Documentation du projet : Gestion de séminaire (PHP / API / JSON)

Contexte

Ce projet a été réalisé dans le cadre de ma formation en BTS SIO option SLAM. Il s'agit d'un site web en PHP permettant de gérer les différentes caractéristiques liées à l'organisation d'un séminaire : étudiants, classe, etc. Le stockage des données se fait dans un fichier JSON, et l'application utilise une API REST pour permettre les opérations de création, lecture, mise à jour et suppression (CRUD) sur ces données.

Objectifs du projet

- Concevoir une application web légère et autonome (pas de base de données SQL).
- Créer une API REST en PHP lisant et écrivant dans un fichier JSON.
- Permettre la gestion complète des étudiant d'une classe (CRUD).
- Favoriser la modularité et la réutilisabilité du code (API indépendante).

Technologies utilisées

- Backend : PHP
- API : REST (GET, POST, PUT, DELETE)
- Format des données : JSON
- Frontend : HTML / CSS
- Outils annexes : Postman (pour les tests d'API)

Fonctionnalités de l'API

Méthode HTTP	URI	Description
GET	/api/étudiants	Récupère tous les étudiants
GET	/api/etudiant?id=1	Récupère un étudiant spécifique
POST	/api/classe	Crée une classe
PUT	/api/etudiant?id=1	Met à jour un étudiant existant
DELETE	/api/etudiant?id=1	Supprime un étudiant

Extrait de code PHP - Lecture (GET)

```
header("Content-Type: application/json");
$data = json_decode(file_get_contents("seminaire.json"), true);
echo json_encode($data);
```

Documentation du projet : Gestion de séminaire (PHP / API / JSON)

Tests réalisés

- Tests des routes avec Postman (vérification du bon fonctionnement de chaque méthode HTTP).
- Vérification de la persistance des données dans le fichier JSON.
- Tests de validation des données (ex : champs obligatoires non vides).

Résultats obtenus

- L'API fonctionne correctement pour les 4 opérations CRUD.
- L'interface HTML permet à un utilisateur de consulter ou ajouter un séminaire facilement.
- Le projet est autonome, portable, et peut évoluer vers un backend avec base SQL ou une application mobile.

Améliorations possibles

- Intégration d'un formulaire HTML pour modifier ou supprimer directement depuis l'interface.
- Ajout d'un système d'authentification (ex : pour sécuriser les accès à l'API).
- Passage vers une base de données MySQL si le volume de données augmente.