

DOCUMENTATION DE L'AUTHENTIFICATION



Préambule

Dans le projet ToDoAndCo l'accès au site se fait par une page de login dans laquelle on entre un [username](#) et un [mot de passe](#)

Les données liées aux utilisateurs auxquels se réfèrent le username et le mot de passe sont stockées dans une base de données.

Les mots de passe sont cryptés avant d'être stockés dans la base de données

Pour se connecter à la base de données il faut fournir les identifiants définis ci-dessus via le formulaire de login.

Lorsque ces données sont envoyées par soumission du formulaire, elles vont être analysées par le système mis en place dans le projet.

A) Configuration du fichier de sécurité : **security.yml**

Pour ce faire il faut aller dans le fichier **security.yml** qui se trouve dans **app/config**

A.1 Section **encoders**

Renseigner la section **encoders** avec le chemin vers l'entité User qui contient le mot de passe (password) à encoder et on y indiquera le mode de cryptage.

security:

encoders:

AppBundle\Entity\User: bcrypt

A.2 Section **encoders**

Dans la partie **providers** on indiquera où se trouve l'entité liée à l'utilisateur (User) pour vérifier les identifiants qui seront fournis pour authentication.

providers:

doctrine:

entity:
class: AppBundle\User
property: username

Ensuite il faut indiquer par quel moyen on souhaite accéder à l'application, pour ce faire on va renseigner la section **firewalls** sous rubrique **main**.

A.3 Section firewalls

firewalls:
dev:
 pattern: ^/(_(profiler|wdt)|css|images|js)/
 security: false

main:
 anonymous: ~
 pattern: ^/
 form_login:
 login_path: login
 check_path: login_check
 always_use_default_target_path: true
 default_target_path: /
 logout: ~
 logout_on_user_change: true

Pour le cas du projet la connexion se fera par un formulaire de login

On indiquera donc le champ **form_login** et les sous champs **login_path** et **check_path**

A.4 Section access_control

On veillera à permettre l'accès à ce formulaire à tout utilisateur en renseignant la section **access_control**

access_control:
- { path: ^/login, roles: IS_AUTHENTICATED_ANONYMOUSLY }

Ensuite il faut il faut définir une route dans un contrôleur ce qui va permettre à Symfony de prendre en charge l'authentification

P8 - Projet ToDoAndCo

On crée une classe **SecurityController.php** que l'on placera dans **src\Appbundle\Controller**

B) Création du SecurityController

Pour valider ou non l'authentification on crée un contrôleur dédié nommé **SecurityController.php**

Celui-ci contiendra la méthode login() dont le code métier est fourni par la documentation Symfony

NOTA: le code existant date de la version initiale du projet

https://symfony.com/doc/3.1/security/form_login_setup.html

et pourra être mis à jour pour suivre l'upgrade de la version Symfony 4.4

https://symfony.com/doc/4.4/security/form_login_setup.html