FP-Identification-Symfony

Lien vers Auth0 qui contient la base pour l’authentification dans symfony :

https://auth0.com/blog/creating-your-first-symfony-app-and-adding-authentication/

Dans un premier créé l’entité User :

```
symfony console make:user
```

suivre les instructions :
-User
-yes
-email
-yes

copier ce code dans l’entité User :

```php
<?php
namespace App\Entity;
use Doctrine\ORM\Mapping as ORM;
use Symfony\Component\Security\Core\User\UserInterface;
/**
 * @ORM\Entity(repositoryClass="App\Repository\UserRepository")
 */
class User implements UserInterface
{
    /**
     * @ORM\Id()
     * @ORM\GeneratedValue()
     * @ORM\Column(type="integer")
     */
    private $id;
    /**
     * @ORM\Column(type="string", length=180, unique=true)
     */
    private $email;
    /**
     * @ORM\Column(type="json")
     */
    private $roles = [];

    /**
     * @var string The hashed password
     * @ORM\Column(type="string")
     */
    private $password;
    /**
     * @ORM\Column(type="string", length=255)
     */
    private $name;
    public function getId(): ?int
    {
        return $this->id;
    }
    public function getEmail(): ?string
    {
        return $this->email;
    }
    public function setEmail(string $email): self
    {
        $this->email = $email;
        return $this;
    }
    /**
     * A visual identifier that represents this user.
     *
     * @see UserInterface
     */
    public function getUsername(): string
    {
        return (string) $this->email;
    }
    /**
     * @see UserInterface
     */
    public function getRoles(): array
    {
        $roles = $this->roles;
        // guarantee every user at least has ROLE_USER
        $roles[] = 'ROLE_USER';
        return array_unique($roles);
    }
    public function setRoles(array $roles): self
    {
        $this->roles = $roles;
        return $this;
    }
    /**
     * @see UserInterface
     */
    public function getPassword(): string
    {
        return (string) $this->password;
    }
    public function setPassword(string $password): self
    {
        $this->password = $password;
        return $this;
    }
    /**
     * @see UserInterface
     */
    public function getSalt()
    {
        // not needed when using the "bcrypt" algorithm in security.yaml
    }
    /**
     * @see UserInterface
     */
    public function eraseCredentials()
    {
        // If you store any temporary, sensitive data on the user, clear it here
        // $this->plainPassword = null;
    }
    public function getName(): ?string
    {
        return $this->name;
    }
    public function setName(string $name): self
    {
        $this->name = $name;
        return $this;
    }
}
```

créé un RegistrationController pour pouvoir enregistrer un nouvel utilisateur :

symfony console make:controller

inséré ce code dans le RegistrationController :

```php
<?php

namespace App\Controller;

use App\Entity\User;
use App\Form\UserType;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\Routing\Annotation\Route;
use Symfony\Component\Security\Core\Encoder\UserPasswordEncoderInterface;
class RegistrationController extends AbstractController
{
    private $passwordEncoder;
    public function __construct(UserPasswordEncoderInterface $passwordEncoder)
    {
        $this->passwordEncoder = $passwordEncoder;
    }
    /**
     * @Route("/registration", name="registration")
     */
    public function index(Request $request)
    {
        $user = new User();
        $form = $this->createForm(UserType::class, $user);
        $form->handleRequest($request);
        if ($form->isSubmitted() && $form->isValid()) {
            // Encode the new users password
            $user->setPassword($this->passwordEncoder->encodePassword($user, $user->getPassword()));
            // Set their role
            $user->setRoles(['ROLE_USER']);
            // Save
            $em = $this->getDoctrine()->getManager();
            $em->persist($user);
            $em->flush();
            return $this->redirectToRoute('app_login');
        }
        return $this->render('registration/index.html.twig', [
            'form' => $form->createView(),
        ]);
    }
}
```

faire un formulaire pour récupéré les données des utilisateurs :

symfony console make:form

rentrez ce code :

```
class UserType extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder, array $options)
    {
        $builder
            ->add('email', EmailType::class)
            ->add('name', TextType::class)
            ->add('password', RepeatedType::class, [
                'type' => PasswordType::class,
                'first_options' => ['label' => 'Password'],
                'second_options' => ['label' => 'Confirm Password']
            ])
        ;
    }

    public function configureOptions(OptionsResolver $resolver)
    {
        $resolver->setDefaults([
            'data_class' => User::class,
        ]);
    }
}
```

