# Documentation d’architecture Ishak’School

Table des matières

[Documentation d’architecture Ishak’School 1](#_Toc201672072)

[Présentation rapide 1](#_Toc201672073)

[Vue d’ensemble de l’architecture 1](#_Toc201672074)

[Les conteneurs Docker 2](#_Toc201672075)

[Communication entre services 2](#_Toc201672076)

[Structure base de données 2](#_Toc201672077)

[Environnement de développement 2](#_Toc201672078)

[Sécurité 3](#_Toc201672079)

[Tests & CI 3](#_Toc201672080)

## Présentation rapide

Ishak’School est une application web pédagogique visant à illustrer un système CRUD complet sur 4 entités : students, teachers, courses, enrollments. L’objectif est de manipuler les données via une interface réactive (React) connectée à une API Symfony, le tout déployé dans un environnement conteneurisé (Docker).

## Vue d’ensemble de l’architecture

Une image contenant texte, capture d’écran, diagramme, blanc

Le contenu généré par l’IA peut être incorrect.

## Les conteneurs Docker

|  |  |  |  |
| --- | --- | --- | --- |
| **Conteneur** | **Rôle** | **Ports** | **Image utilisée** |
| **frontend** | Interface utilisateur avec React | 5173 | node:alpine |
| **backend** | API Symfony (php-fpm + API Platform) | 9000 | php:8.2-fpm |
| **nginx** | Reverse proxy HTTP/HTTPS | 80, 443 | nginx:alpine |
| **db** | Base de données MariaDB | 3306 | mariadb:11 |
| **phpmyadmin** | Interface BDD | 8080 | phpmyadmin/phpmyadmin |

## Communication entre services

|  |  |  |  |
| --- | --- | --- | --- |
| **Source** | **Destination** | **Protocole** | **Description** |
| navigateur | nginx | HTTP | Proxy de toutes les requêtes |
| nginx | frontend | HTTP | Routes statiques React |
| nginx | backend | HTTP | API (/api/...) vers Symfony |
| backend | db | TCP | Connexion à la base de données |
| phpmyadmin | db | TCP | Interface graphique BDD |
| frontend | backend (via nginx) | HTTP | Requêtes via axios/fetch (API) |

## Structure base de données

4 entités principales :

* **Student** : nom, prénom, date de naissance, email
* **Teacher** : nom, prénom, email
* **Course** : titre, description, FK vers Teacher
* **Enrollment** : FK vers Student & Course, date d’inscription

Toutes les relations sont en ManyToOne, les id sont auto-incrémentés.

## Environnement de développement

* Lancement par docker compose up --build
* Base de données initialisée avec ishak + mot de passe root
* Les entités Symfony exposent automatiquement des endpoints RESTful via API Platform
* React utilise Axios pour communiquer avec /api/... (proxifié via Nginx)
* Données inspectables via PhpMyAdmin (localhost:8080)

## Sécurité

* API sécurisée par JWT via LexikJWTAuthenticationBundle
* CORS configuré pour n’accepter que les appels du frontend
* Données validées côté Symfony (Validator)

## Tests & CI

* Tests React via Vitest
* Tests Symfony via PHPUnit
* Lint JS et PHP
* GitHub Actions exécute les tests à chaque push ou pull request