Développement front avancé

MMI 3 – TP#5 S6

IUT CERGY-PONTOISE

Danielo **JEAN-LOUIS**

Symfony

- Framework PHP gratuit et open source
- Développé en France
- Première version : janvier 2007
  - Dernière version (01/2025) : Version 7.2

Source(s) :
- https://symfony.com/ - anglais

Symfony

- Complet et populaire
  - Fonctionne aussi bien pour un blog qu’un projet complexe
- Utilise le moteur de template Twig
- Sert de base à de nombreux projets :
  - Laravel, Ez Publish, PrestaShop...

Source(s) :

- https://symfony.com/ - anglais

Symfony

- Propose de nombreux modules
  - Communauté très développée
- Nécessite PHP 8.2+
- Utilise la POO
  - Programmation Orientée Objet
- Basé sur le patron MVC

Source(s) :
- https://symfony.com/ - anglais

MVC – Modèle-vue-contrôleur

- Patron d’architecture logicielle
  - Un des plus connus (pour le pas dire le plus connu)
- Chaque partie a sa responsabilité
  - Le code est mieux structuré et plus lisible

MVC – Modèle-vue-contrôleur

- Composé de trois parties :
  - Modèle : Données à afficher
  - Vue : Interface graphique
  - Contrôleur : Contient la logique. Fait le lien entre le modèle et la vue
    - Injecte les données dans la vue
    - Modifie / lit le modèle

MVC – Modèle-vue-contrôleur

MVC – Avantages

- Découplage du code
  - Chaque partie a sa responsabilité
  - Code plus facile à faire évoluer
- Code plus facilement testable au niveau des tests unitaires

MVC – Cas d’usage : Connexion utilisateur

- Formulaire de connexion (IHM)
  - Vue
- Récupération des données du formulaire (POST) et envoie les données vers le modèle
  - Contrôleur
- Vérifie si la paire utilisateur / mdp est bonne
  - Modèle

Installation

- Utilisation de la ligne de commandes
  - Commandes différentes en fonction de l’OS

Source(s) :
- https://symfony.com/download - anglais

Pratiquons ! - Symfony (Partie 1)

Pré-requis :

- Avoir la ressource ressources/symfony

A télécharger ici :
https://github.com/DanYellow/cours/raw/refs/heads/main/developpement-web-et-dispositif-interactif-s6/travaux-pratiques/numero-5/developpement-web-et-dispositif-interactif-s6_travaux-pratiques_numero-5.ressources.zip

Sécurité

- Symfony possède une couche de sécurité protégeant des failles suivantes :
  - XSS, CSRF, Injection SQL
- Seuls les assets et templates (via leur route) sont accessibles depuis le navigateur grâce au routing

Source(s) :

- https://fr.wikipedia.org/wiki/Cross-site_scripting
- https://fr.wikipedia.org/wiki/Cross-site_request_forgery
- https://fr.wikipedia.org/wiki/Injection_SQL

Routing

- Aiguillage du projet
  - Appelle un contrôleur en fonction de l’URL courante. Les deux sont liés
- Gère l’URL rewriting
  - URL plus élégantes
  - URL mieux référençables

Source(s) :

- https://symfony.com/doc/current/routing.html

Routing

```
http://localhost:8000/region/bretagne/ville/brest
```

Le routing de symfony permet d'utiliser cette url au lieu de celle en dessous

```
http://localhost:8000/?region=bretagne&ville=brest
```

Source(s) :

- https://symfony.com/doc/current/routing.html

Routing

- Nommage possible des routes
  - Manipulation aisée dans les templates
- Possibilité de :
  - Passer des paramètres et des query string
  - Définir des conditions sur les paramètres

Source(s) :

- https://symfony.com/doc/current/routing.html

Routing

```php
class MyController
{
    #[Route('/region/{region}/ville/{city}', name: 'my_city_details')]
    public function city_details($region, $city): Response
    {
        // [...]
    }
}
```

Exemple de route avec deux paramètres **obligatoires** nommée “my_city_details”

Source(s) :

- https://symfony.com/doc/current/routing.html

Pratiquons ! - Symfony (Partie 2)

Pré-requis :

- Avoir la ressource ressources/symfony

A télécharger ici :
https://github.com/DanYellow/cours/raw/refs/heads/main/developpement-web-et-dispositif-interactif-s6/travaux-pratiques/numero-5/developpement-web-et-dispositif-interactif-s6_travaux-pratiques_numero-5.ressources.zip

Routing - Notes

- Il est possible d’écrire les routes dans d’autres formats (php, yml, xml), préférez les attributs
- Si l’URL courante n’est pas “capturée” par le routeur → page 404

Source(s) :

- https://symfony.com/doc/current/routing.html

Routing - Notes

- La première URL qui est “capturée” voit sa fonction appelée
  - Pensez à ordonner vos routes

Source(s) :
- https://symfony.com/doc/current/routing.html

Contrôleur

- Gestionnaire de la logique d’une app sf
- Classe PHP
  - Suffixée “Controller” par convention
- Doit impérativement retourner une Reponse
  - Reponse → classe Symfony

