

# ChoiceType, DateType et CountryType

## L'élément `ChoiceType`

L'élément `ChoiceType` permet de créer un champ récupérant le ou les choix d'un utilisateur.

Il se matérialise, suivant la configuration, en cases à cocher, liste d'options ou en boutons radio.

Pour créer les options disponibles pour l'utilisateur, il suffit de passer une option `choices` qui prend un tableau associatif avec en clés les valeurs affichées et en valeurs, les valeurs retournées.

## Par défaut : liste déroulante

Si vous ne précisez aucune configuration, `Symfony` créera une liste déroulante :

```
<?php
```

```
namespace App\Controller;
```

```
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\Form\Extension\Core\Type\CheckboxType;
use Symfony\Component\Form\Extension\Core\Type\ChoiceType;
use Symfony\Component\Form\Extension\Core\Type\SubmitType;
use Symfony\Component\Form\Extension\Core\Type\TextType;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\Routing\Annotation\Route;
```

```
class DefaultController extends AbstractController
{
    function __construct()
    {
    }

    #[Route('/', name: 'index')]
    public function index(Request $request)
    {
        $todo = new Todo();
        $form = $this->createFormBuilder($todo)
```

```

->add('content', TextType::class, [
    'label' => 'Un super label',
    'attr' => [
        'placeholder' => 'Contenu de la todo'
    ],
    'help' => 'Indiquez ce que vous avez à faire'
])
->add('done', CheckboxType::class, ['required' => false])
->add('type', ChoiceType::class, [
    'choices' => [
        'Loisir' => 'loisir',
        'Travail' => 'travail',
        'Repos' => 'repos'
    ],
])
->add('submit', SubmitType::class)
->getForm();

$form->handleRequest($request);

if ($form->isSubmitted() && $form->isValid()) {
    dd($todo);
}

return $this->render('page1.html.twig', [
    'myform' => $form->createView()
]);
}
}

class Todo
{
    function __construct(
        public string $content = '',
        public ?string $type = null,
        public bool $done = false
    ) {
    }
}

```

## Liste déroulante à choix multiples

Pour obtenir une liste déroulante avec plusieurs options sélectionnables, il suffit d'ajouter l'option `multiple` et de lui passer `true` :

```

<?php
// ...

$todo = new Todo('');
$form = $this->createFormBuilder($todo)
    ->add('content', TextType::class, [
        'label' => 'Un super label',
        'attr' => [
            'placeholder' => 'Contenu de la todo'
        ],
        'help' => 'Indiquez ce que vous avez à faire'
    ])
    ->add('done', CheckboxType::class, ['required' => false])
    ->add('type', ChoiceType::class, [
        'multiple' => true,
        'choices' => [
            'Loisir' => 'loisir',
            'Travail' => 'travail',
            'Repos' => 'repos'
        ],
    ])
    ->add('submit', SubmitType::class)
    ->getForm();

$form->handleRequest($request);

if ($form->isSubmitted() && $form->isValid()) {
    dd($todo);
}

return $this->render('page1.html.twig', [
    'myform' => $form->createView()
]);
}
}

class Todo
{
    function __construct(
        public string $content,
        public array $type = [],
        public bool $done = false
    ) {

```

```
}  
}
```

## Bouton radio

Pour obtenir des boutons radio, il suffit de mettre l'option `expanded` à `true` :

```
<?php  
// ...  
$todo = new Todo();  
$form = $this->createFormBuilder($todo)  
    ->add('content', TextType::class, [  
        'label' => 'Un super label',  
        'attr' => [  
            'placeholder' => 'Contenu de la todo'  
        ],  
        'help' => 'Indiquez ce que vous avez à faire'  
    ])  
    ->add('done', CheckboxType::class, ['required' => false])  
    ->add('type', ChoiceType::class, [  
        'expanded' => true,  
        'choices' => [  
            'Loisir' => 'loisir',  
            'Travail' => 'travail',  
            'Repos' => 'repos'  
        ],  
    ])  
    ->add('submit', SubmitType::class)  
    ->getForm();  
  
$form->handleRequest($request);  
  
if ($form->isSubmitted() && $form->isValid()) {  
    dd($todo);  
}  
  
return $this->render('page1.html.twig', [  
    'myform' => $form->createView()  
]);  
}  
}  
  
class Todo  
{
```

```
function __construct(
    public string $content = '',
    public ?string $type = null,
    public bool $done = false
) {
}
}
```

