Authentification

Documentation technique - Implémentation de l'authentification sur Symfony 6.4

Cette documentation fournit une explication complète de l'implémentation de l'authentification dans notre application Symfony 6.4. Elle est destinée aux développeurs juniors rejoignant l'équipe et fournira des informations détaillées sur les fichiers modifiés, le processus d'authentification, et où sont stockés les utilisateurs.

Fichier security.yaml

Configurations globales

Dans le fichier `config/packages/security.yaml`, vous trouverez les configurations globales de sécurité de notre application. Voici ce que chaque partie signifie :

- `password_hashers` : Cette section spécifie comment les mots de passe des utilisateurs doivent être hachés. Dans notre cas, la valeur `"auto"` est utilisée, ce qui signifie que Symfony détermine automatiquement le moyen le plus sécurisé de hacher les mots de passe en fonction des fonctionnalités disponibles sur votre système.
- `providers` : Cette section définit le fournisseur de l'utilisateur pour charger les utilisateurs à partir de la base de données. Dans cet exemple, le fournisseur `app_user_provider` utilise une entité `App\Entity\User` et recherche les utilisateurs par le champ `username`.

Firewalls

Les firewalls sont des éléments clés de la sécurité dans Symfony. Ils déterminent comment les utilisateurs sont authentifiés et autorisés à accéder aux ressources de notre application. Voici les firewalls configurés dans notre fichier `security.yaml` :

- `dev` : Ce firewall est utilisé pour les environnements de développement. Il permet un accès sans restriction aux ressources statiques et aux outils de

débogage. Cela permet aux développeurs de travailler facilement sans se soucier de l'authentification.

- `main` : C'est le principal firewall de l'application. Il utilise le fournisseur d'utilisateurs `app_user_provider` défini précédemment. Pour l'authentification, il utilise un formulaire de connexion `(form_login)` avec les chemins de connexion (`login_path` et `check_path`) définis sur `app_login`.

Contrôle d'accès

Le contrôle d'accès définit quelles parties de l'application sont accessibles à quels utilisateurs. Voici les règles de contrôle d'accès définies dans notre fichier `security.yaml` :

- Les chemins `/login` et `/users/create` sont accessibles à tous les utilisateurs `(PUBLIC_ACCESS)`, ce qui signifie que ces pages sont ouvertes à tous les visiteurs, même s'ils ne sont pas authentifiés.
- Le chemin `/users` est réservé aux administrateurs `(ROLE_ADMIN)`. Seuls les utilisateurs possédant ce rôle auront accès à cette partie de l'application.
- Tous les autres chemins `(^/)` sont accessibles uniquement aux utilisateurs authentifiés `(ROLE_USER)`. Cela signifie que les utilisateurs doivent être connectés pour accéder à ces parties de l'application.

Contrôleur SecurityController

Le `SecurityController` est responsable de la gestion des actions liées à l'authentification dans notre application. Voici ce que chaque action du contrôleur fait :

- `login()` : Cette action est associée à la route `/login`. Elle est utilisée pour afficher le formulaire de connexion et gérer la soumission du formulaire. Si un utilisateur est déjà connecté et tente d'accéder à la page de connexion, il sera redirigé vers la page d'accueil.
- `logout()` : Cette action est associée à la route `/logout`. Elle est utilisée pour gérer la déconnexion de l'utilisateur. Dans cet exemple, la méthode ne fait rien de spécifique, mais Symfony gère automatiquement le processus de déconnexion pour vous.

Cela résume l'implémentation de l'authentification dans notre application Symfony 6.4. Les utilisateurs sont stockés dans une entité `User`, l'authentification est gérée via des firewalls et des contrôles d'accès configurés dans le fichier `security.yaml`, et les actions spécifiques sont gérées par le `SecurityController`.