1. **Traduction de la page en anglais**
2. **Extrait de la page en anglais**

Dans le cadre de mon application j’ai réalisé une fonctionnalité me permettant de mettre à jours le champ « LastLogin » dans la base de données afin de pouvoir disposer de l’information de la dernière connexion d’un utilisateur.

**Background information**

This might be documented somewhere already, and probably also blogged about already. But at the time when I was implementing this myself my Google-fu was failing me.

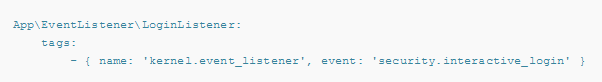
So I thought I'd document this seemingly very common feature, in case someone else is stuck and is searching for an answer.

Keep in mind this is for Symfony 4. At least that's the version I wrote and tested this for.

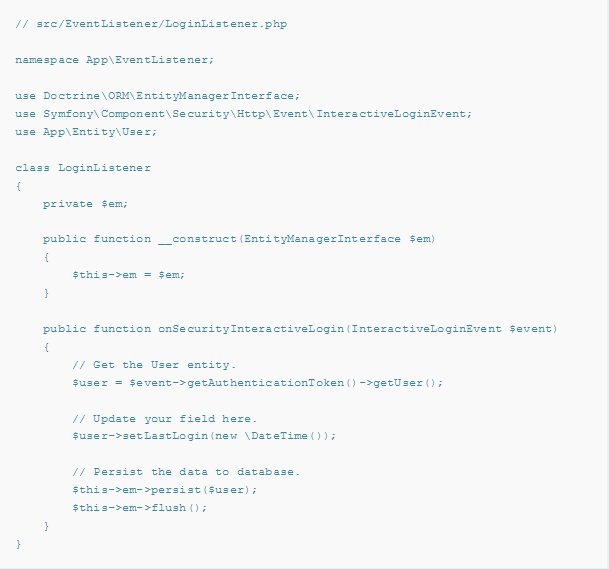
**Event Listeners**

While your application is being executed Symfony triggers a lot of event notifications, and your application can listen to these events and respond to them as well.

First you need to create an event listener that listens to  event. You do so by adding the following piece to :



Now that we have the event listener configured, let's write the code to be executed:

**In Summary**

It's that simple, and I'm amazed by the amount of thought and work has gone into Symfony to make things easier and quicker to develop.

Some might argue that having too much “magic” in a framework might not be good.

But from my experience so far with Symfony 4 I'm constantly amazed at how much more efficient I'm with this framework and how little “boilerplate” code I have to write to get things going.

Instead, I can concentrate on creating my application.

1. **Traduction de la page**

**Contexte**

Cela doit déjà être documenté quelque part, et il y a probablement aussi un sujet de blog qui parle de cela. Mais au moment où je l’ai implémenté moi-même mon Google-Fu m’a fait défaut.

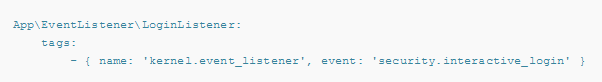
J’ai donc pensé à documenter ce qui semblait être une fonction très commune, au cas où quelqu’un d’autre serait bloqué et chercherais une réponse.

Gardez à l’esprit que cela concerne Symfony 4. Au moins que c’est sur cette version que je l’ai écrit et testé.

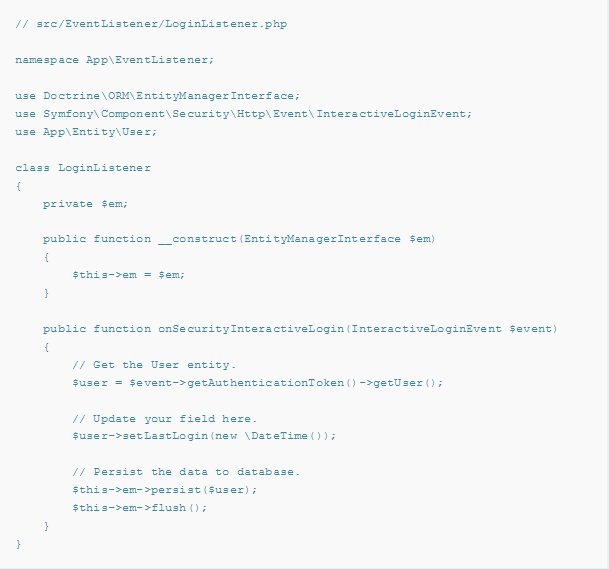
**Ecoute d’évènements**

Durant l’exécution de votre application, Symfony déclenche un certain nombre d’évènements, et votre application peut écouter ces évènements et peut très bien leur répondre.

Tout d’abords vous avez besoin de créer un « écouteur d’évènement » qui écoutera l’évènement . Faites cela en ajouter l’extrait suivant à  :



Maintenant que nous avons « l’écouteur d’évènement » configuré, allons écrire le code à exécuter :

****

**En résumé**

C’est si simple, et je suis surpris par la réflexion et le travail qui a été fait dans Symfony pour rendre les choses plus simple et plus rapide à développer.

Certains diront que d’avoir trop de « magie » dans un « cadriciel » n’est pas une bonne chose.

De mon expérience jusqu’à présent avec Symfony 4, Je suis constamment surpris de l’efficience dont je fais preuve avec ce « cadriciel » et comment les petites « boites à outil » de code que j’ai à écrire pour faire fonctionner les choses.

Pendant ce temps, je peux me concentrer sur la création de mon application.