

Ecole Supérieure Privée Technologies & Ingénierie

Type d'épreuve	: <input type="checkbox"/> Devoir	<input checked="" type="checkbox"/> Examen
Enseignant	: Wissem ELJAOUED	
Matière	: Framework PHP (Symfony)	
Année Universitaire	: 2022-2023	Semestre : 1
Classe	: CII-2-GLSI-A-B-C-D, CII-2-DSEN-A	
Documents	: <input type="checkbox"/> Autorisés	<input checked="" type="checkbox"/> Non autorisés
Date	: 04-01-2023	Durée : 1h :30
Nombre de pages	: 8 pages	

Exercice 01 : (10 pts)

A. QCM (5 pts)

- Dans Symfony, la configuration des routes peut être écrite en_____?
a) YAML
b) PHP
c) XML
d) Toutes les réponses sont vraies
- Quel est le rôle du routeur dans Symfony ?
a) Lister toutes les URL d'un site Internet
b) Associer des paramètres à une URL
c) Exécuter un contrôleur à partir d'une URL
d) Rediriger les URLs
- En Twig, pour appliquer un filtre sur un attribut *attribut* d'un objet *\$objet*, quelle est la bonne syntaxe ?
a) {{ objet->attribut | filtre }}
b) {% objet.attribut | filtre %}
c) {{ objet.attribut | filtre }}
d) {{ filtre(objet.attribut) }}
- Parmi les codes du contrôleur suivant, lequel supprime correctement une entité ?

a) <code>\$em= \$this->getDoctrine()->getManager(); \$salle = \$em->getRepository('NBundle:Salle') -> find(\$id); \$em->remove(\$salle); \$em->flush();</code>	c) <code>\$em= \$this->getDoctrine()->getManager(); \$em->remove(); \$em->flush();</code>
b) <code>\$em= \$this->getDoctrine()->getManager(); \$salle = \$em->getRepository('NBundle:Salle') -> find(\$id); \$em->remove (\$salle);</code>	d) <code>\$salle = \$em-> getRepository('NBundle:Salle') -> find(\$id); \$em->delete (\$salle); \$em->flush()</code>

5- Supposons qu'on a deux templates Twig : *enfant.twig* qui étend *parent.twig* qui contient un block appelé **body**. Si l'on surcharge ce block dans *enfant.twig*, comment retrouver le contenu de ce block provenant de *parent.twig* ?

- a) {% parent() % }
- b) {{ parent() }}
- c) {% parent(body) % }
- d) {{ parent(body) }}

B. Question/Réponse (5 pts)

1- Convertir le code PHP suivant en Twig.

```
<?php
echo '<ul>';
if (count($list_films) > 0)
{ foreach ($list_films as $film )
{ echo '<li>'. $film->getNom(). '</li>' ; }
} else { ?>
<li> Pas de film trouvé </li>
< ?php }
echo '</ul>'; ?>
```

.....

.....

.....

.....

.....

.....

2- Dessiner et expliquer l'architecture de Symfony en précisant tous les composants nécessaires.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

3- Comment faire la redirection vers la page qui a comme route « **liste_chiens** » en utilisant la méthode **redirect** (sachant que **redirect** prend comme argument une URL)

.....

.....

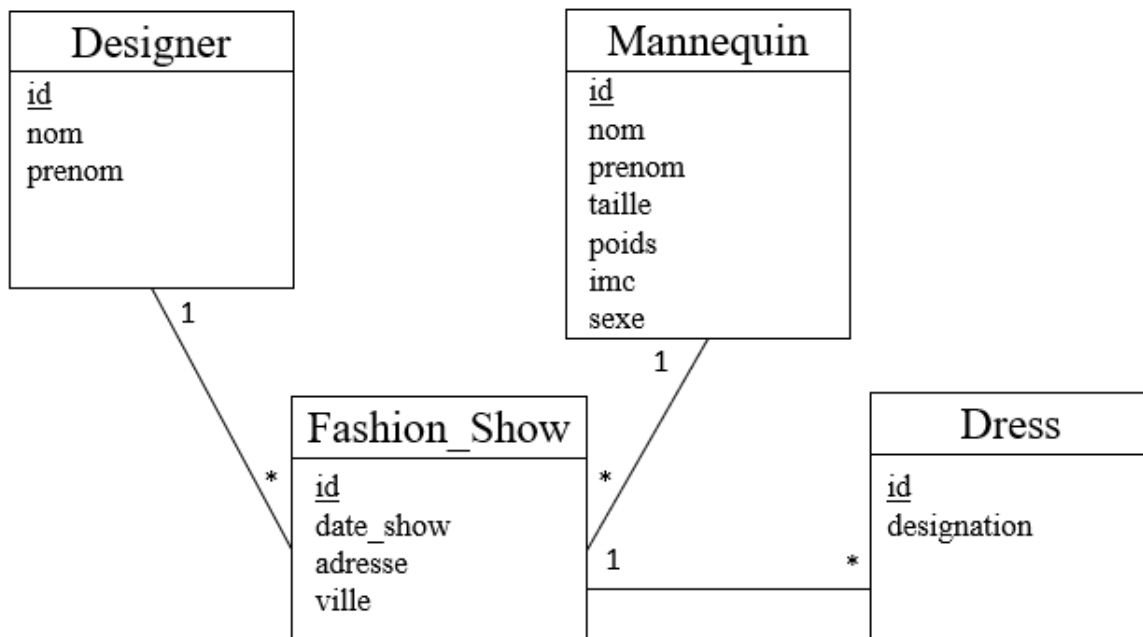
.....



