



Test Laravel backend

Instructions

Mise en contexte

Le but premier de ce test est d'évaluer la qualité du code, le respect des consignes et le souci du détail. Nous pensons que, grâce à ton expérience, tu pourras compléter ce projet démo en environ 4h.

Dans le but de respecter les bonnes pratiques de Laravel, on t'invite à porter une attention particulière aux détails. Il y a plusieurs façons de parvenir à compléter les tâches demandées.

N'hésites pas à nous écrire en cours de route si tu as des questions.

Objectifs

Réaliser une petite application Laravel permettant l'accès à quatre endpoints GET, un endpoint POST et un endpoint DELETE, qui pourront être testés depuis Postman.

Au moins un des concepts suivants devra être utilisé pour une fonctionnalité du projet démo. Si aucune des fonctionnalités demandées ne permet d'utiliser ces concepts de façon adéquate, tu pourras développer une nouvelle fonctionnalité qui saura en tirer profit.

- Events + Listeners
- Observers
- Queues
- Mailables

Le code final pourra nous être envoyé sous forme de fichier .zip ou encore mis en ligne sur Github. Nous ne nous attendons pas à ce que ces éléments soient présents :

- authentification à l'API
- traductions

Requis techniques

Mise en place de l'application

À partir d'une installation fraîche de Laravel à jour (incluant une connexion à une base de données), créer les migrations et les modèles pour contenir au minimum ces structures de données :

- Équipes
 - Nom
- Membres
 - Nom
 - Courriel
 - Équipe

Les modèles devront supporter la suppression réversible (*soft-delete*); la liste des champs de chaque modèle ne se limitera donc pas à la liste ci-dessus.

Création de données démo

Insérer des données démo pour ces deux modèles dans la base de données, de manière programmatique.

Endpoints GET

Développer quatre endpoints GET, permettant de récupérer respectivement les équipes et les membres.

- Équipes :
 - Retourner toutes les équipes

- Pour chaque équipe, retourner le nom d'équipe et la liste des membres qu'elle contient
- Équipe :
 - Possibilité de passer un identifiant d'équipe en paramètre pour filtrer la recherche
 - Pour chaque équipe, retourner le nom d'équipe et la liste des membres qu'elle contient
- Membres :
 - Retourner tous les membres
 - Pour chaque membre, savoir à quelle équipe il appartient
- Membre :
 - Possibilité de passer un identifiant de membre en paramètre pour filtrer la recherche
 - Pour chaque membre, savoir à quelle équipe il appartient

Endpoint POST

Créer un endpoint POST permettant de créer un nouveau membre d'équipe (incluant toutes les informations pertinentes).

Endpoint DELETE

Le endpoint DELETE devra permettre de *soft-delete* une équipe. Lors de la suppression d'une équipe, tous les membres qui y sont rattachés devront être *soft-deletés* également.