

# My Project

Généré par Doxygen 1.13.2



<b>1 Index des espaces de nommage</b>	<b>1</b>
1.1 Liste des espaces de nommage . . . . .	1
<b>2 Index hiérarchique</b>	<b>3</b>
2.1 Hiérarchie des classes . . . . .	3
<b>3 Index des classes</b>	<b>5</b>
3.1 Liste des classes . . . . .	5
<b>4 Index des fichiers</b>	<b>7</b>
4.1 Liste des fichiers . . . . .	7
<b>5 Documentation des espaces de nommage</b>	<b>9</b>
5.1 Référence de l'espace de nommage App . . . . .	9
5.2 Référence de l'espace de nommage App\Controller . . . . .	9
5.3 Référence de l'espace de nommage App\Entity . . . . .	10
5.4 Référence de l'espace de nommage App\Form . . . . .	10
5.5 Référence de l'espace de nommage App\Repository . . . . .	11
5.6 Référence de l'espace de nommage App\Security . . . . .	11
5.7 Référence de l'espace de nommage App\Service . . . . .	11
<b>6 Documentation des classes</b>	<b>13</b>
6.1 Référence de la classe App\Controller\AdminController . . . . .	13
6.1.1 Description détaillée . . . . .	13
6.1.2 Documentation des fonctions membres . . . . .	14
6.1.2.1 delete() . . . . .	14
6.1.2.2 edit() . . . . .	14
6.1.2.3 index() . . . . .	14
6.2 Référence de la classe App\Security\AuthAuthenticator . . . . .	15
6.2.1 Description détaillée . . . . .	16
6.2.2 Documentation des constructeurs et destructeur . . . . .	16
6.2.2.1 __construct() . . . . .	16
6.2.3 Documentation des fonctions membres . . . . .	16
6.2.3.1 authenticate() . . . . .	16
6.2.3.2 getLoginUrl() . . . . .	16
6.2.3.3 onAuthenticationSuccess() . . . . .	17
6.2.4 Documentation des données membres . . . . .	17
6.2.4.1 LOGIN_ROUTE . . . . .	17
6.3 Référence de la classe App\Controller\CartController . . . . .	17
6.3.1 Description détaillée . . . . .	18
6.3.2 Documentation des fonctions membres . . . . .	18
6.3.2.1 addToCart() . . . . .	18
6.3.2.2 cart() . . . . .	18
6.3.2.3 removeFromCart() . . . . .	18

6.4 Référence de la classe App\Form\ChangePasswordFormType . . . . .	19
6.4.1 Description détaillée . . . . .	19
6.4.2 Documentation des fonctions membres . . . . .	19
6.4.2.1 buildForm() . . . . .	19
6.4.2.2 configureOptions() . . . . .	20
6.5 Référence de la classe App\Security\EmailVerifier . . . . .	20
6.6 Référence de la classe App\Controller\HomeController . . . . .	20
6.6.1 Description détaillée . . . . .	20
6.6.2 Documentation des fonctions membres . . . . .	21
6.6.2.1 index() . . . . .	21
6.7 Référence de la classe App\Kernel . . . . .	21
6.8 Référence de la classe App\Controller\LoginController . . . . .	21
6.8.1 Documentation des fonctions membres . . . . .	22
6.8.1.1 login() . . . . .	22
6.9 Référence de la classe App\Controller\PaymentController . . . . .	22
6.9.1 Description détaillée . . . . .	23
6.9.2 Documentation des constructeurs et destructeur . . . . .	23
6.9.2.1 __construct() . . . . .	23
6.9.3 Documentation des fonctions membres . . . . .	23
6.9.3.1 cancel() . . . . .	23
6.9.3.2 checkout() . . . . .	23
6.9.3.3 success() . . . . .	24
6.9.4 Documentation des données membres . . . . .	24
6.9.4.1 \$session . . . . .	24
6.10 Référence de la classe App\Controller\ProductController . . . . .	24
6.10.1 Description détaillée . . . . .	24
6.10.2 Documentation des fonctions membres . . . . .	24
6.10.2.1 show() . . . . .	24
6.11 Référence de la classe App\Controller\RegistrationController . . . . .	25
6.11.1 Description détaillée . . . . .	25
6.11.2 Documentation des constructeurs et destructeur . . . . .	26
6.11.2.1 __construct() . . . . .	26
6.11.3 Documentation des fonctions membres . . . . .	26
6.11.3.1 register() . . . . .	26
6.11.3.2 verifyUserEmail() . . . . .	26
6.11.4 Documentation des données membres . . . . .	27
6.11.4.1 \$authenticator . . . . .	27
6.11.4.2 \$userAuthenticator . . . . .	27
6.12 Référence de la classe App\Form\RegistrationFormType . . . . .	27
6.12.1 Description détaillée . . . . .	28
6.12.2 Documentation des fonctions membres . . . . .	28
6.12.2.1 buildForm() . . . . .	28

6.12.2.2 configureOptions()	28
6.13 Référence de la classe App\Controller\ResetPasswordController	28
6.13.1 Description détaillée	29
6.13.2 Documentation des constructeurs et destructeur	29
6.13.2.1 __construct()	29
6.13.3 Documentation des fonctions membres	29
6.13.3.1 checkEmail()	29
6.13.3.2 processSendingPasswordResetEmail()	30
6.14 Référence de la classe App\Entity\ResetPasswordRequest	30
6.14.1 Description détaillée	31
6.14.2 Documentation des constructeurs et destructeur	31
6.14.2.1 __construct()	31
6.14.3 Documentation des fonctions membres	31
6.14.3.1 getId()	31
6.14.3.2 getUser()	32
6.14.4 Documentation des données membres	32
6.14.4.1 \$id	32
6.14.4.2 \$user	32
6.15 Référence de la classe App\Form\ResetPasswordRequestFormType	32
6.15.1 Description détaillée	33
6.15.2 Documentation des fonctions membres	33
6.15.2.1 buildForm()	33
6.15.2.2 configureOptions()	33
6.16 Référence de la classe App\Repository\ResetPasswordRequestRepository	33
6.16.1 Description détaillée	34
6.16.2 Documentation des constructeurs et destructeur	34
6.16.2.1 __construct()	34
6.16.3 Documentation des fonctions membres	34
6.16.3.1 createResetPasswordRequest()	34
6.17 Référence de la classe App\Controller\SecurityController	35
6.17.1 Description détaillée	35
6.17.2 Documentation des fonctions membres	35
6.17.2.1 login()	35
6.17.2.2 logout()	35
6.18 Référence de la classe App\Controller\ShopController	36
6.18.1 Description détaillée	36
6.18.2 Documentation des fonctions membres	36
6.18.2.1 index()	36
6.19 Référence de la classe App\Service\StripeService	37
6.19.1 Description détaillée	37
6.19.2 Documentation des constructeurs et destructeur	37
6.19.2.1 __construct()	37

6.19.3 Documentation des fonctions membres	37
6.19.3.1 createCheckoutSession()	37
6.20 Référence de la classe App\Entity\Sweatshirt	38
6.20.1 Description détaillée	40
6.20.2 Documentation des fonctions membres	40
6.20.2.1 getId()	40
6.20.2.2 getImage()	40
6.20.2.3 getName()	40
6.20.2.4 getPrice()	40
6.20.2.5 getStockL()	40
6.20.2.6 getStockM()	40
6.20.2.7 getStockS()	41
6.20.2.8 getStockXL()	41
6.20.2.9 getStockXS()	41
6.20.2.10 isFeatured()	41
6.20.2.11 setFeatured()	41
6.20.2.12 setImage()	41
6.20.2.13 setName()	41
6.20.2.14 setPrice()	42
6.20.2.15 setStockL()	42
6.20.2.16 setStockM()	42
6.20.2.17 setStockS()	42
6.20.2.18 setStockXL()	42
6.20.2.19 setStockXS()	42
6.20.3 Documentation des données membres	42
6.20.3.1 \$featured	42
6.20.3.2 \$id	43
6.20.3.3 \$image	43
6.20.3.4 \$name	43
6.20.3.5 \$price	43
6.20.3.6 \$stockL	43
6.20.3.7 \$stockM	43
6.20.3.8 \$stockS	43
6.20.3.9 \$stockXL	43
6.20.3.10 \$stockXS	43
6.21 Référence de la classe App\Repository\SweatshirtRepository	44
6.21.1 Description détaillée	44
6.21.2 Documentation des constructeurs et destructeur	44
6.21.2.1 __construct()	44
6.22 Référence de la classe App\Form\SweatshirtType	45
6.22.1 Description détaillée	45
6.22.2 Documentation des fonctions membres	45

6.22.2.1 buildForm()	45
6.22.2.2 configureOptions()	45
6.23 Référence de la classe App\Entity\User	46
6.23.1 Description détaillée	47
6.23.2 Documentation des fonctions membres	48
6.23.2.1 eraseCredentials()	48
6.23.2.2 getActivationToken()	48
6.23.2.3 getDeliveryAddress()	48
6.23.2.4 getEmail()	48
6.23.2.5 getId()	48
6.23.2.6 getPassword()	48
6.23.2.7 getRoles()	49
6.23.2.8 getUserIdentifier()	49
6.23.2.9 getUsername()	49
6.23.2.10 isVerified()	49
6.23.2.11 setActivationToken()	49
6.23.2.12 setDeliveryAddress()	49
6.23.2.13 setEmail()	50
6.23.2.14 setIsVerified()	50
6.23.2.15 setPassword()	50
6.23.2.16 setRoles()	50
6.23.2.17 setUsername()	50
6.23.3 Documentation des données membres	50
6.23.3.1 \$activationToken	50
6.23.3.2 \$deliveryAddress	51
6.23.3.3 \$email	51
6.23.3.4 \$id	51
6.23.3.5 \$isVerified	51
6.23.3.6 \$password	51
6.23.3.7 \$roles	51
6.23.3.8 \$username	51
6.24 Référence de la classe App\Repository\UserRepository	51
6.24.1 Description détaillée	52
6.24.2 Documentation des constructeurs et destructeur	52
6.24.2.1 __construct()	52
6.24.3 Documentation des fonctions membres	52
6.24.3.1 upgradePassword()	52
<b>7 Documentation des fichiers</b>	<b>55</b>
7.1 Référence du fichier src/Controller/AdminController.php	55
7.2 Référence du fichier src/Controller/CartController.php	55
7.3 Référence du fichier src/Controller/HomeController.php	55

