

[Démarrer un projet Web] PHP avec le Framework Symfony

Contexte "Le p'tit jardinier"

Célia Baylet

Présentation du contexte

La société Le p'tit Jardinier a pour activité l'entretien des jardins pour les particuliers et les organisations. Etant contactée fréquemment pour des devis de taille de haies, elle souhaite mettre en place une application Web permettant à toute personne d'obtenir un tarif.

Langages et technologies utilisées

Symfony (framework php)

PHP

Doctrine ORM (pour la gestion de la base de données)

Twig (templates intégré à Symfony)

Html / css

Architecture

Le projet suit l'architecture **MVC** (Modèle-Vue-Contrôleur) :

- **Controllers** : Gèrent les requêtes HTTP et la logique métier.
- **Entities** : Représentent les objets métiers et interagissent avec la base de données.
- **Repositories** : Facilitent l'accès aux données stockées.
- **Templates (Twig)** : Gèrent l'affichage des pages HTML.

Authentification des utilisateurs

- Inscription et connexion via email/password
- Gestion des rôles (ROLE_USER, ROLE_ADMIN)

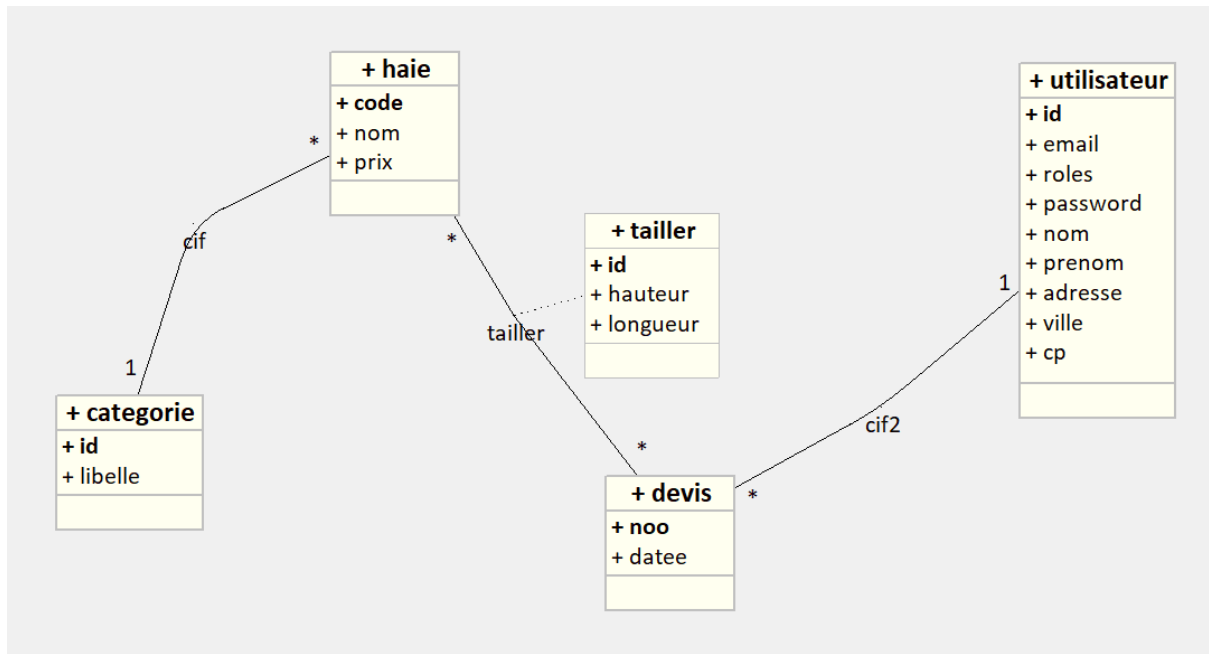
Gestion des devis

- Création de devis à partir d'un formulaire
- Calcul du prix en fonction de la longueur, hauteur et type de haie
- Remise de 10% pour les entreprises
- Stockage des devis en base de données avec l'id de l'utilisateur

Gestion des haies

- Chaque haie a un prix unitaire
- Les tarifs varient en fonction de la hauteur (>1.50m : majoration x1.5)

MODELE UML :



