

« Day 4 »

La semaine dernière, nous avons manipulé le modèle par l'intermédiaire de la vue à l'aide des formulaires.

Ces formulaires d'ajouts et de modifications doivent maintenant être accessibles directement de votre page CV ! Pour cela, vous devez insérer des balises html `<a>` ou `<button>` dans votre page `index.html.twig` :

```
<a href="https://www.w3schools.com/" target="_blank">Visit W3Schools!</a>
```

https://www.w3schools.com/html/html_links.asp

NB : Penser à utiliser l'attribut « **_target** » si besoin.

Ensuite, il va falloir remplacer l'url dans l'attribut « **href** » par l'adresse de votre contrôleur que vous souhaitez appeler. Cela est possible avec la commande twig « **path** » :

```
<a href="{% path('<nom de l'action de votre contrôleur>') %}">Home</a>
```

<https://symfony.com/doc/current/templating.html#linking-to-pages>

Réaliser tous les formulaires, réaliser les boutons Create/Edit/Delete pour toutes les catégories.

Pour réaliser un « **remove** » avec doctrine, il suffit de renseigner dans une action d'un contrôleur avec un id comme fait auparavant pour l'édition :

```
$loisir = $eManager->getRepository("AppBundle:Loisir")->findOneBy(["id" => $id]);

$eManager = $this->getDoctrine()->getManager();

$eManager->remove($loisir);

$eManager->flush();
```

Essayer de supprimer un id qui existe, un id qui n'existe pas... Tenter de gérer par un message clair à l'utilisateur que l'id n'existe pas... Pour ceux qui souhaitent réaliser la notification avec une librairie adaptée : <https://ned.im/noty/#/>

Sécuriser vos pages

En suivant le tutorial suivant, sécurisez vos boutons pour que seul un admin authentifié puissent les voir et les utiliser.

<https://symfony.com/doc/3.4/security.html>

NB : On utilisera une base `in_memory` pour les utilisateurs pour les plus curieux regardez autour de `FosUserBundle` & `FosAuthBundle`.

