

Table des matières

[1. Le composant sécurité de Symfony 3](#_Toc13058916)

[2. Entité User. 4](#_Toc13058917)

[3. Création du formulaire d’inscription. 4](#_Toc13058918)

[4.Hasher les mots de passe. 5](#_Toc13058919)

[5. Rendre les User uniques. 5](#_Toc13058920)

[6. Création de la page Login. 6](#_Toc13058921)

[7.La déconnexion. 7](#_Toc13058922)

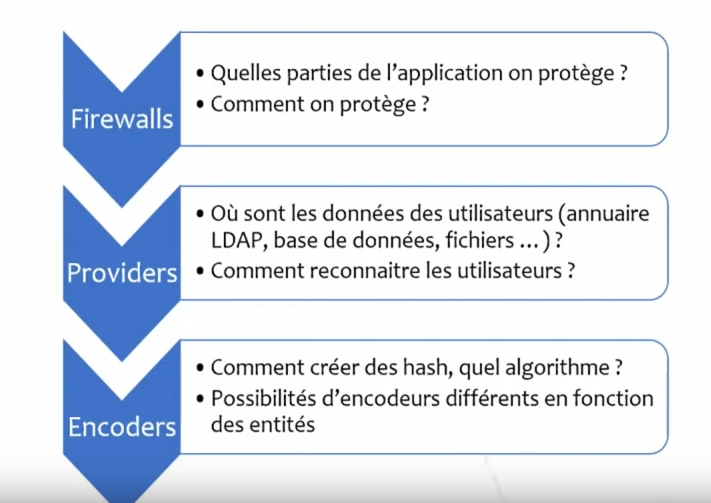
[8. Variable d’environnement. 7](#_Toc13058923)

[9. Formulaire de commentaire avec réquisition de connexion. 8](#_Toc13058924)

[10. Astuces dignes de ce nom. 9](#_Toc13058925)

Tuto 4/4, blog :

# 1. Le composant sécurité de Symfony



On retrouve le dossier sécurité dans config/package/security.yaml

providers:  
 in\_memory: { memory: ~ }  
firewalls:  
 Dev:  
 pattern: ^/(\_(profiler|wdt)|css|images|js)/  
 security: false  
 main:  
 anonymous: true

# 2. Entité User.

Création de l’entité :

php bin/console make:entity

Avec un email, un username, un password.

Ensuite on envoi le tout via une migration pour mettre à jour la base.

# 3. Création du formulaire d’inscription.

Création du formulaire via la console, RegistrationType.

php bin/console make:form RegistrationType

Ensuite création du controller :

php bin/console make:controller

Le SecurityController.

*// PasswordType pour masquer le mot de passe*->add('password', PasswordType::*class*)

On ajout le composant sécurité dans l’entité User :

*use* Symfony\Component\Validator\Constraints *as* Assert;

Ajout d’une contrainte pour le password :

*/\*\*  
 \** ***@ORM\Column****(type="string", length=255)  
 \** ***@Assert\Length****(min="8", minMessage="Votre mot de passe doit faire 8 caractères")  
 \*/  
private* $password;

Ajout de la contrainte password = confirm\_password :

*\** ***@Assert\EqualTo****(propertyPath="confirm\_password")*

Ne pas oublier d’ajouter le message pour ne pas dévoiler le MDP :

*\** ***@Assert\EqualTo****(propertyPath="confirm\_password",message="Le mot de passe ne correspond pas")*

# 4.Hasher les mots de passe.

Dans le fichier security.yaml :

security:  
 encoders:  
 App\Entity\User:  
 algorithm: bcrypt

Ajout dans le contrôleur le UserpasswordEncoder

*public function* registration (Request $request, ObjectManager $manager,  
UserPasswordEncoderInterface $encoder){

Ensuite implément de l’interface dans la class User avec les méthodes :