Haie --0,n---cif---1.1---- Tailler---1.1---cif---0,n---devis

Tailler vers haie manyToOne

Car on peut tailler plusieurs fois une haie

Devis vers user manyToOne

Et oneToMany client vers devis

Entités Principales

User.php :

```
#[ORM\Entity(repositoryClass: UserRepository::class)]
#[ORM\UniqueConstraint(name: 'UNIQ_IDENTIFIER_EMAIL', fields:
['email'])]
#[UniqueEntity(fields: ['email'], message: 'There is already an account
with this email')]
class User implements UserInterface, PasswordAuthenticatedUserInterface
{
```

```

#[ORM\Id]
#[ORM\GeneratedValue]
#[ORM\Column]
private ?int $id = null;

#[ORM\Column(length: 180)]
private ?string $email = null;

/**
 * @var list<string> The user roles
 */
#[ORM\Column]
private array $roles = [];

/**
 * @var string The hashed password
 */
#[ORM\Column]
private ?string $password = null;

```

Devis.php:

```

#[ORM\Entity(repositoryClass: DevisRepository::class)]
class Devis
{
    #[ORM\Id]
    #[ORM\Column(length: 255)]
    private ?string $no = null;

    #[ORM\Column(type: Types::DATE_MUTABLE)]
    private ?\DateTimeInterface $date = null;

    /**
     * @var Collection<int, Tailler>
     */
    #[ORM\OneToMany(targetEntity: Tailler::class, mappedBy: 'Devis')]
    private Collection $taillers;

    #[ORM\ManyToOne(inversedBy: 'devis')]
    #[ORM\JoinColumn(referencedColumnName:'id')]
    private ?User $user = null;
}

```

Contrôleur Devis

```
class DevisController extends AbstractController
{
    #[Route('/devis', name: 'app_devis')]
    public function index(
        Request $request,
        HaieRepository $haieRepository,
        UserRepository $userRepository,
        EntityManagerInterface $entityManager
    ): Response {
// Récupération de l'utilisateur connecté
// Récupération de l'utilisateur en base de données
// Récupération des données envoyées par le formulaire
// Calcul du prix
// Vérification du type de client
// Création et persistance du devis
// Création et persistance de l'objet Tailler
// Sauvegarde en bdd
$entityManager->flush();
    }
```

Vue Devis

```
<h2>Détails du devis</h2>
    <ul>
        <li><strong>Type de client :</strong> {{ choix ==
'entreprise' ? 'Entreprise' : 'Particulier' }}</li>
        <li><strong>Numéro de devis :</strong> {{ devis.no
}}</li>
        <li><strong>Date du devis :</strong> {{
devis.date|date('d/m/Y') }}</li>

        {% for tailler in tailleurs %}
            <li><strong>Type de haie :</strong> {{
tailler.haie.code }}</li>
            <li><strong>Longueur :</strong> {{ tailler.longueur
}} m</li>
        {% endfor %}
    </ul>
```

```

                <li><strong>Hauteur :</strong> {{ tailler.hauteur
}} m</li>

                {% endfor %}

                <li><strong>Prix total :</strong> {{ prix }} €</li>
            </ul>

            <a href="{{ path('app_accueil') }}">Retour</a>

```

Les fonctionnalités :

- **Inscription**

```
#[Route('/register', name: 'app_register')]
```

- **Connexion**

```
#[Route(path: '/login', name: 'app_login')]
```

- **Déconnexion**

```
#[Route(path: '/logout', name: 'app_logout')]
```

- **Création des haies**

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

- **Consultation des haies**

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

- **Modification des haies**

```
#[Route('modifier/haie/{code}', name: 'app_modifier_haie')]
```

- **Création d'un devis (type de haie / longueur / hauteur)**

```
#[Route('/devis', name: 'app_devis')]
```

Certaines routes ne sont pas accessibles en tant qu'utilisateurs (non-administrateurs)

Commandes de base :

Connexion lancer serveur web :

symfony server:start

//ou bien

symfony serve

On peut voir le site dans un navigateur, à l'adresse <http://127.0.0.1:8000>

Créer un contrôleur :

symfony console make:controller

sera créer sous la forme nomController

Commande de migrations :

php bin/console make:migration

php bin/console doctrine:migrations:migrate