

KundApp

Résumé du Cahier des Charges Projet

réalisé dans le cadre du titre professionnel Concepteur Développeur d'Application 2024-2025

Présenté par *Jacques Rouillet*

Introduction

KundApp est une plateforme web conçue pour la gestion de cours de bien-être (kundalini, yoga, méditation). Elle a pour but de connecter des professeurs indépendants avec des élèves. Le projet s'appuie sur une architecture microservices pour assurer la modularité et l'évolutivité. L'ensemble des services back-end est développé avec Spring Boot, tandis que l'interface est rendue côté serveur avec Thymeleaf. L'écosystème est entièrement conteneurisé et orchestré via Docker Compose. Le microservice ms-webapp agit comme unique point d'entrée, gérant l'interface utilisateur et la sécurité basée sur les sessions.

Objectifs

- Gestion des cours – CRUD complet pour les sessions créées par les professeurs.
- Système d'inscription – Processus simple pour les élèves, basé sur un système de crédits.
- Gestion des utilisateurs – Administration centralisée des profils (élèves, professeurs, admin).
- Interface utilisateur – Navigation claire et adaptée à chaque rôle.
- Notifications automatisées – Envoi d'emails pour les événements clés (inscription, annulation, etc.).
- Sécurité & architecture – Architecture microservices découplée, sécurisée par Spring Security (sessions & CSRF), et entièrement conteneurisée avec Docker.

Fonctionnalités

- Portail de connexion – Inscription et authentification des utilisateurs.
- Tableaux de bord par rôle – Vues dédiées pour l'élève, le professeur et l'administrateur.
- Catalogue de cours – Affichage des sessions disponibles avec leurs détails (sujet, date, places restantes).
- Gestion des sessions (Professeur) – Création, modification et annulation de ses propres cours.
- Réservation (Élève) – Inscription et désinscription à un cours, avec déduction et restitution automatique des crédits.
- Espace administrateur – Supervision complète des utilisateurs (gestion des crédits, suppression) et de toutes les sessions.

Rôles et Permissions

- Visiteur non authentifié – Accès : pages de connexion et d'inscription.
- Client (Élève) authentifié – Actions : consulter et s'inscrire aux cours, gérer son profil, voir ses cours à venir.
- Professeur authentifié – Actions : créer et gérer ses propres cours, voir la liste des participants.
- Administrateur – Supervision complète : gestion de tous les utilisateurs, de tous les cours et des crédits.

Architecture Technique & Technologies

- Front-end : Thymeleaf, HTML, CSS, JavaScript natif.
- Back-end (microservices Spring Boot) : ms-webapp (façade web et orchestration), ms-identity (utilisateurs & authentification), ms-course-mgmt (logique des cours), ms-notification (envoi d'emails), ms-discovery (Eureka), ms-configserver (configuration centralisée).
- Bases de données : MySQL (données métier), MongoDB (notifications).
- Déploiement & orchestration : Docker Compose pour l'environnement de développement et de production.
- Communication inter-services : Spring Cloud OpenFeign.

Tests & Qualité

Tests unitaires avec JUnit 5 et Mockito sur les couches service des microservices pour valider la logique métier de manière isolée.

Schéma architectural

[UI (Thymeleaf/JS)] → [ms-webapp (Gateway & Sécurité Session)] → {microservices (ms-identity, ms-course-mgmt, ms-notification)} → {MySQL / MongoDB}