



# Formation PHP - Symfony

## D06 - Authentication & Authorization

Rares Serban [rares.serban@pitechplus.com](mailto:rares.serban@pitechplus.com)  
42 Staff [pedago@staff.42.fr](mailto:pedago@staff.42.fr)

*Résumé: Cette journée est consacrée aux composantes de Symfony relatives au système de sécurité, authentification et autres autorisations.*

# Table des matières

<b>I</b>	<b>Préambule</b>	<b>2</b>
<b>II</b>	<b>Consignes</b>	<b>3</b>
<b>III</b>	<b>Règles spécifiques de la journée</b>	<b>4</b>
<b>IV</b>	<b>Exercice 01</b>	<b>5</b>
<b>V</b>	<b>Exercice 02</b>	<b>6</b>
<b>VI</b>	<b>Exercice 03</b>	<b>7</b>
<b>VII</b>	<b>Exercice 04</b>	<b>8</b>
<b>VIII</b>	<b>Exercice 05</b>	<b>9</b>
<b>IX</b>	<b>Exercice 06</b>	<b>10</b>
<b>X</b>	<b>Exercice 07</b>	<b>11</b>

# Chapitre I

## Préambule

Aujourd’hui vous allez découvrir le composant *Security* du framework Symfony et comment il peut être utilisé pour créer les flux d’authentification (enregistrement, connexion) dans une application..

Vous apprendrez aussi comment sécuriser certaines parties de votre site web pour les utilisateurs authentifiés et même restreindre certaines fonctionnalité aux utilisateurs avec certains rôles.

# Chapitre II

## Consignes

Sauf contradiction explicite, les consignes suivantes seront valables pour tous les jours de cette Piscine.

- Seul ce sujet sert de référence : ne vous fiez pas aux bruits de couloir.
- Le sujet peut changer jusqu'à une heure avant le rendu.
- Les exercices sont très précisément ordonnés du plus simple au plus complexe. En aucun cas nous ne porterons attention ni ne prendrons en compte un exercice complexe si un exercice plus simple n'est pas parfaitement réussi.
- Attention aux droits de vos fichiers et de vos répertoires.
- Vous devez suivre la procédure de rendu pour tous vos exercices. L'url de votre dépôt GIT pour cette journée est disponible sur votre intranet.
- Vos exercices seront évalués par vos camarades de Piscine.
- En plus de vos camarades, vous pouvez être évalués par un programme appelé la Moulinette. La Moulinette est très stricte dans sa notation car elle est totalement automatisée. Il est donc impossible de discuter de sa note avec elle. Soyez d'une rigueur irréprochable pour éviter les mauvaises surprises.
- Les exercices shell doivent s'exécuter avec `/bin/sh`.
- Vous ne devez laisser aucun autre fichier que ceux explicitement spécifiés par les énoncés des exercices dans votre dépôt de rendu.
- Vous avez une question ? Demandez à votre voisin de droite. Sinon, essayez avec votre voisin de gauche.
- Toutes les réponses à vos questions techniques se trouvent dans les `man` ou sur Internet.
- Pensez à discuter sur le forum Piscine de votre Intra et sur Slack !
- Lisez attentivement les exemples car ils peuvent vous permettre d'identifier un travail à réaliser qui n'est pas précisé dans le sujet à première vue.
- Réfléchissez. Par pitié, par Thor, par Odin !


# Chapitre III

## Règles spécifiques de la journée

- Pour cette journée, votre repository doit contenir seulement une application Symfony.
- Vous devrez respecter les bonnes pratiques du framework Symfony.
- Vous pouvez utiliser des bundles ou libraries tierces.
- Chaque exercice aura son propre nom de bundle de type *ExxBundle*, où *xx* est le numéro de l'exercice.
- Chaque exercice aura sa propre route sous la forme */eXX*, où *XX* est le numéro de l'exercice. Cependant les actions et contrôleurs peuvent avoir les noms que vous souhaitez.

# Chapitre IV

## Exercice 01


	Exercice :
Exercice 01 : Login & Création de compte	
Dossier de rendu : <i>ex/</i>	
Fichiers à rendre : <b>Fichiers et répertoires de votre application</b>	
Fonctions Autorisées : <b>Toute méthode</b>	
Remarques : <b>n/a</b>	

Premièrement, créer une nouvelle application Symfony puis créer un bundle appelé E01Bundle. Toute les demande de cet exercice seront incluse dans ce bundle. Dans la même logique, les exercices suivants seront résolus dans leur bundles nommés respectivement.

Ensuite, mettre en place la sécurité Symfony et ajouter un formulaire de connexion et d'enregistrement. Vos utilisateurs devront être chargés depuis une base de données. Ajouter ensuite une page d'accueil qui affichera un message de bienvenue contenant le nom de l'utilisateur actuellement identifié et un bouton de déconnexion. Si aucun utilisateur n'est connecté la page devra afficher des liens vers la page contenant le formulaire de connexion et vers la page d'enregistrement si l'utilisateur ne dispose pas encore de compte.

