

Documentation Technique – Le P'tit Jardinier

1. Introduction

Ce projet Symfony, intitulé « Le P'tit Jardinier », est une application web permettant la gestion de clients, la création et la gestion de devis liés à l'entretien de haies, ainsi que l'administration des utilisateurs via un système d'authentification. Il est développé dans le cadre d'un BTS SIO, avec un accent sur la programmation web, l'interaction avec une base de données via Doctrine, et l'utilisation des fonctionnalités complètes du framework Symfony.

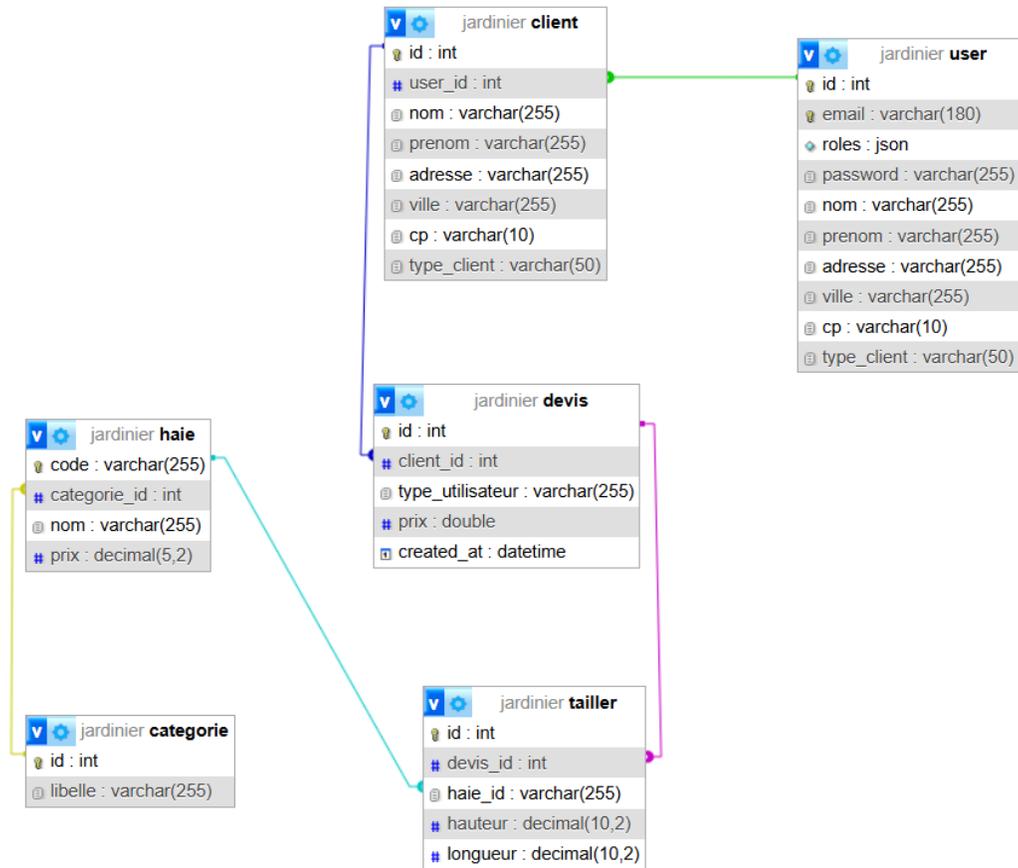


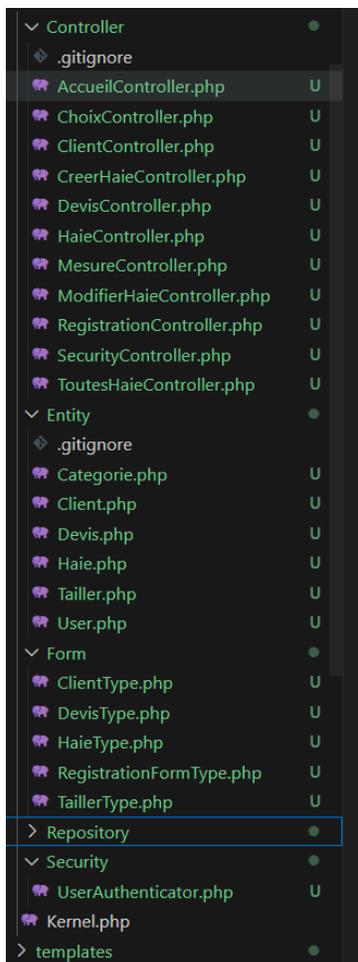
2. Structure du projet

Voici la structure globale du projet avec ses principaux composants :

- Contrôleurs : gestion des routes et de la logique métier.
- Entity : définition des modèles de données Doctrine.
- Forms : création de formulaires Symfony pour la saisie utilisateur.