Créez un controller SecurityController qui servira plus tard :

symfony console make:controller SecurityController

Généré l’authentification :

symfony console make:auth
suivre les indication :
- 1
- LoginFormAuthenticator
- SecurityController
- yes

Ouvrir le SecurityController dans `src/Controller/SecurityController.php et le remplir avec :`

```php
<?php

namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;
use Symfony\Component\Security\Http\Authentication\AuthenticationUtils;

class SecurityController extends AbstractController
{
   /**
    * @Route("/", name="app_login")
    */
   public function login(AuthenticationUtils $authenticationUtils): Response
   {
      // get the login error if there is one
      $error = $authenticationUtils->getLastAuthenticationError();
      // last username entered by the user
      $lastUsername = $authenticationUtils->getLastUsername();

      return $this->render('security/login.html.twig', ['last_username' => $lastUsername, 'error' => $error]);
   }

   /**
    * @Route("/logout", name="app_logout")
    */
   public function logout()
   {
      throw new \Exception('This method can be blank - it will be intercepted by the logout key on your firewall');
   }
}
```

Créez une base de données, utilisez cette commande pour créé les tables :

```
php bin/console doctrine:schema:update --force
```

Ouvrez le fichier Security.yalm dans /config/packages/security.yalm et le remplir avec :

```
security:
    enable_authenticator_manager: true
    # https://symfony.com/doc/current/security.html#registering-the-user-hashing-passwords
    password_hashers:
        Symfony\Component\Security\Core\User\PasswordAuthenticatedUserInterface: 'auto'
    # https://symfony.com/doc/current/security.html#loading-the-user-the-user-provider
    encoders:
        App\Entity\User:
            algorithm: auto
    providers:
        # used to reload user from session & other features (e.g. switch_user)
        app_user_provider:
            entity:
                class: App\Entity\User
                property: email #propriété utiliser pour différencier les utilisateur(peut être username ou uuid)
    firewalls:
        dev:
            pattern: ^/(_(profiler|wdt)|css|images|js)/
            security: false
        main:
            lazy: true
            provider: app_user_provider
            custom_authenticator: App\Security\LoginFormAuthenticator
            logout: #ligne qui gère la deconnexion
                path: app_logout
                target: /
                # where to redirect after logout
                # target: app_any_route

            # activate different ways to authenticate
            # https://symfony.com/doc/current/security.html#the-firewall

            # https://symfony.com/doc/current/security/impersonating_user.html
            # switch_user: true

    # Easy way to control access for large sections of your site
    # Note: Only the *first* access control that matches will be used
    access_control: #permet de définir quel rôle aura accès à quel page grâce au chemin de la page et au nom du rôle
        # - { path: ^/admin, roles: ROLE_ADMIN }
        # - { path: ^/profile, roles: ROLE_USER }


when@test:
    security:
        password_hashers:
            # By default, password hashers are resource intensive and take time. This is
            # important to generate secure password hashes. In tests however, secure hashes
            # are not important, waste resources and increase test times. The following
            # reduces the work factor to the lowest possible values.
            Symfony\Component\Security\Core\User\PasswordAuthenticatedUserInterface:
                algorithm: auto
                cost: 4 # Lowest possible value for bcrypt
                time_cost: 3 # Lowest possible value for argon
                memory_cost: 10 # Lowest possible value for argon
```

Si un username ou un uuid à été choisi pour a=différencier les utilisateur au moment de faire un make:user modifier la ligne property: email en property: username ou property: username

et modifier plus tard le /*templates*/registration/index.html;twig en conséquence.

