TD 1 : Créer une commande Artisan

Objectifs :

- Créer une commande exécutable en ligne de commande
- Utiliser des paramètres et des options en entrée de la commande
- Interagir avec l'utilisateur
- Afficher des messages à l'utilisateur
- Mettre en place une barre de progression
- Valider les données

Ressources :

- https://laravel.com/docs/9.x/artisan#generating-commands
- https://laravel.com/docs/9.x/artisan#defining-input-expectations
- https://laravel.com/docs/9.x/artisan#retrieving-input
- https://laravel.com/docs/9.x/artisan#asking-for-confirmation
- https://laravel.com/docs/9.x/artisan#progress-bars
- https://laravel.com/docs/9.x/validation#manually-creating-validators

Créer une commande **import:json.**

Cette commande prendra en paramètre les arguments suivants :

fileName : Le nom du fichier à importer
table : La table dans laquelle nous importos les données (Valeur par défaut : Product)

Elle prendra en option l'argument suivant :

dry-run : Possibilité de lancer sans insertion en base de données

Au début du traitement un message viendra demander confirmation à l'utilisateur s'il souhaite réaliser un import de données.

Vérifier les arguments en entrée :

Le fichier doit exister : Storage::fileExists($filePath) permet de verifier si un fichier existe. La variable filePath est le chemin relatif depuis le dossier storage/app
La table doit être dans la liste suivante : Product, Category, Size

Les erreurs de validation d'arguments seront affichées à l'utilisateur et provoqueront un arrêt du traitement avec un code retour INVALID

Mettre en place une barre de progression afin de connaitre l'avancée du traitement. Pour voir la progression il est possible de ralentir le traitement grâce à la fonction sleep.

Pour lire un fichier utiliser la méthode suivante

```
private function readFile(string $fileName): array

{

    $datas =
json_decode(Storage::get(self::IMPORT_FILE_FOLDER.$fileName), true);
```

```
        return $datas;
    }
```

Import des produits.

Le slug sera généré à partir du titre du produit.
Règles de validation des Produits :

- title : obligatoire, longueur maximum 200
- slug : unique, obligatoire
- description : obligatoire
- price_in_cents : obligatoire, supérieur à 0
- category : obligatoire, slug existant en base
- sizes : Tableau de code existant en base

Le stock pour chaque taille sera de 10.
En cas d'erreurs de validation, afficher un message à l'utilisateur et passer à la ligne suivante.
Créer le produit uniquement si l'option dry-run est à false.

Facultatif

Import des catégories

Le slug sera généré à partir du libellé
Règle de validation des catégories

- slug : obligatoire, unique
- libelle : obligatoire, longueur maximum 100

En cas d'erreurs de validation, afficher un message à l'utilisateur et passer à la ligne suivante.
Créer la catégorie uniquement si l'option dry-run est à false.

Import des tailles

Le code sera généré à partir du libellé
Règle de validation des tailles

- code : obligatoire, unique
- libelle : obligatoire, longueur maximum 6

En cas d'erreurs de validation, afficher un message à l'utilisateur et passer à la ligne suivante.
Créer la taille uniquement si l'option dry-run est à false.

Ajouter une option pour nettoyer la base de données avant l'import.