- Repositories : requêtes personnalisées sur les entités.
- Templates : vues HTML générées avec Twig.





3. Fonctionnalités principales

- Authentification et gestion des utilisateurs (inscription, connexion, sécurité).
- Gestion complète des clients (CRUD).
- Gestion des types de haies et de leurs caractéristiques.
- Création de devis avec ajout dynamique de haies.
- Exportation des devis au format PDF.
- Interface utilisateur avec des vues Twig stylisées.

4. Modules techniques

4.1 Authentification

Gérée via le SecurityController et Symfony Security. Utilisateur enregistré via RegistrationController.

4.2 Gestion des clients

Implémentée dans ClientController avec affichage, ajout, modification et suppression.

4.3 Gestion des haies

HaieController, CreerHaieController, ModifierHaieController permettent de gérer les haies.

4.4 Création de devis

DevisController permet de créer un devis avec des haies personnalisées (relation avec l'entité Tailler).

4.5 Affichage des devis

Devis affichés dans des templates Twig (mes_devis.html.twig, tous_les_devis.html.twig).

4.6 Génération PDF

Le template pdf.html.twig génère le rendu PDF à partir d'un devis donné.

5. Structure des entités

Voici les principales entités utilisées dans le projet :

- User : représente les utilisateurs enregistrés.
- Client : client particulier ou entreprise.
- Haie : type de haie, avec un prix au m².
- Tailler : relation entre un devis et les haies à tailler (hauteur, longueur).
- Devis : regroupe plusieurs haies à tailler, calcul du prix total.
- Categorie : permet de classer les haies.

6. Interfaces utilisateur (Twig)

Les interfaces sont construites à l'aide du moteur de templates Twig et stylisées avec du CSS simple :

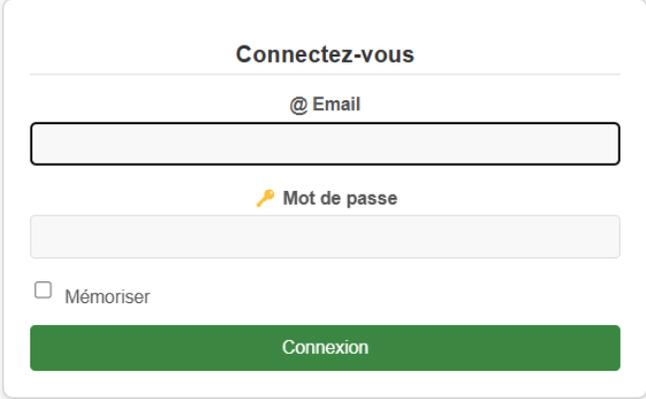
- Formulaires de création/édition pour les entités.
- Tableaux dynamiques pour les listes de clients et de devis.

- Affichage responsive et boutons interactifs pour améliorer l'expérience utilisateur.

7. Annexes

Quelques exemples de routes importantes :

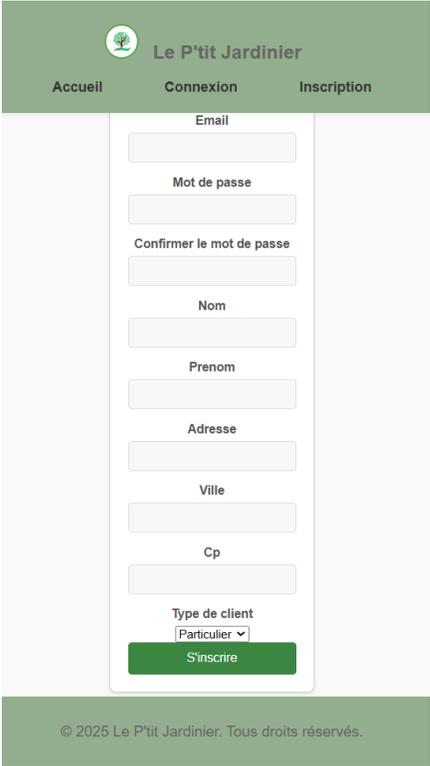
- `/login` : connexion utilisateur



The screenshot shows a login form with the following elements:

- Title: **Connectez-vous**
- Input field for email, labeled "@ Email".
- Input field for password, labeled "Mot de passe" with a key icon.
- Checkbox labeled "Mémoriser".
- Green button labeled "Connexion".

- `/register` : inscription



The screenshot shows a registration form for "Le P'tit Jardinier" with the following elements:

- Header: **Le P'tit Jardinier** with a tree icon.
- Navigation: Accueil, Connexion, Inscription.
- Form fields: Email, Mot de passe, Confirmer le mot de passe, Nom, Prenom, Adresse, Ville, Cp.
- Dropdown menu: Type de client (Particulier).
- Green button: S'inscrire.
- Footer: © 2025 Le P'tit Jardinier. Tous droits réservés.

- `/client` : liste des clients

- `/devis/new` : création d'un devis

Créer un nouveau devis

Type de haie ▾

Hauteur (m)

Longueur (m)

[+ Ajouter une haie](#)

[Enregistrer le devis](#)

Haies à tailler

- `/devis/{id}/pdf` : génération PDF d'un devis

PDF

[Télécharger en PDF](#)

8. Extraits de code significatifs

8.1 Exemple de connexion utilisateur (SecurityController)

```
public function login(AuthenticationUtils $authenticationUtils): Response
{
    // get the login error if there is one
    $error = $authenticationUtils->getLastAuthenticationError();

    // last username entered by the user
    $lastUsername = $authenticationUtils->getLastUsername();

    return $this->render('security/login.html.twig', [
        'last_username' => $lastUsername,
        'error' => $error,
    ]);
}
```

8.2 Ajout dynamique de haies dans un devis (DevisController)

```
$form = $this->createForm(DevisType::class, $devis);
$form->handleRequest($request);

if ($form->isSubmitted() && $form->isValid()) {
    foreach ($devis->getTailleurs() as $tailleur) {
        $tailleur->setDevis($devis);
        $entityManager->persist($tailleur);
    }
    $devis->setClient($this->getUser()->getClient());
    $devis->calculerPrixTotal();
    $entityManager->persist($devis);
    $entityManager->flush();

    return $this->redirectToRoute('mes_devis');
}
```

8.3 Entité Tailleur (relation Devis ↔ Haie)

```
/**
 * @ORM\Entity
 */
class Tailleur
{
    /**
     * @ORM\ManyToOne(targetEntity=Devis::class,
     inversedBy="tailleurs")
     */
    private $devis;

    /**
     * @ORM\ManyToOne(targetEntity=Haie::class)
     */
    private $haie;

    private $hauteur;
    private $longueur;
}
```

8.4 Génération PDF d'un devis (pdf.html.twig)

```

<h1>Devis n°{{ devis.id }}</h1>
<p><strong>Client :</strong> {{ devis.client.nom }} {{ devis.client.prenom }}</p>
<p><strong>Date :</strong> {{ devis.createdAt|date('d/m/Y H:i') }}</p>
<p><strong>Type :</strong> {{ devis.typeUtilisateur }}</p>

<table>
  <thead>
    <tr>
      <th>Type de haie</th>
      <th>Hauteur (m)</th>
      <th>Longueur (m)</th>
      <th>Prix unitaire</th>
      <th>Total</th>
    </tr>
  </thead>
  <tbody>
    {% for t in devis.taillers %}
    <tr>
      <td>{{ t.haie.nom }}</td>
      <td>{{ t.hauteur }}</td>
      <td>{{ t.longueur }}</td>
      <td>{{ t.haie.prix }} €/m2</td>
      <td>{{ (t.hauteur * t.longueur * t.haie.prix)|number_format(2, ',', ',') }} €</td>
    </tr>
    {% endfor %}
  </tbody>
</table>

```