Dans le LoginFormAuthenticator.php configurez la fonction onAuthenticationSuccess comme ceci:

```
public function onAuthenticationSuccess(Request $request, TokenInterface $token, string $firewallName): ?Response
    {
        if ($targetPath = $this->getTargetPath($request->getSession(), $firewallName)) {
            return new RedirectResponse($targetPath);
        }

        // For example:
        // return new RedirectResponse($this->urlGenerator->generate('some_route'));
//        throw new \Exception('TODO: provide a valid redirect inside '.__FILE__);
        return new RedirectResponse($this->urlGenerator->generate('list'));
    }
```

Cette fonction nous permet de rediriger l'utilisateur vers une page lorsqu'il se connecte avec succès.

Configurer les vues twig :

Si on veux qu'un utilisateur ai accès à un contenu une fois connecter utiliser :

```
{% if app.user %}

{% endif %}
```

Par exemple :

```
{% if app.user %}
    <li><a class="nav-link" href="{{ path('list') }}">View List</a></li>
    <li><a class="nav-link" href="{{ path('app_logout') }}">Logout</a></li>
{% endif %}
```

Ce qui permet à un utilisateur qui s'est connecter de voir les boutton permettant d'accéder à une liste et de se déconnecter

ou :

```
{% endif %} {% if app.user %}
    <div class="mb-3">
        You are logged in as {{ app.user.name }},
        <a href="{{ path('app_logout') }}">Logout</a>
    </div>
{% endif %}
```

permet de récupéré le nom de l'utilisateur connecter et de lui afficher un message et un lien pour se déconnecter.

On peu aussi utiliser `{{ app.user.userIdentifier }}` :

```
 {% if app.user %}
    <div class="mb-3">
```

```
        You are logged in as {{ app.user.userIdentifier }},
        <a href="{{ path('app_logout') }}">Logout</a>
    </div>

{% endif %}
```

Par contre qui on veux qu’un utilisateur ai accès à un contenu sans être connecter utiliser :

```
{% if not app.user %}

{% endif %}
```

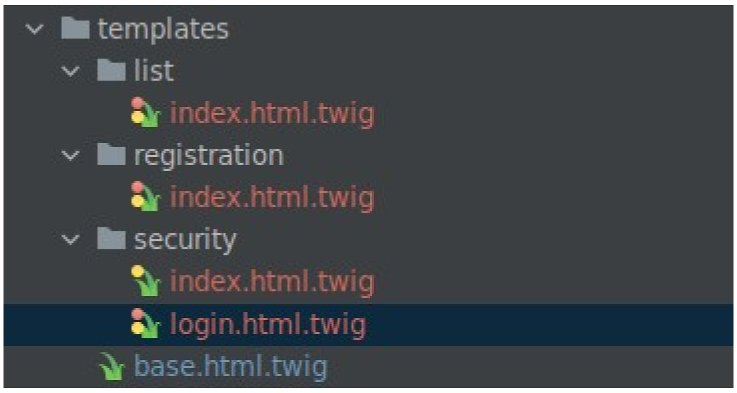
Par exemple :

```
{% if not app.user %}
    <li><a class="nav-link" href="{{ path('app_login') }}">Login</a></li>
    <li>
        <a class="nav-link" href="{{ path('registration') }}">Register</a>
    </li>
{% endif %}
```

Bloc pour récupéré un erreur :

```
{% if (error) %}
    <div class="alert alert-danger">
        {{ error.messageKey|trans(error.messageData, 'security') }}
    </div>
{% endif %}
```

Exemple d’arborescence de templates avec le code :



```
templates
    list
        index.html.twig
    registration
        index.html.twig
    security
        index.html.twig
        login.html.twig
    base.html.twig
```

