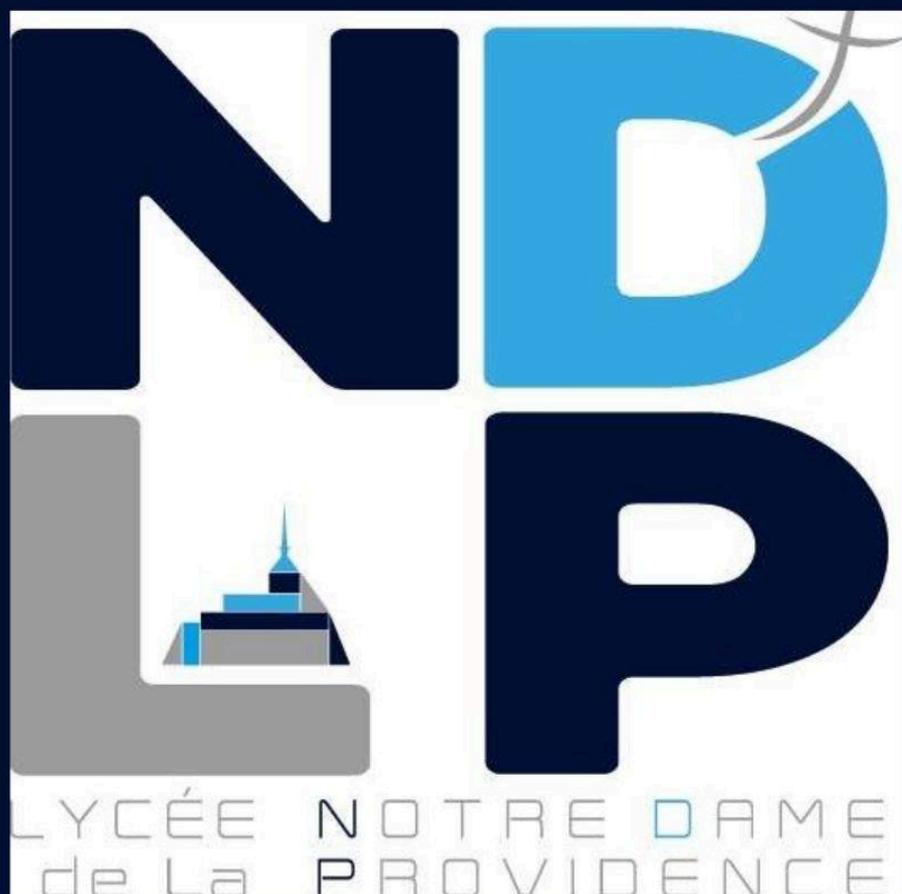


LAVEILLE
MATHIS

EMERIC
GRALL



DOCUMENTATION TECHNIQUE

NOTRE DAME DE LA
PROVIDENCE AVRANCHES

Documentation Technique - Stage Direct

1. Introduction.....	2
2. Architecture Technique.....	3
Technologies utilisées.....	3
Organisation du code.....	4
3. Base de Données.....	6
Modèle Conceptuel des Données (MCD).....	6
4. API et Routes.....	7
Endpoints principaux.....	7
5. Fonctionnalités Avancées.....	9
Authentification et Sécurité.....	9
Recherche avancée dynamique.....	9
Exportation des données.....	9
6. Contraintes et Bonnes Pratiques.....	10
7. Installation du projet Symfony.....	11
8. Conclusion.....	11

1. Introduction

Stage Direct est une application web développée avec Symfony visant à faciliter la gestion des stages scolaires. Cette documentation détaille l'architecture technique, la structure de la base de données, les API et les principales fonctionnalités avancées.

