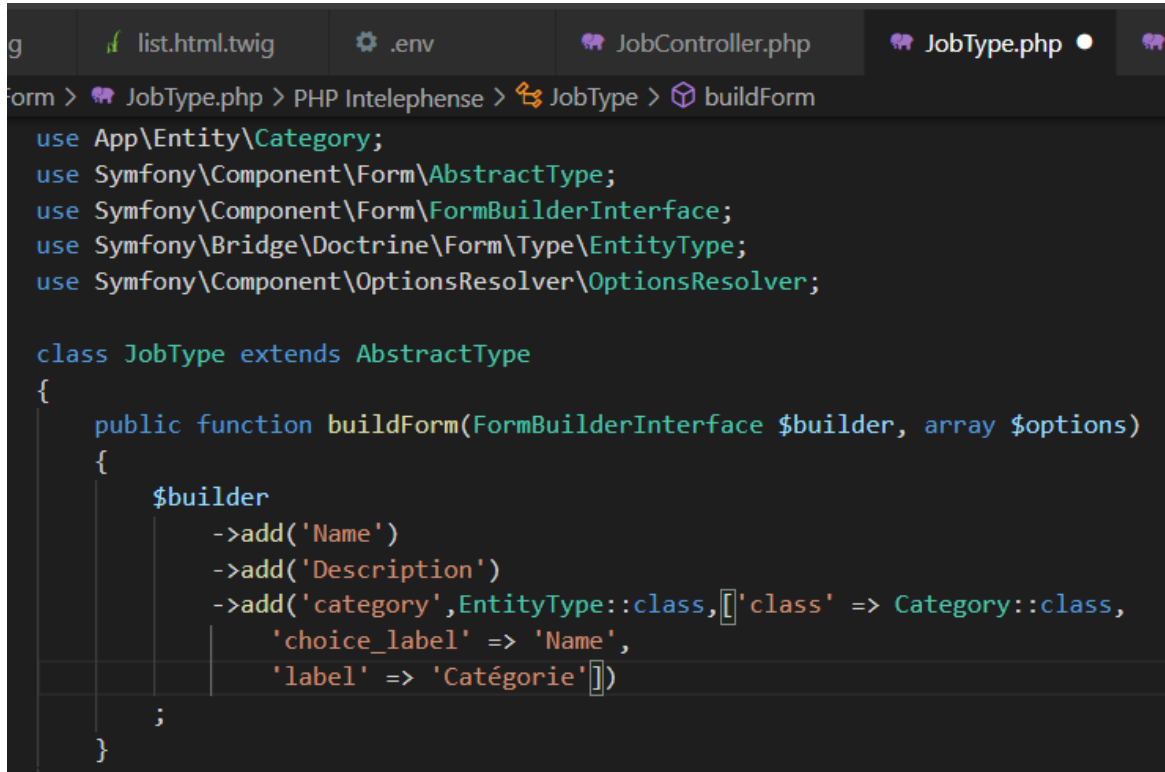


## 1. Modifier le formulaire de création d'un job

Pour ajouter une liste déroulante afin d'affecter une catégorie à un job, modifier la fonction `buildForm` de la classe `Form/JobType.php` comme suit :



```
g  list.html.twig  .env  JobController.php  JobType.php  ●  
Form > JobType.php > PHP Intelephense > JobType > buildForm  
  
use App\Entity\Category;  
use Symfony\Component\Form\AbstractType;  
use Symfony\Component\Form\FormBuilderInterface;  
use Symfony\Bridge\Doctrine\Form\Type\EntityType;  
use Symfony\Component\OptionsResolver\OptionsResolver;  
  
class JobType extends AbstractType  
{  
    public function buildForm(FormBuilderInterface $builder, array $options)  
    {  
        $builder  
            ->add('Name')  
            ->add('Description')  
            ->add('category', EntityType::class, ['class' => Category::class,  
                'choice_label' => 'Name',  
                'label' => 'Catégorie'])  
    }  
}
```