/list/index :

```
{# templates/list/index.html.twig #} {% extends 'base.html.twig' %} {% block body %}
<div class="container">
    <div class="row">
        <div class="col-md-12">
            <div class="card bg-light mb-3 mt-3">
                <div class="card-body">
                    <div class="card-header">List of top technology companies</div>
                    {% if app.user %}
                    <table class="table">
                        <tr>
                            <th>Company Name</th>
                            <th>Market Value</th>
                        </tr>
                        {% for key, item in companies %}
                        <tr>
                            <td>{{ key }}</td>
                            <td>{{ item }}</td>
                        </tr>
                        {% endfor %}
                    </table>
                    {% endif %}
                </div>
            </div>
            {% if not app.user %}
                <a href="{{ path('app_login') }}" class="btn btn-info">
                    You need to login to see the list 😜 😜  >></a
                >

            {% endif %}
        </div>
    </div>
</div>
{% endblock %}
```

/registration/index :

```
{# templates/registration/index.html.twig #} {% extends 'base.html.twig' %} {%
    block body %}
    <div class="container">
        <div class="row">
            <div class="col-md-10 ml-md-auto">
                <div class="card bg-light mb-3 mt-5" style="width: 800px">
                    <div class="card-body">
                        <div class="card-header mb-3">Registration Form</div>
                        {{ form_start(form) }}
                        <div class="form_group">
                            <div class="col-md-12 mb-3">
                                {{ form_row(form.name, {'attr': {'class': 'form-control'}}) }}
                            </div>
                        </div>
                        <div class="form_group">
                            <div class="col-md-12 mb-3">
                                {{ form_row(form.email, {'attr': {'class': 'form-control'}}) }}
                            </div>
                        </div>
                        <div class="form_group">
                            <div class="col-md-12 mb-3">
                                {{ form_row(form.password.first, {'attr': {'class':
                                    'form-control'}}) }}
                            </div>
                        </div>
                        <div class="form_group">
                            <div class="col-md-12 mb-3">
                                {{ form_row(form.password.second, {'attr': {'class':
                                    'form-control'}}) }}
                            </div>
                        </div>
                        <div class="form-group">
                            <div class="col-md-8 col-md-offset-4" style="margin-top:5px;">
                                <button type="submit" class="btn btn-primary">
                                    <i class="fa fa-btn fa-user"></i> Register
                                </button>
                            </div>
                        </div>
                        {{ form_end(form) }}
                    </div>
                </div>
            </div>
        </div>
    </div>
{% endblock %}
```

security/index :

```
{# templates/security/login.html.twig #} {% extends 'base.html.twig' %} {% block
    title %}Log in!{% endblock %} {% block body %}
    <div class="container">
        <div class="row">
            <div class="col-md-10 ml-md-auto">
                <div class="">
                    <div class="card bg-light mb-3 mt-5" style="width: 800px;">
                        <div class="card-body">
                            <form class="form-horizontal" role="form" method="post">
                                {% if not app.user %}
                                    <div class="alert alert-danger">
                                        {{ error.messageKey|trans(error.messageData, 'security') }}
                                    </div>

                                {% endif %} {% if app.user %}
                                    <div class="mb-3">
                                        You are logged in as {{ app.user.name }},
                                        <a href="{{ path('app_logout') }}">Logout</a>
                                    </div>

                                {% endif %}
                                <div class="card-header mb-3">Please sign in</div>
                                <div class="form-group">
                                    <label for="email" class="col-md-4 control-label"
                                    >E-Mail Address</label
                                    >
                                    <div class="col-md-12">
                                        <label for="inputEmail"></label><input
                                                id="inputEmail"
                                                type="email"
                                                class="form-control"
                                                name="email"
                                                value="{{ last_username }}"
                                                required
                                                autofocus
                                        />
                                    </div>
                                </div>
                                <div class="form-group">
                                    <label for="password" class="col-md-4 control-label"
                                    >Password</label
                                    >
                                    <div class="col-md-12">
                                        <label for="inputPassword"></label><input
                                                id="inputPassword"
                                                type="password"
                                                class="form-control"
                                                name="password"
                                                required
                                        />
                                    </div>
                                </div>
                                <input
                                        type="hidden"
                                        name="_csrf_token"
                                        value="{{ csrf_token('authenticate') }}"
                                />
                                <div class="form-group">
                                    <div class="col-md-12">
                                        <button type="submit" class="btn btn-primary">
                                            <i class="fa fa-btn fa-sign-in"></i> Login
                                        </button>
                                    </div>
                                </div>
                            </form>
                        </div>
                    </div>
                </div>
            </div>
        </div>
{% endblock %}
```

