

JELLYSMACK



Sujet

Votre mission (si vous l'acceptez...) est de créer une API REST qui permet à un utilisateur de faire des commentaires sur les personnages et les épisodes dans l'univers Rick&Morty.

Contexte

Les contraintes techniques pour ce projet sont les suivantes:

- Utilisation de Python 3.7+
- Utilisation d'un SGDB relationnel
- Utilisation d'un des deux frameworks suivants:
 - Flask
 - FastAPI
- Respecter les conventions REST

Vous disposez également de deux fichiers JSON contenant les données à utiliser pour votre projet.

Partie obligatoire

Feature 1

Pour cette première partie nous voulons être capable de voir les données que nous avons, il vous faudra donc faire les deux choses suivantes:

1. Script d'import des données depuis les fichiers JSON vers les tables relatives aux épisodes et aux personnages.
2. Exposer deux routes API:
 - a. une route permettant de lister les épisodes
 - b. une route permettant de lister les personnages

Feature 2

Maintenant que nous pouvons lister nos différentes ressources nous voulons mettre en place un système permettant à nos utilisateurs de poster des commentaires. Il vous faudra implémenter les opérations d'API CRUD pour cette nouvelle ressource.

Il est à noter qu'un commentaire peut être associé à:

- un épisode
- un personnage
- un personnage dans un épisode.

Feature 3

Nous avons maintenant une API qui nous permet d'effectuer tout ce qui était prévu à l'origine mais nous faisons maintenant face à une nouvelle problématique, nous retournons trop de données et nous voulons réduire ça.

Il vous faudra mettre en place les deux fonctionnalités suivantes:

1. Un système de pagination sur les commentaires et personnages
2. Mise en place de filtres

Partie optionnelle

Les étapes suivantes étant optionnelles vous n'êtes pas dans l'obligation de respecter l'ordre que vous trouverez dans ce sujet.

Feature 4

A présent que notre MPV est fonctionnel, nous aimerions l'ouvrir à un premier groupe d'utilisateurs.

Pour cela nous il faut implémenter un système d'authentification basé sur JWT:

- Un utilisateur doit pouvoir se login et se logout.
- API CRUD sur les users

Feature 5

Après plusieurs semaines d'utilisation de la part des premiers utilisateurs, notre PO souhaite exploiter certaines données pour cela il aimerait disposer d'un export csv des commentaires.

Proposer une route API pour l'export des commentaires sous format csv ou xls le but est de permettre à l'équipe data de faire des analyses et d'extraire des tendances.

Backlog

A la suite de la dernière réunion avec le PO, il a été identifié le backlog suivant:

1. Récupération de metada depuis https://www.imdb.com/title/tt2861424/?ref=tt_ov_inf:
 - a. récupérer la description de chaque épisode et la stocker en base de données.
 - b. récupérer le thumbnail de chaque épisode.
2. Mise en place d'un workflow de modération des commentaires:
 - a. mise en place des rôles; admin, user, modérateur
 - b. status sur les commentaires; *new*, *in review*, *rejected*, *approved*. Ainsi que la relation de commentaire de commentaire.
3. Les indicateurs statistiques suivants (au format csv):
 - a. nombre de commentaires par épisode
 - b. nombre de commentaires au statut "rejected" par épisode
 - c. la longueur moyenne (nombre de lettres) des commentaires par épisode.

Critères d'évaluation

Nous serons attentifs aux éléments suivant:

1. Qualité du code
 - a. Respect des conventions
 - b. Nommage des variables/fonctions/classe perninant
 - c. Test et couverture du code
2. Délivrabilité de la solution:
 - a. Documentation de la solution
3. Design de la solution
 - a. Design applicatif

- b. Design de la base de données.

Délivrabilité

Le projet devra être rendu dans un repo GitHub public. Commitez aussi souvent que possible et commentez vos commits pour détailler votre chemin de pensée.

Le [README.md](#) doit contenir la documentation pour lancer le projet, la documentation des routes, le temps passé et tout ce que vous jugerez nécessaire de nous faire savoir.

N'hésitez pas à nous faire des retours et nous expliquer les éventuelles problématiques bloquantes que vous auriez rencontrées durant le développement vous empêchant de finir.

Bon code \o/