2. Architecture Technique

Technologies utilisées

- **Back-end** : PHP 8 avec Symfony



- **Base de données** : PostgreSQL / Néon



- **Front-end** : Twig, HTML, CSS, JavaScript



- **Authentication** : Security Component de Symfony



- **Gestion des dépendances** : Composer



- **Environnement de développement** : Git



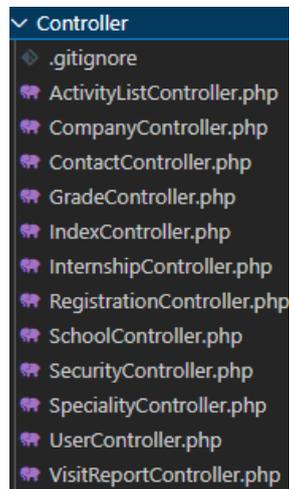
Organisation du code

Le projet est structuré selon l'architecture MVC de Symfony :

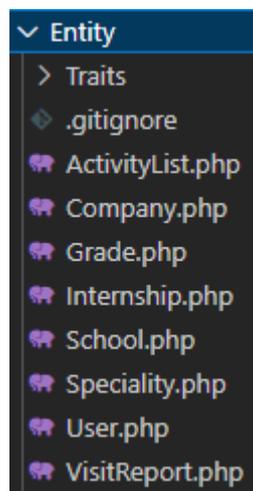
stage-direct-final/

├── config/	# Configuration Symfony
├── migrations/	# Migrations base de données
├── public/	# Fichiers publics
├── src/	# Code source
│ ├── Controller/	# Contrôleurs
│ ├── Entity/	# Entités
│ ├── Form/	# Formulaires
│ ├── Repository/	# Repositories
│ └── Security/	# Sécurité
├── templates/	# Templates Twig
└── tests/	# Tests unitaires

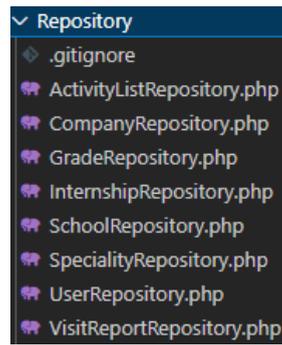
- **src/Controller/** : Contient les contrôleurs gérant les requêtes HTTP.



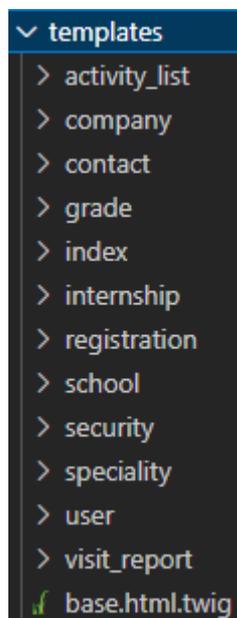
- **src/Entity/** : Définit les modèles de données.



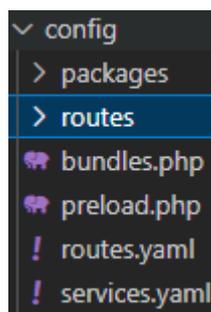
- **src/Repository/** : Gère les requêtes vers la base de données.



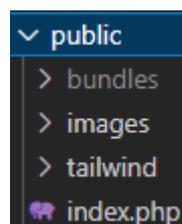
- **templates/** : Contient les vues Twig pour le rendu HTML.



- **config/** : Paramétrage de l'application.



- **public/** : Point d'entrée du projet.



