

Module : Technologies Web2.0

Enseignants : UP Web

Classes : 3A-4SE

Documents autorisés : OUI ☒ NON ☐ Nombre de pages : 5
Calculatrice autorisée : OUI ☐ NON ☒ Internet autorisée : OUI ☐ NON ☒

Date : 11/11/2023

Heure : 09h00

Durée : 01h30

Problème

Nous proposons de créer une application web destinée à la gestion des données liées aux pièces de théâtre (**TheatrePlay**) ainsi qu'à leurs spectacles associés (**Show**). Il s'agit d'une plateforme où les utilisateurs peuvent accéder à des informations sur les spectacles à venir. Cette application va être développée avec le Framework Symfony 5.4 et les données vont être enregistrées dans une base de données MySQL.

Soit le **diagramme de classe** suivant :

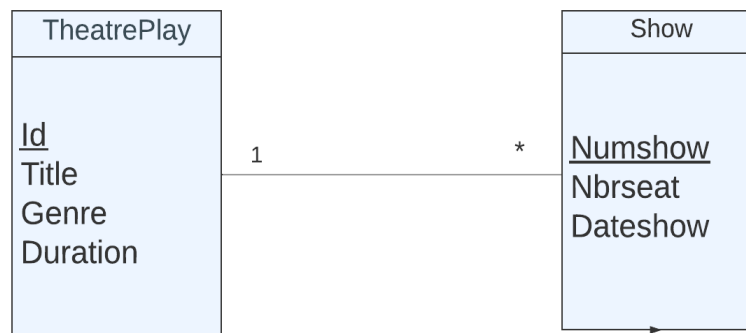


Figure 1

L'entité **TheatrePlay** est caractérisée par les attributs suivants :

- « **Id** » : c'est un identifiant **auto_increment**
- « **Title** » : c'est le titre de la pièce de théâtre et de type String
- « **Genre** » : c'est le genre de la pièce de théâtre et de type String
- « **Duration** » : c'est la durée de la pièce de théâtre et de type String

L'entité **Show** est caractérisée par les attributs suivants :

- « **Numshow** » : c'est un identifiant mais qui n'est pas **auto_increment**
- « **Nbrseat** » : c'est le nombre de places disponibles pour un spectacle et de type Integer
- « **Dateshow** » : c'est la date du spectacle et de type DateTime.

Travail demandé

Nb : Créer les deux entités « TheatrePlay » et « Show » avec les jointures nécessaires puis effectuer une migration vers la base de données nommée « theatre ».

1. Quel est le nom de fichier de migration que vous avez créé ainsi que le but des fichiers migrations dans un projet Symfony 5.4 ? (1 point)
2. Tester et décrire exactement le rendu des commandes suivantes : (1.5 points)
 - a. `symfony console doctrine:migrations:list` ou `php bin/console doctrine:migrations:list`
 - b. `symfony console doctrine:migrations:up-to-date` ou `php bin/console doctrine:migrations:up-to-date`
 - c. `symfony console doctrine:migrations:generate` ou `php bin/console doctrine:migrations: generate`
- 3.a. Compléter, dans le Controller, la fonction "DisplayPlay()", permettant la sélection de la liste des pièces de théâtre « TheatrePlay ». (1.5 points)

```
#[Route('/displayplay', name: 'app_displayplay')]
public function DisplayPlay([1]): Response
{
    [2]
    return $this->render ('TheatrePlay/list.html.twig',
    [
        [3]
    ]);
}
```

- b. Ajouter le code **Twig** qui permet d'afficher la liste des théâtres avec les titres en majuscule comme le montre la figure 2. (2 points)

Id	Title	Genre	Duration
1	HAMLET	Tragedy	1h30
2	OUR TOWN	Drama	2h
3	CYMBELINE	Romance	1h45

Figure 2

- c. Appliquer et décrire le rendu de ce filtre sur l'attribut titre :
{ { theatrePlay.title|slice(1,2)} } (0.5 point)

4. a. Ajouter un nouveau lien « **AddShow** » devant chaque « **TheatrePlay** » qui permet d’afficher le formulaire d’ajout d’un nouveau « **Show** » comme indique la figure 3. (0.5 point)

Id	Title	Genre	Duration	Action
1	HAMLET	Tragedy	1h30	Add Show
2	OUR TOWN	Drama	2h	Add Show
3	CYMBELINE	Romance	1h45	Add Show