# Chapitre V

## Exercice 02


	Exercice :
Exercice 02 : Administration & roles des utilisateurs	
Dossier de rendu : <i>ex/</i>	
Fichiers à rendre : <b>Fichiers et répertoires de votre application</b>	
Fonctions Autorisées : <b>Toute méthode</b>	
Remarques : <b>n/a</b>	

Créer un rôle administrateur pouvant être accordé à vos utilisateurs. Les administrateurs auront accès à une page où l'on pourra voir tous les utilisateurs enregistrés (ayant un compte).

Sur cette page l'administrateur aura la possibilité de supprimer un utilisateur mais ne devrait pas pouvoir supprimer son propre compte utilisateur.

# Chapitre VI

## Exercice 03

	Exercice :
Exercice 03 : Publications	
Dossier de rendu : <i>ex/</i>	
Fichiers à rendre : Fichiers et répertoires de votre application	
Fonctions Autorisées : Toute méthode	
Remarques : n/a	

Créer une nouvelle entité nommée **Post** qui devra avoir au moins les champs suivants : *title*, *content*, *created* et *author*. La publication (post) devra être associée à l'utilisateur qui l'a créé.


La page d'accueil devra maintenant afficher une liste de toutes les publications créées ordonnées de la plus récente à la plus ancienne. La liste affichera le titre (title) de la publication, le nom de son auteur et la date de création.

Chaque titre devra avoir un lien hypertexte permettant de visualiser les détails de la publication. Ce lien ne sera actif que pour les utilisateurs authentifiés. La page d'accueil aura également un lien, pour les utilisateurs authentifiés seulement, vers un formulaire permettant de créer une nouvelle publication.



# Chapitre VII

## Exercice 04


	Exercice :
Exercice 04 : Utilisateurs anonymes	
Dossier de rendu : <i>ex/</i>	
Fichiers à rendre : Fichiers et répertoires de votre application	
Fonctions Autorisées : Toute méthode	
Remarques : n/a	

Les sessions anonymes devront avoir une durée de vie de 1 minute seulement. Quand un utilisateur non authentifié accède le site, il se verra assigné un nom aléatoire, qui proviendra d'une liste de noms d'animaux (en anglais) que vous aurez définie auparavant, précédé par la mot *anonymous*. Ex : anonymous dog.

Ce nom devra être affiché sur la page d'accueil. Un message sera également affiché sur la page d'accueil montrant le temps (en secondes) depuis que la dernière requête a été faite.

# Chapitre VIII

## Exercice 05

	Exercice :
Exercice 05 : Votes	
Dossier de rendu : <i>ex/</i>	
Fichiers à rendre : Fichiers et répertoires de votre application	
Fonctions Autorisées : Toute méthode	
Remarques : n/a	


Ajouter la possibilité pour les utilisateurs authentifiés de voter pour ou contre une publication (like & unlike). Ce lien pour voter sera présent uniquement sur la page présentant les détails de la publication. Un utilisateur ne peut voter qu'une fois pour une publication.

La liste des publications sur la page d'accueil devra afficher le nombre de like et de unlike pour chaque publication de la liste.

Chaque fois qu'un utilisateur accède le site, son score de *réputation* devra être affiché à côté de son nom. Ce score est constitué du nombre de vote positifs, moins le nombre de votes négatifs que l'ensemble de ses publication reçoivent.

# Chapitre IX

## Exercice 06


	Exercice :
Exercice 06 : Modification de publications	
Dossier de rendu : <i>ex/</i>	
Fichiers à rendre : Fichiers et répertoires de votre application	
Fonctions Autorisées : Toute méthode	
Remarques : n/a	

Les utilisateurs authentifiés devrait pouvoir modifier une publication. Sur la page des détails d'une publication on affichera ainsi un bouton *Modifier*.

A la fin de la page des détails d'une publication on affichera un text indiquant quel utilisateur à modifié la publication en dernier et à quelle date et heure, si la publication a été modifiée. Logged in users should be able to edit a post. On the detail page of a post an edit button should be shown.

# Chapitre X

## Exercice 07

	Exercice :
Exercice 07 : Privilèges utilisateurs	
Dossier de rendu : <i>ex/</i>	
Fichiers à rendre : <b>Fichiers et répertoires de votre application</b>	
Fonctions Autorisées : <b>Toute méthode</b>	
Remarques : n/a	

Selon le score de réputation d'un utilisateur, il bénéficiera des privilèges suivants :

- 0 points (nouvel utilisateur) - droit de créer une publication et modifier leurs propres publications seulement
- 3 points - droit d'aimer les publications
- 6 points - droit de ne pas aimer les publications
- 9+ points - droit d'éditer des publications ne lui appartenant pas

Un admin aura tous ces privilèges quelque soit son score de réputation.

Utilisez **Doctrine Fixtures** afin de peupler la base de données avec des utilisateurs de tests, des publications et des votes afin de faciliter le test des fonctionnalités de votre application.