*class* User *implements* UserInterface  
{  
 *public function* getRoles()  
 {  
 *return* ['ROLE\_USER'];  
 }  
  
 *public function* getSalt()  
 {  
 *// TODO: Implement getSalt() method.* }  
  
 *public function* eraseCredentials()  
 {  
 *// TODO: Implement eraseCredentials() method.* }

# 5. Rendre les User uniques.

Ajout de l’assert UniqueEntity dans la class User :

*/\*\*  
 \** ***@ORM\Entity****(repositoryClass="App\Repository\UserRepository")  
 \** ***@UniqueEntity****()  
 \*/*

Et la contrainte liée au mail

*\** ***@UniqueEntity****(  
\* fields={"email},  
\* message ="l'email que vous avez indiqué est déja utilisé")*

# 6. Création de la page Login.

Création de la route dans le Controller.

*/\*\*  
 \** ***@Route*** *("/connexion", name="security\_login"  
 \*/  
public function* login(){  
 *return* $this->render('security\login.html.twig');  
}

Le formulaire de Login



property: email

Ceci précise que c’est le champ email qui permet de se connecter et qu’il est nécessaire à vérifier l’authentification.

providers:  
 in\_memory: { memory: ~ }  
 in\_database:  
 entity:  
 class: App\Entity\User  
 property: email

La vue :

{% *extends* 'base.html.twig' %}  
{% *block* body %}  
 <h1>Connexion</h1>  
  
 <form *action*="{{ path('security\_login') }}" *method*="post">  
 <div *class*="form-group">  
 <input *placeholder*="Adresse mail..." *required name*="\_username" *type*="text"  
 *class*="form-control">  
 </div>  
 <div *class*="form-group">  
 <input *placeholder*="Mot de passe" *required name*="\_password" *type*="password"  
 *class*="form-control">  
 </div>  
 <div *class*="form-group">  
 <button *type*="submit" *class*="btn btn-success">Connexion !</button>  
 </div>  
 </form>  
{% *endblock* %}

# 7. La déconnexion.

Ajout d’une route qui ne mène à rien dans le controller :

*/\*\*  
 \** ***@Route****("/deconnexion", name="security\_logout")  
 \*/  
 public function* logout(){}

Dans le security.yaml ou ajout le logout :

form\_login:  
 login\_path: security\_login  
 check\_path: security\_login  
  
logout:  
 path: security\_logout  
 target: /

# 8. Variable d’environnement.

Dans la vue ‘base.html.twig’

{% *if* app.user %}

Ou

{% *if not* app.user %}

On l’applique aux boutons :

{% *if not* app.user %}  
 <li *class*="nav-item">  
 <a *class*="nav-link" *href*="{{ path('security\_login') }}"> Connexion</a>  
 </li>  
  
{% *else %}*  
 <li *class*="nav-item">  
 <a *class*="nav-link" *href*="{{ path('security\_logout') }}"> Déconnexion</a>  
 </li>  
{% *endif* %}

# 9. Formulaire de commentaire avec réquisition de connexion.

Formulaire Comment :

Php bin/console make : form

Dans le contrôleur :

*//affichage d'un seul article  
/\*\*  
 \** ***@Route****("blog/show/{id}", name="blog\_show")  
 \*/  
public function* show(Article $article, Request $request, ObjectManager $manager){  
 $comment =*new* Comment();  
 $form = $this -> createForm(CommentType::*class*, $comment);  
  
 $form ->handleRequest($request);  
 *if* ($form -> isSubmitted() && $form->isValid()){  
 $manager->persist($comment);  
 }  
  
 *return* $this -> render ('blog/show.html.twig',[  
 'article'=> $article,  
 'commentForm' =>$form -> createView()  
 ]);  
}

Le show :

