Commentaires

English

<https://symfony.com/doc/current/forms.html>

**Creating a Simple Form**

Suppose you're building a simple todo list application that will need to display "tasks". Because your users will need to edit and create tasks, you're going to need to build a form. But before you begin, first focus on the generic Task class that represents and stores the data for a single task:

This class is a "plain-old-PHP-object" because, so far, it has nothing to do with Symfony or any other library. It's a normal PHP object that directly solves a problem inside *your* application (i.e. the need to represent a task in your application). By the end of this article, you'll be able to submit data to a Task instance (via an HTML form), validate its data and persist it to the database.

<https://symfony.com/doc/current/security.html>

**The "User Provider"**

In addition to your User class, you also need a "User provider": a class that helps with a few things, like reloading the User data from the session and some optional features, like [remember me](https://symfony.com/doc/current/security/remember_me.html) and [impersonation](https://symfony.com/doc/current/security/impersonating_user.html).

Fortunately, the make:user command already configured one for you in your security.yaml file under the providers key.

If your User class is an entity, you don't need to do anything else. But if your class is not an entity, then make:user will also have generated a UserProvider class that you need to finish. Learn more about user providers here: [User Providers](https://symfony.com/doc/current/security/user_provider.html).

French

Création d’un formulaire simple

Supposons que vous voulez construire une application basique avec un affichage d’une liste de tache. Parce que vos utilisateurs vont devoir éditer et créer des taches, vous allez donc avoir besoin d’un formulaire. Mais avant de commencer, tout d’abord concentrez-vous sur la class ‘‘Tache’’ qui affiche et stocke les données d’une tache unique.

Cette classe est un ‘‘pur objet PHP’’, car pour le moment elle n’a rien à voir ni avec Synfony ni aucune autre bibliothèque. C’est un objet PHP normal qui va résoudre directement le problème dans votre application (c’est-à-dire elle représentera l’entité ‘‘Tache’’ dans votre application). Dans la fin de cet article, vous allez devoir soumettre ces données à une instance de ‘‘Tache’’ (via un formulaire en HTML), valider celle-ci et les enregistrer dans la base de données.

Le contrôleur d’utilisateurs

En supplément de votre classe ‘‘User’’, vous avez seulement besoin d’un contrôleur d’utilisateurs : une classe qui aide pour des fonctions optionnelles, telles que la mémorisation des informations de l’utilisateur dans une session, la fonction ‘‘se souvenir de moi’’ et ??.

Heureusement, la commande make :user configure tout pour vous dans votre fichier Security.yaml sous la clé du contrôleur.

Si votre classe ‘‘User’’ est une entité, vous n’avez pas besoin de faire autre chose. Mais si votre classe n’est pas une entité, alors make :user va générer un contrôleur d’utilisateurs qu’il faudra finaliser. En savoir plus sur les contrôleur d’utilisateurs.