

# Afficher un champ en lecture seul dans un formulaire.

C'est une question qui revient régulièrement sous symfony et qu'il n'est pas évident de traiter. Comment afficher une donnée en lecture seul.

En effet, en règle générale, on va essayer de torturer les pauvres widget, qui n'ont pas été conçu pour cela, pour afficher la donnée.

La première idée étant de positionner la propriété "readonly" de (x)HTML. Ceci fonctionne très bien, à la première lecture du formulaire, mais, horreur, les données disparaissent lors d'un ré affichage lié à une erreur de saisie dans un champ libre de saisie.

En fait, il n'y a pas de solution simple au niveau du formulaire ("form") de symfony, la meilleur solution étant de réécrire le formulaire "à la main".

Au niveau de la méthode "executeEdit" correspondant au formulaire il n'y a rien de particulier à dire, elle se doit d'initialiser le formulaire et de récupérer les données de l'enregistrement.

C'est dans le template que tous va ce jouer, il faut le faire "presque" à l'ancienne. Il faut construire entièrement le tableau "à la main".

L'exemple de template suivant va nous afficher deux champs, un champ *nom*, qui sera modifiable et soumis à un "validator" dans le "form" de symfony et un champ *commentaire* qui ne sera qu'afficher et non modifiable pour l'utilisateur. Les lignes en gras seront commentées ci-dessous.

```
<h1>Exemple d'affichage sans modification</h1>
```

## Afficher un champ en lecture seul dans un formulaire

Écrit par Michel Rotta

Vendredi, 03 Septembre 2010 09:29 - Mis à jour Vendredi, 03 Septembre 2010 10:01

---

```
<form id="saisie" action="<?php echo url_for('@profil_update') ?>" method="post" >
  <?php echo $form->renderHiddenFields() ?>
  <table>
    <tr>
      <?php if ($form->hasGlobalErrors()): ?>
      <td colspan="2">
        <?php echo $form->renderGlobalErrors() ?>
      </td>
      <?php endif ; ?>
    </tr>
    <tr><?php echo $form['nom']->renderRow() ?></tr>
    <tr>
      <th><?php echo $form['commentaire']->renderLabel() ?></th>
      <td class="disabled"><?php echo $form->getObject()->getCommentaire() ?></td>
    </tr>
    <tr>
      <td colspan="2">
        <input type="submit"
          value="Enregistrer"
          title="Enregistrez vos modifications" />
      </td>
    </tr>
  </table>
</form>
```

- On déclare un formulaire dans un tableau à deux colonnes, classiquement (pour symfony).
- "->renderHiddenFields()" va nous permettre de renseigner les champs cachés (obligatoire)
- Ensuite on crée une ligne sur la largeur du tableau pour afficher d'éventuelles erreurs globales. Je recommande de la mettre en place systématiquement, même si aucun validateur global n'a été défini dans le "form" du formulaire. Si nous devons arriver, plus tard, à en définir un, il sera alors affiché sans que nous ayons à nous casser la tête pour comprendre ce qui ce passe avec ce validateur.
- L'affichage d'une ligne normale ce fait simplement par "->renderRow()", méthode qui va nous générer les deux colonnes, une pour le libellé, l'autre pour les données, l'aide à la saisie (si définie) et le message d'erreur issu du validateur (si besoin).
- Pour les données simplement affichée, le "truc" est ici. "echo \$form['commentaire']->renderLabel()" qui récupère dans le "form" le label définit suivi d'un "echo \$form->getObject()->getCommentaire()" qui va nous récupérer la valeur à afficher.
- Il ne reste plus qu'à mettre classiquement un bouton de saisie ainsi qu'à fermer les balises "table" et "form" et notre formulaire est près à l'emploi.

ATTENTION ! Pour éviter tout envoi intempestif de données par un utilisateur mal intentionné

## Afficher un champ en lecture seul dans un formulaire

Écrit par Michel Rotta

Vendredi, 03 Septembre 2010 09:29 - Mis à jour Vendredi, 03 Septembre 2010 10:01

---

sur le champ *commentaire* il convient de supprimé le validateur dans l'objet "form", ainsi, une donnée supplémentaire générerait un message d'erreur sur un champ qui n'existe pas (et qui apparaîtrait dans les erreur global, ce qui en justifie l'initialisation dans le formulaire.