SOUTENANCE SERVICE

Documentation Technique

Portail de Suivi du Doctorat

Architecture Microservices

Information	Valeur
Microservice	soutenance-service
Port	8083
Base de données	MySQL - soutenance_db
Framework	Spring Boot 3.x
Communication	Kafka (Events) + Feign (REST)
Date	01/01/2026

TABLE DES MATIÈRES

1. PRÉSENTATION GÉNÉRALE DU MICROSERVICE

1.1 Rôle Fonctionnel

Le **soutenance-service** est le microservice responsable de la gestion complète du processus de soutenance de thèse : de la soumission initiale par le doctorant jusqu'à l'enregistrement du résultat final, en passant par la validation des prérequis, la composition du jury et la planification.

■ **En termes simples :** Ce service gère tout le parcours de la soutenance. C'est comme un chef d'orchestre qui coordonne le doctorant, le directeur de thèse, l'administration et les membres du jury pour que la soutenance se déroule dans les règles.

1.2 Responsabilités Principales

- **Soumission de Demande** : Le doctorant soumet sa demande avec manuscrit et rapport anti-plagiat
- **Validation des Prérequis** : Vérification des publications, conférences et heures de formation
- **Gestion du Jury** : Composition, proposition et validation du jury de soutenance
- **Planification** : Fixation de la date, heure et lieu de soutenance
- **Résultats** : Enregistrement de la note, mention et félicitations du jury
- **Notifications** : Publication d'événements Kafka pour informer les parties prenantes

1.3 Prérequis pour Soutenir (Cahier des Charges)

Prérequis	Minimum Requis	Description
Publications Q1/Q2	≥ 2	Articles publiés dans des revues indexées
Conférences	≥ 2	Communications dans des conférences scientifiques
Heures de Formation	≥ 200h	Formations doctorales cumulées

■ **Pourquoi ces prérequis ?** Ils garantissent que le doctorant a acquis les compétences nécessaires pendant sa thèse : capacité à publier, à communiquer ses recherches, et formation continue dans son domaine.

2. ARCHITECTURE INTERNE

2.1 Organisation du Code

Couche	Package	Rôle
Controllers	controllers	Endpoints REST (API)
Services	services	Logique métier et workflow
Repositories	repositories	Accès aux données JPA
Entities	entities	Modèles de données
Events	events	Événements Kafka
Clients	clients	Communication Feign avec user-service
Config	config	Configuration Kafka
Enums	enums	Statuts et rôles

2.2 Communication avec les Autres Services

Service	Type	Usage
user-service	REST (Feign)	Récupérer infos doctorant/directeur
notification-service	Kafka (Async)	Envoyer emails (création, planification, invitations jury)

■ **Pourquoi Feign + Kafka ?** Feign pour les données synchrones (on a besoin de l'info immédiatement), Kafka pour les notifications asynchrones (l'email peut partir quelques secondes plus tard, pas bloquant).

3. FICHIERS DE CONFIGURATION

3.1 SoutenanceServiceApplication.java

Rôle : Point d'entrée du microservice.

Annotations : @SpringBootApplication + @EnableFeignClients (pour appeler user-service)

3.2 application.properties

Paramètre	Valeur	Description
server.port	8083	Port d'écoute
spring.datasource.url	jdbc:mysql://.../soutenance_db	Base de données MySQL
spring.kafka.bootstrap-servers	localhost:9092	Broker Kafka
spring.servlet.multipart.max-file-size	50MB	Taille max fichier (manuscrit)
spring.servlet.multipart.max-request-size	60MB	Taille max requête totale
feign.circuitbreaker.enabled	true	Résilience si user-service down

■ **Pourquoi 50MB pour les fichiers ?** Un manuscrit de thèse peut faire plusieurs dizaines de Mo (images, graphiques). 50MB est une limite raisonnable pour éviter les abus tout en permettant les vrais manuscrits.

3.3 KafkaTopics.java

Rôle : Constantes définissant les noms des topics Kafka utilisés.