## Case à cocher (checkbox)

Pour obtenir des cases à cocher, il suffit de mettre l'option `expanded` à `true` et l'option `multiple` à `true` :

```
<?php
// ...

$todo = new Todo();
$form = $this->createFormBuilder($todo)
    ->add('content', TextType::class, [
        'label' => 'Un super label',
        'attr' => [
            'placeholder' => 'Contenu de la todo'
        ],
        'help' => 'Indiquez ce que vous avez à faire'
    ])
    ->add('done', CheckboxType::class, ['required' => false])
    ->add('type', ChoiceType::class, [
        'expanded' => true,
        'multiple' => true,
        'choices' => [
            'Loisir' => 'loisir',
            'Travail' => 'travail',
            'Repos' => 'repos'
        ],
    ])
    ->add('submit', SubmitType::class)
    ->getForm();

$form->handleRequest($request);

if ($form->isSubmitted() && $form->isValid()) {
    dd($todo);
}

return $this->render('page1.html.twig', [
```

```

        'myform' => $form->createView()
    ]);
}
}

```

```

class Todo
{
    function __construct(
        public string $content = '',
        public array $type = [],
        public bool $done = false
    ) {
    }
}

```

## Exemple avancé pour grouper les choix

Il est possible d'utiliser l'option `group_by` qui prend une fonction de rappel permettant de trier et de regrouper les choix suivant les conditions que vous souhaitez.

Par exemple :

```

<?php
// ...

$todo = new Todo();
$form = $this->createFormBuilder($todo)
    ->add('content', TextType::class, [
        'label' => 'Un super label',
        'attr' => [
            'placeholder' => 'Contenu de la todo'
        ],
        'help' => 'Indiquez ce que vous avez à faire'
    ])
    ->add('done', CheckboxType::class, ['required' => false])
    ->add('dueDate', ChoiceType::class, [
        'choices' => [
            'Demain' => new \DateTime('+1 day'),
            'Dans une semaine' => new \DateTime('+1 week'),
            'Dans deux semaines' => new \DateTime('+2 weeks'),
            'Dans un mois' => new \DateTime('+1 month'),
            'Dans 6 mois' => new \DateTime('+6 months'),
        ],
        'group_by' => function ($choix, $cle, $valeur) {
            if ($choix <= new \DateTime('+3 days')) {

```

```

        return 'Bientôt';
    } elseif ($choix <= new \DateTime('+2 weeks')) {
        return 'Dans moins de deux semaines';
    } else {
        return 'Plus tard';
    }
},
])
->add('submit', SubmitType::class)
->getForm();

$form->handleRequest($request);

if ($form->isSubmitted() && $form->isValid()) {
    dd($todo);
}

return $this->render('page1.html.twig', [
    'myform' => $form->createView()
]);
}
}

class Todo
{
    function __construct(
        public string $content = '',
        public ?DateTime $dueDate = null,
        public bool $done = false
    ) {
    }
}

```

Les possibilités sont vraiment énormes, nous verrons donc d'autres exemples plus tard dans le projet.

## L'élément `DateType`

L'élément `DateType` permet de faire choisir ou modifier une date par les utilisateurs. :

```
<?php
```

```
namespace App\Controller;
```

```

use DateTime;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\Form\Extension\Core\Type\CheckboxType;
use Symfony\Component\Form\Extension\Core\Type\DateType;
use Symfony\Component\Form\Extension\Core\Type\SubmitType;
use Symfony\Component\Form\Extension\Core\Type\TextType;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\Routing\Annotation\Route;

class DefaultController extends AbstractController
{
    function __construct()
    {
    }

    #[Route('/', name: 'index')]
    public function index(Request $request)
    {
        $todo = new Todo();
        $form = $this->createFormBuilder($todo)
            ->add('content', TextType::class, [
                'label' => 'Un super label',
                'attr' => [
                    'placeholder' => 'Contenu de la todo'
                ],
                'help' => 'Indiquez ce que vous avez à faire'
            ])
            ->add('done', CheckboxType::class, ['required' => false])
            ->add('dueDate', DateType::class, ['widget' => 'single_text',])
            ->add('submit', SubmitType::class)
            ->getForm();

        $form->handleRequest($request);

        if ($form->isSubmitted() && $form->isValid()) {
            dd($todo);
        }

        return $this->render('page1.html.twig', [
            'myform' => $form->createView()
        ]);
    }
}

class Todo

```



```
{
    function __construct(
        public string $content = '',
        public ?DateTime $dueDate = null,
        public bool $done = false
    ) {
    }
}
```

L'option `'widget'` permet de choisir comment afficher le champ de choix de la date à l'utilisateur. Le plus courant est `'single_text'` qui affiche le `date picker` du navigateur.