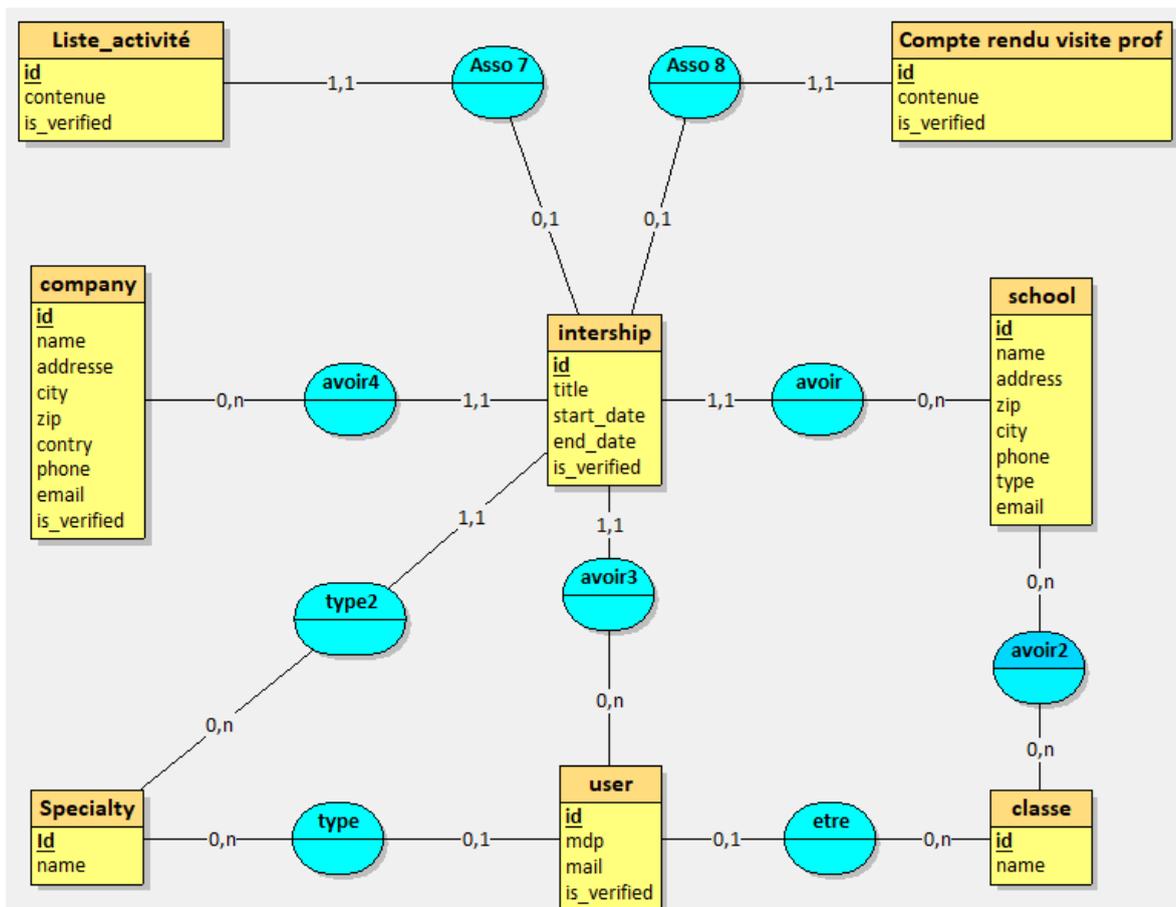
3. Base de Données

Modèle Conceptuel des Données (MCD)

Le MCD inclut les principales entités :

- **Utilisateur** (id, mdp, email, rôle, is_verified)
- **Speciality** (id, name)
- **Classe** (id, name)
- **Internship** (id, title, start-date, end-date, is_verified)
- **Company** (id, name, city, zip, country, phone, email, is_verified)
- **School** (id, name, address, zip, phone, type, email)
- **Liste_activité** (id, contenu, is_verified)
- **Visit_report** (id, contenu, is_verified)

Sans compter les clés étrangères dans les tables.



4. API et Routes

Endpoints principaux

Un **endpoint** est une URL spécifique d'une API qui permet à un client (comme une application web ou mobile) d'interagir avec un serveur. Par exemple, un endpoint peut être utilisé pour récupérer des données, en envoyer ou effectuer des modifications.

Gestion des utilisateurs / Accès : Modérateurs & Administrateurs

- /user → Liste des utilisateurs
- /user/new → Création d'un utilisateur
- /user/{id} → Affichage d'un utilisateur
- /user/{id}/edit → Modification d'un utilisateur
- /user/{id} → Suppression d'un utilisateur
- /user/pending → Liste des utilisateurs en attente de validation
- /user/{id}/verify → Validation d'un utilisateur
- /user/{id}/refuse → Refus d'un utilisateur

Gestion des stages / Accès : Utilisateurs authentifiés

- /internship → Liste des stages
- /internship/new → Création d'un stage
- /internship/{id} → Affichage d'un stage
- /internship/{id}/edit → Modification d'un stage
- /internship/{id} → Suppression d'un stage
- /internship/pending → Liste des stages en attente
- /internship/verify/{id} → Validation d'un stage
- /internship/refuse/{id} → Refus d'un stage
- /internship/export-pdf → Export PDF des stages
- /internship/export → Export Excel des stages

