

Tâches (TODO) du formulaire de validation de création des produits

Principe de validation des champs

statut : état actuel

Les utilisateurs peuvent écrire ce qu'ils veulent dans les champs ; il n'y a pas d'indication ou de contraintes

Le contrôle des informations obligatoires pour créer un produit dans l'inventaire n'est pas implémenté.

Si le formulaire est vide quand on clique sur OK, une page vide s'ouvre (pas d'informations pour l'utilisateur).

Il faut :

- [ajouter des notifications](#) invitant les utilisateurs à corriger leurs erreurs ;
- [empêcher la validation](#) du formulaire si :
 - les champs obligatoires ne sont pas renseignés,
 - les champs ne sont pas valides (le respect des contraintes : le champ Prix ne doit contenir que des chiffres).

État actuel du code

La classe ProductController retourne des messages à la vue.

Les messages en question sont déjà définis.

La méthode Create(ProductViewModel product) dans la classe ProductController a été créée.

Elle est utilisée comme point de départ.

La méthode boucle sur une collection de messages d'erreur.

(modelErrors)

Les messages d'erreur doivent être fournis par une méthode dans la classe ProductService.

Pas encore créée, il faut définir l'implémentation, le nom, etc.

Quelques exemples se trouvent déjà dans la classe OrderController.

Réutiliser la logique d'ajout des erreurs dans le ModelState, principalement au niveau du Controller.

Tableau des spécifications du système de gestion de l'inventaire

Spécifications

Message d'erreur

1. Le produit doit porter un nom

MissingName (Nom manquant)

- | | |
|--|--|
| 2. Le champ Prix est obligatoire | MissingPrice (Prix manquant) |
| 3. Le prix doit être un nombre décimal | PriceNotANumber (Prix pas nombre) |
| 4. Le prix doit être supérieur à zéro | PriceNotGreaterThanZero (Prix pas supérieur à zéro) |
| 5. Le champ Quantité est obligatoire | MissingQuantity (Quantité manquante) |
| 6. La quantité doit être un nombre entier | QuantityNotAnInteger (Quantité pas nombre entier) |
| 7. La quantité doit être supérieure à zéro | QuantityNotGreaterThanZero (Quantité pas supérieure à zéro) |
-

Tests unitaires

statut : **non fait**

Fichier de test créé pour la nouvelle fonctionnalité

Il faut implémenter au moins une méthode par spécification.

J'ai commencé à écrire des tests dans la classe ProductServiceTests, mais les parties Act et Assert sont incomplètes.

Il faut écrire le code pour la partie Act, et paramétrer correctement les vérifications dans la partie Assert.

Tests d'intégration

statut : **non fait**

Vérifier si les modifications apportées aux produits du côté administrateur sont répercutées dans la liste des produits du côté client/utilisateur. Informations sur les produits mises à jour ? Produits disponibles pour l'achat/le panier ?

Il faut vérifier l'absence d'incohérences du côté admin lors de l'ajout/la suppression de produits.