

| | |
|--|----------|
| CREER L'ENTITY PERSON | 2 |
| COMMANDE | 2 |
| CHOISIR NEW FEILD DE TYPE RELATION MANYToMANY | 2 |
| CREER LE CONTROLLER DE LA CLASS PERSON ET LES VUES | 2 |
| TRAITEMENT DE L'ERREUR CATCHABLE FATAL ERROR | 2 |
| <i>La cause</i> | 2 |
| <i>Solution</i> | 2 |
| PERSONNALISER LE CONTROLEUR | 2 |
| MODIFIER LES ROUTES | 3 |
| CREER LES VUES DE GESTION DE L'INDEX PERSON | 4 |
| AJOUTER LE CHAMP PERSON DANS LA VUE D'EDITION D'UNE RESSOURCE | 4 |
| AJOUTER DANS FORM/RESOURCEType.PHP, : | 4 |
| AJOUTER DANS TEMPLATES/ADMIN/RESOURCE/_FORM.HTML.TWIG, | 5 |
| DECLARER L'ENTITE RESOURCE COMME ETANT PROPRIETAIRE DE LA RELATION | 5 |
| <i>Voir la documentation pour la version stable</i> | 5 |
| <i>Indiquer dans Entity/Resource.php « inversedBy="resources" »</i> | 5 |
| <i>Indiquer dans Entity/Person mappedBy="persons" (l'entité Person comme n'etant pas propriétaire)</i> | 5 |
| <i>Erreur, il demande que la table de liaison doit s'appeler 'portail.resource_person'</i> | 5 |
| <i>Faire une retour de migration : Rollback</i> | 5 |
| AJOUTER L'OPTION RECHERCHE PAR PERSON DANS LE FORMULAIRE DE RECHERCHE | 6 |
| INTEGRER LA LIBRAIRIE JAVASCRIPT SELECT2 | 6 |
| <i>Chercher select2</i> | 6 |
| <i>Ajouter dans tempalates/Admin/resource/edit.html.twig et dans</i> <i>tempalates/Admin/resource/new.html.twig</i> | 6 |
| AJOUTER L'OPTION PERSON DANS LE FORMULAIRE DE RECHERCHE ET DANS LA VUE DE CONSULTATION D'UNE RESSOURCE | 6 |
| <i>Ajouter dans src/form/ResourceSearchType :</i> | 6 |
| AJOUTER DANS L'ENTITE RESOURCE | 6 |
| MODIFIER RESOURCERepository.PHP POUR QUE LA RECHERCHE PAR TAG PERSONS SOIT POSSIBLE. | 7 |
| <i>Ajouter dans src/Repository/ResourceRepository.php</i> | 7 |
| AFFICHER L'INDEX PERSONS DANS LE FORMULAIRE DE LA NOTICE DETAILLEE | 7 |

https://www.youtube.com/watch?v=gZ4R_Ooq4LA

Créer l'entity Person

Commande

```
bin/console make:entity.
```

Choisir new feild de type relation ManyToMany

Créer le Controller de la class Person et les vues

```
bin/console make:crud Person
```

```
created: src/Controller/PersonController.php
created: src/Form/PersonType.php
created: templates/person/_delete_form.html.twig
created: templates/person/_form.html.twig
created: templates/person/edit.html.twig
created: templates/person/index.html.twig
created: templates/person/new.html.twig
created: templates/person/show.html.twig
Next: Check your new CRUD by going to /person/
```

Traitement de l'erreur Catchable Fatal Error

Catchable Fatal Error: Object of class App\Entity\Resource could not be converted to string

La cause

- Resources est défini dans le PersonType :

```
public function buildForm(FormBuilderInterface $builder, array $options)
{
    $builder
        ->add('name')
        ->add('resources')
    ;
}
```

Solution

- Commenter la ligne ->add('resources')