7.4 Référence du fichier src/Controller/LoginController.php . . . . .	56
7.5 Référence du fichier src/Controller/PaymentController.php . . . . .	56
7.6 Référence du fichier src/Controller/ProductController.php . . . . .	56
7.7 Référence du fichier src/Controller/RegistrationController.php . . . . .	57
7.8 Référence du fichier src/Controller/ResetPasswordController.php . . . . .	57
7.9 Référence du fichier src/Controller/SecurityController.php . . . . .	57
7.10 Référence du fichier src/Controller/ShopController.php . . . . .	57
7.11 Référence du fichier src/Entity/ResetPasswordRequest.php . . . . .	58
7.12 Référence du fichier src/Entity/Sweatshirt.php . . . . .	58
7.13 Référence du fichier src/Entity/User.php . . . . .	58
7.14 Référence du fichier src/Form/ChangePasswordFormType.php . . . . .	59
7.15 Référence du fichier src/Form/RegistrationFormType.php . . . . .	59
7.16 Référence du fichier src/Form/ResetPasswordRequestFormType.php . . . . .	59
7.17 Référence du fichier src/Form/SweatshirtType.php . . . . .	59
7.18 Référence du fichier src/Kernel.php . . . . .	60
7.19 Référence du fichier src/Repository/ResetPasswordRequestRepository.php . . . . .	60
7.20 Référence du fichier src/Repository/SweatshirtRepository.php . . . . .	60
7.21 Référence du fichier src/Repository/UserRepository.php . . . . .	61
7.22 Référence du fichier src/Security/AuthAuthenticator.php . . . . .	61
7.23 Référence du fichier src/Security/EmailVerifier.php . . . . .	61
7.24 Référence du fichier src/Service/CartService.php . . . . .	61
7.25 Référence du fichier src/Service/StripeService.php . . . . .	61



# Chapitre 1

## Index des espaces de nommage

### 1.1 Liste des espaces de nommage

Liste de tous les espaces de nommage avec une brève description:

<a href="#">App</a> . . . . .	9
<a href="#">App\Controller</a> . . . . .	9
<a href="#">App\Entity</a> . . . . .	10
<a href="#">App\Form</a> . . . . .	10
<a href="#">App\Repository</a> . . . . .	11
<a href="#">App\Security</a> . . . . .	11
<a href="#">App\Service</a> . . . . .	11



## Chapitre 2

# Index hiérarchique

### 2.1 Hiérarchie des classes

Cette liste d'héritage est classée approximativement par ordre alphabétique :

AbstractController	
App\Controller\AdminController . . . . .	13
App\Controller\CartController . . . . .	17
App\Controller\HomeController . . . . .	20
App\Controller>LoginController . . . . .	21
App\Controller\PaymentController . . . . .	22
App\Controller\ProductController . . . . .	24
App\Controller\RegistrationController . . . . .	25
App\Controller\ResetPasswordController . . . . .	28
App\Controller\SecurityController . . . . .	35
App\Controller\ShopController . . . . .	36
AbstractLoginFormAuthenticator	
App\Security\AuthAuthenticator . . . . .	15
AbstractType	
App\Form\ChangePasswordFormType . . . . .	19
App\Form\RegistrationFormType . . . . .	27
App\Form\ResetPasswordRequestFormType . . . . .	32
App\Form\SweatshirtType . . . . .	45
BaseKernel	
App\Kernel . . . . .	21
App\Security>EmailVerifier . . . . .	20
PasswordAuthenticatedUserInterface	
App\Entity\User . . . . .	46
PasswordUpgraderInterface	
App\Repository\UserRepository . . . . .	51
ResetPasswordRequestInterface	
App\Entity\ResetPasswordRequest . . . . .	30
ResetPasswordRequestRepositoryInterface	
App\Repository\ResetPasswordRequestRepository . . . . .	33
ServiceEntityRepository	
App\Repository\ResetPasswordRequestRepository . . . . .	33
App\Repository\ResetPasswordRequestRepository . . . . .	33
App\Repository\SweatshirtRepository . . . . .	44
App\Repository\SweatshirtRepository . . . . .	44
App\Repository\UserRepository . . . . .	51

App\Repository\UserRepository . . . . .	51
App\Service\StripeService . . . . .	37
App\Entity\Sweatshirt . . . . .	38
UserInterface	
App\Entity\User . . . . .	46

## Chapitre 3

# Index des classes

### 3.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

<a href="#">App\Controller\AdminController</a>	
Contrôleur responsable de la gestion des opérations administratives sur les sweat-shirts . . .	13
<a href="#">App\Security\AuthAuthenticator</a>	
Authenticator personnalisé pour la connexion des utilisateurs . . . . .	15
<a href="#">App\Controller\CartController</a>	
Contrôleur responsable de la gestion du panier d'achats . . . . .	17
<a href="#">App\Form\ChangePasswordFormType</a>	
Formulaire permettant à un utilisateur de changer son mot de passe . . . . .	19
<a href="#">App\Security&gt;EmailVerifier</a>	
. . . . .	20
<a href="#">App\Controller\HomeController</a>	
Contrôleur responsable de l'affichage de la page d'accueil . . . . .	20
<a href="#">App\Kernel</a>	
. . . . .	21
<a href="#">App\Controller&gt;LoginController</a>	
. . . . .	21
<a href="#">App\Controller\PaymentController</a>	
Contrôleur responsable de la gestion des paiements via Stripe . . . . .	22
<a href="#">App\Controller\ProductController</a>	
Contrôleur responsable de l'affichage des détails d'un sweat-shirt . . . . .	24
<a href="#">App\Controller\RegistrationController</a>	
Contrôleur de gestion des inscriptions et de la vérification des emails . . . . .	25
<a href="#">App\Form\RegistrationFormType</a>	
Formulaire d'inscription d'un nouvel utilisateur . . . . .	27
<a href="#">App\Controller\ResetPasswordController</a>	
Contrôleur gérant le processus de réinitialisation de mot de passe utilisateur . . . . .	28
<a href="#">App\Entity\ResetPasswordRequest</a>	
Entité représentant une demande de réinitialisation de mot de passe . . . . .	30
<a href="#">App\Form\ResetPasswordRequestFormType</a>	
Formulaire de demande de réinitialisation de mot de passe . . . . .	32
<a href="#">App\Repository\ResetPasswordRequestRepository</a>	
Repository pour l'entité ResetPasswordRequest . . . . .	33
<a href="#">App\Controller\SecurityController</a>	
Contrôleur de sécurité gérant l'authentification des utilisateurs . . . . .	35
<a href="#">App\Controller\ShopController</a>	
Contrôleur de la boutique affichant la liste des sweatshirts . . . . .	36
<a href="#">App\Service\StripeService</a>	
Service pour interagir avec l'API Stripe, en particulier pour créer une session de paiement . . .	37

<a href="#">App\Entity\Sweatshirt</a>	
Représente un sweat-shirt disponible dans la boutique . . . . .	38
<a href="#">App\Repository\SweatshirtRepository</a>	
Repository pour l'entité Sweatshirt . . . . .	44
<a href="#">App\Form\SweatshirtType</a>	
Formulaire pour la création ou modification d'un Sweatshirt . . . . .	45
<a href="#">App\Entity\User</a>	
Représente un utilisateur de l'application . . . . .	46
<a href="#">App\Repository\UserRepository</a>	
Repository pour l'entité User . . . . .	51

# Chapitre 4

## Index des fichiers

### 4.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

src/Kernel.php	60
src/Controller/AdminController.php	55
src/Controller/CartController.php	55
src/Controller/HomeController.php	55
src/Controller/LoginController.php	56
src/Controller/PaymentController.php	56
src/Controller/ProductController.php	56
src/Controller/RegistrationController.php	57
src/Controller/ResetPasswordController.php	57
src/Controller/SecurityController.php	57
src/Controller/ShopController.php	57
src/Entity/ResetPasswordRequest.php	58
src/Entity/Sweatshirt.php	58
src/Entity/User.php	58
src/Form/ChangePasswordFormType.php	59
src/Form/RegistrationFormType.php	59
src/Form/ResetPasswordRequestFormType.php	59
src/Form/SweatshirtType.php	59
src/Repository/ResetPasswordRequestRepository.php	60
src/Repository/SweatshirtRepository.php	60
src/Repository/UserRepository.php	61
src/Security/AuthAuthenticator.php	61
src/Security/EmailVerifier.php	61
src/Service/CartService.php	61
src/Service/StripeService.php	61





## Chapitre 5

# Documentation des espaces de nommage

### 5.1 Référence de l'espace de nommage App

#### Espaces de nommage

- namespace [Controller](#)
- namespace [Entity](#)
- namespace [Form](#)
- namespace [Repository](#)
- namespace [Security](#)
- namespace [Service](#)

#### Classes

- class [Kernel](#)

### 5.2 Référence de l'espace de nommage App\Controller

#### Classes

- class [AdminController](#)  
*Contrôleur responsable de la gestion des opérations administratives sur les sweat-shirts.*
- class [CartController](#)  
*Contrôleur responsable de la gestion du panier d'achats.*
- class [HomeController](#)  
*Contrôleur responsable de l'affichage de la page d'accueil.*

— class [LoginController](#)

— class [PaymentController](#)

*Contrôleur responsable de la gestion des paiements via Stripe.*

— class [ProductController](#)

*Contrôleur responsable de l'affichage des détails d'un sweat-shirt.*

— class [RegistrationController](#)

*Contrôleur de gestion des inscriptions et de la vérification des emails.*

— class [ResetPasswordController](#)

*Contrôleur gérant le processus de réinitialisation de mot de passe utilisateur.*

— class [SecurityController](#)

*Contrôleur de sécurité gérant l'authentification des utilisateurs.*

— class [ShopController](#)

*Contrôleur de la boutique affichant la liste des sweatshirts.*

## 5.3 Référence de l'espace de nommage App\Entity

### Classes

— class [ResetPasswordRequest](#)

*Entité représentant une demande de réinitialisation de mot de passe.*

— class [Sweatshirt](#)

*Représente un sweat-shirt disponible dans la boutique.*

— class [User](#)

*Représente un utilisateur de l'application.*

## 5.4 Référence de l'espace de nommage App\Form

### Classes

— class [ChangePasswordFormType](#)

*Formulaire permettant à un utilisateur de changer son mot de passe.*

— class [RegistrationFormType](#)

*Formulaire d'inscription d'un nouvel utilisateur.*

- class [ResetPasswordRequestFormType](#)

*Formulaire de demande de réinitialisation de mot de passe.*

- class [SweatshirtType](#)

*Formulaire pour la création ou modification d'un Sweatshirt.*

## 5.5 Référence de l'espace de nommage App\Repository

### Classes

- class [ResetPasswordRequestRepository](#)

*Repository pour l'entité ResetPasswordRequest.*

- class [SweatshirtRepository](#)

*Repository pour l'entité Sweatshirt.*

- class [UserRepository](#)

*Repository pour l'entité User.*

## 5.6 Référence de l'espace de nommage App\Security

### Classes

- class [AuthAuthenticator](#)

*Authenticator personnalisé pour la connexion des utilisateurs.*

- class [EmailVerifier](#)

## 5.7 Référence de l'espace de nommage App\Service

### Classes

- class [StripeService](#)

*Service pour interagir avec l'API Stripe, en particulier pour créer une session de paiement.*



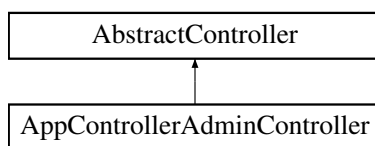
## Chapitre 6

# Documentation des classes

### 6.1 Référence de la classe App\Controller\AdminController

Contrôleur responsable de la gestion des opérations administratives sur les sweat-shirts.

