Authentication : User guide

The document explains how the authentication works on this application.

User Entity

The user is represented by the user entity in ***/src/Entity/User.php.***
It has to implement the interface ***UserInterface*** to be able to manage user login, security etc...

The *UserInterface* needs 5 methods:

- ***getRoles()***: return the user's role
- ***getPassword()***: return the hash password
- ***getSalt()***: return the salt used for the password hash
- ***getUsername()***: return the unique username
- ***eraseCredentials()***: erase sensitive data

Security Setup

The security system is configured in ***/config/packages/security.yaml***.

1. Encoders

This part describes the hash used to hash the user's password.

```
encoders:
    App\Entity\User: bcrypt
```

2. Providers

The provider explains where the User's information is and how to access it.

```
providers:
    doctrine:
        entity:
            class: App:User
            property: username
```

3.Firewalls

The firewalls config allows you to set up the way to login and logout. It's our authentication system.

```
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: ~
```

4.Access Control

On the access_control part we can define the authorisation access for each page of the website and the users' roles allowed.

```
access_control:
        - { path: ^/tasks, roles: ROLE_USER}
        - { path: ^/users/list, roles: ROLE_ADMIN}
        - { path: ^/login, roles: IS_AUTHENTICATED_ANONYMOUSLY }
```

We can see that we need at least a ROLE_USER for access to the tasks part of the website. The users/list is only accessible by Admin and everybody is allowed on the login page.