Implémentation de l'authentification

Cette Documentation permet de comprendre:

- quel(s) fichier(s) il faut modifier et pourquoi
- comment s'opère l'authentification
- et où sont stockés les utilisateurs.

Informations:

- Edition: fabien Vernières février 2023
- Projet de formation: Améliorez une application existante de ToDo & Co
- Application: todoco.fabienvernieres.com

L'entité User

Pour sécuriser notre application nous utilisons le **SecurityBundle** qui nous fournit des outils pour les sessions sécurisées.

On exécute la commande : `composer require symfony/security-bundle`

Grâce à **Symfony Flex** un fichier de configuration **security.yaml** est créé, comprenant trois parties principales. Le **provider** (fournisseur d'utilisateurs), le **firewall** (coeur de la sécurité) et **l'acces_control** (gestion des autorisations).

Avec le **MakerBundle** on crée la classe User: `php bin/console make:user`

La classe **User** est créée ainsi que son **Repository**. Le fichier **security.yaml** est mis à jour en utilisant notre classe User comme entité et une de ses propriétés (ex: username).

L'entité User est à présent connecté au système d'authentification.

Formulaire de connexion

On exécute la commande : `symfony console make:auth`

- sélectionne 1 pour générer une classe d'authentification
- nomme la classe d'authentification **AppAuthenticator**
- le contrôleur **SecurityController**
- crée une URL **/logout (yes)**

Un formulaire (template) de connexion et une classe d'authentification (authenticator) sont créés.

Le fichier **security.yaml** est mis à jour pour lier ces classes.

Nous pouvons personnaliser la classe **AppAuthenticator** et plus particulièrement la méthode **onAuthenticationSuccess()** pour choisir la page de destination après une connexion réussie: `return new RedirectResponse($this->urlGenerator->generate('app_default'));`

L'authentification

A chaque requête sur l'application, le **firewall** contrôle si cette dernière a besoin d'une authentification et redirige le cas échéant l'utilisateur vers le formulaire de connexion. Ce dernier peut également vouloir se connecter en accédant directement à cette page de login (bouton Se connecter).

Après soumission du formulaire, notre **AppCustomAuthenticator** retourne un objet de type **Passport**. Ce dernier contient:

- un **UserBadge** qui va récupérer notre utilisateur
- un **PasswordCredentials** qui valide les informations d'identification

Ensuite Symfony utilise l'UserProvider que l'on a paramétré dans le fichier **security.yaml**. Il charge l'utilisateur depuis la base de données à l'aide de l'identifiant (username) fournit par le **UserBadge**.

Symfony envoie un **Token** à la session pour permettre la persistance de la connexion utilisateur qui pourra à présent naviguer sur les pages de l'application tout en restant authentifié.