Graphe d'héritage de App\Controller\AdminController:



#### Fonctions membres publiques

- [index](#) (Request \$request, EntityManagerInterface \$em, SluggerInterface \$slugger)  
*Affiche la liste des sweat-shirts et permet d'ajouter un nouveau sweat-shirt.*
- [edit](#) ([Sweatshirt](#) \$sweatshirt, Request \$request, EntityManagerInterface \$em, SluggerInterface \$slugger)  
*Permet de modifier un sweat-shirt existant.*
- [delete](#) ([Sweatshirt](#) \$sweatshirt, EntityManagerInterface \$em)  
*Permet de supprimer un sweat-shirt de la base de données.*

#### 6.1.1 Description détaillée

Contrôleur responsable de la gestion des opérations administratives sur les sweat-shirts.

## 6.1.2 Documentation des fonctions membres

### 6.1.2.1 delete()

```
App\Controller\AdminController::delete (
    Sweatshirt $sweatshirt,
    EntityManagerInterface $em)
```

Permet de supprimer un sweat-shirt de la base de données.

Cette méthode gère la suppression d'un sweat-shirt spécifique.

`@Route("/admin/delete/{id}", name="app_admin_delete")`

#### Paramètres

Sweatshirt	<i>\$sweatshirt</i>	L'entité sweat-shirt à supprimer.
EntityManagerInterface	<i>\$em</i>	L'interface pour gérer les entités Doctrine.

#### Renvoie

Response La réponse redirigeant vers la page d'administration après la suppression.

### 6.1.2.2 edit()

```
App\Controller\AdminController::edit (
    Sweatshirt $sweatshirt,
    Request $request,
    EntityManagerInterface $em,
    SluggerInterface $slugger)
```

Permet de modifier un sweat-shirt existant.

Cette méthode gère l'édition des données du sweat-shirt et le téléchargement d'une nouvelle image.

`@Route("/admin/edit/{id}", name="app_admin_edit")`

#### Paramètres

Sweatshirt	<i>\$sweatshirt</i>	L'entité sweat-shirt à modifier.
Request	<i>\$request</i>	La requête HTTP.
EntityManagerInterface	<i>\$em</i>	L'interface pour gérer les entités Doctrine.
SluggerInterface	<i>\$slugger</i>	Le service pour générer un slug à partir du nom de fichier.

#### Renvoie

Response La réponse contenant la vue avec le formulaire de modification.

### 6.1.2.3 index()

```
App\Controller\AdminController::index (
    Request $request,
    EntityManagerInterface $em,
    SluggerInterface $slugger)
```

Affiche la liste des sweat-shirts et permet d'ajouter un nouveau sweat-shirt.

Cette méthode gère le formulaire d'ajout de sweat-shirt et le téléchargement de l'image.

`@Route("/admin", name="app_admin")`

## Paramètres

Request	<i>\$request</i>	La requête HTTP.
EntityManagerInterface	<i>\$em</i>	L'interface pour gérer les entités Doctrine.
SluggerInterface	<i>\$slugger</i>	Le service pour générer un slug à partir du nom de fichier.

## Renvoie

Response La réponse contenant la vue du formulaire et la liste des sweat-shirts.

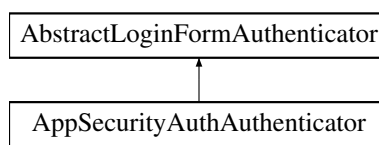
La documentation de cette classe a été générée à partir du fichier suivant :

— src/Controller/[AdminController.php](#)

## 6.2 Référence de la classe App\Security\AuthAuthenticator

Authenticator personnalisé pour la connexion des utilisateurs.

Graphe d'héritage de App\Security\AuthAuthenticator:



## Fonctions membres publiques

- [\\_\\_construct](#) (private UrlGeneratorInterface \$urlGenerator)
- [authenticate](#) (Request \$request)  
*Crée le Passport utilisé pour authentifier un utilisateur.*
- [onAuthenticationSuccess](#) (Request \$request, TokenInterface \$token, string \$firewallName)  
*Gère la redirection après une authentification réussie.*

## Attributs publics

- const [LOGIN\\_ROUTE](#) = 'app\_login'

## Fonctions membres protégées

- [getLoginUrl](#) (Request \$request)  
*Retourne l'URL de la page de connexion.*

## 6.2.1 Description détaillée

Authenticator personnalisé pour la connexion des utilisateurs.

Étend `AbstractLoginFormAuthenticator` pour gérer le processus d'authentification via un formulaire.

## 6.2.2 Documentation des constructeurs et destructeur

### 6.2.2.1 `__construct()`

```
App\Security\AuthAuthenticator::__construct (
    private UrlGeneratorInterface $urlGenerator)
```

## 6.2.3 Documentation des fonctions membres

### 6.2.3.1 `authenticate()`

```
App\Security\AuthAuthenticator::authenticate (
    Request $request)
```

Crée le Passport utilisé pour authentifier un utilisateur.

Ce Passport contient :

- le badge `UserBadge` avec le nom d'utilisateur
- les credentials de mot de passe
- un badge `CSRF` pour vérifier le token
- un badge `RememberMe` pour activer l'option "se souvenir de moi"

Paramètres

Request	<i>\$request</i>	
---------	------------------	--

Renvoie

Passport

### 6.2.3.2 `getLoginUrl()`

```
App\Security\AuthAuthenticator::getLoginUrl (
    Request $request) [protected]
```

Retourne l'URL de la page de connexion.

Utilisé pour rediriger l'utilisateur si l'authentification échoue.

Paramètres

Request	<i>\$request</i>	
---------	------------------	--

Renvoie

string



### 6.2.3.3 onAuthenticationSuccess()

```
App\Security\AuthAuthenticator::onAuthenticationSuccess (
    Request $request,
    TokenInterface $token,
    string $firewallName)
```

Gère la redirection après une authentification réussie.

Redirige vers la page initialement demandée (targetPath), si elle existe, sinon vers la page d'accueil.

#### Paramètres

Request	<i>\$request</i>	
TokenInterface	<i>\$token</i>	
string	<i>\$firewallName</i>	

#### Renvoie

Response|null

## 6.2.4 Documentation des données membres

### 6.2.4.1 LOGIN\_ROUTE

```
const App\Security\AuthAuthenticator::LOGIN_ROUTE = 'app_login'
```

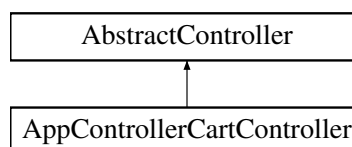
La documentation de cette classe a été générée à partir du fichier suivant :

— src/Security/[AuthAuthenticator.php](#)

## 6.3 Référence de la classe App\Controller\CartController

Contrôleur responsable de la gestion du panier d'achats.

Graphe d'héritage de App\Controller\CartController:



### Fonctions membres publiques

— [addToCart](#) (Request \$request, [Sweatshirt](#) \$sweatshirt, SessionInterface \$session)

*Permet d'ajouter un sweat-shirt au panier.*

— [cart](#) (SessionInterface \$session)

*Affiche les articles dans le panier et le total de la commande.*

— [removeFromCart](#) (SessionInterface \$session, int \$index, Request \$request)

*Permet de supprimer un sweat-shirt du panier.*

### 6.3.1 Description détaillée

Contrôleur responsable de la gestion du panier d'achats.

### 6.3.2 Documentation des fonctions membres

#### 6.3.2.1 addToCart()

```
App\Controller\CartController::addToCart (
    Request $request,
    Sweatshirt $sweatshirt,
    SessionInterface $session)
```

Permet d'ajouter un sweat-shirt au panier.

Cette méthode récupère les informations du sweat-shirt, la taille choisie par l'utilisateur, et les ajoute dans le panier stocké en session.

```
@Route("/cart/add/{id}", name="app_cart_add", methods={"POST"})
```

##### Paramètres

Request	<i>\$request</i>	La requête HTTP contenant les informations envoyées par le formulaire.
Sweatshirt	<i>\$sweatshirt</i>	L'entité sweat-shirt à ajouter au panier.
SessionInterface	<i>\$session</i>	L'interface pour gérer les données de session.

##### Renvoie

Response Redirige vers la page du panier après l'ajout du sweat-shirt.

#### 6.3.2.2 cart()

```
App\Controller\CartController::cart (
    SessionInterface $session)
```

Affiche les articles dans le panier et le total de la commande.

Cette méthode récupère les éléments du panier depuis la session et calcule le total.

```
@Route("/cart", name="app_cart")
```

##### Paramètres

SessionInterface	<i>\$session</i>	L'interface pour gérer les données de session.
------------------	------------------	--

##### Renvoie

Response Affiche la vue du panier avec les articles et le total.

#### 6.3.2.3 removeFromCart()

```
App\Controller\CartController::removeFromCart (
    SessionInterface $session,
    int $index,
    Request $request)
```

Permet de supprimer un sweat-shirt du panier.

Cette méthode récupère le panier depuis la session et supprime l'élément à l'index spécifié. Elle vérifie également le token CSRF pour assurer la sécurité de la suppression.

```
@Route("/cart/remove/{index}", name="app_cart_remove", methods={"POST"})
```

## Paramètres

SessionInterface	<i>\$session</i>	L'interface pour gérer les données de session.
int	<i>\$index</i>	L'index de l'article à supprimer dans le panier.
Request	<i>\$request</i>	La requête HTTP contenant le token CSRF.

## Renvoie

Response Redirige vers la page du panier après la suppression de l'article.

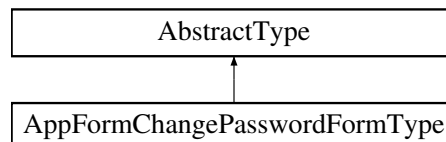
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/Controller/CartController.php](#)

## 6.4 Référence de la classe App\Form\ChangePasswordFormType

Formulaire permettant à un utilisateur de changer son mot de passe.