N'oublier pas `use` dans la classe « `JobType` »

```
use Symfony\Bridge\Doctrine\Form\Type\EntityType;  
use App\Entity\Category;
```

## Affichage catégorie dans la liste :

### List.html.twig :

```
{% block body %}

<table class="table">
  <thead>
    <tr>
      <th scope="col">Name</th>
      <th scope="col">Description</th>
      <th scope="col">categorie</th>
    </tr>
  </thead>
  <tbody>
    {% for a in list_jobs %}
      <tr>
        <th scope="row">{{a.Name }}</th>
        <th scope="row">{{a.Description }}</th>
        <th scope="row">{{a.categorie.Name}}</th>
      </tr>
    {% endfor %}
  </tbody>
</table>

{% endblock %}
```

### Modifier Job :

```
/**
 * @Route("/edit/{id}", name="edit_job")
 */
public function edit(Request $request, $id) {
    //nouvelle instance
    $job = new Job();
    //On fait une recherche de l'entité qu'on va modifier
    $job = $this->getDoctrine()->getRepository(Job::class)->find($id);
    //création du formulaire
    $form = $this->createForm(JobType::class,$job);
    $form->handleRequest($request);
    if($form->isSubmitted() && $form->isValid()) {
        $entityManager = $this->getDoctrine()->getManager();
        //synchronise l'état des instances de l'entité avec celui de la table dans la BDR.
        $entityManager->flush();
        return $this->redirectToRoute('list'); }
    return $this->render('job/edit.html.twig', ['form' => $form->createView()]); }
```

### Ajouter un message flash

```
// affichage d'un message flash

$this->addFlash('success', 'Entity Updated! ');
```

### Base.html.twig

```

    </div>
</nav>
    {% block body %}{% endblock %}
    {% for message in app.flashes('success') %}
        <div class="alert alert-success">
            {{ message }}
        </div>
    {% endfor %}
</body>
</html>

```

### edit.html.twig

```

{% extends 'base.html.twig' %}
{% block title %}Modifier un Article{% endblock %}
{% block body %}
    {{ form_start(form) }}
    {{ form_widget(form) }}
    <button type="submit" class="btn btn-success">Modifier</button>
    {{ form_end(form) }}
{% endblock %}

```

### List.html.twig

```

{% extends 'base.html.twig' %}

{% block title %}Hello JobController!{% endblock %}

{% block body %}

<table class="table">
    <thead>
        <tr>
            <th scope="col">Name</th>
            <th scope="col">Description</th>
            <th scope="col">categorie</th>
            <th scope="col">Edit</th>
            <th scope="col">Supprimer</th>
        </tr>
    </thead>
    <tbody>
        {% for a3
in list_jobs %}
        <tr>
            <th scope="row">{{a.Name }}</th>
            <th scope="row">{{a.Description }}</th>
            <th scope="row">{{a.categorie.Name}}</th>
            <th class="green"><a class="btn btn-
info" href="{{path("edit_job",{ 'id':a.id}) }}">Modifier</a></th>

```

```
<th class="green"><a class = "btn btn-  
danger" href="{{path('Supprimer_job',{'id':a.id})}}">Supprimer</a></td></tr>  
    {% endfor %}  
</tbody>  
</table>  
{% endblock %}
```

## Supprimer Job

```
/**  
 * @Route("/supprimer/{id}", name="Supprimer_job")  
 */  
public function Supprimer(Request $request, $id) {  
    //nouvelle instance  
    $job = new Job();  
    //On fait une Recherche de l'entité qu'on va supprimer  
    $job = $this->getDoctrine()->getRepository(Job::class)->find($id);  
    $entityManager = $this->getDoctrine()->getManager();  
    //Permet de supprimer l'instance d'une entité en argument  
    $entityManager->remove($job);  
    //synchronise l'état des instances de l'entité avec celui de la table  
    dans la BDR.  
    $entityManager->flush();  
    return $this->redirectToRoute('list');  
}
```

## Créer une nouvelle entité « Stage » dont les propriétés

- Name
- Description
- Durée

Taper la commande suivante **php bin/console make :crud**