Documentation Technique : Implémentation de l'Authentification sous Symfony 7.1

Cette documentation est destinée aux développeurs juniors qui rejoindront l'équipe. Elle vise à vous aider à comprendre comment l'authentification est implémentée dans notre projet Symfony 7.1, quels fichiers sont concernés, comment le processus d'authentification s'opère, et où les utilisateurs sont stockés.

1. Structure des Fichiers Concernés par l'Authentification

a) `config/packages/security.yaml`

Le fichier principal à modifier pour configurer l'authentification est `security.yaml`. Ce fichier définit les éléments suivants :

- Pare-feu (firewalls) : Gère les points d'entrée de l'application où l'authentification est requise.
- Access Control : Définit les permissions d'accès aux différentes routes selon les rôles des utilisateurs.
- Fournisseurs d'utilisateurs : Spécifie d'où viennent les utilisateurs (base de données, API, etc.).
- Encodage des mots de passe : Détermine comment les mots de passe sont sécurisés.

b) `src/Controller/SecurityController.php`

Ce fichier contient les actions relatives à la connexion et à la déconnexion. Il peut également contenir des méthodes pour enregistrer les utilisateurs.

- `login()` : Gère la logique de connexion.
- `logout()` : Gère la déconnexion.

c) `src/Entity/User.php`

Cette entité représente les utilisateurs dans la base de données. Elle contient les propriétés telles que le nom d'utilisateur, le mot de passe, les rôles, etc.

- Propriétés : Contiennent les informations de l'utilisateur.
- Méthodes d'interface (UserInterface) : Implémentées pour que l'entité soit compatible avec le système de sécurité de Symfony.

d) `templates/security/login.html.twig`

Ce fichier gère l'interface utilisateur de la page de connexion. Il contient le formulaire de connexion qui envoie les informations d'authentification à Symfony.

2. Processus d'Authentification

L'authentification sous Symfony 7.1 s'opère de la manière suivante :

1. Soumission du Formulaire : L'utilisateur soumet ses identifiants via le formulaire de connexion (`login.html.twig`).
2. Traitement par le Firewall : Le pare-feu (firewall) configuré dans `security.yaml` intercepte la requête et la redirige vers la méthode de connexion définie dans `SecurityController`.
3. Vérification des Identifiants : Symfony compare les identifiants soumis avec ceux stockés dans la base de données. Cette vérification se fait via le fournisseur d'utilisateurs configuré (généralement lié à l'entité `User`).
4. Création d'un Token de Sécurité : Si les identifiants sont corrects, Symfony crée un token de sécurité pour l'utilisateur, ce qui lui permet d'accéder aux ressources protégées.
5. Redirection : L'utilisateur est redirigé vers la page d'accueil après une connexion réussie.
6. Déconnexion : Lorsqu'un utilisateur se déconnecte, le token de sécurité est invalidé, et l'utilisateur est redirigé vers une page publique.

3. Stockage des Utilisateurs

Les utilisateurs sont stockés dans la base de données sous forme d'entité `User`. Cette entité est généralement mappée à une table SQL via Doctrine ORM. Chaque utilisateur a des propriétés comme un nom d'utilisateur, un mot de passe (crypté), un email, et un ou plusieurs rôles.

- Base de données : La table associée à l'entité `User` contient les informations d'identification et de profil des utilisateurs.
- Doctrine ORM : Gère les interactions entre l'application et la base de données, permettant de récupérer, créer, mettre à jour, et supprimer des utilisateurs.
- Mot de passe : Les mots de passe sont stockés sous forme chiffrée à l'aide de l'algorithme défini dans `security.yaml`.

Conclusion

En résumé, l'implémentation de l'authentification dans Symfony 7.1 repose sur la configuration du fichier `security.yaml`, l'utilisation de l'entité `User` pour gérer les

utilisateurs, et des contrôleurs pour orchestrer le processus de connexion et de déconnexion. En comprenant ces éléments, vous serez bien équipé pour maintenir et étendre le système d'authentification du projet.