

Documentation

Route vers l'Admin : <http://127.0.0.1:8000/editor/product/>

Mail : hh@gmail.com

Pass : azerty

HomeController

Namespace: App\Controller

Ce contrôleur gère les pages d'accueil, de visualisation des produits, de filtrage des produits et les produits mis en avant.

- **Routes:**
 - `/`: Page d'accueil.
 - `/home/product/{id}/show`: Affiche les détails d'un produit spécifique.
 - `/home/product/filter`: Filtre les produits en fonction de la fourchette de prix.
 - `/highlighted`: Affiche les produits mis en avant.
-

ProductController

Namespace: App\Controller

Ce contrôleur gère les opérations CRUD sur les produits, y compris la gestion des stocks et l'affichage de l'historique des ajouts de stock.

- **Routes:**
 - `/editor/product`: Liste des produits.
 - `/editor/product/boutique`: Affiche la boutique avec les produits.
 - `/editor/product/new`: Ajoute un nouveau produit.
 - `/editor/product/{id}`: Affiche les détails d'un produit spécifique.
 - `/editor/product/{id}/edit`: Édite un produit existant.
 - `/editor/product/{id}/delete`: Supprime un produit.
 - `/editor/product/add/product/{id}/stock`: Ajoute du stock à un produit.
 - `/editor/product/add/product/{id}/stock/history`: Affiche l'historique des ajouts de stock pour un produit.
-

RegistrationController

Namespace: App\Controller

Ce contrôleur gère l'inscription des utilisateurs.

- **Route:**
 - `/register`: Page d'inscription.
-

UserController

Namespace: `App\Controller`

Ce contrôleur gère la gestion des utilisateurs et leurs rôles.

- **Routes:**
 - `/admin/user`: Liste des utilisateurs.
 - `/admin/user/{id}/to/editor`: Attribue le rôle d'éditeur à un utilisateur.
 - `/admin/user/{id}/remove/editor/role`: Retire le rôle d'éditeur d'un utilisateur.
 - `/admin/user/{id}/remove/`: Supprime un utilisateur.
-

CartController

Namespace: `App\Controller`

Ce contrôleur gère les opérations sur le panier d'achat.

- **Routes:**
 - `/cart`: Affiche le panier.
 - `/cart/add/{id}/`: Ajoute un produit au panier.
 - `/cart/remove/{id}/{size}`: Supprime un produit spécifique du panier.
 - `/cart/remove/`: Vide le panier.
-

ProductType

Namespace: `App\Form`

Formulaire pour la création et la modification des produits.

- **Champs:**
 - `name`: Nom du produit.
 - `price`: Prix du produit.
 - `size`: Taille du produit.
 - `highlighted`: Checkbox pour mettre le produit en avant.
 - `stock`: Quantité en stock.
 - `image`: Image du produit.

Exemples de Fonctionnalités

Mise en avant d'un produit

1. Ajout d'un champ "highlighted" dans le formulaire de produit:

```
php
Copier le code
->add('highlighted', CheckboxType::class, [
    'label' => 'Mettre en avant',
    'required' => false,
])
```

2. Route pour afficher les produits mis en avant:

```
php
Copier le code
#[Route('/highlighted', name: 'app_product_highlighted', methods:
['GET'])]
public function highlightedProducts(ProductRepository
$productRepository): Response
{
    $highlightedProducts = $productRepository->findBy(['highlighted'
=> true]);

    return $this->render('home/index.html.twig', [
        'highlightedProducts' => $highlightedProducts,
        'page_title' => 'Highlighted Products',
    ]);
}
```

Filtrage des Produits

1. Route pour le filtrage des produits:

```
php
Copier le code
#[Route('/home/product/filter', name: 'app_home_product_filter',
methods: ['GET'])]
public function filter(Request $request, ProductRepository
$productRepository): Response
{
    // Récupérer tous les produits
    $products = $productRepository->findAll();

    // Filtrer les produits par fourchette de prix
    $minPrice = $request->query->get('min_price');
    $maxPrice = $request->query->get('max_price');

    // Effectuer le filtrage en fonction de la fourchette de prix
    if ($minPrice !== null && $maxPrice !== null) {
        $filteredProducts = array_filter($products, function
($product) use ($minPrice, $maxPrice) {
```

```
        return $product->getPrice() >= $minPrice && $product->getPrice() <= $maxPrice;
    });
    } else {
        // Si aucune fourchette de prix n'est spécifiée, retourner tous les produits
        $filteredProducts = $products;
    }

    return $this->render('product/boutique.html.twig', [
        'products' => $filteredProducts,
        'page_title' => 'Filtrage par Prix',
    ]);
}
```

Considérations de Sécurité

- **Sanitization et Validation:** Utiliser les fonctionnalités de Symfony pour la validation et la sanitization des données du formulaire.
 - **Permissions:** Assurer que les routes sensibles sont protégées par des contrôles d'accès appropriés (`is_granted`).
-

Conclusion

Ce projet Symfony offre une gestion complète des produits, des utilisateurs et du panier d'achat avec des fonctionnalités avancées telles que la mise en avant des produits et le filtrage des produits par prix. La structure est bien organisée avec des contrôleurs distincts pour chaque domaine de responsabilité, facilitant ainsi la maintenance et l'évolution du code.