Constante	Topic	Déclencheur
SOUTENANCE_CREATED	soutenance-created	Nouvelle demande soumise
SOUTENANCE_STATUS_CHANGED	soutenance-status-changed	Tout changement de statut
SOUTENANCE_SCHEDULED	soutenance-scheduled	Date de soutenance fixée
JURY_INVITATION	jury-invitation	Invitation envoyée à un membre du jury

4. MODÈLE DE DONNÉES

4.1 Énumérations

4.1.1 StatutSoutenance.java

Rôle : Définit les 8 états possibles d'une soutenance dans le workflow.

Statut	Description	Qui agit ensuite ?
BROUILLON	En cours de saisie	Doctorant
SOUMIS	Demande soumise	Directeur
PREREQUIS_VALIDES	Prérequis vérifiés OK	Admin
AUTORISEE	Autorisée par admin	Directeur
JURY_PROPOSE	Jury proposé	Admin
PLANIFIEE	Date fixée	Tous (attente jour J)
TERMINEE	Soutenance effectuée	Fin
REJETEE	Refusée	Fin

4.1.2 RoleJury.java

Rôle : Définit les différents rôles des membres du jury.

Rôle	Description
PRESIDENT	Préside la soutenance et annonce le résultat
RAPPORTEUR	Évalue le manuscrit et rédige un rapport (min. 2 requis)
EXAMINATEUR	Pose des questions lors de la soutenance
DIRECTEUR_THESE	Membre de droit, ne vote pas

■ **Composition minimale d'un jury :** 1 Président + 2 Rapporteurs + Examinateurs (optionnel). Le directeur de thèse assiste mais ne participe pas à la décision.

4.2 Entités JPA

4.2.1 Soutenance.java (Entité Principale)

Table : soutenances

Champ	Type	Description
id	Long (PK)	Identifiant unique
doctorantId	Long	Référence vers user-service
directeurId	Long	Référence vers le directeur
titreThese	String	Titre de la thèse
statut	StatutSoutenance	État actuel dans le workflow
prerequis	Prerequis (Embedded)	Publications, conférences, heures
cheminManuscrit	String	Chemin du fichier PDF
membresJury	List<MembreJury>	Composition du jury
dateSoutenance	LocalDate	Date prévue
noteFinale	Double	Note obtenue (0-20)
mention	String	Très Honorable, Honorable...
felicitationsJury	Boolean	Félicitations accordées ?

Champs @Transient : doctorantInfo et directeurInfo (UserDTO) - non persistés, enrichis dynamiquement via Feign lors de la récupération.

4.2.2 Prerequis.java (@Embeddable)

Rôle : Objet embarqué dans Soutenance pour stocker les prérequis.

Champs : nombreArticlesQ1Q2, nombreConferences, heuresFormation, prerequisValides

Méthode clé : verifierPrerequisMinimaux() → true si ≥2 articles, ≥2 conférences, ≥200h

■ **@Embeddable, c'est quoi ?** Les champs de Prerequis sont stockés directement dans la table 'soutenances' (pas de table séparée). C'est une composition : un Prerequis n'existe pas sans sa Soutenance.

4.2.3 MembreJury.java

Table : membres_jury

Champs : id, soutenance (FK), role, nom, prenom, email, etablissement, grade, rapportSoumis, avisFavorable, commentaireRapport

Relation : @ManyToOne avec Soutenance (@JsonIgnore pour éviter boucle infinie)

4.2.4 JuryDisponible.java

Table : jurys_disponibles

Rôle : Catalogue de personnes pouvant être sélectionnées comme membres de jury. Le directeur pioche dans cette liste lors de la composition du jury.

5. COUCHE REPOSITORY

5.1 SoutenanceRepository.java

Méthode	Description
findByDoctorantId()	Soutenances d'un doctorant
findByDirecteurId()	Soutenances supervisées par un directeur
findByStatut()	Filtrage par statut
findByDoctorantIdAndStatut()	Soutenance unique d'un doctorant à un statut donné

5.2 MembreJuryRepository.java

Méthodes : findBySoutenanceId(), findBySoutenanceIsNull() (jurys disponibles), findByRole()

5.3 JuryDisponibleRepository.java

Méthodes : findByRole(), findByEmail(), existsByEmail()

6. ÉVÉNEMENTS KAFKA

Le service publie des événements vers Kafka pour déclencher des notifications dans le notification-service.

6.1 BaseEvent.java

Rôle : Classe abstraite parente avec eventId (UUID), timestamp, source.