Source(s) :
- https://symfony.com/doc/current/controller.html

Contrôleur

- Constitué de méthodes php qui réagissent (ou non) en fonction de l'URL courante
- Peut hériter de la classe "AbstractController"
  - Ajoute de nouvelles méthodes dont la gestions de templates

Source(s) :
- https://symfony.com/doc/current/controller.html

Contrôleur

```php
<?php
namespace App\Controller;

// [...]
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;

class FrontController extends AbstractController
{
    #[Route('/')]
    public function index(): Response
    {
        return $this->render('index.html.twig');
    }
}
```

Source(s) :

- https://symfony.com/doc/current/controller.html

Templates

- Responsables de la Vue dans le modèle MVC
- Gère l’affichage de la donnée, affiché dans le navigateur
- Peut recevoir des données en provenance du contrôleur

Source(s) :

- https://symfony.com/doc/current/templates.html

Templates

- Placés dans le dossier templates/
- Plusieurs formats possibles :
  - php, html et twig (préférez ce dernier)

Source(s) :
- https://symfony.com/doc/current/templates.html

Templates / Contrôleur - Connexion

```php
<?php
namespace App\Controller;

// [...]
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;

class FrontController extends AbstractController
{
    #[Route('/lucky')]
    public function index(): Response
    {
        // [...]
        return $this->render('index.html.twig', [
            'formation' => "MMI",
            'iut' => "CY Paris Université - Site de Sarcelles",
            'students_list' => $studentsList,
        ]);
    }
}
```

Templates / Contrôleur - Connexion

```twig
<ul>
    {% for student in students_list %}
        <li>{{ student.lastname }} - {{ iut }}</li>
    {% endfor %}
</ul>
```

On affiche les données du contrôleur dans le template twig

Twig dans Symfony

- Symfony injecte des variables globales dans tous les templates
  - Ex : app.request.method
- Possède des fonctions supplémentaires
  - Ex : path() → génère des urls relatives

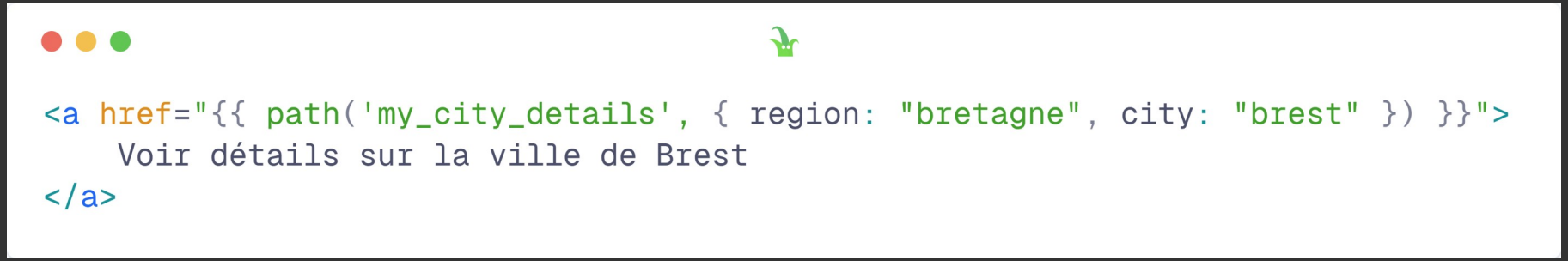
```twig
<a href="{{ path('my_city_details', { region: "bretagne", city: "brest" }) }}">
    Voir détails sur la ville de Brest
</a>
```

Source(s) :

- https://symfony.com/doc/current/templates.html#the-app-global-variable
- https://symfony.com/doc/current/templates.html#linking-to-pages

Twig dans Symfony

- Possède des fonctions supplémentaires
  - path() → génère des urls relatives



```twig
<a href="{{ path('my_city_details', { region: "bretagne", city: "brest" }) }}">
    Voir détails sur la ville de Brest
</a>
```

  - asset() → accès au contenu du dossier public/
  - form() → génère un formulaire

Source(s) :
- https://symfony.com/doc/current/templates.html#the-app-global-variable
- https://symfony.com/doc/current/templates.html#linking-to-pages

Pratiquons ! - Symfony (Partie 3)

Pré-requis :

- Avoir la ressource ressources/symfony

A télécharger ici :
https://github.com/DanYellow/cours/raw/refs/heads/main/developpement-web-et-dispositif-interactif-s6/travaux-pratiques/numero-5/developpement-web-et-dispositif-interactif-s6_travaux-pratiques_numero-5.ressources.zip

Templates / Twig – Notes

- On utilisera la snake_case pour nommer templates et dossiers de templates
- Impossible d'utiliser du php dedans mais Twig propose un ensemble de fonctions
  - Vous pouvez créer les vôtres

Source(s) :

- https://symfony.com/doc/current/templates.html
- https://fr.wikipedia.org/wiki/Snake_case
- https://twig.symfony.com/

Templates / Twig – Notes

- Inutile de mettre “templates” dans le chemin, Symfony cherchera toujours dans ce dossier
- Le même template peut être réutilisé

Source(s) :

- https://symfony.com/doc/current/templates.html
- https://fr.wikipedia.org/wiki/Snake_case
- https://twig.symfony.com/

Questions ?