Graphes d'héritage de App\Form\ChangePasswordFormType:



## Fonctions membres publiques

— [buildForm](#) (FormBuilderInterface \$builder, array \$options)

*Construit le formulaire de changement de mot de passe.*

— [configureOptions](#) (OptionsResolver \$resolver)

*Configure les options par défaut du formulaire.*

### 6.4.1 Description détaillée

Formulaire permettant à un utilisateur de changer son mot de passe.

Ce formulaire utilise un champ de mot de passe répété pour forcer la confirmation du mot de passe. Le mot de passe doit avoir une longueur minimale de 8 caractères et ne doit pas être vide. Le champ n'est pas mappé à une propriété de l'entité (mapped: false), car l'encodage du mot de passe est généralement géré manuellement dans le contrôleur.

### 6.4.2 Documentation des fonctions membres

#### 6.4.2.1 buildForm()

```

App\Form\ChangePasswordFormType::buildForm (
    FormBuilderInterface $builder,
    array $options)
  
```

Construit le formulaire de changement de mot de passe.

## Paramètres

FormBuilderInterface	<i>\$builder</i>	Le constructeur de formulaire
array	<i>\$options</i>	Les options du formulaire

## 6.4.2.2 configureOptions()

```
App\Form\ChangePasswordFormType::configureOptions (
    OptionsResolver $resolver)
```

Configure les options par défaut du formulaire.

## Paramètres

OptionsResolver	<i>\$resolver</i>	Le résolveur d'options
-----------------	-------------------	------------------------

La documentation de cette classe a été générée à partir du fichier suivant :

— [src/Form/ChangePasswordFormType.php](#)

## 6.5 Référence de la classe App\Security&gt;EmailVerifier

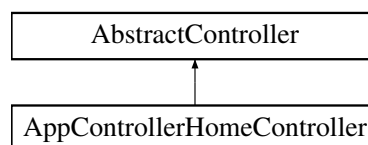
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/Security/EmailVerifier.php](#)

## 6.6 Référence de la classe App\Controller\HomeController

Contrôleur responsable de l'affichage de la page d'accueil.

Graphe d'héritage de App\Controller\HomeController:



## Fonctions membres publiques

— [index](#) ([SweatshirtRepository](#) \$sweatshirtRepository)

*Affiche la page d'accueil avec les sweat-shirts en avant.*

## 6.6.1 Description détaillée

Contrôleur responsable de l'affichage de la page d'accueil.

## 6.6.2 Documentation des fonctions membres

### 6.6.2.1 index()

```
App\Controller\HomeController::index (
    SweatshirtRepository $sweatshirtRepository)
```

Affiche la page d'accueil avec les sweat-shirts en avant.

Cette méthode récupère les sweat-shirts qui sont marqués comme "en avant" dans la base de données et les affiche sur la page d'accueil.

```
@Route("/", name="app_home")
```

#### Paramètres

SweatshirtRepository	<i>\$sweatshirtRepository</i>	Le repository des sweat-shirts pour interroger la base de données.
----------------------	-------------------------------	--

#### Renvoie

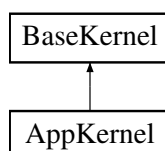
Response La réponse HTTP qui contient la vue avec les sweat-shirts en avant.

La documentation de cette classe a été générée à partir du fichier suivant :

— [src/Controller/HomeController.php](#)

## 6.7 Référence de la classe App\Kernel

Graphe d'héritage de App\Kernel:

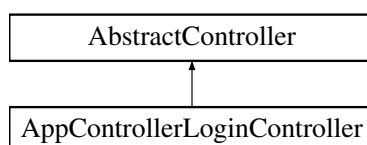


La documentation de cette classe a été générée à partir du fichier suivant :

— [src/Kernel.php](#)

## 6.8 Référence de la classe App\Controller\LoginController

Graphe d'héritage de App\Controller\LoginController:



## Fonctions membres publiques

— [login](#) ()

## 6.8.1 Documentation des fonctions membres

### 6.8.1.1 login()

`App\Controller\LoginController::login ()`

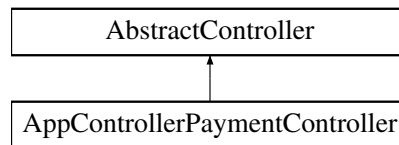
La documentation de cette classe a été générée à partir du fichier suivant :

— `src/Controller/LoginController.php`

## 6.9 Référence de la classe App\Controller\PaymentController

Contrôleur responsable de la gestion des paiements via Stripe.

Graphe d'héritage de App\Controller\PaymentController:



## Fonctions membres publiques

— [\\_\\_construct](#) (RequestStack \$requestStack)

*Constructeur du contrôleur de paiement.*

— [checkout](#) (StripeService \$stripeService)

*Crée une session de paiement Stripe à partir des éléments du panier.*

— [success](#) ()

*Affiche la page de confirmation après un paiement réussi.*

— [cancel](#) ()

*Affiche la page d'annulation de paiement si l'utilisateur quitte Stripe sans valider.*

## Attributs privés

— [\\$session](#)

## 6.9.1 Description détaillée

Contrôleur responsable de la gestion des paiements via Stripe.

## 6.9.2 Documentation des constructeurs et destructeur

### 6.9.2.1 \_\_construct()

```
App\Controller\PaymentController::__construct (  
    RequestStack $requestStack)
```

Constructeur du contrôleur de paiement.

#### Paramètres

RequestStack	<i>\$requestStack</i>	Pile de requêtes pour accéder à la session.
--------------	-----------------------	---

## 6.9.3 Documentation des fonctions membres

### 6.9.3.1 cancel()

```
App\Controller\PaymentController::cancel ()
```

Affiche la page d'annulation de paiement si l'utilisateur quitte Stripe sans valider.

```
@Route("/cancel", name="app_payment_cancel")
```

#### Renvoie

Response La vue d'annulation du paiement.

### 6.9.3.2 checkout()

```
App\Controller\PaymentController::checkout (  
    StripeService $stripeService)
```

Crée une session de paiement Stripe à partir des éléments du panier.

Redirige vers la page de paiement Stripe si le panier contient des articles, sinon redirige vers la page du panier avec un message d'avertissement.

```
@Route("/create-checkout-session", name="app_create_checkout_session")
```

#### Paramètres

StripeService	<i>\$stripeService</i>	Service personnalisé pour la gestion de Stripe.
---------------	------------------------	---

#### Renvoie

Response Redirection vers l'URL de paiement Stripe ou la page panier.

### 6.9.3.3 success()

```
App\Controller\PaymentController::success ()
```

Affiche la page de confirmation après un paiement réussi.

Vide le panier de la session.

```
@Route("/success", name="app_payment_success")
```

Renvoie

Response La vue de confirmation du paiement.

## 6.9.4 Documentation des données membres

### 6.9.4.1 \$session

```
App\Controller\PaymentController::$session [private]
```

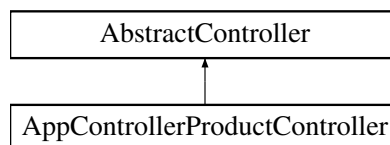
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/Controller/PaymentController.php](#)

## 6.10 Référence de la classe App\Controller\ProductController

Contrôleur responsable de l'affichage des détails d'un sweat-shirt.

Graphe d'héritage de App\Controller\ProductController:



### Fonctions membres publiques

— [show](#) ([Sweatshirt](#) \$sweatshirt)

*Affiche la fiche produit d'un sweat-shirt.*

### 6.10.1 Description détaillée

Contrôleur responsable de l'affichage des détails d'un sweat-shirt.

## 6.10.2 Documentation des fonctions membres

### 6.10.2.1 show()

```
App\Controller\ProductController::show (
    Sweatshirt $sweatshirt)
```

Affiche la fiche produit d'un sweat-shirt.

Cette méthode récupère un sweat-shirt à partir de son identifiant (grâce au ParamConverter de Symfony) et retourne une vue détaillée.

```
@Route("/product/{id}", name="app_product")
```



## Paramètres

Sweatshirt	<code>\$sweatshirt</code>	L'entité du sweat-shirt à afficher.
------------	---------------------------	-------------------------------------

## Renvoi

Response La vue contenant les détails du sweat-shirt.

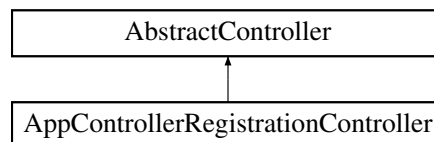
La documentation de cette classe a été générée à partir du fichier suivant :

— `src/Controller/ProductController.php`

## 6.11 Référence de la classe App\Controller\RegistrationController

Contrôleur de gestion des inscriptions et de la vérification des emails.

Graphe d'héritage de App\Controller\RegistrationController:



## Fonctions membres publiques

— `__construct` (UserAuthenticatorInterface \$userAuthenticator, AuthAuthenticator \$authenticator)

*Constructeur.*

— `register` (Request \$request, UserPasswordHasherInterface \$userPasswordHasher, EntityManagerInterface \$entityManager, TokenGeneratorInterface \$tokenGenerator, MailerInterface \$mailer)

*Page d'inscription et envoi de l'email de vérification.*

— `verifyUserEmail` (int \$id, string \$token, string \$signature, int \$expires, Request \$request, TranslatorInterface \$translator, EntityManagerInterface \$entityManager,)

*Vérifie le lien de confirmation d'email et connecte l'utilisateur.*

## Attributs privés

— UserAuthenticatorInterface `$userAuthenticator`

— AuthAuthenticator `$authenticator`

### 6.11.1 Description détaillée

Contrôleur de gestion des inscriptions et de la vérification des emails.

## 6.11.2 Documentation des constructeurs et destructeur

### 6.11.2.1 \_\_construct()

```
App\Controller\RegistrationController::__construct (
    UserAuthenticatorInterface $userAuthenticator,
    AuthAuthenticator $authenticator)
```

Constructeur.

Paramètres

UserAuthenticatorInterface	<i>\$userAuthenticator</i>	Gestion de l'authentification utilisateur.
AuthAuthenticator	<i>\$authenticator</i>	Authenticator personnalisé.

