

Documentation du code Symfony

Les dépendances que j'ai installé :

composer create-project symfony/skeleton stubborn_c

composer require symfony/maker-bundle --dev

composer require doctrine/orm

composer require security

composer require symfony/form

php bin/console make:user

php bin/console make:migration

composer require symfony/asset

composer require validator twig-bundle

composer require form orm security

symfony console make:registration

composer require symfonycasts/verify-email-bundle

composer require symfony/mailer

symfony console make:auth

AdminController

Ce contrôleur gère les fonctionnalités liées à l'administration du site.

Méthodes

index(Request \$request): Response

Cette méthode affiche le tableau de bord de l'administrateur et gère les actions telles que l'ajout, la modification et la suppression de produits.

Paramètres:

\$request: Requête HTTP.

Rôles nécessaires:

ROLE_ADMIN.

- Actions:
- Affiche le tableau de bord de l'administrateur.
- Gère les actions de l'utilisateur, y compris l'ajout, la modification et la suppression de produits.

Retourne:

- Une réponse HTTP.

AuthController

Ce contrôleur gère les fonctionnalités liées à l'authentification des utilisateurs.

Méthodes

forgotPassword(Request \$request): Response

Cette méthode permet à un utilisateur d'initier le processus de réinitialisation de mot de passe en soumettant son adresse e-mail.

Paramètres:

\$request: Requête HTTP.

Actions:

- Vérifie si l'e-mail soumis existe dans la base de données.
- Redirige vers une page de confirmation si l'e-mail est trouvé.

Retourne:

- Une réponse HTTP.

confirmResetPassword(Request \$request, string \$email, UserPasswordHasherInterface \$userPasswordHasher): Response

Cette méthode permet à un utilisateur de confirmer la réinitialisation de son mot de passe en soumettant un nouveau mot de passe.

Paramètres:

\$request: Requête HTTP.

\$email: Adresse e-mail de l'utilisateur.

\$userPasswordHasher: Interface pour hasher les mots de passe utilisateur.

Actions:

- Met à jour le mot de passe dans la base de données.
- Redirige vers la page de connexion.

Retourne:

- Une réponse HTTP.

CartController

Ce contrôleur gère les fonctionnalités liées au panier d'achat des utilisateurs.

Méthodes

cart(Request \$request): Response

Cette méthode affiche le contenu du panier d'achat de l'utilisateur.

Paramètres:

\$request: Requête HTTP

Actions:

Récupère les produits du panier depuis la session.

Calcule le total du panier.

Retourne:

- Une réponse HTTP

HomeController

Ce contrôleur gère les fonctionnalités liées à la page d'accueil du site.

Méthodes

index(): Response

Cette méthode affiche la page d'accueil du site avec les produits mis en avant.

Actions:

- Récupère les produits mis en avant depuis la base de données.
- Affiche la page d'accueil avec les produits mis en avant.

Retourne:

- Une réponse HTTP.

RegistrationController

Ce contrôleur gère les fonctionnalités liées à l'enregistrement des utilisateurs.

Méthodes

register(Request \$request, UserPasswordHasherInterface \$userPasswordHasher, EntityManagerInterface \$entityManager): Response

Cette méthode permet à un utilisateur de s'inscrire sur le site.

Paramètres:

\$request: Requête HTTP.

\$userPasswordHasher: Interface pour hasher les mots de passe utilisateur.

\$entityManager: Gestionnaire d'entités Doctrine.

Actions:

- Traite le formulaire d'inscription.
- Envoie un e-mail de confirmation.

Retourne:

- Une réponse HTTP.

verifyUserEmail(Request \$request, UserRepository \$userRepository): Response

Cette méthode permet à un utilisateur de vérifier son adresse e-mail après l'inscription.

Paramètres:

\$request: Requête HTTP.

\$userRepository: Référentiel des utilisateurs.

Actions:

- Valide le lien de confirmation d'e-mail.

Retourne:

- Une réponse HTTP.

SecurityController

Ce contrôleur gère les fonctionnalités liées à la sécurité, telles que la connexion et la déconnexion des utilisateurs.

Méthodes

login(AuthenticationUtils \$authenticationUtils): Response

Cette méthode affiche le formulaire de connexion et gère le processus de connexion des utilisateurs.

Paramètres:

\$authenticationUtils: Utilitaires d'authentification.

Actions:

- Affiche le formulaire de connexion.
- Gère les erreurs de connexion.

Retourne:

- Une réponse HTTP.

logout(): void

Cette méthode gère la déconnexion des utilisateurs.

Actions:

- Vide la session de l'utilisateur.

Retourne:

- Rien.

ShopController

Ce contrôleur gère les fonctionnalités liées à la boutique en ligne.

Méthodes

- index(Request \$request): Response
- Cette méthode affiche la liste des produits disponibles à l'achat.

Paramètres:

\$request: Requête HTTP.

Actions:

- Récupère les produits depuis la base de données.
- Filtre les produits par prix si une plage de prix est spécifiée.

Retourne:

- Une réponse HTTP.

addToCart(Request \$request, ProductRepository \$productRepository, \$id): Response

Cette méthode permet à un utilisateur d'ajouter un produit à son panier d'achat.

Paramètres:

\$request: Requête HTTP.

\$productRepository: Référentiel des produits.

\$id: Identifiant du produit à ajouter au panier.

Actions:

- Vérifie la disponibilité du produit dans la taille sélectionnée.
- Ajoute le produit au panier.

Retourne:

- Une réponse HTTP.

removeFromCart(Request \$request, int \$cartId): Response

Cette méthode permet à un utilisateur de retirer un produit de son panier d'achat.

Paramètres:

\$request: Requête HTTP.

\$cartId: Identifiant du produit dans le panier.

Actions:

- Supprime le produit du panier.

Retourne:

- Une réponse HTTP.