{% *extends* 'base.html.twig' %}  
  
{% *block* body %}  
  
 <article>  
 <h2 >{{ article.title }} </h2>  
 <div *class*="metadata"> Ecrit le {{ article.createdAt | date('d/m/Y') }} à {{ article.createdAt | date ('H:i') }} dans la catégorie  
 {{ article.category.title }}</div>  
 <div *content*>  
 <img *src*='{{ article.image }}' *alt*="">  
 {{ article.content | raw }}  
 </div>  
 </article>  
  
 <section *id*="commentaires">  
 <h1>{{ article.comments|length }} commentaires :</h1>  
 {% *for* comment *in* article.comments %}  
 <div *class*="comment">  
 <div *class*="row">  
 <div *class*="col-3">  
 {{ comment.author }} (<small>{{ comment.createdAt | date ('d/m/Y à H:i') }}</small>)  
 </div>  
 <div *class*="col">  
 {{ comment.content | raw }}  
 </div>  
 </div>  
 {% *endfor* %}  
  
 {% *if* app.user %}  
 {{ form\_start(commentForm) }}  
 {{ form\_row(commentForm.author,{'attr':{'placeholder':"Votre nom" }})}}  
 {{ form\_row(commentForm.content,{'attr':{'placeholder':"Votre commentaire"}}) }}  
 <button *type*="submit" *class*="btn btn-success">Commenter</button>  
 {{ form\_end(commentForm) }}  
 {% *else* %}  
 <h2>Vous ne pouvez pas commenter si vous n'etes pas connecté</h2>  
 <a *href*="{{ path('security\_login') }}" *class*="btn btn-primary">  
 Connexion  
 </a>

# 10. Astuces dignes de ce nom.

🔗 Site officiel de Symfony (et sa documentation) : [https://symfony.com/](https://www.youtube.com/redirect?q=https%3A%2F%2Fsymfony.com%2F&event=video_description&v=_GjHWa9hQic&redir_token=7BLNGnNH42bsm5YwVEUKt3XGZNB8MTU2MjIyOTg4MkAxNTYyMTQzNDgy)

🔗 Tutoriel officiel de présentation du composant Security : [https://symfony.com/doc/current/secur...](https://www.youtube.com/redirect?q=https%3A%2F%2Fsymfony.com%2Fdoc%2Fcurrent%2Fsecurity.html&event=video_description&v=_GjHWa9hQic&redir_token=7BLNGnNH42bsm5YwVEUKt3XGZNB8MTU2MjIyOTg4MkAxNTYyMTQzNDgy)

🔗 Mieux comprendre les Firewalls : [https://symfony.com/doc/current/secur...](https://www.youtube.com/redirect?q=https%3A%2F%2Fsymfony.com%2Fdoc%2Fcurrent%2Fsecurity%2Ffirewall_restriction.html&event=video_description&v=_GjHWa9hQic&redir_token=7BLNGnNH42bsm5YwVEUKt3XGZNB8MTU2MjIyOTg4MkAxNTYyMTQzNDgy)

🔗 Documentation sur l'option form\_login : [https://symfony.com/doc/current/secur...](https://www.youtube.com/redirect?q=https%3A%2F%2Fsymfony.com%2Fdoc%2Fcurrent%2Fsecurity%2Fform_login_setup.html&event=video_description&v=_GjHWa9hQic&redir_token=7BLNGnNH42bsm5YwVEUKt3XGZNB8MTU2MjIyOTg4MkAxNTYyMTQzNDgy)

🔗 Comment charger les utilisateurs à partir de la base de données : [https://symfony.com/doc/current/secur...](https://www.youtube.com/redirect?q=https%3A%2F%2Fsymfony.com%2Fdoc%2Fcurrent%2Fsecurity%2Fentity_provider.html&event=video_description&v=_GjHWa9hQic&redir_token=7BLNGnNH42bsm5YwVEUKt3XGZNB8MTU2MjIyOTg4MkAxNTYyMTQzNDgy)

