

TP Symfony – Gestion de Tâches

Contexte du TP

Dans ce TP, vous allez développer une application web simple de gestion de tâches. L'application permettra de :

- **Ajouter, afficher, modifier et supprimer des tâches.**
- Marquer une tâche comme terminée.
- Consulter le détail d'une tâche.

Ce mini-projet fil rouge vous permettra de vous familiariser avec :

- La structure d'un projet Symfony.
- La création d'entités et de contrôleurs.
- La manipulation de la base de données avec Doctrine.
- La création et la gestion de formulaires avec Twig.

Pour en savoir plus sur le contexte et l'architecture de Symfony, consultez la [documentation Symfony - Structure du projet](#).

Étapes Progressives

Partie 1 : Initialisation du Projet et Découverte de la Structure

Objectifs :

- Comprendre la structure d'un projet Symfony.
- Découvrir les dossiers clés (`src/` , `templates/` , `config/` , etc.).
- Lancer le serveur intégré pour visualiser l'application.

Instructions :

1. Création du projet :

Exécutez la commande suivante pour créer un nouveau projet Symfony avec Flex :

```
symfony new gestion-taches --webapp
```

Documentation utile :
[Installation de Symfony.](#)

2. Exploration de la structure :

- Ouvrez le projet dans votre éditeur favori.
- Repérez les dossiers essentiels :
 - `src/` : Contient le code métier (contrôleurs, entités, formulaires, etc.).
 - `templates/` : Contient les vues Twig.
 - `config/` : Contient la configuration du projet. *Documentation utile :*
[Comprendre la structure d'un projet Symfony.](#)

3. Lancement du serveur :

Utilisez la commande suivante pour démarrer le serveur de développement :

```
symfony server:start
```

Ensuite, ouvrez votre navigateur et rendez-vous sur `http://localhost:8000` pour visualiser la page d'accueil par défaut.

Documentation utile :
[Le serveur web intégré de Symfony.](#)

Ce que vous apprenez :

- La configuration initiale d'un projet Symfony.
- La compréhension de l'architecture et des dossiers essentiels d'un projet Symfony.

Partie 2 : Création de l'Entité "Task" avec Doctrine

Objectifs :

- Créer une entité pour modéliser une tâche.
- Configurer et migrer la base de données.

Instructions :

1. Création de l'entité Task :

Utilisez la commande suivante pour générer l'entité :

```
php bin/console make:entity Task
```

Ajoutez les propriétés suivantes :

- `id` (généré automatiquement)
- `title` (string, 255 caractères)
- `description` (text, nullable)
- `isDone` (boolean, par défaut false)
- `createdAt` (datetime)

Documentation utile :

[Doctrine et Symfony : Créer une entité.](#)

2. Création et exécution de la migration :

Générez la migration avec :

```
php bin/console make:migration
```

Puis exécutez-la avec :

```
php bin/console doctrine:migrations:migrate
```

Documentation utile :

[Migrations Doctrine dans Symfony.](#)

Ce que vous apprenez :

- La création et la configuration d'une entité avec Doctrine.
- La gestion des migrations pour synchroniser la base de données avec vos entités.

Partie 3 : Création d'un Contrôleur et de Vues avec Twig

Objectifs :

- Mettre en place un contrôleur pour gérer les actions liées aux tâches.
- Créer des templates Twig pour afficher la liste et le détail des tâches.

Instructions :

1. Création du contrôleur `TaskController` :

Utilisez la commande suivante pour générer un nouveau contrôleur :

```
php bin/console make:controller TaskController
```

Dans ce contrôleur, ajoutez des actions pour :

- Afficher la liste des tâches (méthode `index`).
- Afficher le détail d'une tâche (méthode `show`).

Documentation utile :

[Création d'un contrôleur avec Symfony](#).

2. Création des vues avec Twig :

- Dans le dossier `templates/task/` , créez `index.html.twig` pour afficher toutes les tâches.
- Créez également `show.html.twig` pour afficher le détail d'une tâche.
- Utilisez les boucles et conditions de Twig pour rendre les données dynamiques.

Documentation utile :

[Twig et Symfony : les bases des templates](#).

Ce que vous apprenez :

- La mise en place d'un contrôleur et la définition de routes.
- La création de vues dynamiques avec Twig pour afficher des données provenant du contrôleur.

Partie 4 : Gestion des Formulaires et Opérations CRUD

Objectifs :

- Créer et manipuler des formulaires Symfony pour l'ajout et la modification des tâches.
- Implémenter les opérations CRUD (Créer, Lire, Mettre à jour, Supprimer).

Instructions :

1. Création d'un formulaire TaskType :

Utilisez la commande suivante pour générer un formulaire :

```
php bin/console make:form TaskType
```

Dans ce formulaire, ajoutez les champs correspondant à l'entité Task (`title` , `description` , etc.).

Documentation utile :

[Les formulaires dans Symfony.](#)

2. Intégration du formulaire dans le contrôleur :

- Ajoutez une action dans le `TaskController` pour créer une nouvelle tâche. Gérez la soumission du formulaire, la validation, et la persistance via Doctrine.
- Ajoutez également une action pour modifier une tâche existante en pré-remplissant le formulaire.
- Proposez enfin une option pour supprimer une tâche.

Documentation utile :

[Contrôleurs et formulaires](#)

et

[Opérations CRUD avec Doctrine.](#)

3. Affichage et validation côté client :

- Intégrez les formulaires dans vos vues Twig.
- Affichez les messages de validation en cas d'erreurs ou de succès.

Documentation utile :

[Affichage et gestion des erreurs de formulaire dans Twig.](#)

Ce que vous apprenez :

- La création et la gestion de formulaires dans Symfony.
- La mise en œuvre complète d'un cycle CRUD avec persistance en base de données.