

Objectif : Gestion des erreurs avec des exceptions dans un service Express + MongoDB

 Propriétaire	 Marine
 Étiquettes	

Dans cet exercice, vous allez implémenter un service de gestion des utilisateurs dans une API avec Express et MongoDB. Vous allez apprendre à gérer les erreurs à l'aide des exceptions pour assurer un meilleur contrôle des flux dans votre application. Voici les trois routes que vous devez implémenter :

1. Route de recherche d'utilisateur par email

La première route permet de rechercher un utilisateur par son adresse e-mail.

Étapes à suivre :

- Créez une fonction `findUserByEmail(email)` dans votre service de gestion des utilisateurs.
- Cette fonction devra accepter un email en paramètre.
- **Erreur à gérer :** Si aucun email n'est fourni, lève une exception `ArgumentRequiredException`.
- Une fois l'email fourni, il faut rechercher l'utilisateur dans la base de données.
- **Erreur à gérer :** Si l'utilisateur n'est pas trouvé en base, lève une exception `UserNotFound`.

2. Route de creation d'utilisateur

- **Erreur à gérer :** Si aucun email n'est fourni, lève une exception `ArgumentRequiredException`.
- Une fois l'email fourni, il faut rechercher l'utilisateur dans la base de données.

- **Erreur à gérer** : Si l'utilisateur est trouvé en base, lève une exception `UserAlreadyExistsException`.
 - Si l'email n'existe pas, vous pourrez créer l'utilisateur dans la base de données.
 - Après la création, vous devrez renvoyer un message de succès ou l'objet de l'utilisateur créé.

3. Gestion des exceptions

Pour chaque exception que vous devez gérer, vous devez d'abord définir vos propres exceptions personnalisées :

- `ArgumentRequiredException`
- `UserNotFoundException`
- `AlreadyExistingException`
 - Où et comment pouvez-vous définir ces exceptions dans votre code pour qu'elles soient réutilisables ?
 - Comment gérer la propagation de ces exceptions dans votre service ? Utilisez-vous un mécanisme global pour intercepter et gérer les erreurs (par exemple, un middleware de gestion des erreurs) ?
 - Comment renvoyer des réponses appropriées (statut HTTP, messages d'erreur) au client en cas d'exception ?

Étapes détaillées pour l'implémentation :

1. **Mise en place des exceptions personnalisées** : Avant de commencer à coder vos routes, commencez par définir vos exceptions personnalisées. Vous pouvez créer une classe pour chaque type d'exception qui étend `Error`, et chaque classe devra recevoir un message d'erreur approprié lors de son instantiation.

Exemples de réflexion :

- Comment structurer ces classes pour qu'elles soient réutilisables et extensibles ?
- Quel type d'informations souhaitez-vous inclure dans les exceptions (par exemple, un code d'erreur spécifique) ?