Gestion des entreprises / Accès : Utilisateurs authentifiés

- /company → Liste des entreprises
- /company/new → Création d'une entreprise
- /company/{id} → Affichage d'une entreprise
- /company/{id}/edit → Modification d'une entreprise
- /company/{id} → Suppression d'une entreprise
- /company/pending → Liste des entreprises en attente
- /company/verify/{id} → Validation d'une entreprise
- /company/refuse/{id} → Refus d'une entreprise
- /company/export-pdf → Export PDF des entreprises
- /company/export → Export Excel des entreprises

Gestion des écoles / Accès : Administrateurs

- /school → Liste des écoles
- /school/new → Création d'une école
- /school/{id} → Affichage d'une école
- /school/{id}/edit → Modification d'une école
- /school/{id} → Suppression d'une école

Gestion des classes / Accès : Administrateurs

- /grade → Liste des classes
- /grade/new → Création d'une classe
- /grade/{id} → Affichage d'une classe
- /grade/{id}/edit → Modification d'une classe
- /grade/{id} → Suppression d'une classe

Gestion des rapports de visite / Accès : Utilisateurs authentifiés

- /visit-report → Liste des rapports de visite
- /visit-report/{id} → Affichage d'un rapport
- /visit-report/{id}/edit → Modification d'un rapport
- /visit-report/{id} → Suppression d'un rapport

5. Fonctionnalités Avancées

Authentification et Sécurité

- Connexion sécurisée via e-mail/mot de passe avec Symfony Security.
- Gestion des rôles avec Symfony pour restreindre l'accès aux fonctionnalités.
- Vérification email obligatoire
- Protection CSRF sur les formulaires

Recherche avancée dynamique

- Possibilité d'ajouter/supprimer dynamiquement des critères de recherche sur les entreprises et stages.
- Filtrage par nom, ville, statut, date...
- Implémentation en JavaScript côté front et requêtes optimisées côté back.

Exportation des données

Format d'export disponibles :

- Export PDF (mise en page optimisée pour l'impression)
- Export Excel (format XLSX pour manipulation des données)

Fonctionnalités d'export :

- Export PDF :
 - Utilisation de Dompdf pour la génération
 - Support du format A4 paysage
 - Mise en page personnalisée via templates Twig
 - En-têtes de colonnes personnalisables
 - Filtres de données conservés lors de l'export
- Export EXCEL :
 - Utilisation de PhpSpreadsheet
 - Format XLSX compatible avec Excel
 - Colonnes formatées selon le type de données
 - Conservation des filtres de recherche

6. Contraintes et Bonnes Pratiques

- **Sécurité :**
 - Utilisation de Hashing pour les mots de passe.
 - Protection CSRF sur les formulaires.
 - Validation des entrées utilisateur pour éviter les injections SQL/XSS et autres erreurs.

 - **Performances :**
 - Utilisation de Doctrine QueryBuilder pour optimiser les requêtes.
 - Pagination sur les listes de stages et entreprises.

 - **Déploiement :**
 - Hébergement interne à l'établissement.
 - Mise en place d'un environnement de test avant la production.
 - Utilisation de Git pour la gestion des versions.
-

7. Installation du projet Symfony

- **Cloner le dépôt du projet :**
 - git clone <https://github.com/Spitskyyy/stage-direct-final>
 - Ouvrir le projet
- **Installer les dépendances avec Composer :**
 - composer install
- **Configurer l'environnement :**
 - Copier le fichier .env en .env.local
 - Modifier .env.local avec vos informations de connexion à la base de données.
- **Créer la base de données (si non existante) :**
 - php bin/console doctrine:database:create
- **Exécuter les migrations :**
 - php bin/console doctrine:migrations:migrate
- **Faire les fixtures :**
 - php bin/console doctrine:fixtures:load

8. Conclusion

Cette documentation technique couvre les principaux aspects du projet **Stage Direct**. Pour toute mise à jour ou amélioration, se référer au dépôt Git et aux bonnes pratiques de développement Symfony.

Dernière mise à jour : 29/04/2025