

Travaux Pratiques : Création d'une petite application avec Symfony

Objectifs : À la fin de ce TP, vous aurez créé une petite application qui affiche une liste de produits et de clients, avec la possibilité d'afficher les détails pour chaque entité. Vous utiliserez les Méthodes Helper de l'AbstractController, comprendrez les bases de Twig, manipulerez l'objet Request et utiliserez le système de routage dans Symfony 6.

Prérequis :

- Installation de Symfony 6 et compréhension de base de son fonctionnement.
- Serveur local configuré (par exemple XAMPP, MAMP, WAMP, ou Docker).

Partie 1 : Les Méthodes Helper de l'AbstractController

1. Créer un nouveau contrôleur **ProductController** et utiliser le helper **render()** pour afficher une vue **products.html.twig** qui listera des produits factices.

Partie 2 : Appréhender les Bases de Twig dans Symfony 6

2. Dans la vue **products.html.twig**, créer une structure HTML basique avec un tableau pour lister les produits.
3. Utiliser la syntaxe Twig pour faire une boucle sur les produits factices et afficher leurs noms et prix.

Partie 3 : Manipulation de l'objet Request dans Symfony 6

4. Ajouter une méthode dans **ProductController** qui gère l'affichage du détail d'un produit. Utilisez l'objet **Request** pour récupérer l'ID du produit passé en paramètre dans l'URL.

Partie 4 : Le système de routage

5. Configurer la route pour la méthode de détail du produit en utilisant les annotations de route avec un paramètre **{id}**.
6. Dans **products.html.twig**, ajouter des liens vers la page de détail de chaque produit en utilisant la méthode **path()** de Twig.

Partie 5 : Utilisation Twig

7. Créer un nouveau template **customers.html.twig** et un nouveau contrôleur **CustomerController** avec une méthode pour afficher une liste de clients factices.
8. Dans **customers.html.twig**, utilisez Twig pour afficher les clients dans une liste ordonnée.

Partie 6 : Création de liens et utilisation avancée des routes

9. Dans le contrôleur **HomeController**, créer une méthode **index()** qui affichera un tableau de bord simple (**dashboard.html.twig**) avec des liens vers les pages de produits et de clients en utilisant la méthode **path()** de Twig.

Partie 7 : Révision et intégration

10. Tester toutes les routes en utilisant la commande **php bin/console debug:router** et en accédant à ces routes via le navigateur.

11. Veiller à ce que toutes les pages soient bien interconnectées grâce à des liens, permettant une navigation fluide entre le tableau de bord, la liste des produits, la liste des clients et les détails de chaque entité.

Point Bonus : Mise en page avec Bootstrap

- Intégrer Bootstrap à votre projet Symfony pour améliorer l'interface utilisateur de votre application. Utilisez la documentation officielle de Bootstrap pour vous aider à comprendre comment utiliser le Framework CSS.

Livraison :

- Les étudiants doivent soumettre le projet via un système de gestion de versions comme Git, en hébergeant leur code sur une plateforme comme GitHub ou GitLab.
- Ils doivent également fournir un fichier README.md avec des instructions sur la manière de configurer et de lancer le projet, ainsi que des captures d'écran des différentes pages de l'application.

Critères d'évaluation :

- Respect des consignes données.
- Qualité du code (lisibilité, respect des conventions).
- Fonctionnement correct de toutes les routes.
- Qualité de l'interaction utilisateur (UX/UI) dans les templates Twig.
- Présence et qualité du fichier README.md.

Conseils :

- Utilisez les ressources du cours pour vous aider à compléter les tâches.
- Testez régulièrement votre application pour vous assurer que chaque partie fonctionne comme prévu.
- N'hésitez pas à ajouter une touche personnelle à votre application pour la rendre unique.