Travail pratique #2 : API Authentification

Pondération : 25 %

Travail devant être réalisé **en équipe de 2**

Remise : Au plus tard **dimanche le 4 avril** avant minuit

1 Objectif

Développer et héberger une API REST sécuritaire incluant des tests en utilisant Laravel.

2 Conditions de réalisation du travail

Votre remise consiste à livrer toutes les routes accompagnées de tests.

Vous devez utiliser votre TP1 comme départ. Ce n'est pas grave si votre TP1 n'était pas complet, vous n'avez pas à le corriger car vous ne serez pas évalué de nouveau sur les mêmes choses.

Le modèle de la base de données est disponible en annexe.

De plus, vous devrez travailler en équipe en utilisant **GitLab** en continu et de façon adéquate. Vous devrez faire un *merge request* à votre coéquipier pour chaque fonctionnalité ou modification de code.

Éléments importants de l'API REST :

- Contrôleurs
- Modèles et entité(s)
    - Relations appropriées
- Ressource (DTO)
- Validation (*Request*) pour les actions *store* et *update*
- Structure et données de tests (migrations et *seeds*)
- Documentation de l'API avec Swagger
- Tests de la base de données

- Tests suivants pour toutes les actions :
    - Les requêtes valides (200 et 201)
    - Les requêtes avec données invalides (422)
    - L'authentification (401)
    - L’autorisation (403)

3 API – Routes à développer

- **Opérations CRUD** pour un film :
    - **Ajout** d’un film (seulement si admin)
    - **Modification** d’un film; tous les champs sont modifiables à l’exception de l’identifiant du film, bien sûr (seulement si admin)
    - **Suppression** d’un film (seulement si admin)
- **Ajout** d’une critique (seulement si membre connecté)
- **Opérations CRUD** pour un *user* :
    - **Consultation** des informations d’un certain *user* (seulement si on est connecté avec ce *user*)
    - **Modification** d’un *user* existant (seulement si on est connecté avec ce *user*)

4 Critères d'évaluation

Éléments	Pondération
• **Routes d'API demandées** o Autorisation o Planification des routes, requêtes, validations et réponses o Considération de sécurité	40%
• **Qualité du code** o Bonne utilisation des patrons de conception ▪ MVC, DTO, Entité, Modèles, Relations, Request, etc. o Bien découpé et indenté o Code clair et simple o Principes SOLID o Respecte la structure de Laravel	20%
• **Qualité des tests** o Données de tests suffisantes (*seeds*) o Tests suffisants o Teste la fiabilité et la robustesse o Tests pour les codes suivants : 200, 201, 204, 422, 401, 403, 405	30%
• **Utilisation adéquate de GitLab** o Branches par fonctionnalité, *merge request*, participation, etc.	10%

Pour chaque faute de français, 0,5% sera soustrait de la note finale (pour un maximum de 20%).

5 Documents à réaliser et à remettre

Veuillez remettre sur « LÉA » une archive « zip » contenant tous les fichiers du projet :

- Lien GitLab avec accès pour l'utilisateur **filiat00**
- Code (sans les dépendances – **sans le dossier *vendor***)

Bon succès!

Annexe – Base de données