Personnaliser le Controleur

- déplacer le fichier PersonController dans Admin/PersonController
- renommer le en AdminPersonController
- modifier le namespace

```
namespace App\Controller => namespace App\Controller\Admin
```
- modifier les chemins :

```
person/index.html.twig => admin/person/index.html.twig
person/new.html.twig => admin/person/new.html.twig
person/show.html.twig => admin/person/show.html.twig
person/edit.html.twig => admin/person/edit.html.twig
```

Modifier les routes

```
/**
 * @Route("/person")
 */
=>
/**
 * @Route("/admin/person")
 */
/**
 * @Route("/", name="person_index", methods={"GET"})
 */
=>
/**
 * @Route("/", name="person.index", methods={"GET"})
 */
/**
 * @Route("/new", name="person_new", methods={"GET", "POST"})
 */
=>
/**
 * @Route("/new", name="person.new", methods={"GET", "POST"})
 */
/**
 * @Route("/{id}", name="person_show", methods={"GET"})
 */
=>
/**
 * @Route("/{id}", name="person.show", methods={"GET"})
 */
/**
 * @Route("/{id}/edit", name="person_edit", methods={"GET", "POST"})
 */
=>
/**
 * @Route("/{id}/edit", name="person.edit", methods={"GET", "POST"})
 */
/**
 * @Route("/{id}", name="person_delete", methods={"DELETE"})
 */
=>
/**
 * @Route("/{id}", name="person.delete", methods={"DELETE"})
 */
```

Créer les vues de gestion de l'index Person

- Créer une répertoire : templates/Admin/person
- Dupliquer les vues templates/Admin/resource/* dans templates/Admin/person
- Modifier edit.html.twig

```
    {{ include ('admin/resource/_form.html.twig', { button: 'Valider'}) }}
=>
    {{ include ('admin/person/_form.html.twig', { button: 'Valider'}) }}
```

index.html.twig

```
    {{ path('admin.resource.edit', {id: person.id}) }}
=>
    {{ path('admin.person.edit', {id: person.id}) }}
```

{% for resource in resources %}

=>

{% for person in persons %}

{{ resource.title }}

=>

{{ person.name }}

etc
- Modifier new.html.twig

```
    {{ include ('admin/resource/_form.html.twig', {button: 'Ajouter'}) }}
=>
    {{ include ('admin/person/_form.html.twig', {button: 'Ajouter'}) }}
```
- Modifier _form.twig.html

```
// uniquement form_start et form_end et form_rest pour afficher le bouton à la fin
    {{ form_start(form) }}
    {{ form_rest(form) }}
        <button class="btn btn-primary">{{ button|default('Enregistrer') }}</button>
    }
    {{ form_end(form) }}
```

Ajouter le champ Person dans la vue d'édition d'une resource

Ajouter dans Form/ResourceType.php, :

```
->add('persons', EntityType::class, [
    'class' => Person::class,
    'required' => false,
    'choice_label' => 'name',
    'multiple' => true
])
```

Ajouter dans templates/Admin/resource/_form.html.twig,
<div class="col-md-4">{{ form_row(form.persons) }}</div>

Déclarer l'entité Resource comme étant propriétaire de la relation

Voir la documentation pour la version stable

- Aller à <https://www.doctrine-project.org/projects/orm.html>
<https://www.doctrine-project.org/projects/doctrine-orm/en/2.6/index.html>
- Aller à la page association mapping
<https://www.doctrine-project.org/projects/doctrine-orm/en/2.6/reference/association-mapping.html#association-mapping>
- Aller à la page many-to-many bidirectionnelle
<https://www.doctrine-project.org/projects/doctrine-orm/en/2.6/reference/association-mapping.html#many-to-many-bidirectional>
- Voir la partie "Owning and Inverse Side on a ManyToMany Association" qui explique que l'un des parties de la relation va être le propriétaire

Indiquer dans Entity/Resource.php « inversedBy="resources" »

```
/**  
 * @ORM\ManyToMany(targetEntity="App\Entity\Person", inversedBy="resources")  
 */  
private $persons;
```

