

Le P'tit Jardinier – Documentation Technique (Symfony)

1. Présentation du Projet

Cet atelier professionnel a pour but de réaliser une application web, à l'aide du Framework PHP Symfony. Cette application reprend le contexte d'un site de jardinerie, qui doit permettre de pouvoir créer des devis avec un ou plusieurs taillages de haies dans le même devis.

Fonctionnalités principales :

- Gestion des haies et de leurs catégories
 - Gestion des clients
 - Création de devis (calcul automatique du tarif)
 - Espace membre avec authentification
-

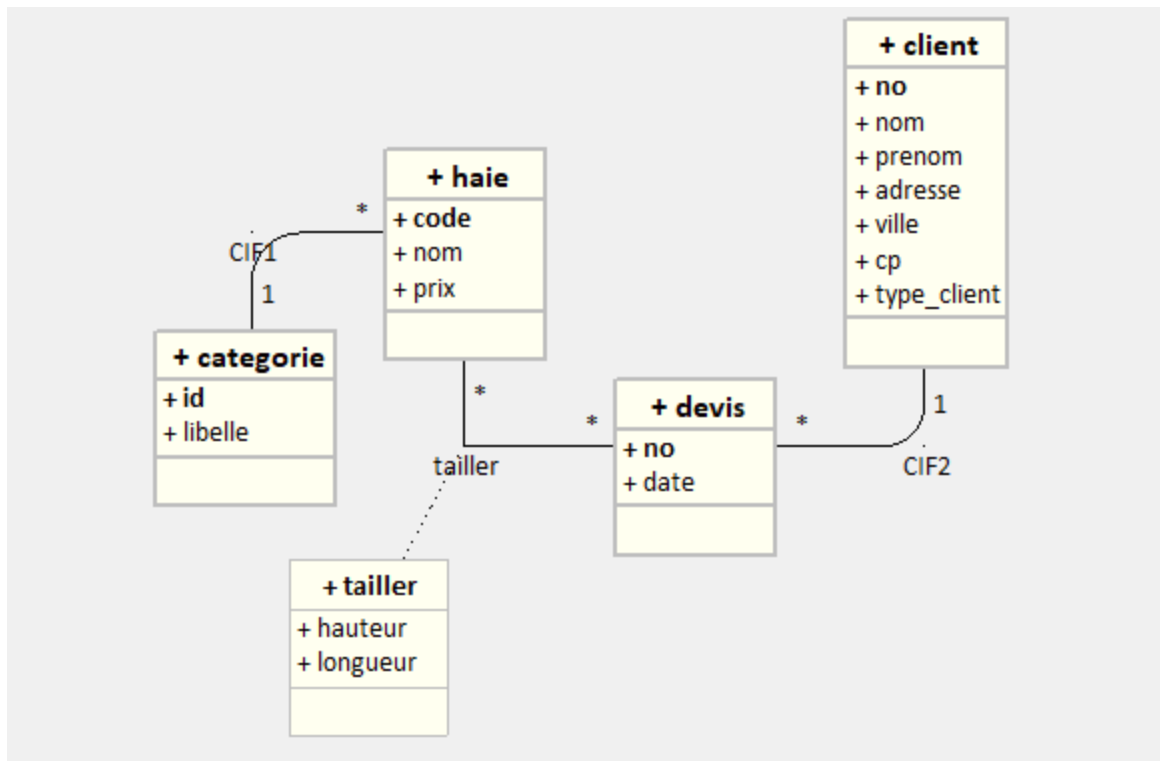
2. Environnement de Développement

- Symfony CLI
- PHP 8.x
- WAMP (serveur local Apache/MySQL)
- PHPMysqlAdmin (gestion de la base)
- Visual Studio Code (éditeur de code)
- SGBD : MySQL
- Interface basée sur Twig

Architecture utilisée :

- MVC (Modèle - Vue - Contrôleur)

Modèle conceptuel :



Entités et Relations

Exemple de relation Haie - Catégorie :

```
#[ORM\ManyToOne(inversedBy: 'haies')]
#[ORM\JoinColumn(nullable: false)]
private ?Categorie $categorie = null;
```

Génération des Migrations

Après modification des entités, générer et exécuter les migrations :

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

3. Configuration Symfony

Base de données

Dans le fichier .env :

```
DATABASE_URL="mysql://user:password@127.0.0.1:3306/le_ptit_jardinier"
```

Doctrine est utilisé comme ORM (Object Relational Mapper).

4. Routage

Les routes sont définies par annotations :

```
[Route('/haie', name: 'haie')]
```

Elles permettent d'associer une URL à une méthode de contrôleur.

5. Contrôleurs et Vues

Contrôleur : Exemple de création d'une haie

```
#[Route('/new', name: 'app_haie_new', methods: ['GET', 'POST'])]
public function new(Request $request, EntityManagerInterface $entityManager):
    Response
    {
        // verifie si un utilisateur est connecté
        if (!$this->getUser()) {
            return $this->redirectToRoute('app_login');
        }

        $haie = new Haie();
        $form = $this->createForm(HaieType::class, $haie);
        $form->handleRequest($request);
        if ($form->isSubmitted() && $form->isValid()) {
            $entityManager->persist($haie);
            $entityManager->flush();
            return $this->redirectToRoute('app_haie_index', [],
Response::HTTP_SEE_OTHER);
        }
        return $this->render('haie/new.html.twig', [
            'haie' => $haie,
            'form' => $form,
```