🔗 Comment utiliser le UserPasswordEncoder : [https://symfony.com/doc/current/secur...](https://www.youtube.com/redirect?q=https%3A%2F%2Fsymfony.com%2Fdoc%2Fcurrent%2Fsecurity%2Fpassword_encoding.html&event=video_description&v=_GjHWa9hQic&redir_token=7BLNGnNH42bsm5YwVEUKt3XGZNB8MTU2MjIyOTg4MkAxNTYyMTQzNDgy)

🔗 Créer un lien de déconnexion : [https://symfony.com/doc/current/secur...](https://www.youtube.com/redirect?q=https%3A%2F%2Fsymfony.com%2Fdoc%2Fcurrent%2Fsecurity.html%23security-logging-out&event=video_description&v=_GjHWa9hQic&redir_token=7BLNGnNH42bsm5YwVEUKt3XGZNB8MTU2MjIyOTg4MkAxNTYyMTQzNDgy)

🔗 Comment accéder à l'utilisateur connecté dans Twig : [https://symfony.com/doc/current/templ...](https://www.youtube.com/redirect?q=https%3A%2F%2Fsymfony.com%2Fdoc%2Fcurrent%2Ftemplating%2Fapp_variable.html%23content_wrapper&event=video_description&v=_GjHWa9hQic&redir_token=7BLNGnNH42bsm5YwVEUKt3XGZNB8MTU2MjIyOTg4MkAxNTYyMTQzNDgy) 🔗 Documentation complète du fichier de configuration du package Security : [https://symfony.com/doc/current/refer...](https://www.youtube.com/redirect?q=https%3A%2F%2Fsymfony.com%2Fdoc%2Fcurrent%2Freference%2Fconfiguration%2Fsecurity.html&event=video_description&v=_GjHWa9hQic&redir_token=7BLNGnNH42bsm5YwVEUKt3XGZNB8MTU2MjIyOTg4MkAxNTYyMTQzNDgy)

💪🔥 BONUS : -----------------------

🔗 Comment gérer une option "Se rappeler de moi" : [https://symfony.com/doc/current/secur...](https://www.youtube.com/redirect?q=https%3A%2F%2Fsymfony.com%2Fdoc%2Fcurrent%2Fsecurity%2Fremember_me.html&event=video_description&v=_GjHWa9hQic&redir_token=7BLNGnNH42bsm5YwVEUKt3XGZNB8MTU2MjIyOTg4MkAxNTYyMTQzNDgy)

🔗 Comment gérer les accès dans le controller : [https://symfony.com/doc/current/secur...](https://www.youtube.com/redirect?q=https%3A%2F%2Fsymfony.com%2Fdoc%2Fcurrent%2Fsecurity%2Fsecuring_services.html&event=video_description&v=_GjHWa9hQic&redir_token=7BLNGnNH42bsm5YwVEUKt3XGZNB8MTU2MjIyOTg4MkAxNTYyMTQzNDgy)

🔗 Comment rediriger là où on veut après le login : [https://symfony.com/doc/current/secur...](https://www.youtube.com/redirect?q=https%3A%2F%2Fsymfony.com%2Fdoc%2Fcurrent%2Fsecurity%2Fform_login.html&event=video_description&v=_GjHWa9hQic&redir_token=7BLNGnNH42bsm5YwVEUKt3XGZNB8MTU2MjIyOTg4MkAxNTYyMTQzNDgy)

🔗 Comment utiliser les annotations pour sécuriser les controllers : [https://symfony.com/doc/master/bundle...](https://www.youtube.com/redirect?q=https%3A%2F%2Fsymfony.com%2Fdoc%2Fmaster%2Fbundles%2FSensioFrameworkExtraBundle%2Fannotations%2Fsecurity.html&event=video_description&v=_GjHWa9hQic&redir_token=7BLNGnNH42bsm5YwVEUKt3XGZNB8MTU2MjIyOTg4MkAxNTYyMTQzNDgy)