Indiquer dans Entity/Person mappedBy="persons" (l'entité Person comme n'étant pas propriétaire)

```
/**  
 * @ORM\ManyToMany(targetEntity="App\Entity\Resource", mappedBy="persons")  
 */  
private $resources;
```

Erreur, il demande que la table de liaison doit s'appeler 'portail.resource_person'

An exception occurred while executing 'SELECT t0.id AS id_1, t0.name AS name_2 FROM person t0 INNER JOIN resource_person ON t0.id = resource_person.person_id WHERE resource_person.resource_id = ?' with params [3267]:

SQLSTATE[42S02]: Base table or view not found: 1146 Table 'portail.resource_person' doesn't exist

Faire une retour de migration : Rollback

- Savoir où en est la migration
`bin/console doctrine:migration:status`
- Récupère le timestamp de la version précédente :
`>> Previous Version: 2019-03-27 18:55:20 (20190327185520)`
- Exécuter un Rollback
`bin/console doctrine:migration:migrate 20190327184017`
- Execution d'une nouvelle migration
`bin/console make:migration`
`bin/console doctrine:migrations:migrate`

La table change de nom : resource_person

Ajouter l'option recherche par Person dans le formulaire de recherche

Dans templates/resource/index.html.twig, ajouter :

```
<div class="form-row align-items-end">
  <div class="col">
    {{ form_row(form.persons) }}
  </div>
</div>
```

Intégrer la librairie javascript select2

- Aller à cdn.js
<https://cdnjs.com/>

Chercher select2

<https://cdnjs.com/libraries/select2>

Ajouter dans templates/Admin/resource/edit.html.twig et dans

templates/Admin/resource/new.html.twig

```
{% block stylesheets %}
```

```
<link rel="stylesheet"
```

```
href="https://cdnjs.cloudflare.com/ajax/libs/select2/4.0.5/css/select2.min.css">
```

```
{% endblock %}
```

```
{% block javascripts %}
```

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/select2/4.0.5/js/select2.min.js"></script>
```

```
<script>
```

```
    $('select').select2();
```

```
</script>
```

```
{% endblock %}
```

Ajouter l'option Person dans le formulaire de recherche et dans la vue de consultation d'une ressource

Ajouter dans src/form/ResourceSearchType :

```
->add('persons', EntityType::class, [
    'required' => false,
    'label' => false,
    'class' => Person::class,
    'choice_label' => 'name',
    'multiple' => true
])
```

Ajouter dans l'entité Resource

```
/**
```

```
 * @var ArrayCollection
```

```
 */
```

```
private $persons;
```

```
function __construct()
```

```
{
```

```
    $this->persons = new ArrayCollection();
```

```
}
```

```

/**
 * @return \Doctrine\Common\Collections\ArrayCollection
 */
public function getPersons()
{
    return $this->persons;
}

/**
 * @param \Doctrine\Common\Collections\ArrayCollection $persons
 */
public function setPersons($persons)
{
    $this->persons = $persons;
}

```

Modifier `ResourceRepository.php` pour que la recherche par tag Persons soit possible.

Voir la doc sur DQL (*Doctrine Query Language*)

<https://www.doctrine-project.org/projects/doctrine-orm/en/2.6/reference/dql-doctrine-query-language.html#doctrine-query-language>

Pour voir les exemples de conditions. Notamment la fonctionnalité *Members of "Get all users who are members of \$group."*

Ajouter dans `src/Repository/ResourceRepository.php`

```

if ($search->getPersons()->count() > 0) {
    foreach ($search->getPersons() as $k => $person) {
        $query = $query
            ->andWhere(":person$k MEMBER OF r.persons")
            ->setParameter("person$k", $person)
        ;
    }
}

```

Afficher l'index Persons dans le formulaire de la notice détaillée

```

{% if resource.persons|length > 0 %}

    {% for person in resource.persons %}
    {% set counter = ( counter | default(0) ) + 1 %}
    <tr>
        <td>Fiche personne {{ counter}}</td>
        <td>{{ person.name }}</td>
    </tr>
    {% endfor %}
{% endif %}

```