## 6.11.3 Documentation des fonctions membres

### 6.11.3.1 register()

```
App\Controller\RegistrationController::register (
    Request $request,
    UserPasswordHasherInterface $userPasswordHasher,
    EntityManagerInterface $entityManager,
    TokenGeneratorInterface $tokenGenerator,
    MailerInterface $mailer)
```

Page d'inscription et envoi de l'email de vérification.

```
@Route("/register", name="app_register")
```

Paramètres

Request	<i>\$request</i>	Requête HTTP.
UserPasswordHasherInterface	<i>\$userPasswordHasher</i>	Service de hachage du mot de passe.
EntityManagerInterface	<i>\$entityManager</i>	Gestionnaire d'entité Doctrine.
TokenGeneratorInterface	<i>\$tokenGenerator</i>	Générateur de jetons sécurisés.
MailerInterface	<i>\$mailer</i>	Service d'envoi de mails.

Renvoie

Response La vue d'inscription ou de vérification.

### 6.11.3.2 verifyUserEmail()

```
App\Controller\RegistrationController::verifyUserEmail (
    int $id,
    string $token,
    string $signature,
    int $expires,
    Request $request,
    TranslatorInterface $translator,
    EntityManagerInterface $entityManager)
```

Vérifie le lien de confirmation d'email et connecte l'utilisateur.

```
@Route("/verify/email/{id}/{token}/{signature}/{expires}", name="app_verify_email")
```

## Paramètres

int	<i>\$id</i>	Identifiant de l'utilisateur.
string	<i>\$token</i>	Jeton de vérification.
string	<i>\$signature</i>	Signature de sécurité.
int	<i>\$expires</i>	Timestamp d'expiration du lien.
Request	<i>\$request</i>	Requête HTTP.
TranslatorInterface	<i>\$translator</i>	Service de traduction (non utilisé ici).
EntityManagerInterface	<i>\$entityManager</i>	Gestionnaire d'entité.

## Renvoie

Response Redirection vers la page d'accueil ou d'erreur.

## 6.11.4 Documentation des données membres

## 6.11.4.1 \$authenticator

```
AuthAuthenticator App\Controller\RegistrationController::$authenticator [private]
```

## 6.11.4.2 \$userAuthenticator

```
UserAuthenticatorInterface App\Controller\RegistrationController::$userAuthenticator [private]
```

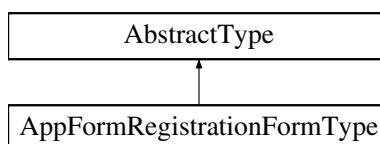
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/Controller/RegistrationController.php](#)

## 6.12 Référence de la classe App\Form\RegistrationFormType

Formulaire d'inscription d'un nouvel utilisateur.

Graphes d'héritage de App\Form\RegistrationFormType:



## Fonctions membres publiques

— [buildForm](#) (FormBuilderInterface \$builder, array \$options)

*Construit le formulaire d'inscription.*

— [configureOptions](#) (OptionsResolver \$resolver)

*Configure les options par défaut du formulaire, notamment la classe de données liée.*

### 6.12.1 Description détaillée

Formulaire d'inscription d'un nouvel utilisateur.

Ce formulaire permet de collecter les informations nécessaires à la création d'un compte utilisateur : email, nom d'utilisateur, adresse de livraison et mot de passe (avec confirmation). Les mots de passe ne sont pas mappés à l'entité User afin d'être encodés dans le contrôleur.

### 6.12.2 Documentation des fonctions membres

#### 6.12.2.1 buildForm()

```
App\Form\RegistrationFormType::buildForm (
    FormBuilderInterface $builder,
    array $options)
```

Construit le formulaire d'inscription.

##### Paramètres

FormBuilderInterface	<i>\$builder</i>	Le constructeur de formulaire
array	<i>\$options</i>	Les options passées au formulaire

#### 6.12.2.2 configureOptions()

```
App\Form\RegistrationFormType::configureOptions (
    OptionsResolver $resolver)
```

Configure les options par défaut du formulaire, notamment la classe de données liée.

##### Paramètres

OptionsResolver	<i>\$resolver</i>	Le résolveur d'options
-----------------	-------------------	------------------------

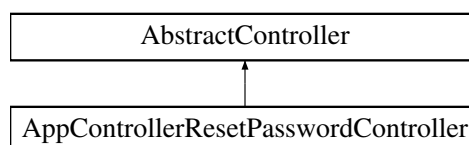
La documentation de cette classe a été générée à partir du fichier suivant :

— src/Form/[RegistrationFormType.php](#)

## 6.13 Référence de la classe App\Controller\ResetPasswordController

Contrôleur gérant le processus de réinitialisation de mot de passe utilisateur.

Graphe d'héritage de App\Controller\ResetPasswordController:



## Fonctions membres publiques

- `__construct` (private ResetPasswordHelperInterface \$resetPasswordHelper, private EntityManagerInterface \$entityManager)

*Constructeur du contrôleur.*

- `checkEmail` ()

*Affiche une page de confirmation après envoi de l'e-mail de réinitialisation.*

## Fonctions membres privées

- `processSendingPasswordResetEmail` (string \$emailFormData, MailerInterface \$mailer, TranslatorInterface \$translator)

*Génère un token de réinitialisation et envoie un e-mail à l'utilisateur concerné.*

### 6.13.1 Description détaillée

Contrôleur gérant le processus de réinitialisation de mot de passe utilisateur.

Ce contrôleur utilise le bundle SymfonyCasts ResetPassword pour gérer :

- La demande de réinitialisation (formulaire et envoi d'e-mail)
- L'affichage d'une page de confirmation
- La réinitialisation effective du mot de passe avec token sécurisé

### 6.13.2 Documentation des constructeurs et destructeur

#### 6.13.2.1 `__construct()`

```
App\Controller\ResetPasswordController::__construct (  
    private ResetPasswordHelperInterface $resetPasswordHelper,  
    private EntityManagerInterface $entityManager)
```

Constructeur du contrôleur.

#### Paramètres

ResetPasswordHelperInterface	<i>\$resetPasswordHelper</i>	Service de gestion des tokens de réinitialisation
EntityManagerInterface	<i>\$entityManager</i>	Gestionnaire d'entités Doctrine

### 6.13.3 Documentation des fonctions membres

#### 6.13.3.1 `checkEmail()`

```
App\Controller\ResetPasswordController::checkEmail ()
```

Affiche une page de confirmation après envoi de l'e-mail de réinitialisation.

Affiche un faux token si l'utilisateur n'existe pas, pour des raisons de sécurité.

#### Renvoie

Response

### 6.13.3.2 processSendingPasswordResetEmail()

```
App\Controller\ResetPasswordController::processSendingPasswordResetEmail (
    string $emailFormData,
    MailerInterface $mailer,
    TranslatorInterface $translator) [private]
```

Génère un token de réinitialisation et envoie un e-mail à l'utilisateur concerné.

Si l'utilisateur n'existe pas, redirige vers la page de confirmation.

#### Paramètres

string	<i>\$emailFormData</i>	Adresse email saisie par l'utilisateur
MailerInterface	<i>\$mailer</i>	Service d'envoi d'e-mails
TranslatorInterface	<i>\$translator</i>	Service de traduction

#### Renvoie

RedirectResponse

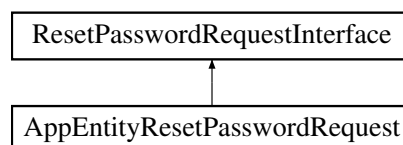
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/Controller/ResetPasswordController.php](#)

## 6.14 Référence de la classe App\Entity\ResetPasswordRequest

Entité représentant une demande de réinitialisation de mot de passe.

Graphes d'héritage de App\Entity\ResetPasswordRequest:



#### Fonctions membres publiques

— [\\_\\_construct](#) ([User](#) \$user, \DateTimeInterface \$expiresAt, string \$selector, string \$hashedToken)

*Constructeur de la demande de réinitialisation de mot de passe.*

— [getId](#) ()

*Retourne l'identifiant de la demande.*

— [getUser](#) ()

*Retourne l'utilisateur associé à la demande.*

### Attributs privés

- int `$id` = null
- User `$user` = null

#### 6.14.1 Description détaillée

Entité représentant une demande de réinitialisation de mot de passe.

Cette entité est utilisée par le bundle SymfonyCasts ResetPassword pour gérer la logique de création, validation et expiration des Tokens de réinitialisation.

#### 6.14.2 Documentation des constructeurs et destructeur

##### 6.14.2.1 `__construct()`

```
App\Entity\ResetPasswordRequest::__construct (
    User $user,
    \DateTimeInterface $expiresAt,
    string $selector,
    string $hashedToken)
```

Constructeur de la demande de réinitialisation de mot de passe.

##### Paramètres

User	<i>\$user</i>	L'utilisateur qui a demandé la réinitialisation
\DateTimeInterface	<i>\$expiresAt</i>	La date d'expiration du jeton
string	<i>\$selector</i>	Le sélecteur utilisé pour retrouver la demande
string	<i>\$hashedToken</i>	Le Token de réinitialisation hashé

#### 6.14.3 Documentation des fonctions membres

##### 6.14.3.1 `getId()`

```
App\Entity\ResetPasswordRequest::getId ()
```

Retourne l'identifiant de la demande.

##### Renvoie

int|null

### 6.14.3.2 getUser()

```
App\Entity\ResetPasswordRequest::getUser ()
```

Retourne l'utilisateur associé à la demande.

Renvoie

User

## 6.14.4 Documentation des données membres

### 6.14.4.1 \$id

```
int App\Entity\ResetPasswordRequest::$id = null [private]
```

### 6.14.4.2 \$user

```
User App\Entity\ResetPasswordRequest::$user = null [private]
```

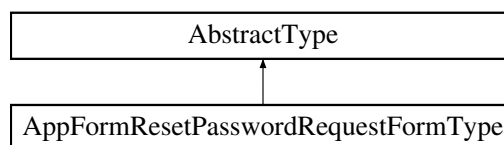
La documentation de cette classe a été générée à partir du fichier suivant :

— src/Entity/[ResetPasswordRequest.php](#)

## 6.15 Référence de la classe App\Form\ResetPasswordRequestFormType

Formulaire de demande de réinitialisation de mot de passe.

Graphe d'héritage de App\Form\ResetPasswordRequestFormType:



### Fonctions membres publiques

— [buildForm](#) (FormBuilderInterface \$builder, array \$options)

*Construit le formulaire de demande de réinitialisation.*

— [configureOptions](#) (OptionsResolver \$resolver)

*Configure les options par défaut du formulaire.*



### 6.15.1 Description détaillée

Formulaire de demande de réinitialisation de mot de passe.

Ce formulaire permet à l'utilisateur de saisir son adresse email afin de recevoir un lien de réinitialisation de mot de passe. Il contient un seul champ obligatoire.