6.2 Événements Spécialisés

Événement	Topic	Données Clés
SoutenanceCreatedEvent	soutenance-created	doctorantEmail, sujetThese, directeurEmail
SoutenanceStatusChangedEvent	soutenance-status-changed	oldStatus, newStatus, dateSoutenance, lieu
JuryInvitationEvent	jury-invitation	membreJuryEmail, roleJury, dateSoutenance

■ **Que fait notification-service avec ces events ?** Il les consomme et envoie des emails : "Votre demande a été soumise", "Vous êtes invité comme rapporteur", "Soutenance planifiée le..."

7. COMMUNICATION INTER-SERVICES (FEIGN)

7.1 UserServiceClient.java

Annotation : @FeignClient(name = "USER-SERVICE", fallback = UserServiceClientFallback.class)

Méthode	Endpoint Appelé	Usage
getUserById(Long id)	GET /api/users/{id}/dto	Récupérer infos doctorant/directeur
getUserByUsername(String)	GET /api/users/username/{}/dto	Récupérer par matricule

7.2 UserServiceClientFallback.java

Rôle : Si user-service est indisponible, retourne un UserDTO par défaut (nom: "INCONNU", email: "unavailable@service.com").

■■ **Circuit Breaker Pattern :** Le fallback empêche le service de planter si user-service est down. La soutenance peut être créée, mais les infos utilisateur seront manquantes temporairement.

8. COUCHE SERVICE (LOGIQUE MÉTIER)

8.1 SoutenanceServiceImpl.java

Rôle : Implémente toute la logique métier du workflow de soutenance.

Catégorie	Méthodes
CRUD	createSoutenance(), updateSoutenance(), getSoutenanceById()
Soumission	soumettreDemande() - Upload manuscrit + rapport
Étape 1: Directeur	validerPrerequisDirecteur(), rejeterParDirecteur()
Étape 2: Admin	autoriserSoutenance()
Étape 3: Jury	ajouterMembreJury(), proposerJury(), validerJury(), refuserJury()
Étape 4: Planif	planifierSoutenance(), proposerDateSoutenance()
Étape 5: Résultat	enregistrerResultat()
Utilitaires	enrichirAvecInfosUtilisateurs(), publishStatusChangedEvent()

8.2 SoutenanceEventPublisher.java

Rôle : Publie les événements vers Kafka via KafkaTemplate.

Méthodes : publishSoutenanceCreated(), publishSoutenanceStatusChanged(), publishSoutenanceScheduled(), publishJuryInvitation(), publishAllJuryInvitations()

9. COUCHE CONTROLLER (API REST)

9.1 SoutenanceController.java

Base URL : /api/soutenances

Méthode	Endpoint	Description
POST	/soumettre (multipart)	Soumettre demande avec fichiers
GET	/	Lister toutes les soutenances
GET	/{id}	Détails d'une soutenance
GET	/doctorant/{id}	Soutenances d'un doctorant
GET	/directeur/{id}	Soutenances d'un directeur
PUT	/{id}/valider-prerequis	Directeur valide prérequis
PUT	/{id}/autoriser	Admin autorise
POST	/{id}/jury	Ajouter membre au jury
PUT	/{id}/proposer-jury	Directeur propose le jury
PUT	/{id}/valider-jury	Admin valide le jury
PUT	/{id}/planifier	Admin planifie date/lieu
PUT	/{id}/resultat	Enregistrer résultat final
GET	/jury/disponibles	Liste des jurys disponibles

9.2 SoutenanceFileController.java

Base URL : /api/soutenances/files

Endpoints : GET /{filename} et GET /download?path=... pour télécharger manuscrits et rapports.

Fonctionnalités : Extraction du nom de fichier, détection MIME automatique (PDF, DOC, images).

