Authentication

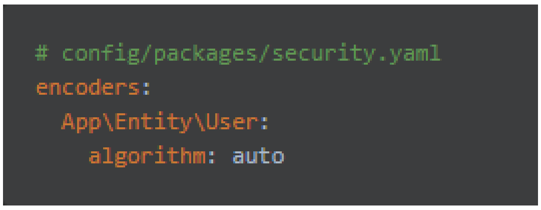
Introduction

The authentication system is located in the "security.yaml" file and is based on the "User" entity.
Official documentation available here: https://symfony.com/doc/current/security.html

Providers

In addition to your User class, you also need a "User provider": a class that helps with a few things, like reloading the User data from the session and some optional features, like remember me and impersonation.

Example :

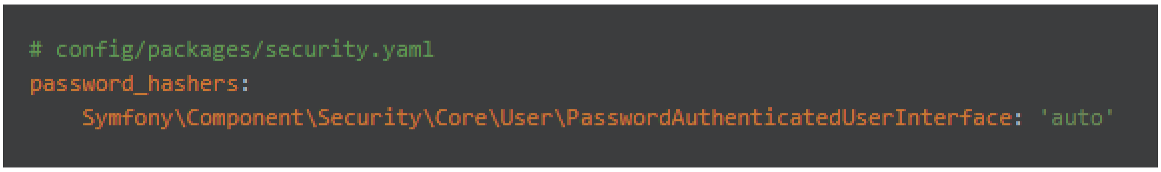


```yaml
# config/packages/security.yaml
encoders:
  App\Entity\User:
    algorithm: auto
```

Password Hasher

Use native password hasher, which auto-selects and migrates the best possible hashing algorithm (which currently is "bcrypt").

Example :

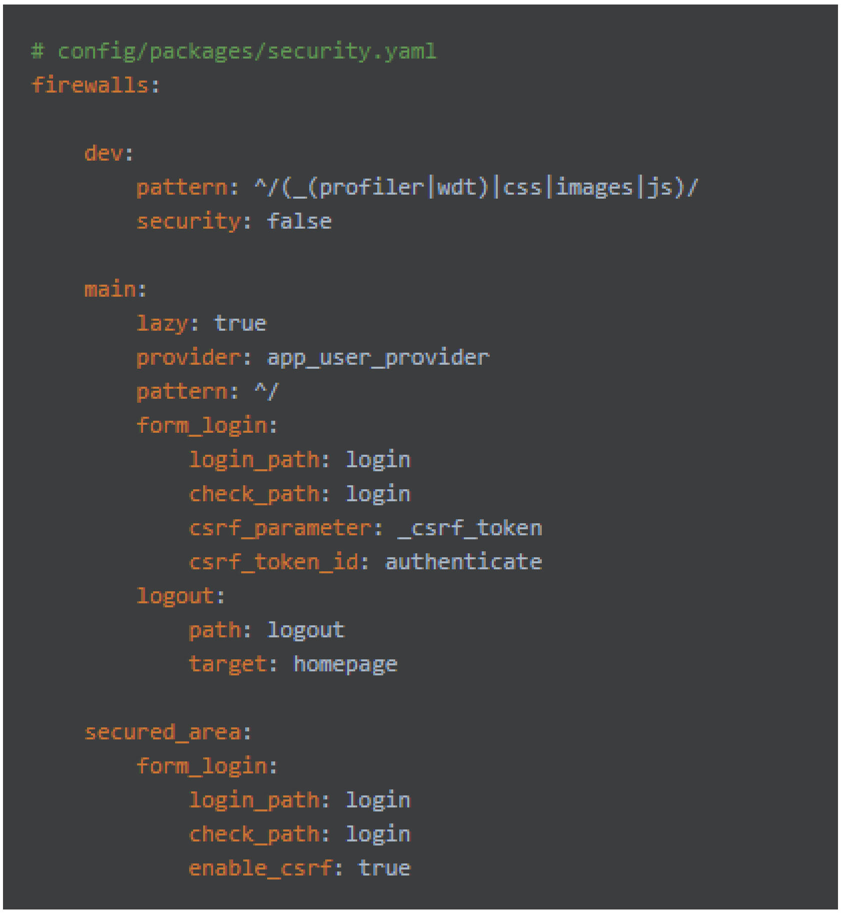


```yaml
# config/packages/security.yaml
password_hashers:
    Symfony\Component\Security\Core\User\PasswordAuthenticatedUserInterface: 'auto'
```

Firewall

The firewalls section of config/packages/security.yaml is the most important section. A "firewall" is your authentication system: the firewall defines which parts of your application are secured and how your users will be able to authenticate (e.g. login form, API token, etc).

Example :

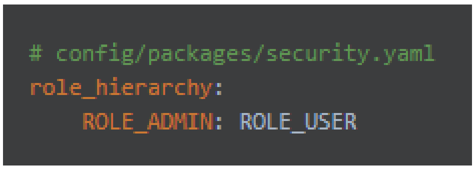


```yaml
# config/packages/security.yaml
firewalls:

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

    main:
        lazy: true
        provider: app_user_provider
        pattern: ^/
        form_login:
            login_path: login
            check_path: login
            csrf_parameter: _csrf_token
            csrf_token_id: authenticate
        logout:
            path: logout
            target: homepage

    secured_area:
        form_login:
            login_path: login
            check_path: login
            enable_csrf: true
```

Role Hierarchy

Instead of giving many roles to each user, you can define role inheritance rules by creating a role hierarchy.

Example :

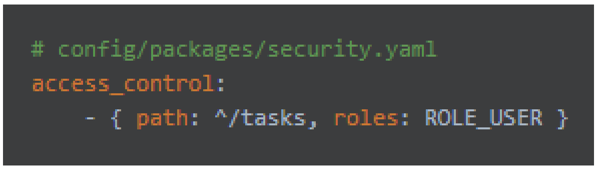


```yaml
# config/packages/security.yaml
role_hierarchy:
    ROLE_ADMIN: ROLE_USER
```

Access Control

The access control fine tunes the authorization needed to access certain paths, for example some paths can be made accessible to any user or only to admins users.

Example :



```yaml
# config/packages/security.yaml
access_control:
    - { path: ^/tasks, roles: ROLE_USER }
```