### 6.15.2 Documentation des fonctions membres

#### 6.15.2.1 buildForm()

```
App\Form\ResetPasswordRequestFormType::buildForm (
    FormBuilderInterface $builder,
    array $options)
```

Construit le formulaire de demande de réinitialisation.

##### Paramètres

FormBuilderInterface	<i>\$builder</i>	Le constructeur de formulaire
array	<i>\$options</i>	Les options passées au formulaire

#### 6.15.2.2 configureOptions()

```
App\Form\ResetPasswordRequestFormType::configureOptions (
    OptionsResolver $resolver)
```

Configure les options par défaut du formulaire.

##### Paramètres

OptionsResolver	<i>\$resolver</i>	Le résolveur d'options
-----------------	-------------------	------------------------

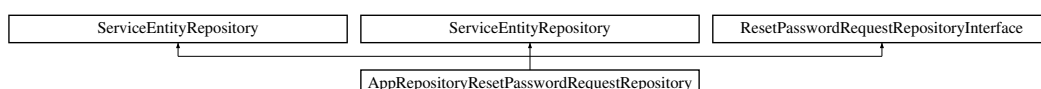
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/Form/ResetPasswordRequestFormType.php](#)

## 6.16 Référence de la classe App\Repository\ResetPasswordRequestRepository

Repository pour l'entité ResetPasswordRequest.

Graphe d'héritage de App\Repository\ResetPasswordRequestRepository:



## Fonctions membres publiques

- `__construct` (ManagerRegistry \$registry)

Constructeur du repository.

- `createResetPasswordRequest` (object \$user, \DateTimeInterface \$expiresAt, string \$selector, string \$hashedToken)

Crée une instance de demande de réinitialisation de mot de passe.

### 6.16.1 Description détaillée

Repository pour l'entité ResetPasswordRequest.

Ce repository gère les demandes de réinitialisation de mot de passe en implémentant les fonctionnalités nécessaires via l'interface ResetPasswordRequestRepositoryInterface et le trait ResetPasswordRequestRepositoryTrait.

<ResetPasswordRequest>

### 6.16.2 Documentation des constructeurs et destructeur

#### 6.16.2.1 \_\_construct()

```
App\Repository\ResetPasswordRequestRepository::__construct (
    ManagerRegistry $registry)
```

Constructeur du repository.

##### Paramètres

ManagerRegistry	\$registry	Le registre de gestionnaire d'entités Doctrine
-----------------	------------	--

### 6.16.3 Documentation des fonctions membres

#### 6.16.3.1 createResetPasswordRequest()

```
App\Repository\ResetPasswordRequestRepository::createResetPasswordRequest (
    object $user,
    \DateTimeInterface $expiresAt,
    string $selector,
    string $hashedToken)
```

Crée une instance de demande de réinitialisation de mot de passe.

##### Paramètres

User	\$user	L'utilisateur pour lequel la demande est créée
\DateTimeInterface	\$expiresAt	Date d'expiration de la demande
string	\$selector	Sélecteur unique de la demande
string	\$hashedToken	Jeton sécurisé, déjà haché

##### Renvoie

ResetPasswordRequestInterface L'objet ResetPasswordRequest prêt à être persisté

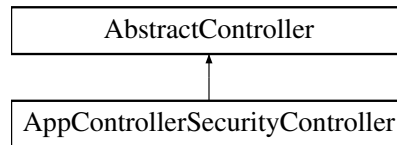
La documentation de cette classe a été générée à partir du fichier suivant :

- `src/Repository/ResetPasswordRequestRepository.php`

## 6.17 Référence de la classe App\Controller\SecurityController

Contrôleur de sécurité gérant l'authentification des utilisateurs.

Graphes d'héritage de App\Controller\SecurityController:



### Fonctions membres publiques

— [login](#) (AuthenticationUtils \$authenticationUtils)

*Affiche la page de connexion et gère les erreurs éventuelles de login.*

— [logout](#) ()

*Route de déconnexion.*

### 6.17.1 Description détaillée

Contrôleur de sécurité gérant l'authentification des utilisateurs.

### 6.17.2 Documentation des fonctions membres

#### 6.17.2.1 login()

```
App\Controller\SecurityController::login (
    AuthenticationUtils $authenticationUtils)
```

Affiche la page de connexion et gère les erreurs éventuelles de login.

#### Paramètres

AuthenticationUtils	<i>\$authenticationUtils</i>	Utilitaire pour récupérer les informations d'authentification
---------------------	------------------------------	---

#### Renvoie

Response

#### 6.17.2.2 logout()

```
App\Controller\SecurityController::logout ()
```

Route de déconnexion.

Cette méthode ne sera jamais exécutée, car Symfony intercepte cette route automatiquement via la configuration du pare-feu (`firewall.logout`).

#### Renvoie

void

## Exceptions

--	--

LogicException

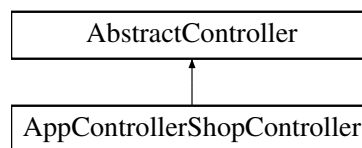
La documentation de cette classe a été générée à partir du fichier suivant :

— src/Controller/[SecurityController.php](#)

## 6.18 Référence de la classe App\Controller\ShopController

Contrôleur de la boutique affichant la liste des sweatshirts.

Graphes d'héritage de App\Controller\ShopController:



### Fonctions membres publiques

— [index](#) ([SweatshirtRepository](#) \$sweatshirtRepository, Request \$request)

*Affiche la liste des sweatshirts disponibles.*

### 6.18.1 Description détaillée

Contrôleur de la boutique affichant la liste des sweatshirts.

### 6.18.2 Documentation des fonctions membres

#### 6.18.2.1 index()

```
App\Controller\ShopController::index (
    SweatshirtRepository $sweatshirtRepository,
    Request $request)
```

Affiche la liste des sweatshirts disponibles.

Cette méthode peut filtrer les produits par plage de prix via un paramètre GET `priceRange` (ex: 20-50).

#### Paramètres

SweatshirtRepository	<i>\$sweatshirtRepository</i>	Le repository pour accéder aux sweatshirts
Request	<i>\$request</i>	La requête HTTP contenant éventuellement le paramètre de filtre

#### Renvoie

Response La réponse contenant le rendu de la page boutique

La documentation de cette classe a été générée à partir du fichier suivant :

— src/Controller/[ShopController.php](#)

## 6.19 Référence de la classe App\Service\StripeService

Service pour interagir avec l'API Stripe, en particulier pour créer une session de paiement.

### Fonctions membres publiques

— `__construct` (private string \$stripeSecretKey)

*Constructeur du service, initialise la clé secrète Stripe pour l'authentification.*

— `createCheckoutSession` (array \$lineItems, string \$successUrl, string \$cancelUrl)

*Crée une session de paiement avec Stripe Checkout.*

### 6.19.1 Description détaillée

Service pour interagir avec l'API Stripe, en particulier pour créer une session de paiement.

### 6.19.2 Documentation des constructeurs et destructeur

#### 6.19.2.1 `__construct()`

```
App\Service\StripeService::__construct (  
    private string $stripeSecretKey)
```

Constructeur du service, initialise la clé secrète Stripe pour l'authentification.

#### Paramètres

string	<code>\$stripeSecretKey</code>	La clé secrète de l'API Stripe pour effectuer les requêtes.
--------	--------------------------------	---

### 6.19.3 Documentation des fonctions membres

#### 6.19.3.1 `createCheckoutSession()`

```
App\Service\StripeService::createCheckoutSession (  
    array $lineItems,  
    string $successUrl,  
    string $cancelUrl)
```

Crée une session de paiement avec Stripe Checkout.

Cette méthode génère une session de paiement que l'utilisateur peut utiliser pour compléter son achat en ligne. Elle inclut les éléments du panier, les URLs de succès et d'annulation.

**Paramètres**

array	<i>\$lineItems</i>	Les éléments du panier pour le paiement (par exemple, articles, prix).
string	<i>\$successUrl</i>	L'URL où l'utilisateur sera redirigé après un paiement réussi.
string	<i>\$cancelUrl</i>	L'URL où l'utilisateur sera redirigé si le paiement est annulé.

**Renvoie**

Session La session de paiement Stripe créée.

La documentation de cette classe a été générée à partir du fichier suivant :

— src/Service/[StripeService.php](#)

## 6.20 Référence de la classe App\Entity\Sweatshirt

Représente un sweat-shirt disponible dans la boutique.

**Fonctions membres publiques**

— [getId](#) ()

*Retourne l'identifiant du sweat-shirt.*

— [getName](#) ()

*Retourne le nom du sweat-shirt.*

— [setName](#) (string \$name)

*Définit le nom du sweat-shirt.*

— [getPrice](#) ()

*Retourne le prix du sweat-shirt.*

— [setPrice](#) (float \$price)

*Définit le prix du sweat-shirt.*

— [getStockXS](#) ()

*Retourne le stock XS.*

— [setStockXS](#) (int \$stockXS)

*Définit le stock XS.*

— [getStockS](#) ()

*Retourne le stock S.*

— [setStockS](#) (int \$stockS)

*Définit le stock S.*

— `getStockM()`

*Retourne le stock M.*

— `setStockM(int $stockM)`

*Définit le stock M.*

— `getStockL()`

*Retourne le stock L.*

— `setStockL(int $stockL)`

*Définit le stock L.*

— `getStockXL()`

*Retourne le stock XL.*

— `setStockXL(int $stockXL)`

*Définit le stock XL.*

— `getImage()`

*Retourne le nom ou l'URL de l'image.*

— `setImage(?string $image)`

*Définit le nom ou l'URL de l'image.*

— `isFeatured()`

*Indique si le produit est mis en avant.*

— `setFeatured(bool $featured)`

*Définit si le produit doit être mis en avant.*

### Attributs privés

— `int $id = null`

— `string $name = null`

— `float $price = null`

— `int $stockXS = null`

— `int $stockS = null`

— `int $stockM = null`

— `int $stockL = null`

— `int $stockXL = null`

- string `$image` = null
- bool `$featured` = false

### 6.20.1 Description détaillée

Représente un sweat-shirt disponible dans la boutique.

### 6.20.2 Documentation des fonctions membres

#### 6.20.2.1 getId()

```
App\Entity\Sweatshirt::getId ()
```

Retourne l'identifiant du sweat-shirt.

#### 6.20.2.2 getImage()

```
App\Entity\Sweatshirt::getImage ()
```

Retourne le nom ou l'URL de l'image.

#### 6.20.2.3 getName()

```
App\Entity\Sweatshirt::getName ()
```

Retourne le nom du sweat-shirt.

#### 6.20.2.4 getPrice()

```
App\Entity\Sweatshirt::getPrice ()
```

Retourne le prix du sweat-shirt.