## Limiter les dates affichées

Vous pouvez limiter les dates qui peuvent être choisies par l'utilisateur en utilisant les options `days`, `months` et `years` :

```
<?php
```

```
namespace App\Controller;
```

```
use DateTime;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\Form\Extension\Core\Type\CheckboxType;
use Symfony\Component\Form\Extension\Core\Type\DateType;
use Symfony\Component\Form\Extension\Core\Type\SubmitType;
use Symfony\Component\Form\Extension\Core\Type\TextType;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\Routing\Annotation\Route;
```

```
class DefaultController extends AbstractController
```

```
{
    function __construct()
    {
    }

    #[Route('/', name: 'index')]
    public function index(Request $request)
    {
        $todo = new Todo();
        $form = $this->createFormBuilder($todo)
            ->add('content', TextType::class, [
                'label' => 'Un super label',
            ])
            ->add('dueDate', DateType::class, [
                'label' => 'Date d\'expiration',
            ])
            ->add('done', CheckboxType::class, [
                'label' => 'Terminé',
            ])
            ->getForm();
    }
}
```

```

        'attr' => [
            'placeholder' => 'Contenu de la todo'
        ],
        'help' => 'Indiquez ce que vous avez à faire'
    ])
    ->add('done', CheckboxType::class, ['required' => false])
    ->add('dueDate', DateType::class, [
        'years' => range(date('Y'), date('Y') + 2),
        'months' => range(date('m'), 12),
        'days' => range(date('d'), 31),
    ])
    ->add('submit', SubmitType::class)
    ->getForm();

    $form->handleRequest($request);

    if ($form->isSubmitted() && $form->isValid()) {
        dd($todo);
    }

    return $this->render('page1.html.twig', [
        'myform' => $form->createView()
    ]);
}
}

class Todo
{
    function __construct(
        public string $content = '',
        public ?DateTime $dueDate = null,
        public bool $done = false
    ) {
    }
}

```

Pour modifier la locale par défaut de votre application, et afficher ici les dates en français, modifiez `config/packages/framework.yaml` :

```

#....
framework:
    #....

```

```
default_locale: fr
#....
```

## L'élément `CountryType`

L'élément `CountryType` permet de créer une liste déroulante pour choisir un pays.

Les pays sont affichés dans la langue de l'utilisateur (ou celle configurée par défaut).

La valeur renvoyée est le code du pays en deux lettres (par exemple `FR` ou `US` ou `ES`).

```
<?php
```

```
namespace App\Controller;

use DateTime;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\Form\Extension\Core\Type\CheckboxType;
use Symfony\Component\Form\Extension\Core\Type\CountryType;
use Symfony\Component\Form\Extension\Core\Type\SubmitType;
use Symfony\Component\Form\Extension\Core\Type\TextType;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\Routing\Annotation\Route;

class DefaultController extends AbstractController
{
    function __construct()
    {
    }

    #[Route('/', name: 'index')]
    public function index(Request $request)
    {
        $todo = new Todo();
        $form = $this->createFormBuilder($todo)
            ->add('content', TextType::class, [
                'label' => 'Un super label',
                'attr' => [
                    'placeholder' => 'Contenu de la todo'
                ],
                'help' => 'Indiquez ce que vous avez à faire'
            ])
            ->add('done', CheckboxType::class, ['required' => false])
            ->add('country', CountryType::class)
```

```

        ->add('submit', SubmitType::class)
        ->getForm();

$form->handleRequest($request);

if ($form->isSubmitted() && $form->isValid()) {
    dd($todo);
}

return $this->render('page1.html.twig', [
    'myform' => $form->createView()
]);
}
}

class Todo
{
    function __construct(
        public string $content = '',
        public ?string $country = null,
        public bool $done = false
    ) {
    }
}

```