

ADRAR\o/ **PÔLE NUMÉRIQUE**

- > **INFRASTRUCTURES** SYSTÈMES & RÉSEAUX
- > **CYBERSÉCURITÉ** INFRASTRUCTURES & APPLICATIONS
- > **DEVOPS / SCRIPTING** & AUTOMATISATION
- > **DEVELOPPEMENT** WEB & MOBILE
- > **TRANSFORMATION NUMERIQUE DES ENTREPRISES**

www.adrar-numerique.com

Affichage, mise à jour et suppression d'une donnée Symfony

Comme en PHP natif, vous serez amenés à vouloir créer une interface pour gérer vos entités et les données de vos tables.

Pour se faire, nous allons voir comment récupérer une donnée en GET/POST de manière très simple et comment gérer son traitement.

Affichage, mise à jour et suppression d'une donnée Symfony

Imaginons que je souhaite récupérer les données de mon utilisateur au travers de cette URL
type pour pré-remplir un formulaire: `/users/user/[id]`

Nous allons devoir passer par plusieurs étapes:

- réaliser notre formulaire (`src/Form/UserFormType.php`)
- créer une route dans notre contrôleur (`src/Controller/UserController.php`)
- utiliser la méthode `save` de notre *repository* associé (`src/Repository/UserRepository.php`)

Affichage, mise à jour et suppression d'une donnée Symfony

SELECT

```
#[Route(path: '/users/user/{id}', name: 'app_user')]  
public function user(int $id, UserRepository $userRepository): Response  
{  
    $user = $userRepository->find($id);  
  
    return $this->render('users/user.html.twig', [  
        'user' => $user,  
    ]);  
}
```

UserController.php

Affichage, mise à jour et suppression d'une donnée

Symfony

SELECT

```

{% extends "base.html.twig" %}
{% block css %}
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">
{% endblock %}
{% block js %}
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.min.js"></script>
{% endblock %}
{% block body %}
<table class="table">
  <thead>
    <tr>
      <th>ID</th><th>Email</th><th>Image</th><th>Actions</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>{{ user.id }}</td>
      <td>{{ user.email }}</td>
      <td></td>
      <td>
        <a href="{{ path('app_delete_user', { 'id': user.id }) }}"><i class="bi bi-trash-fill"></i></a>
        <a href="{{ path('app_edit_user', { 'id': user.id }) }}"><i class="bi bi-pencil-square"></i></a>
      </td>
    </tr>
  </tbody>
</table>
{% endblock %}

```

users/user.html.twig

Affichage, mise à jour et suppression d'une donnée Symfony

EDIT

```
class UserFormType extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder, array $options): void
    {
        $builder
            ->add('email', EmailType::class, [
                'attr' => ['class' => 'register-input']
            ])
            ->add('plainPassword', PasswordType::class, [
                'mapped' => false,
                'attr' => ['autocomplete' => 'new-password', 'class' => 'register-input']
            ])
            ->add('nom', TextType::class, [
                'attr' => ['class' => 'register-input']
            ])
            ->add('prenom', TextType::class, [
                'attr' => ['class' => 'register-input']
            ])
            ->add('image', FileType::class, [
                'label' => 'Image de profil (tout types)',
                'mapped' => false, // Cette ligne signifie que ce champs ne sera associé à aucune propriété de l'entité
                'required' => false, // Permet de ne pas rendre ce champs obligatoire
                'attr' => ['accept' => 'image/*'] // Accepte tous les fichiers de type image
            ])
            ->add('save', SubmitType::class, [
                'label' => 'Mettre à jour'
            ])
        ;
    }

    public function configureOptions(OptionsResolver $resolver): void
    {
        $resolver->setDefaults([
            'data_class' => User::class,
        ]);
    }
}
```

UserFormType.php

Affichage, mise à jour et suppression d'une donnée Symfony

EDIT

```
#[Route(path: '/users/user/edit/{id}', name: 'app_users')] UserController.php
public function user(int $id, Request $request, UserRepository $userRepository): Response
{
    // $username = $request->get('username'); // Si non passé en paramètre
    $user = $userRepository->find($id);

    $form = $this->createForm(UserFormType::class, $user);

    $form->handleRequest($request);
    if ($form->isSubmitted() && $form->isValid()) {
        $user = $form->getData();
        $userRepository->save($user, true); // Le true permet de m à j la donnée en BDD également

        return $this->redirectToRoute('task_success');
    }

    return $this->render('users/edit.html.twig', [
        'form' => $form,
    ]);
}
```


Affichage, mise à jour et suppression d'une donnée Symfony

EDIT

```
{% extends "base.html.twig" %}                                     users/user.html.twig

{% block css %}
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">
{% endblock %}

{% block js %}
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.min.js"></script>
{% endblock %}

{% block body %}
    {{ form(form) }}
{% endblock %}
```


Affichage, mise à jour et suppression d'une donnée Symfony

DELETE

```
#[Route('/user/delete/{id}', name: 'user_delete')]
public function deleteUser(Request $request, User $user): void
{
    $entityManager = $this->getDoctrine()->getManager();
    $userRepository = $entityManager->getRepository(User::class);
    $userRepository->remove($user);
    $entityManager->flush();

    // Rediriger l'utilisateur vers une page appropriée (par exemple, la liste des utilisateurs)
    return $this->redirectToRoute('user_list');
}
```

UserController.php