#### 6.20.2.5 getStockL()

```
App\Entity\Sweatshirt::getStockL ()
```

Retourne le stock L.

#### 6.20.2.6 getStockM()

```
App\Entity\Sweatshirt::getStockM ()
```

Retourne le stock M.



### 6.20.2.7 getStockS()

```
App\Entity\Sweatshirt::getStockS ()
```

Retourne le stock S.

### 6.20.2.8 getStockXL()

```
App\Entity\Sweatshirt::getStockXL ()
```

Retourne le stock XL.

### 6.20.2.9 getStockXS()

```
App\Entity\Sweatshirt::getStockXS ()
```

Retourne le stock XS.

### 6.20.2.10 isFeatured()

```
App\Entity\Sweatshirt::isFeatured ()
```

Indique si le produit est mis en avant.

### 6.20.2.11 setFeatured()

```
App\Entity\Sweatshirt::setFeatured (  
    bool $featured)
```

Définit si le produit doit être mis en avant.

### 6.20.2.12 setImage()

```
App\Entity\Sweatshirt::setImage (  
    ?string $image)
```

Définit le nom ou l'URL de l'image.

### 6.20.2.13 setName()

```
App\Entity\Sweatshirt::setName (  
    string $name)
```

Définit le nom du sweat-shirt.

#### 6.20.2.14 `setPrice()`

```
App\Entity\Sweatshirt::setPrice (  
    float $price)
```

Définit le prix du sweat-shirt.

#### 6.20.2.15 `setStockL()`

```
App\Entity\Sweatshirt::setStockL (  
    int $stockL)
```

Définit le stock L.

#### 6.20.2.16 `setStockM()`

```
App\Entity\Sweatshirt::setStockM (  
    int $stockM)
```

Définit le stock M.

#### 6.20.2.17 `setStockS()`

```
App\Entity\Sweatshirt::setStockS (  
    int $stockS)
```

Définit le stock S.

#### 6.20.2.18 `setStockXL()`

```
App\Entity\Sweatshirt::setStockXL (  
    int $stockXL)
```

Définit le stock XL.

#### 6.20.2.19 `setStockXS()`

```
App\Entity\Sweatshirt::setStockXS (  
    int $stockXS)
```

Définit le stock XS.

### 6.20.3 Documentation des données membres

#### 6.20.3.1 `$featured`

```
bool App\Entity\Sweatshirt::$featured = false [private]
```

### 6.20.3.2 \$id

```
int App\Entity\Sweatshirt::$id = null [private]
```

### 6.20.3.3 \$image

```
string App\Entity\Sweatshirt::$image = null [private]
```

### 6.20.3.4 \$name

```
string App\Entity\Sweatshirt::$name = null [private]
```

### 6.20.3.5 \$price

```
float App\Entity\Sweatshirt::$price = null [private]
```

### 6.20.3.6 \$stockL

```
int App\Entity\Sweatshirt::$stockL = null [private]
```

### 6.20.3.7 \$stockM

```
int App\Entity\Sweatshirt::$stockM = null [private]
```

### 6.20.3.8 \$stockS

```
int App\Entity\Sweatshirt::$stockS = null [private]
```

### 6.20.3.9 \$stockXL

```
int App\Entity\Sweatshirt::$stockXL = null [private]
```

### 6.20.3.10 \$stockXS

```
int App\Entity\Sweatshirt::$stockXS = null [private]
```

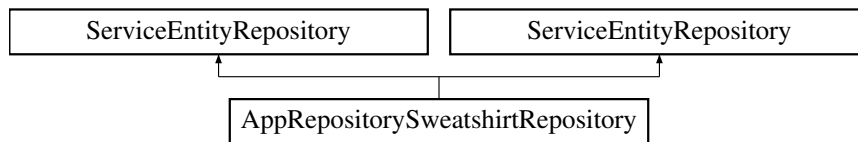
La documentation de cette classe a été générée à partir du fichier suivant :

— src/Entity/[Sweatshirt.php](#)

## 6.21 Référence de la classe App\Repository\SweatshirtRepository

Repository pour l'entité Sweatshirt.

Graphe d'héritage de App\Repository\SweatshirtRepository:



### Fonctions membres publiques

— [\\_\\_construct](#) (ManagerRegistry \$registry)

*Constructeur du repository Sweatshirt.*

#### 6.21.1 Description détaillée

Repository pour l'entité Sweatshirt.

Ce repository fournit des méthodes d'accès à la base de données pour les objets de type Sweatshirt. Il hérite des méthodes standard de Doctrine telles que find(), findAll(), findBy(), findOneBy(), etc.

<Sweatshirt>

#### 6.21.2 Documentation des constructeurs et destructeur

##### 6.21.2.1 \_\_construct()

```
App\Repository\SweatshirtRepository::__construct (
    ManagerRegistry $registry)
```

Constructeur du repository Sweatshirt.

##### Paramètres

ManagerRegistry	\$registry	Le registre du gestionnaire Doctrine
-----------------	------------	--------------------------------------

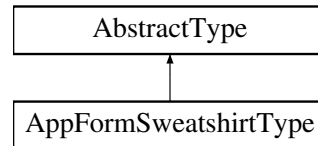
La documentation de cette classe a été générée à partir du fichier suivant :

— src/Repository/[SweatshirtRepository.php](#)

## 6.22 Référence de la classe App\Form\SweatshirtType

Formulaire pour la création ou modification d'un Sweatshirt.

Graphe d'héritage de App\Form\SweatshirtType:



### Fonctions membres publiques

- [buildForm](#) (FormBuilderInterface \$builder, array \$options)

*Construit le formulaire du Sweatshirt.*

- [configureOptions](#) (OptionsResolver \$resolver)

*Configure les options du formulaire, notamment la classe de données associée.*

### 6.22.1 Description détaillée

Formulaire pour la création ou modification d'un Sweatshirt.

Ce formulaire permet de renseigner :

- le nom du sweatshirt,
- son prix,
- le stock disponible pour chaque taille (XS à XL),
- une image facultative,
- une option pour le mettre en avant.

### 6.22.2 Documentation des fonctions membres

#### 6.22.2.1 buildForm()

```
App\Form\SweatshirtType::buildForm (
    FormBuilderInterface $builder,
    array $options)
```

Construit le formulaire du Sweatshirt.

#### Paramètres

FormBuilderInterface	<i>\$builder</i>	Le constructeur de formulaire
array	<i>\$options</i>	Les options du formulaire

#### 6.22.2.2 configureOptions()

```
App\Form\SweatshirtType::configureOptions (
    OptionsResolver $resolver)
```

Configure les options du formulaire, notamment la classe de données associée.

## Paramètres

OptionsResolver	<i>\$resolver</i>	Le résolveur d'options
-----------------	-------------------	------------------------

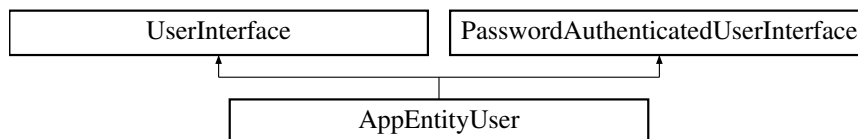
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/Form/SweatshirtType.php](#)

## 6.23 Référence de la classe App\Entity\User

Représente un utilisateur de l'application.

Graphe d'héritage de App\Entity\User:



### Fonctions membres publiques

— [getId](#) ()

*Retourne l'identifiant de l'utilisateur.*

— [getEmail](#) ()

*Retourne l'email de l'utilisateur.*

— [setEmail](#) (string \$email)

*Définit l'email de l'utilisateur.*

— [getUserIdentifier](#) ()

*Retourne l'identifiant unique de l'utilisateur (utilisé par le système de sécurité).*

— [getRoles](#) ()

*Retourne la liste des rôles de l'utilisateur.*

— [setRoles](#) (array \$roles)

*Définit les rôles de l'utilisateur.*

— [getPassword](#) ()

*Retourne le mot de passe haché de l'utilisateur.*

— [setPassword](#) (string \$password)

*Définit le mot de passe haché de l'utilisateur.*

- `eraseCredentials ()`  
*Méthode utilisée pour effacer les données sensibles temporaires.*
- `getUsername ()`  
*Retourne le nom d'utilisateur.*
- `setUsername (string $username)`  
*Définit le nom d'utilisateur.*
- `getDeliveryAddress ()`  
*Retourne l'adresse de livraison.*
- `setDeliveryAddress (string $deliveryAddress)`  
*Définit l'adresse de livraison.*
- `isVerified ()`  
*Vérifie si l'utilisateur a confirmé son compte.*
- `setIsVerified (bool $isVerified)`  
*Définit si le compte utilisateur est vérifié.*
- `getActivationToken ()`  
*Retourne le Token d'activation du compte.*
- `setActivationToken (?string $activationToken)`  
*Définit le Token d'activation du compte.*

### Attributs privés

- `int $id = null`
- `string $email = null`
- `array $roles = []`
- `string $password = null`
- `string $username = null`
- `string $deliveryAddress = null`
- `bool $isVerified = false`
- `string $activationToken = null`

#### 6.23.1 Description détaillée

Représente un utilisateur de l'application.

## 6.23.2 Documentation des fonctions membres

### 6.23.2.1 eraseCredentials()

```
App\Entity\User::eraseCredentials ()
```

Méthode utilisée pour effacer les données sensibles temporaires.

Voir également

UserInterface

### 6.23.2.2 getActivationToken()

```
App\Entity\User::getActivationToken ()
```

Retourne le Token d'activation du compte.

### 6.23.2.3 getDeliveryAddress()

```
App\Entity\User::getDeliveryAddress ()
```

Retourne l'adresse de livraison.

### 6.23.2.4 getEmail()

```
App\Entity\User::getEmail ()
```

Retourne l'email de l'utilisateur.

### 6.23.2.5 getId()

```
App\Entity\User::getId ()
```

Retourne l'identifiant de l'utilisateur.

### 6.23.2.6 getPassword()

```
App\Entity\User::getPassword ()
```

Retourne le mot de passe haché de l'utilisateur.

Voir également

PasswordAuthenticatedUserInterface



### 6.23.2.7 getRoles()

```
App\Entity\User::getRoles ()
```

Retourne la liste des rôles de l'utilisateur.

Voir également

UserInterface

Renvoie

list<string>

### 6.23.2.8 getUserIdentifier()

```
App\Entity\User::getUserIdentifier ()
```

Retourne l'identifiant unique de l'utilisateur (utilisé par le système de sécurité).

Voir également

UserInterface

### 6.23.2.9 getUsername()

```
App\Entity\User::getUsername ()
```

Retourne le nom d'utilisateur.

