l’Authentification - projet Todo & Co

Implémentation technique & références

Emplacement des fichiers de configuration

la configuration pour les composants de sécurité se trouve à l’emplacement :
/config/packages/security.yaml

Emplacement des fichiers relatifs à l’utilisateur

Les utilisateurs en base de données sont représentés par l’entité ‘User’, dans src/entity. La contrainte d’unicité se trouve sur $email. La propriété est utilisée par le provider pour récupérer l’utilisateur et ne doit pas être modifiée à la légère.

Détail des services utilisés par l’application

roles

Les rôles sont un moyen d’effectuer des vérifications de sécurité et de restriction d’accès. un rôle peut être n’importe quelle chaîne de caractère, précédé de ‘ROLE_’. Par défaut, tous les utilisateurs disposent du rôle ‘ROLE_USER’. Les administrateurs disposent en plus du rôle ‘ROLE_ADMIN’ permettant de visiter les pages réservées aux administrateurs. Le role ‘IS_AUTHENTICATED_ANONYMOUSLY’ est un rôle permettant à n’importe quel utilisateur de visiter la route.

encoders

Les encoders permettent de crypter et décrypter les mots de passe utilisateur.
Dans le cas où plusieurs classes implémentant UserInterface existent, il est possible de définir un encoder différent pour chacune d’elles.
pour assigner un encoder à une classe User, notez sous forme ‘clef:valeur’ la dénomination complète de la classe (ex: App/Entity/User) suivie de l’algorithme à utiliser. A moins d’avoir une excellente raison d’en changer, préférez y renseigner la valeur ‘auto’.

user providers

Les providers permettent de charger un objet utilisateur à partir de la session en cours. Sa présence dans le fichier de configuration security permet de définir quel Provider sera chargé de s’occuper de quel objet User. sa définition doit respecter le schéma suivant:
label:
    entity:

class: ***fully_qualified_class_name***
property: ***property***

label: Le nom permettant de faire référence à ce provider dans le reste du code. important pour la configuration du firewall.
fully_qualified_class_name: le nom complet de la classe représentant l'objet utilisateur (ex: App/Entity/User)
property: Le nom de la propriété de l'objet servant au provider pour effectuer la requête en base de données. Prenez soin d'y inscrire une propriété disposant d'une contrainte d'unicité pour éviter les problèmes.

firewalls

Le firewall est le service chargé d'authentifier les utilisateurs et de surveiller les accès. Comme pour l'access control, l'ordre de déclaration des firewalls importe. La première correspondance sera choisie.
Si plusieurs firewall peuvent être définis, un seul peut être appelé par requête pour appliquer les restrictions d'accès ou authentifier l'utilisateur.
L'application dispose d'un seul firewall (main) avec la configuration suivante :

- pattern: le firewall surveille toutes les routes
- form_login: Le firewall utilise une stratégie basée sur un formulaire pour authentifier les utilisateurs.
- login_path: Route utilisée par le firewall pour rediriger les utilisateurs non authentifiés.
- login_check: Route utilisée par le firewall pour authentifier l'utilisateur (vérifier le formulaire soumis).
- default_target_path: Vers quelle route l'utilisateur est-il redirigé une fois que l'authentification a été acceptée. (ici, la page d'accueil)
- provider: Le provider 'users' (défini plus haut dans la configuration) pour récupérer l'objet utilisateur en base de données et permettre au firewall de le comparer au formulaire soumis.

access_control

access_control de la configuration security permet de définir des règles pour donner au firewall les moyens de restreindre l'accès de certaines pages aux utilisateurs ne disposant pas des rôles demandés.

Une nouvelle règle doit être ajoutée selon le format suivant:
- { path: ***path_pattern***, roles: ***ROLE*** }

path: le pattern de la route devant recevoir la restriction. le symbole '^' permet une correspondance pour toutes les routes disposant de caractères avant /login *(ex: '/api/login', '/blog/article/login','/login' toutes trois correspondent)*

Notez qu'en ce qui concerne l'écriture de la valeur de path, l'ordre est crucial. En effet, lors de la résolution de la route, le routeur va faire correspondre le premier critère sans chercher plus avant si un autre critère plus précis existe, aussi faut-il mettre les règles plus précises avant les règles qui le sont moins.
(ex: mettons que deux règles de contrôle d'accès existent, l'une avec le critère '/admin/users' et l'autre avec le critère '/admin'. Mettons également qu'il existe une route '/admin/users'. Pour que cette route bénéficie du contrôle d'accès de la règle '/admin/users', la règle doit figurer <u>avant</u> la règle '/admin' sans quoi la route correspondra à la règle '/admin' et suivra le contrôle d'accès implémenté par celle-ci plutôt que la règle cité précédemment, pourtant plus précise)

roles: Servent aux composant sécurité pour différencier les droit d'accès des utilisateurs. Les utilisateurs ne disposant pas de TOUS les rôles mentionnés ici se verront refuser l'accès. Dans le cas où plusieurs rôles doivent être définis, les différentes valeurs doivent être listées sous forme de tableau.