

Exercices Frameworks Laravel/Symfony

Au travers de cette série d'exercices, vous allez découvrir un framework PHP et apprendre à utiliser les fonctionnalités de base. Ces exercices vous concernent que vous ayez choisi de travailler avec Symfony ou Laravel. Attention cependant, il ne s'agit là que d'une base, ce sera à vous de pousser les tutoriels et de donner libre cours à votre curiosité afin d'apprendre tout ce qu'il y a à savoir sur ces outils !

Au cours des différents exercices qui vont suivre, vous allez réaliser un petit blog très simple mais à l'aide d'un framework

Exercice 1 : Réaliser un système de routing

Toute application a un point de départ et outre la création du dossier contenant les fichiers de votre framework préféré, la première chose que l'on fait est bien souvent de définir des routes. Pour ce premier exercice, vous allez donc créer les routes suivantes, chaque route renverra à une méthode d'un controller qui renverra une vue spécifique :

- Une route qui matchera la racine de votre site ou /index. Ce sera la route par défaut, celle de l'accueil.
- Une route single qui correspondra à la route vers laquelle le visiteur est redirigé quand il clique sur un article. Cette route matchera /article/id. Le paramètre id ne peut être composé que de chiffres et sera mis par défaut à 1 s'il n'est pas rempli.
- Une route pour l'ajout d'un article dans l'espace d'administration qui matchera admin/article/add.

Exercice 2 : Créer des entités/migrations

Vous allez maintenant créer l'entité/migration centrale de votre site, j'ai nommé l'entité article. Elle sera composée des attributs suivants :

- Titre
- Contenu
- Date de création
- Nom de l'auteur associé
- Nom de la catégorie associée
- Nombre de vues

Une fois votre entité/migration créée, mettez à jour votre schéma de base de données, normalement vous devriez voir une nouvelle table apparaître dans MySQL.

Selon votre framework, rajoutez une méthode qui vous retourne les 50 premiers caractères du contenu, elle vous servira à l'exercice suivant

Prenez quelques minutes insérer un jeu de données (vous le faire en mode bricolage dans PhpMyAdmin ou faire les pros et utilisez les outils spécialement conçus à cet effet dans les frameworks) cela nous permettra d'alimenter nos futures vues.

Exercice 3 : Afficher des données

Maintenant que vos routes et votre entité ont été créées, l'heure est venue d'afficher les données sur notre site. Pour cela vous allez réaliser les opérations suivantes :

- Dans la méthode de l'accueil, vous allez récupérer les trois derniers articles insérés en base de données. Vous allez passer ces articles à la vue et dans la vue boucler sur le tableau afin d'afficher

ces trois articles sous formes de cartes. Dans chaque carte affichez toutes les informations de l'article sauf son contenu dont vous n'afficherez que l'extrait.

- Dans la méthode de la route single vous allez récupérer l'article dont l'id est égal au paramètre de votre route, le passer à la vue et afficher toutes ses informations détaillées

Exercice 4 : Créer un service

Votre blog avance plutôt bien mais dès qu'il va grossir vous allez commencer à devoir réemployer des fonctions sur différentes pages pour réaliser plusieurs fois les mêmes actions, bref vous allez dupliquer du code ce qui n'est pas bien.

Créons un service qui vérifie qu'il n'y ait pas d'insultes dans le contenu de nos articles (oui notre blog sera tous publics).

Concrètement vous allez créer un service avec une méthode qui vérifie la présence d'insultes dans le contenu d'un article et les remplace par des dièses si elle en trouve. Vous utiliserez ce service dans le controller de la route single avant l'affichage du contenu de l'article.

Exercice 5 : Installer une extension

Quand on code on n'aime pas vraiment réinventer la roue et c'est bien souvent ce que l'on fait quand on crée un système de gestion des utilisateurs. Dans cet exercice, vous allez rajouter une extension qui vous permettra de gérer vos utilisateurs. Dans Symfony il s'agit d'un bundle (FosUser), dans Laravel on les appelle les packages (Auth).

Regardez la doc pour installer ou activer ces extensions, une fois installée vous devez avoir au minimum:

- Une page pour vous connecter
- Une page pour créer un compte
- Une page de déconnexion

Vous devez donc pouvoir :

- Créer un utilisateur
- Gérer ses rôles, du simple visiteur à l'administrateur avec tous les droits
- Protéger un route en restreignant son accès à un certain type d'utilisateur

Assurez-vous que votre route admin ne soit accessible qu'aux utilisateurs qui sont administrateurs, si quelqu'un essaie d'accéder à cette page sans être connecté et être administrateur il est automatiquement redirigé.

Créez un utilisateur, donnez lui le rôle administrateur, essayez d'accéder à l'administration. Si tout se passe bien vous devez arriver sur la page admin, ensuite déconnectez vous et essayez de retourner sur la page, vous devez être redirigé.

Voilà vous venez de mettre en place un système de gestion des utilisateurs simple qui protège votre administration !

Exercice 6 : Gérer ses formulaires

Pour le moment cependant, notre administration est un peu vide, nous pouvons nous connecter pour y accéder mais on ne peut pas y faire grand-chose. Nous allons donc rajouter un formulaire qui nous permettra d'ajouter des articles en base de données.

Pour Symfony

- Créez un formulaire sur la base de votre entité
- Spécifiez les types de champs du formulaires
- Appelez ce formulaire dans votre controller, passez le à la vue puis affichez le
- Ajoutez un ensemble de contraintes sur votre entité article (titres uniquement composé de lettres et compris entre 2 et 150 caractères, nom de l'auteur composé de lettres uniquement et de maximum 50 caractères)
- Dans votre controller gérez la soumission de votre formulaire en vous assurant qu'il s'agit bien d'une méthode POST et que votre formulaire est valide, créez l'entité correspondante puis enregistrez la.

Normalement, vous devez voir un nouvel article apparaître sur la page d'accueil !

Pour Laravel

- Créer une nouvelle méthode dans le controller pour récupérer et traiter les requêtes de type POST
- Créer votre formulaire dans la vue
- Créer la request contenant les règles de validation du formulaire
- Appeler cette request dans le controller et vérifier la validité des informations envoyées
- Si tout est OK enregistrer l'article en base de données et rediriger

Voilà vous venez de réaliser un système très basique de blogging, libre à vous maintenant de continuer à l'améliorer en rajoutant d'autres entités, d'autres contraintes ou bien de démarrer un autre projet personnel qui vous tient à coeur !