10. WORKFLOW COMPLET DE SOUTENANCE

```
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■■
■ DOCTORANT ■
■ ■■→ Soumet demande (manuscrit + rapport anti-plagiat) ■
■ Statut: SOUMIS ■
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■■
▼
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■■
■ DIRECTEUR ■
■ ■■→ Vérifie prérequis (publications, conférences, heures) ■
■ ✓ Valide → Statut: PREREQUIS_VALIDES ■
■ ✗ Rejette → Statut: REJETEE ■
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■■
▼
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■■
■ ADMIN ■
■ ■■→ Autorise administrativement la soutenance ■
■ Statut: AUTORISEE ■
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■■
▼
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■■
■ DIRECTEUR ■
■ ■■→ Propose le jury (≥3 membres: 1 président + 2 rapporteurs) ■
■ Statut: JURY_PROPOSE ■
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■■
▼
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■■
■ ADMIN ■
■ ■■→ Valide le jury + Fixe date/heure/lieu ■
■ Statut: PLANIFIEE ■
■ ■ Invitations envoyées aux membres du jury ■
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■■
▼
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■■
■ JOUR J - SOUTENANCE ■
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■■
▼
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■■
■ ADMIN ■
■ ■■→ Enregistre le résultat (note, mention, félicitations) ■
■ Statut: TERMINEE ■
```

```
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
■■
```

■ **Transitions possibles :** À chaque étape, un rejet peut renvoyer à l'état précédent (ex: jury refusé → retour à AUTORISEE pour reproportionner). Le doctorant et le directeur sont notifiés par email à chaque changement.

11. QUESTIONS / RÉPONSES POTENTIELLES

Q1 : Pourquoi utiliser @Embeddable pour Prerequis plutôt qu'une table séparée ?

R : Les prérequis n'ont pas d'existence propre sans soutenance. @Embeddable évite une jointure et simplifie le modèle. Les 3 champs sont stockés directement dans la table soutenances.

Q2 : Comment éviter la boucle infinie JSON entre Soutenance et MembreJury ?

R : On utilise @JsonIgnore sur la relation soutenance dans MembreJury, et @JsonIgnoreProperties sur la liste membresJury dans Soutenance. Ainsi, la sérialisation s'arrête proprement.

Q3 : Pourquoi les fichiers sont stockés localement plutôt que dans la BDD ?

R : Les manuscrits peuvent faire 50MB+. Stocker en BDD (BLOB) dégraderait les performances. On stocke le fichier sur disque et on garde juste le chemin en BDD. En production, on utiliserait un stockage cloud (S3, MinIO).

Q4 : Que se passe-t-il si user-service est indisponible lors d'une soumission ?

R : Grâce au fallback Feign, la soutenance est créée avec des infos utilisateur par défaut. L'événement Kafka est publié avec les données disponibles. Le service reste fonctionnel.

Q5 : Pourquoi avoir une table jurys_disponibles séparée ?

R : C'est un catalogue réutilisable. Les mêmes experts peuvent être sollicités pour plusieurs soutenances. Le directeur pioche dans ce catalogue plutôt que de ressaisir les infos à chaque fois.

Q6 : Comment garantir qu'un jury est complet avant planification ?

R : La méthode proposerJury() vérifie que membresJury.size() >= 3. Les méthodes utilitaires juryEstComplet() et tousLesRapportsFavorables() dans l'entité Soutenance ajoutent des contrôles.

12. CONCLUSION

Le **soutenance-service** est le cœur du processus de fin de thèse, orchestrant un workflow complexe impliquant doctorants, directeurs, administration et membres de jury.

Points Forts de l'Architecture :

- **Workflow Multi-Étapes** : 8 statuts couvrant tout le cycle de vie de la soutenance
- **Upload de Fichiers** : Support des manuscrits jusqu'à 50MB avec stockage organisé
- **Gestion du Jury** : Catalogue réutilisable + composition flexible
- **Prérequis Embarqués** : Validation automatique des conditions de soutenance
- **Événements Kafka** : Notifications asynchrones pour toutes les parties prenantes
- **Résilience** : Fallback Feign si user-service indisponible
- **API REST Complète** : Endpoints pour chaque étape et chaque acteur

Valeur dans l'Architecture Globale :

Ce microservice représente l'aboutissement du parcours doctoral. Il s'intègre avec user-service (infos utilisateurs via Feign), notification-service (emails via Kafka), et respecte les règles métier du cahier des charges (prérequis minimaux, composition du jury).

■ **En résumé :** soutenance-service transforme une demande de soutenance en un événement académique structuré, avec traçabilité complète et notifications automatiques à chaque étape.