Documentation technique

Implémentation de l'authentification

Il y a deux parties à distinguer dans le processus de sécurité : - L'authentification qui permet d'identifier l'utilisateur - L'autorisation qui permet de vérifier son identité et autorise, ou non, l'accès au contenu demandé.

Chaque élément de la sécurité est paramétré dans le fichier de configuration `config/packages/security.yaml`. Pour plus d'informations concernant ce fichier et ses différentes parties, n'hésitez pas a consulter la documentation officiel Security.

L'entité User

Avant la mise en place du système d'utilisateurs qui comprend l'inscription, l'authentification, la gestion des rôles, il faut définir la classe User qui implémente l'interface UserInterface du composant security. Par convention, elle s'appelle User et sera créée dans `src/Entity/User.php`.

Encoders

L'objet encodeur va encoder les mots de passe des utilisateurs via `UserPasswordEncoderInterface`. Ici l'algorithme de hachage est automatique et le plus performant possible (bcrypt).

```
security:
    password_hashers:
        Symfony\Component\Security\Core\User\PasswordAuthenticatedUserInterface: 'auto'
```

Providers

Appelé le fournisseur d'utilisateur, il indique où se situe les informations utiles à l'authentification de celui-ci. La configuration ci-dessous utilise Doctrine pour charger l'entité User en utilisant la propriété email comme "user identifier" qui permettra l'authentification.

```
providers:
        app_user_provider:
            entity:
                class: App\Entity\User
                property: email
```

Firewalls

Cette section est la plus importante. Il prend en charge l'authentification, permet de vérifier l'identité de l'utilisateur et lui donne accès ou non à certaines pages de la plateforme.

```yaml
firewalls:
        dev:
            pattern: ^/(_(profiler|wdt)|css|images|js)/
            security: false
        main:
            lazy: true
            provider: app_user_provider
            custom_authenticator: App\Security\AppAuthenticator
            logout:
                path: logout
                target: /login
```

La partie `dev` s'assure que l'on ne bloque pas accidentellement les outils de développement de Symfony qui se trouvent sous des URL tels que `/_profiler`, etc. Toutes les URL réels sont gérés par la partie `main`.

Access Control (Authorization)

Son rôle est de décider si un utilisateur a le rôle nécessaire puis autorise (ou non) l'accès à une ressource. Il est possible de définir différents rôles et d'en attribuer un ou plusieurs pour un utilisateur donné.

```yaml
access_control:
        - { path: ^/login, roles: PUBLIC_ACCESS }
        - { path: ^/admin, roles: ROLE_ADMIN }
        - { path: ^/user, roles: [ROLE_ADMIN, ROLE_USER] }
```

Ci-dessus, on indique que : - /login est accessible sans authentification - /admin n'est accessible avec authentification et rôle admin - /user n'est accessible avec authentification et rôle user