



# Formation PHP - Symfony

Rush01

42 pedago [pedago@42.fr](mailto:pedago@42.fr)

*Résumé: Dans ce rush vous allez créer une application intranet similaire à l'intra de 42.*

# Table des matières

<b>I</b>	<b>Consignes</b>	<b>2</b>
<b>II</b>	<b>Règles spécifiques de la journée</b>	<b>3</b>
<b>III</b>	<b>Préambule</b>	<b>4</b>
<b>IV</b>	<b>Partie obligatoire</b>	<b>5</b>
<b>V</b>	<b>Partie Bonus</b>	<b>7</b>

# Chapitre I

## 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 II

## Règles spécifiques de la journée

- Vous **DEVEZ** fournir un fichier auteur nommé **auteur** a la racine de votre dossier de rendu, comme d'habitude

```
$> cat -e auteur
noich$
thor$
$>
```

- Ce projet sera entièrement corrigé par des humains. Conséquemment, vous êtes libres d'avoir des variations entre l'exemple donné et le rendu de votre programme. Cependant, vous devez respecter l'esprit du sujet et rendre un programme possédant un rendu ne différant pas trop du sujet ainsi que du format de sortie.
- Pour les mêmes raisons, vous êtes libres de choisir vos noms de fichiers et votre hierarchie générale. Mais gardez en tête qu'une organisation de fichier est une bonne pratique du code et peut vous apporter des points bonus.
- Essayez de respecter les standards de codage de Symfony 2.8 du mieux possible, cela peut vous apporter des points bonus.
- D'autres bundles ou libraries tiers peuvent être utilisés.
- Rappelez-vous qu'il n'y a pas de points bonus pour la mise en page de votre application, seules les fonctionnalités et le code comptent. Cependant vous devez au moins organiser le contenu du site web et rendre le tout facile d'accès.

# Chapitre III

## Préambule

Vous aller créer un intranet pour une organisation de votre choix : une société, une école, une association, un gouvernement, etc. L'intranet aura de multiples types d'utilisateurs et une interface d'administration (backend) afin de gérer divers aspects le concernant.

# Chapitre IV

## Partie obligatoire

Comme il s'agit d'un intranet il ne sera utilisé qu'en interne par une organisation. Pour cette raison il n'y aura pas de formulaire d'enregistrement de nouveaux utilisateurs. Cependant les utilisateurs seront créés par les Administrateurs.

Votre application devra être composée de 2 parties : la partie visible aux utilisateurs (front-end ou front-office) et une partie visible seulement aux administrateurs ou autres rôles (back-end ou back-office) où il pourront gérer le site.

Mettre en place la sécurité et un rôle Administrateur qui aura accès au back-end. Depuis le back-end il pourra créer de nouveaux utilisateurs et leur assigner différents rôles. Vous pouvez décider des rôles qui existeront pour votre application. Un utilisateur devra avoir au moins les attributs suivants : prénom, nom, email, mot de passe, date de création et rôle. Les attributs peuvent être optionnels ou pas et ils auront le type que vous considérez approprié.

Vous devrez également créer une commande qui permettra la création de nouveaux utilisateurs de type Administrateur puisqu'au début votre application n'aura pas d'utilisateurs. Après qu'un utilisateur est créé, il devra recevoir une confirmation par email. Depuis l'email il devra pouvoir cliquer sur un lien qui permettra d'activer son compte. Sur la page d'activation il devra pouvoir saisir un mot de passe pour son compte.

Un utilisateur identifié aura accès à la page de son profil ainsi qu'à un formulaire où il pourra modifier ses informations. Assurez vous d'implémenter également une fonctionnalité "mot de passe oublié". D'autres utilisateurs auront accès au profil d'un utilisateur pour visualiser ses informations. Assurez vous aussi d'implémenter correctement la sécurité afin que des utilisateurs non autorisés ne puissent accéder à des actions spécifiques (ex. visualisation, modification).

Selon le type d'organisation choisi, l'utilisateur aura également accès à diverses sections. Par exemple : une liste des groupes de travail / sections / cours dont il fait partie, la possibilité d'adhérer à une groupe / section / cours, une section actualités où les dernières items pourraient être affichés. Chaque section peut avoir ses propres publications / posts qui peuvent être créées par les administrateurs ou d'autres rôles choisis à partir du back-end. A la place des publications, d'autres type de contenus peuvent être utilisés. Ce serait sympa si on peut également charger des images quand on crée un nouveau contenu.

Indépendamment de votre choix, les utilisateurs sans rôle ne devraient pas avoir le droit de commenter certains des contenus de votre application, ni de répondre aux commentaires des autres utilisateurs. N'oubliez pas d'implémenter une pagination pour vos listes de contenus.

Les utilisateurs devraient pouvoir envoyer des messages à d'autres utilisateurs de votre application. Ces messages n'ont pas besoin d'être des messages en temps réel (mais cela est inclus dans la partie bonus). Vous devrez utiliser des test unitaires afin de tester les services utilisés dans votre application.

Vous êtes libres d'utiliser votre imagination et étendre les fonctionnalités de votre application autant que vous le souhaitez. Vous serez récompensés selon les fonctionnalités supplémentaires existantes, à condition que les fonctionnalités de base sont présentes.

# Chapitre V

## Partie Bonus

Implémenter la possibilité que les commentaires postés par vos utilisateurs soit transmis dynamiquement par des appels Ajax. Vous pouvez aussi implémenter des appels Ajax dans d'autres parties de votre application, par exemple pour permettre aux utilisateurs de voir ou modifier leurs informations sans que cela ne demande le rechargement de la page.

Permettez aux utilisateurs de s'inscrire à des sections / groupes / catégories et recevoir des notifications web lorsque du nouveau contenu a été ajouté à la catégorie en question. Les notifications devraient être gérées en temps réel en utilisant des Websockets.

Implémenter un chat en temps réel pour les utilisateurs en utilisant des Websockets et permettez aux utilisateurs d'envoyer des images ou autres fichiers dans le chat. Si une photo est envoyée, une petite prévisualisation devra apparaître dans le chat.