les jeux de plateau soient toujours en disponible puisque non empruntables)

4. Mise en place des fonctionnalités demandés

L'interface Utilisateur (User) n'est qu'une interface de consultation des données, donc une page d'accueil avec tous les médias mais aussi par tri de type de média est disponible.

Pour l'interface Personnel(Staff), la mise en place d'un système de connexion fourni par Django est mis en place. Il n'y a que les utilisateurs qui sont staff (`is_staff = True`) qui peuvent accéder à cette interface. Ce point là, permettra par la suite de créer une interface différente pour les utilisateurs si on souhaite qu'ils puissent s'identifier. L'interface Staff permet de visualiser les médias, de les gérer en les modifiant, en les supprimant, ou en ajoutant des médias. Un "staff" peut également faire l'emprunt d'un média ou son retour, et il a aussi accès à la liste des emprunteurs et leurs statuts.

5. Les tests

Dans `staff/tests.py`, différents tests sont mis en place pour vérifier la logique pour :

- L'emprunt → Passer *disponible de True à False* et enregistrer **dateEmprunt**
- Le retour → Suppression de l'emprunteur, la date et passer *disponible de False à True*
- Blocage →
 - 3 emprunts : l'emprunteur devient automatiquement bloqué
 - 1 semaine : l'emprunteur devient bloqué après que le temps soit passé
- Jeux de plateau : Ne peuvent pas être empruntés.

Pour exécuter les tests, il faut exécuter la commande :

Windows : `py manage.py test staff`
Mac/Linux : `python manage.py test staff`

Un fichier de données tests à également été conçues afin de vérifier sur l'application médiathèque les fonctionnalités via l'interface : 5 données pour chaque média et 5

emprunteurs, et avec des médias déjà empruntés pour avoir une mise en situation la plus proche du réel.

6. Conclusion

L'application réponds au différents besoins demandés :

- une interface utilisateur simple
- une interface "staff" sécurisée, incluant la gestion des emprunteurs, le CRUD (Create, Read, Update, Delete) pour la gestion des médias et les règles de blocage des emprunteurs.

Le projet se veut évolutif puisqu'il est séparé en 2 applications, il sera donc possible d'intégrer plus de fonctionnalités et d'autres interfaces (applications par la suite).