

Documentation Technique (Développeur)

Phrase clé : Fournir un guide de développement et d'intégration.

Livrable : Documentation développeur (API, classes, services)

1. Introduction

Ce document décrit l'architecture technique, les fichiers, les API internes et les bonnes pratiques permettant de maintenir et de faire évoluer le système de gestion du **Centre de Formation**.

Le projet repose sur un environnement local **XAMPP** et une base MySQL.

2. Architecture générale

2.1 Technologies

PHP 8+

MySQL (base : *centre_formation*)

HTML / CSS (*style.css*)

Export CSV / Excel (PHP natif)

Serveur local : Apache (XAMPP)

3. Organisation du code

Le projet est découpé en modules fonctionnels correspondant aux pages PHP :

Module / Fichier	Rôle
db.php	Connexion MySQL (PDO ou MySQLi selon version)
header.php	En-tête du site & navigation
footer.php	Pied de page

style.css	Style graphique global
etudiants.php	CRUD étudiants
formations.php	CRUD formations
sessions.php	Gestion des sessions et planification
presences.php	Saisie des présences des étudiants
inscriptions.php	Inscription d'un étudiant à une session
list_formations.php	Liste filtrée des formations
export_emargement.php	Export du tableau des présences

4. Base de données

Base : **centre_formation**

Tables principales :

etudiants (id, nom, prenom, email...)
formations (id, intitule, description...)
sessions (id, formation_id, date_debut, date_fin...)
inscriptions (id, etudiant_id, session_id)
presences (id, session_id, etudiant_id, date, statut)

5. API internes (endpoints PHP)

Fichier	Méthode	Paramètres	Fonction
etudiants.php	POST	nom, prenom, email...	Ajouter un étudiant
etudiants.php	GET	id	Récupérer un étudiant
formations.php	POST	intitule	Ajouter une formation
sessions.php	POST	formation_id, dates	Créer une session
presences.php	POST	session_id, etudiant_id, statut	Enregistrer une présence
export_emargement.php	GET	session_id	Export CSV/Excel

6. Structure du front-end

`style.css` gère l'uniformité des pages (couleurs bleu/white, tableaux, boutons...)

Layout basé sur :

`.container`

`.btn`

`.notice.ok` / `.notice.err`

7. Bonnes pratiques de développement

Utiliser **requêtes préparées** pour éviter les injections SQL

Factoriser les fonctions d'accès à la BD

Séparer la logique métier et l'affichage

Utiliser une structure MVC simplifiée (bonne pratique future)

8. Guide d'intégration

Copier le projet dans :

`C:/xampp/htdocs/centre_formation/`

Importer la BD via phpMyAdmin

Adapter `db.php` si besoin

Lancer :

http://localhost/centre_formation/