### 6.23.2.10 isVerified()

```
App\Entity\User::isVerified ()
```

Vérifie si l'utilisateur a confirmé son compte.

### 6.23.2.11 setActivationToken()

```
App\Entity\User::setActivationToken (  
    ?string $activationToken)
```

Définit le Token d'activation du compte.

### 6.23.2.12 setDeliveryAddress()

```
App\Entity\User::setDeliveryAddress (  
    string $deliveryAddress)
```

Définit l'adresse de livraison.

#### 6.23.2.13 setEmail()

```
App\Entity\User::setEmail (
    string $email)
```

Définit l'email de l'utilisateur.

#### 6.23.2.14 setIsVerified()

```
App\Entity\User::setIsVerified (
    bool $isVerified)
```

Définit si le compte utilisateur est vérifié.

#### 6.23.2.15 setPassword()

```
App\Entity\User::setPassword (
    string $password)
```

Définit le mot de passe haché de l'utilisateur.

#### 6.23.2.16 setRoles()

```
App\Entity\User::setRoles (
    array $roles)
```

Définit les rôles de l'utilisateur.

##### Paramètres

<i>list&lt;string&gt;</i>	\$roles
---------------------------	---------

#### 6.23.2.17 setUsername()

```
App\Entity\User::setUsername (
    string $username)
```

Définit le nom d'utilisateur.

### 6.23.3 Documentation des données membres

#### 6.23.3.1 \$activationToken

```
string App\Entity\User::$activationToken = null [private]
```

### 6.23.3.2 \$deliveryAddress

```
string App\Entity\User::$deliveryAddress = null [private]
```

### 6.23.3.3 \$email

```
string App\Entity\User::$email = null [private]
```

### 6.23.3.4 \$id

```
int App\Entity\User::$id = null [private]
```

### 6.23.3.5 \$isVerified

```
bool App\Entity\User::$isVerified = false [private]
```

### 6.23.3.6 \$password

```
string App\Entity\User::$password = null [private]
```

### 6.23.3.7 \$roles

```
array App\Entity\User::$roles = [] [private]
```

### 6.23.3.8 \$username

```
string App\Entity\User::$username = null [private]
```

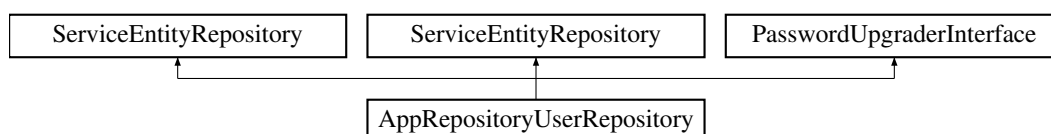
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/Entity/User.php](#)

## 6.24 Référence de la classe App\Repository\UserRepository

Repository pour l'entité User.

Graphe d'héritage de App\Repository\UserRepository:



## Fonctions membres publiques

- `__construct` (ManagerRegistry \$registry)

*Constructeur du repository User.*

- `upgradePassword` (PasswordAuthenticatedUserInterface \$user, string \$newHashedPassword)

*Met à jour automatiquement (rehash) le mot de passe de l'utilisateur.*

### 6.24.1 Description détaillée

Repository pour l'entité User.

Fournit des méthodes d'accès aux utilisateurs enregistrés en base de données. Implémente PasswordUpgrader↔ Interface pour permettre la mise à jour automatique des mots de passe hashés.

<User>

### 6.24.2 Documentation des constructeurs et destructeur

#### 6.24.2.1 \_\_construct()

```
App\Repository\UserRepository::__construct (
    ManagerRegistry $registry)
```

Constructeur du repository User.

##### Paramètres

ManagerRegistry	\$registry	Le registre des gestionnaires Doctrine.
-----------------	------------	---

### 6.24.3 Documentation des fonctions membres

#### 6.24.3.1 upgradePassword()

```
App\Repository\UserRepository::upgradePassword (
    PasswordAuthenticatedUserInterface $user,
    string $newHashedPassword)
```

Met à jour automatiquement (rehash) le mot de passe de l'utilisateur.

Cette méthode est appelée automatiquement par Symfony si la stratégie de mise à jour de mot de passe est activée. Elle sert à rehasher les mots de passe selon les nouvelles règles de sécurité définies.

##### Paramètres

PasswordAuthenticatedUserInterface	\$user	L'utilisateur dont le mot de passe doit être mis à jour.
string	\$newHashedPassword	Le nouveau mot de passe hashé.

## Exceptions

<i>UnsupportedUserException</i>	Si l'utilisateur passé en paramètre n'est pas une instance de User.
---------------------------------	---

La documentation de cette classe a été générée à partir du fichier suivant :

— src/Repository/[UserRepository.php](#)



## Chapitre 7

# Documentation des fichiers

### 7.1 Référence du fichier src/Controller/AdminController.php

#### Classes

— class [App\Controller\AdminController](#)

*Contrôleur responsable de la gestion des opérations administratives sur les sweat-shirts.*

#### Espaces de nommage

— namespace [App\Controller](#)

### 7.2 Référence du fichier src/Controller/CartController.php

#### Classes

— class [App\Controller\CartController](#)

*Contrôleur responsable de la gestion du panier d'achats.*

#### Espaces de nommage

— namespace [App\Controller](#)

### 7.3 Référence du fichier src/Controller/HomeController.php

#### Classes

— class [App\Controller\HomeController](#)

*Contrôleur responsable de l'affichage de la page d'accueil.*

## Espaces de nommage

— namespace [App\Controller](#)

## 7.4 Référence du fichier src/Controller/LoginController.php

### Classes

— class [App\Controller>LoginController](#)

## Espaces de nommage

— namespace [App\Controller](#)

## 7.5 Référence du fichier src/Controller/PaymentController.php

### Classes

— class [App\Controller\PaymentController](#)

*Contrôleur responsable de la gestion des paiements via Stripe.*

## Espaces de nommage

— namespace [App\Controller](#)

## 7.6 Référence du fichier src/Controller/ProductController.php

### Classes

— class [App\Controller\ProductController](#)

*Contrôleur responsable de l'affichage des détails d'un sweat-shirt.*

## Espaces de nommage

— namespace [App\Controller](#)



## 7.7 Référence du fichier src/Controller/RegistrationController.php

### Classes

— class [App\Controller\RegistrationController](#)

*Contrôleur de gestion des inscriptions et de la vérification des emails.*

### Espaces de nommage

— namespace [App\Controller](#)

## 7.8 Référence du fichier src/Controller/ResetPasswordController.php

### Classes

— class [App\Controller\ResetPasswordController](#)

*Contrôleur gérant le processus de réinitialisation de mot de passe utilisateur.*

### Espaces de nommage

— namespace [App\Controller](#)

## 7.9 Référence du fichier src/Controller/SecurityController.php

### Classes

— class [App\Controller\SecurityController](#)

*Contrôleur de sécurité gérant l'authentification des utilisateurs.*

### Espaces de nommage

— namespace [App\Controller](#)

## 7.10 Référence du fichier src/Controller/ShopController.php

### Classes

— class [App\Controller\ShopController](#)

*Contrôleur de la boutique affichant la liste des sweatshirts.*

## Espaces de nommage

— namespace [App\Controller](#)

## 7.11 Référence du fichier src/Entity/ResetPasswordRequest.php

### Classes

— class [App\Entity\ResetPasswordRequest](#)

*Entité représentant une demande de réinitialisation de mot de passe.*

## Espaces de nommage

— namespace [App\Entity](#)

## 7.12 Référence du fichier src/Entity/Sweatshirt.php

### Classes

— class [App\Entity\Sweatshirt](#)

*Représente un sweat-shirt disponible dans la boutique.*

## Espaces de nommage

— namespace [App\Entity](#)

## 7.13 Référence du fichier src/Entity/User.php

### Classes

— class [App\Entity\User](#)

*Représente un utilisateur de l'application.*

## Espaces de nommage

— namespace [App\Entity](#)

## 7.14 Référence du fichier src/Form/ChangePasswordFormType.php

### Classes

— class [App\Form\ChangePasswordFormType](#)

*Formulaire permettant à un utilisateur de changer son mot de passe.*

### Espaces de nommage

— namespace [App\Form](#)

## 7.15 Référence du fichier src/Form/RegistrationFormType.php

### Classes

— class [App\Form\RegistrationFormType](#)

*Formulaire d'inscription d'un nouvel utilisateur.*

### Espaces de nommage

— namespace [App\Form](#)

## 7.16 Référence du fichier src/Form/ResetPasswordRequestFormType.php

### Classes

— class [App\Form\ResetPasswordRequestFormType](#)

*Formulaire de demande de réinitialisation de mot de passe.*

### Espaces de nommage

— namespace [App\Form](#)

## 7.17 Référence du fichier src/Form/SweatshirtType.php

### Classes

— class [App\Form\SweatshirtType](#)

*Formulaire pour la création ou modification d'un Sweatshirt.*

## Espaces de nommage

— namespace [App\Form](#)

## 7.18 Référence du fichier src/Kernel.php

### Classes

— class [App\Kernel](#)

## Espaces de nommage

— namespace [App](#)

## 7.19 Référence du fichier src/Repository/ResetPasswordRequestRepository.php

### Classes

— class [App\Repository\ResetPasswordRequestRepository](#)

*Repository pour l'entité ResetPasswordRequest.*

## Espaces de nommage

— namespace [App\Repository](#)

## 7.20 Référence du fichier src/Repository/SweatshirtRepository.php

### Classes

— class [App\Repository\SweatshirtRepository](#)

*Repository pour l'entité Sweatshirt.*

## Espaces de nommage

— namespace [App\Repository](#)

## 7.21 Référence du fichier src/Repository/UserRepository.php

### Classes

— class [App\Repository\UserRepository](#)

*Repository pour l'entité User.*

### Espaces de nommage

— namespace [App\Repository](#)

## 7.22 Référence du fichier src/Security/AuthAuthenticator.php

### Classes

— class [App\Security\AuthAuthenticator](#)

*Authenticator personnalisé pour la connexion des utilisateurs.*

### Espaces de nommage

— namespace [App\Security](#)

## 7.23 Référence du fichier src/Security/EmailVerifier.php

### Classes

— class [App\Security>EmailVerifier](#)

### Espaces de nommage

— namespace [App\Security](#)

## 7.24 Référence du fichier src/Service/CartService.php

## 7.25 Référence du fichier src/Service/StripeService.php

### Classes

— class [App\Service\StripeService](#)

*Service pour interagir avec l'API Stripe, en particulier pour créer une session de paiement.*

## Espaces de nommage

— namespace [App\Service](#)