Figure 3

- b. Après la création du formulaire « **ShowType** », écrire, dans la partie Controller, le code qui permet d’afficher le formulaire d’ajout d’un Show. (1 point)
- c. En cliquant sur le lien **Addshow**, une erreur ‘*Object of class App\Entity\TheatrePlay could not be converted to string*’ sera affichée (voir la figure 4), expliquer et résoudre cette erreur. (1 point)

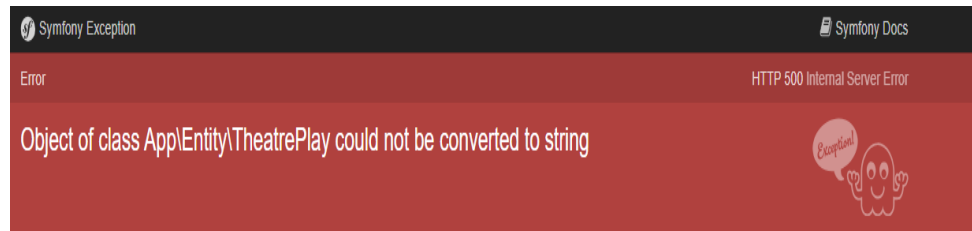


Figure 4

- d. Ajouter la ligne de code qui permet d’initialiser le nombre de places d’un Show « **Nbrseat** » à 30. (1 point)
- e. Quelle est la fonction prédéfinie responsable à la récupération des données saisies dans les inputs du formulaire et quel bundle sera importé pour utiliser la classe « **Request** » ? (1.5point)
5. Compléter le code de la fonction « **UpdateTheatrePlay()** » par les instructions nécessaires. (3.5 points)

```

#[Route('/update/.....[1].....', name: 'update_theatre_play')]
public function UpdateTheatrePlay( .....[2]..... , .....[3]..... , .....[4]..... ,
.....[5]..... )
{

$em=$managerRegistry->getManager() ;
$play=.....[6]..... ;
$form=$this->createForm( .....[7]..... , .....[8]..... ;
$form-> .....[9]..... ;
If( .....[10]..... )
{

$em-> .....[11]..... ;
$em-> .....[12]..... ;

Return $this-> .....[13]..... ;
}

return $this->render('theatre_play/update.html.twig', [
    'f'=>$form.....[14]..... ;
]);
}

```

6. Créer le lien « **Delete** » permettant de supprimer une pièce de théâtre comme indique la figure 5. (0.5 point)

Id	Title	Genre	Duration	Action	
1	HAMLET	Tragedy	1h30	Add Show	Delete
2	OUR TOWN	Drama	2h	Add Show	Delete
3	CYMBELINE	Romance	1h45	Add Show	Delete

Figure 5

7. Écrire le code de la fonction **DeleteTheatrePlay()** qui permet de supprimer un TheatrePlay. (1 point)
8. En cliquant sur le lien « **Delete** », une erreur sera affichée (voir la figure 6). Expliquer et résoudre cette exception. (1 point)

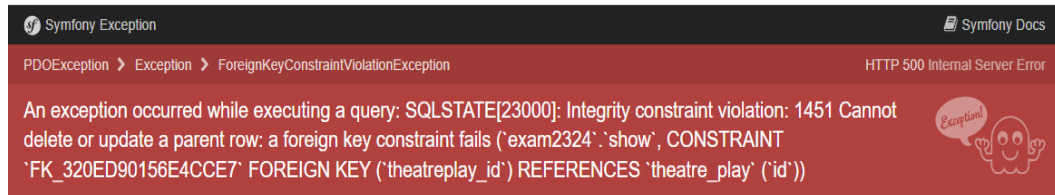


Figure 6

9. Ecrire la requête **QueryBuilder** qui permet de **(1 point)** :

- Ordonner les pièces de théâtre par ordre alphabétique décroissant selon leurs titres. (voir Figure 7)
- Ordonner les pièces de théâtre par ordre décroissant selon l'attribut Duration. (voir Figure 7)

Id	Title	Genre	Duration	Action	
2	OUR TOWN	Drama	2h	Add Show	Delete
1	HAMLET	Tragedy	1h30	Add Show	Delete
3	CYMBELINE	Romance	1h45	Add Show	Delete

Figure 7

10. Ecrire la requête **DQL** permettant d'afficher le nombre total de spectacles pour une pièce de théâtre dont son id est passé en paramètre. **(1.5 point)**

```
public function TotalNumber ($id)
{
....
}
```