Discipline : MDW

Matière : Atelier Framework Côté Serveur **Unité d'Ensg.** : Programmation objet avancée

Niveau : 3^{ème} Année (Licence appliquée : LMD)

Workshop 4: Custom Repository method and Report with Query Builder and Doctrine Query Language

1- Create a new controller named : Report
E:\TP_JEE\Etudiant\PHP\test>php bin/console make:controller Report

Using simple query

2- Add the following query in BorrowingRepository to find the most borrowing books:

```
public function findMostPopularBooks()
{
    $sql = "SELECT b2.title, COUNT(*) AS howMany
        FROM borrowing b1
        INNER JOIN book b2 ON b2.id=b1.book_id
        GROUP BY b2.title
        ORDER BY howMany DESC";

$statement = $this->_em->getConnection()->prepare($sql);
    $result = $statement->executeQuery()->fetchAllAssociative();
    return $result;
}
```

You can verify that em is defined in ServiceEntityRepository witch extends EntityRepository

- 3- Call the method in ReportController
 - a. Add the repository to index method and Import it

```
Atteindre la définition
                                                                                                  F12
<?php
                                                         Atteindre les références
                                                                                              Maj+F12
namespace App\Controller;
                                                         Import Class
                                                                                             Ctrl+Alt+I
use Symfony\Bundle\FrameworkBundle\Controlle
use Symfony\Component\HttpFoundation\Respons
                                                         Import All Classes
                                                                                             Ctrl+Alt+A
use Symfony\Component\Routing\Annotation\Rou
                                                         Expand Class
                                                                                             Ctrl+Alt+S
class ReportController extends AbstractContr
                                                         Highlight Not Imported Classes
                                                                                            Ctrl+Alt+N
                                                         Highlight Not Used Classes
                                                                                            Ctrl+Alt+U
     #[Route('/report', name: 'app_report')]
     public function index(BorrowingRepositor
                                                         Generate namespace for this file
                                                                                            Ctrl+Alt+G
                                                         Kite: Find Related Code From Line
         return $this->render('report/index.h
                                                         Rechercher toutes les références
                                                                                           Maj+Alt+F12
               'controller_name' => 'ReportCont
          1);
                                                         Modifier toutes les occurrences
                                                         Mettre en forme le document avec...
                                                         Mettre le document en forme
                                                                                            Maj+Alt+F
```

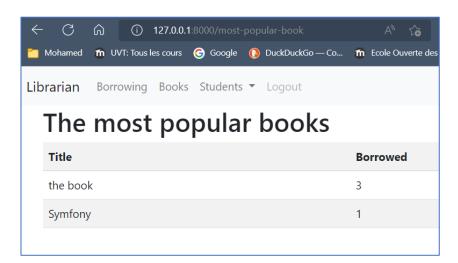
b. Add the definition of \$books proprieties

4- Customize the template

5- Change the name root

```
#[Route('/most-popular-book', name: 'most_popular_book')]
```

Result:



Using QueryBuilder

a. Add new request builder in BorrowinRepository

```
public function findMostPopularBooksQb()
{
    return $this->createQueryBuilder('b')
        ->addSelect('bk.title, COUNT(b) AS howMany')
        ->join('b.book', 'bk')
        ->groupBy('bk.title')
        ->orderBy('howMany','DESC')
        ->getQuery()
        ->getResult();
}
```

b. Add now the new definition of \$books proprieties in ReportController

```
$books = $repository->findMostPopularBooksQb();
```

The result is the same

Using Doctrine Query Language

a. Add new function named findMostPopularBooksDql() in BorrowinRepository

b. Add now the new definition of \$books proprieties in ReportController

```
$books = $repository->findMostPopularBooksDql();
```

The result is the same

Search by book

1. In order to search for all borrowings of a given book, add BookSearch in App\Entity

```
<?php
namespace App\Entity;</pre>
```

```
use Doctrine\ORM\Mapping as ORM;

class BookSearch
{
   private $book;
   public function getBook(): ?Book
   {
      return $this->book;
   }
   public function setBook(?Book $book): self
   {
      $this->book = $book;
      return $this;
   }
}
```

2. Create then in the Form section, BookSearchType class which is a builder for all books

```
<?php
namespace App\Form;
use App\Entity\Book;
use App\Entity\BookSearch;
use Symfony\Component\Form\AbstractType;
use Symfony\Component\Form\FormBuilderInterface;
use Symfony\Component\OptionsResolver\OptionsResolver;
use Symfony\Bridge\Doctrine\Form\Type\EntityType;
class BookSearchType extends AbstractType
    public function buildForm(FormBuilderInterface $builder, array $options): void
        $builder
            ->add('book',EntityType::class,['class' => Book::class,
            'choice_label' => 'title' ,
            'label' => 'Book' ]);
    public function configureOptions(OptionsResolver $resolver): void
        $resolver->setDefaults([
            'data_class' => BookSearch::class,
        ]);
```

}

For example add the root /BoorowingBook that will create an instance of BookSearch and will call BookSearchType to select all borrowings of a given book in a form

```
public function BorrowingBook(Request $request, BorrowingRepository $repository) {
    $bookSearch = new BookSearch();
    $form = $this->createForm(BookSearchType::class,$bookSearch);
    $form->handleRequest($request);
    $borrowings= [];

if($form->isSubmitted() && $form->isValid()) {
    $book = $bookSearch->getBook();
    if ($book!="")
        $borrowings=$repository->findBy( array('book' => $book) );
    else
        $borrowings= $repository->findAll();
}

return $this->render('report/BorrowingBook.html.twig',
        ['form' => $form->createView(),'borrowings' => $borrowings]);
}
```