/security/login :

```
{# templates/security/login.html.twig #} {% extends 'base.html.twig' %} {% block
  title %}Log in!{% endblock %} {% block body %}
  <div class="container">
    <div class="row">
      <div class="col-md-10 ml-md-auto">
        <div class="">
          <div class="card bg-light mb-3 mt-5" style="width: 800px;">
            <div class="card-body">
              <form class="form-horizontal" role="form" method="post">
                {% if (error) %}
                  <div class="alert alert-danger">
                    {{ error.messageKey|trans(error.messageData, 'security') }}
                  </div>

                {% endif %} {% if app.user %}
                  <div class="mb-3">
                    You are logged in as {{ app.user.userIdentifier }},
                    <a href="{{ path('app_logout') }}">Logout</a>
                  </div>

                {% endif %}
                <div class="card-header mb-3">Please sign in</div>
                <div class="form-group">
                  <label for="email" class="col-md-4 control-label"
                  >E-Mail Address</label
```

```
                        >
                        <div class="col-md-12">
                          <label for="inputEmail"></label><input
                              id="inputEmail"
                              type="email"
                              class="form-control"
                              name="email"
                              value="{{ last_username }}"
                              required
                              autofocus
                          />
                        </div>
                      </div>
                      <div class="form-group">
                        <label for="password" class="col-md-4 control-label"
                        >Password</label
                        >
                        <div class="col-md-12">
                          <label for="inputPassword"></label><input
                              id="inputPassword"
                              type="password"
                              class="form-control"
                              name="password"
                              required
                          />
                        </div>
                      </div>
                      <input
                          type="hidden"
                          name="_csrf_token"
                          value="{{ csrf_token('authenticate') }}"
                      />
                      <div class="form-group">
                        <div class="col-md-12">
                          <button type="submit" class="btn btn-primary">
                            <i class="fa fa-btn fa-sign-in"></i> Login
                          </button>
                        </div>
                      </div>
                    </form>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
{% endblock %}
```

base.html.twig

```
{# templates/base.html.twig #}
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8" />
  <title>{% block title %}Welcome!{% endblock %}</title>
  <link
      rel="stylesheet"
      href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
```

```
    />
    {% block stylesheets %}{% endblock %}
</head>
<body>
<nav
        class="navbar navbar-expand-lg navbar-light bg-light"
        style="height: 70px;"
>
    <a class="navbar-brand" href="#">Symfony</a>
    <div class="collapse navbar-collapse" id="navbarSupportedContent"></div>
    <ul class="nav navbar-nav navbar-right">
        {% if app.user %}
            <li><a class="nav-link" href="{{ path('list') }}">View List</a></li>
            <li><a class="nav-link" href="{{ path('app_logout') }}">Logout</a></li>
        {% endif %}

        {% if not app.user %}
            <li><a class="nav-link" href="{{ path('app_login') }}">Login</a></li>
            <li>
                <a class="nav-link" href="{{ path('registration') }}">Register</a>
            </li>
        {% endif %}
    </ul>
</nav>
{% block body %}{% endblock %} {% block javascripts %}{% endblock %}
</body>
</html>
```