

# Les options héritées de FormType

## Fonctionnement des types du composant `Form`

`Symfony` propose environ 40 types pour les formulaires, avec des dizaines d'options. Au lieu de tous les voir en une fois, nous allons voir les options principales puis voir les autres types et options au fur et à mesure.

Nous allons étudier les options utilisables avec un `TextType` (un champ textuel) qui est le type de champ le plus utilisé.

Le système de types des formulaires `Symfony` est basé sur l'héritage. Par exemple, **le type `TextType` hérite de la classe parente `FormType`, qui hérite elle-même de la classe `BaseType`.**

De cette manière, une grande partie des options des types de formulaire sont appelées **des options héritées** (`inherited options`) et sont disponibles sur l'ensemble des types.

## Les options héritées de `FormType`

L'option `attr` permet d'ajouter n'importe quel attribut `HTML` valide sur un élément de formulaire.

Par exemple :

```
<?php

$this->createFormBuilder()
    ->add('content', TextType::class, [
        'attr' => ['class' => 'maClasse'],
    ]);
```

L'option `data` permet de spécifier la valeur d'un champ lors de l'initialisation du formulaire.

L'option `disabled` permet de désactiver un champ.

L'option `empty_data` permet de spécifier la valeur d'un champ qui sera envoyé au serveur si aucune valeur n'est spécifiée par l'utilisateur.

L'option `error_bubbling` permet de remonter l'erreur du champ au champ parent ou au formulaire parent. Autrement dit, cela permet de "remonter" l'erreur d'un cran.

L'option **error\_mapping** permet de définir à quel champ rattacher une erreur. C'est particulièrement utile lorsqu'une erreur n'est rattachée par défaut à aucun champ. Par exemple, dans le cas d'une validation d'email.

L'option **help** permet d'afficher un message d'aide au-dessous du champ. L'option **help\_attr** permettant d'ajouter des attributs sur l'élément d'aide.

L'option **label** permet de définir le label pour le champ. L'option **label\_attr** permettant d'ajouter des attributs sur l'élément **label**.

L'option **required** permet de rendre le champ obligatoire ou optionnel. Par défaut, les champs sont requis. Il faut donc les rendre optionnels si besoin :

```
<?php
```

```
$this->createFormBuilder()  
    ->add('content', TextType::class, [  
        'attr' => ['class' => 'maClasse'],  
        'required' => false  
    ]);
```

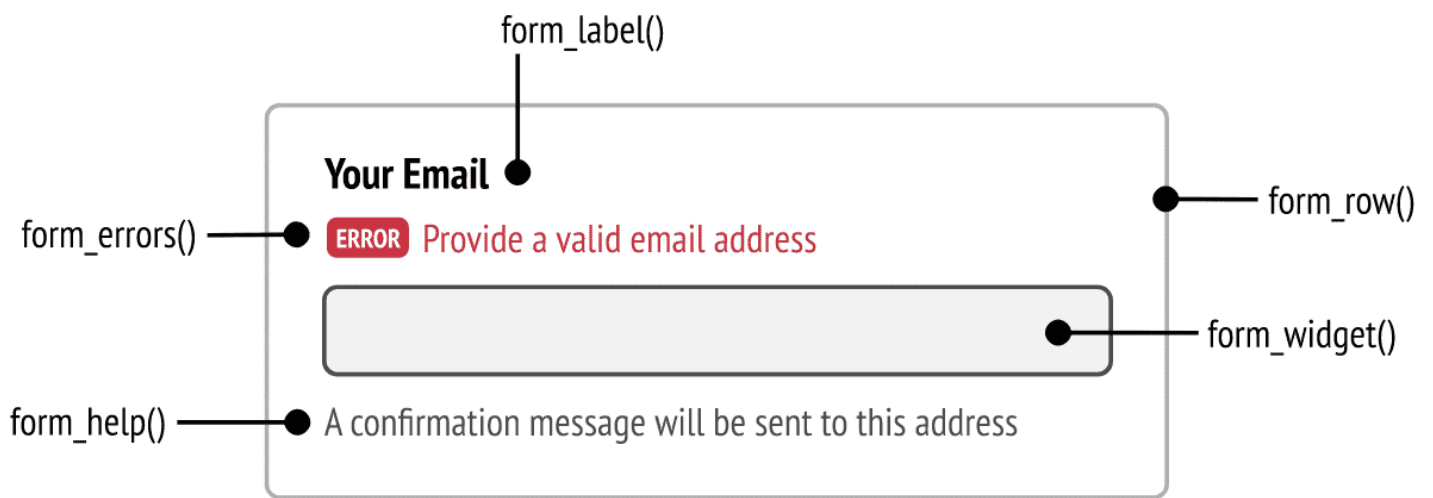
L'option **row\_attr** permet de définir des attributs sur l'élément qui contient tous les éléments du champ. Cet élément est appelé une rangée (ou **row**) pour les formulaires **Symfony** (voir plus bas).

L'option **trim** permet d'enlever les espaces lors de l'envoi. Par défaut, l'option est activée.

## Structure d'un élément de formulaire

Nous reprendrons en détail la structure d'un élément de formulaire lorsque nous étudierons les fonctions d'affichage **Twig** pour les formulaires **Symfony**.

Voici déjà un bref aperçu de la structure des éléments pour pouvoir comprendre les options que nous avons vues jusqu'à présent :



Remarquez l'élément rangée ( [row](#) ) qui contient le label, les erreurs, le champ et le message d'aide.