

# Documentation technique

ToDo and Co

Version : 2.0

# Sommaire

1. [Introduction](#)
2. [Authentication](#)
  - 2.1. [fonctionnement](#)
  - 2.2. [implémentation](#)
    - 2.2.1. [User](#)
    - 2.2.2. [Password Hasher](#)
    - 2.2.3. [Provider](#)
    - 2.2.4. [Firewalls](#)
    - 2.2.5. [Access control](#)

# Introduction

Symfony est un framework php basé sur le modèle, vue et contrôleur, la plupart des fichiers édité pour intégrer et configurer l'authentification se trouve dans le fichier **security.yaml** présent dans le dossier **config/package**.

## Authentication

### 1.1. Fonctionnement

L'authentification est un processus qui permet de définir l'identité de l'utilisateur courant de l'application.

Deux cas se présentent :

- **Authentifié** l'utilisateur peut accéder à la ressource selon son rôle.
- **Non authentifié** l'utilisateur est redirigé vers le formulaire d'authentification.

### 1.2. Implémentation

#### 1.2.1 User

Les données sont identifiées par des classes appelées entités présentes dans **src/Entity/**, dans notre cas l'utilisateur est représenté par **la classe User qui implémente UserInterface** fourni par symfony avec des méthodes pour gérer les utilisateurs.

```
class User implements UserInterface  
{
```

#### 1.2.2 Password Hasher

Password Hasher présent dans **config/packages/security.yaml** est utilisé pour crypter les mots de passe des utilisateurs avec le meilleur algorithme disponible.

```
password_hashers:  
    Symfony\Component\Security\Core\User\PasswordAuthenticatedUserInterface: "auto"  
    App\Entity\User:  
        algorithm: auto
```

### 1.2.3 Provider

Le provider présent dans **config/packages/security.yaml** définit quelle classe utiliser et quelle propriété pour authentifier l'utilisateur.

```
providers:
    # used to reload user from session & other features (e.g. switch_user)
    app_user_provider:
        entity:
            class: App\Entity\User
            property: email
```

### 1.2.4 Firewalls

Le Firewalls présent dans **config/packages/security.yaml** détermine si l'on peut accéder a certain page est de quelle manière selon l'environnement.

- **Dev** Environnement de développement
- **Main** Environnement général

```
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
```

### 1.2.5 Access control

L'access control présent dans **config/packages/security.yaml** autorise ou restreint certains de **vos utilisateurs** à accéder à une ressource **en fonction de leurs rôles**.

```
access_control:
    - { path: ^/admin, roles: ROLE_ADMIN }
```