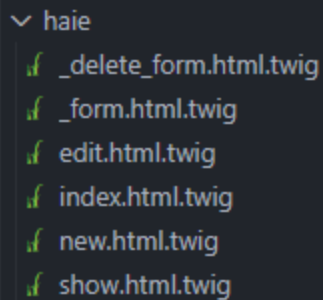
```
    });  
}
```

6. Vue (Twig)

```
{% extends 'base.html.twig' %}  
  
{% block body %}  
    <h1>Ajouter une haie</h1>  
    {{ form(form) }}  
{% endblock %}
```

7. CRUD avec Doctrine

```
php bin/console make:crud
```



```
▼ haie  
  ✓ _delete_form.html.twig  
  ✓ _form.html.twig  
  ✓ edit.html.twig  
  ✓ index.html.twig  
  ✓ new.html.twig  
  ✓ show.html.twig
```

Création

Utilisation de `persist()` et `flush()` :

```
$entityManager->persist($haie);  
$entityManager->flush();
```

JARDINIER

Accueil

Gestion Devis

Créer un Devis

Voir les Devis

Gestion Haie

Créer une haie

Voir les haies

Gestion Catégorie

Créer une catégorie

Voir les catégories

Gestion Users

Créer un User

Voir les users

Créer une nouvelle Haie

Nom

Prix

Categorie

Tuya

Sauvegarder

admin admin

Administrateur

Lecture

Utilisation du Repository :

```
return $this->render('haie/index.html.twig', [  
    'haies' => $haieRepository->findAll(),  
]);
```

JARDINIER

Accueil

Gestion Devis

Créer un Devis

Voir les Devis

Gestion Haie

Créer une haie

Voir les haies

Gestion Catégorie

Créer une catégorie

Voir les catégories

Gestion Users

Créer un User

Voir les users

Gestion des Haie

Nom : Haie 1

Prix : 6.00€

Type : Tuya

VOIR

MODIFIER

SUPPRIMER

Nom : Haie 2

Prix : 10.00€

Type : Tulipe

VOIR

MODIFIER

SUPPRIMER

Nom : Haie 3

Prix : 20.00€

Type : Tuya

VOIR

MODIFIER

SUPPRIMER

Ajouter une Haie

admin admin

Administrateur

Modification

Modifier l'objet et appeler flush() :

```

$form = $this->createForm(HaieType::class, $haie);
$form->handleRequest($request);

if ($form->isSubmitted() && $form->isValid()) {
    $entityManager->flush();
}

```

JARDINIER

- Accueil
- Gestion Devis
 - Créer un Devis
 - Voir les Devis
- Gestion Haie
 - Créer une haie
 - Voir les haies
- Gestion Catégorie
 - Créer une catégorie
 - Voir les catégories
- Gestion Users
 - Créer un User
 - Voir les users

Edit Haie

Nom
Haie 1

Prix
6

Categorie
Tuya

Mettre à jour

Retour à la liste

SUPPRIMER

admin admin
Administrateur

sf

Suppression

```

$entityManager->remove($haie);
$entityManager->flush();

```

8. Formulaires Symfony

Symfony permet la génération automatique de formulaires à partir des entités.

Exemple :

```

$builder
    ->add('nom')
    ->add('prix')
    ->add('categorie', EntityType::class, [
        'class' => Catégorie::class,
        'choice_label' => 'libelle',
    ])

```

```
}  
;
```

Types de champs utilisés au cours du projet :

- TextType
- NumberType
- DateType
- SubmitType ...

9. Authentification & Espace Membre

Création d'un utilisateur :

```
php bin/console make:user
```

10. Gestion des Erreurs

Validation des données avec Symfony Validator :

```
use Symfony\Component\Validator\Constraints\Length;  
#[Assert\Length(  
    min: 8,  
    minMessage: "Le code doit contenir au moins {{ limit }} caractères"  
)]
```

Affichage des erreurs dans la vue Twig :

```
{{ form_errors(form.codePostal) }}
```

 **Se connecter**

Email :

Mot de Passe :

Identifiants invalides.

[Se connecter](#)

Nouvel utilisateur ? [S'enregistrer](#)

11. Génération de Devis

- Sélection des haies à inclure dans le devis
- Saisie des dimensions
- Calcul dynamique du prix total

<http://127.0.0.1:8000/liste/devis>

JARDINIER

Gestion des Devis

[Accueil](#)

[Gestion Devis](#)

[Créer un Devis](#)

[Voir les Devis](#)

[Gestion Haie](#)

[Voir les haies](#)

[Gestion Catégorie](#)

[Voir les catégories](#)

Devis n°2

24/03/2025

[SUPPRIMER](#)

[VOIR](#)

Devis n°4

24/03/2025

[SUPPRIMER](#)

[VOIR](#)

[Créer un nouveau Devis](#)

JARDINIER

Accueil

Gestion Devis

Créer un Devis

Voir les Devis


Gestion Haie

Voir les haies

Gestion Catégorie

Voir les catégories

Créer le : 24/03/2025



DEVIS N°2

Jardinier, Inc.
Rue de la Jardinierie
JardinCity, CP 12345

Vous êtes un Particulier

Type	Prix unitaire	Longueur	Hauteur	
Haie 1	6,00 €	10 cm	20 cm	<div>SUPPRIMER</div>
Haie 3	20,00 €	20 cm	12 cm	<div>SUPPRIMER</div>

Total : 460,00 €

Segalat Adrien
Particulier

RETOUR A LA LISTE DES DEVIS

AJOUTER UNE HAIE AU DEVIS

http://127.0.0.1:8000/mesure/add/2

JARDINIER

Accueil

Gestion Devis

Créer un Devis

Voir les Devis

Gestion Haie

Voir les haies

Gestion Catégorie

Voir les catégories

Ajouter une haie au devis N°2

--Choisir votre type de haie--

Longueur en cm

Hauteur en cm

Ajouter la haie au devis

Segalat Adrien
Particulier