Problème : (10 pts)

Notre objectif est d'implémenter une application web qui gère des fashion shows en précisant pour chaque designer, ses défilés de mode (fashion shows) et les mannequins participants (en leurs associant les robes à porter). Dans ce cadre, on propose de créer une application avec Symfony 5.3.

Soit le diagramme de classes suivant :



A. Configuration des entités :

A.1. Compléter les annotations (et les attributs) manquantes dans les entités suivantes sachant qu'on peut naviguer entre les entités dans les deux sens :

<pre> <?php namespace App\Entity; use Doctrine\ORM\Mapping as ORM; class Designer { /** * @ORM\GeneratedValue * @ORM\..... * @ORM\..... */ private \$id; /** * @ORM\..... </pre>	<pre> <?php namespace App\Entity; use Doctrine\ORM\Mapping as ORM; class FashionShow { /** * @ORM\GeneratedValue * @ORM\..... * @ORM\..... */ private \$id; /** * @ORM\..... </pre>
--	---

<pre> */ private \$nom; /** * @ORM\..... */ private \$prenom; //Getters-Setters } </pre>	<pre> */ private \$date_show; /** * @ORM\..... */ private \$adresse; /** * @ORM\..... */ private \$ville; //Getters-Setters } </pre>
---	--

B. Affichage de la liste des fashion shows :

Soit *FashionShowController*, le contrôleur responsable des actions de gestion des fashion shows.

B.1. Terminer l'action *listFS()* qui permet de retourner la page Twig suivante : '*listFS.html.twig*' (se trouve sous : *Templates/*). Cette page Twig permet de retourner la liste des fashion shows sauvegardés dans notre base de données.

```

<?php
namespace App\Controller;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;

class FashionShowController extends Controller
{
.....

public function listFS()
{
    //EntityManager
.....
}

```

```
//Liste des fashion shows
.....

return $this->render('.....',
    array(
        .....
    ));
}
}
```

B.2. Compléter la page Twig suivante : **'listFS.html.twig'**. Pour Le Designer d'un fashion show, on veut afficher son **nom**, pour son mannequin, afficher son **nom** et pour les robes associées, afficher leurs **désignations**.

[illegible]

C. Ajout d'un Mannequin :

Pour ajouter un mannequin, nous avons besoin de créer un formulaire pour introduire ses données. Soit ***MForm***, le formulaire d'ajout d'un mannequin.

Soit *MController*, le contrôleur responsable des actions de gestion des mannequins.

C.1. Terminer l'action *addM()* qui permet de:

- Rediriger vers la liste des mannequins *Si* l'ajout d'un mannequin est effectué correctement (sachant que le nom de la route de la liste des mannequins est *listeMRoute*).
- *Sinon*, Afficher le formulaire '*MForm*' en utilisant la page Twig suivante : '*addM.html.twig*' (se trouve sous : *Templates/M*).
- *Calculer l'indice de masse corporelle (imc) avant d'ajouter un mannequin sachant que :*
 - $IMC = \text{poids} / \text{taille}$.

```
<?php
namespace App\Controller;
//use
class MController extends Controller
{
.....
.....
.....
public function addM(Request ..... )
{
    // EntityManager
    .....

    // Création d'un mannequin
    .....

    // Création d'un formulaire
    $form = $this->..... (.....::class, .....);
    $form->handleRequest(.....);

    if (.....)
    {
        //Ajout du mannequin
        .....
        .....
        .....
        .....
        .....
        .....
        .....

        return $this-> .....
    }

    return $this->render('.....',
        array(
            .....
        ));
}
}
```

C.2. Compléter la page Twig suivante : '*addM.html.twig*'.

Ajout d'un mannequin

```
<form action=..... method=.....>
```

.....

```
<input type=..... />
</form>
```

D. Ajout d'un Fashion show :

Soit **'FSForm'**, le formulaire d'ajout d'un fashion show.

D.1. Compléter le formulaire ‘FSForm’ :

Afficher les **noms des mannequins** pour la liste déroulante des mannequins.

Afficher les **noms des designers** pour la liste déroulante des designers.

Afficher les **designations des robes** pour la liste déroulante des robes.

[illegible]

