

# ■ PORTAIL DOCTORAT

## Résumé Complet du Projet

### Document de Contexte pour Continuation

Date	26 Décembre 2025
Version	1.0
Statut	En développement
Technologies	Angular 17 + Spring Boot 3 (Microservices)

## ■ Table des Matières

- 1. Vue d'Ensemble du Projet
- 2. Architecture Technique
- 3. Structure des Fichiers Frontend
- 4. Structure des Fichiers Backend
- 5. Modèles de Données (Alignement Backend/Frontend)
- 6. Workflow des Soutenances (CDC)
- 7. Pages Implémentées
- 8. Travail Restant à Faire
- 9. Fichiers Clés et Leur Emplacement
- 10. Prompt de Continuation

# 1. Vue d'Ensemble du Projet

Le **Portail Doctorat** est une application web permettant de gérer l'intégralité du parcours doctoral, depuis la candidature initiale jusqu'à la soutenance de thèse. L'application est construite avec une architecture microservices côté backend (Spring Boot 3) et une application Angular 17 côté frontend.

## Objectifs principaux :

- Dématérialiser les processus administratifs du doctorat
- Suivre le parcours des doctorants (inscriptions, prérequis, dérogations)
- Gérer les demandes de soutenance avec workflow complet
- Fournir des tableaux de bord adaptés à chaque acteur (Admin, Directeur, Doctorant)

## Rôles utilisateurs :

Rôle	Code	Description
Candidat	CANDIDAT	Personne souhaitant s'inscrire en doctorat
Doctorant	DOCTORANT	Candidat validé, inscrit en thèse
Directeur de Thèse	DIRECTEUR_THESE	Encadrant scientifique
Administrateur	ADMIN	Gestionnaire de la scolarité doctorale

## 2. Architecture Technique

### Stack Technologique :

Couche	Technologie
Frontend	Angular 17+ (Standalone Components, Signals)
Backend	Spring Boot 3 (Microservices)
Base de données	MySQL / PostgreSQL
Authentification	JWT (JSON Web Tokens)
Architecture	Microservices avec API Gateway + Eureka
Messaging	Apache Kafka (événements)

### Microservices Backend :

- **eureka-server** - Service Discovery
- **config-server** - Configuration centralisée
- **api-gateway** - Point d'entrée unique
- **user-service** - Gestion utilisateurs et authentification JWT
- **inscription-service** - Candidatures, inscriptions, campagnes
- **soutenance-service** - Demandes de soutenance, jury, prérequis
- **document-service** - Génération PDF (attestations, PV)
- **notification-service** - Emails et notifications (Kafka consumer)

### 3. Structure des Fichiers Frontend

Le frontend est organisé selon une architecture modulaire avec des composants standalone Angular 17.

```
src/app/  
  ■■■ core/  
  ■ ■■■ guards/ (auth.guard.ts, role.guard.ts)  
  ■ ■■■ interceptors/ (auth.interceptor.ts)  
  ■ ■■■ models/ (soutenance.model.ts, user.model.ts, etc.)  
  ■ ■■■ services/ (soutenance.service.ts, auth.service.ts, etc.)  
  ■■■ features/  
  ■ ■■■ admin/  
  ■ ■ ■■■ dashboard/  
  ■ ■ ■■■ derogation-management/  
  ■ ■ ■■■ soutenance-detail/ ← NOUVEAU  
  ■ ■ ■■■ soutenance-list/ ← MIS À JOUR  
  ■ ■ ■■■ user-management/  
  ■ ■ ■■■ admin.routes.ts  
  ■ ■■■ directeur/  
  ■ ■ ■■■ soutenances/  
  ■ ■ ■■■ validation/  
  ■ ■■■ inscriptions/  
  ■ ■■■ soutenances/  
  ■■■ shared/  
  ■ ■■■ components/  
  ■ ■■■ main-layout/  
  ■■■ environments/
```

## 4. Structure des Fichiers Backend

```
portail-doctorat-backend/  
  ■■■■ eureka-server/  
  ■■■■ config-server/  
  ■■■■ api-gateway/  
  ■■■■ user-service/  
  ■ ■■■■ entities/User.java  
  ■ ■■■■ dto/UserDTO.java  
  ■ ■■■■ controllers/AuthController.java, UserController.java  
  ■ ■■■■ security/JwtService.java  
  ■■■■ inscription-service/  
  ■ ■■■■ entities/Inscription.java, Campagne.java, Derogation.java  
  ■ ■■■■ controllers/InscriptionController.java  
  ■■■■ soutenance-service/  
  ■ ■■■■ entities/  
  ■ ■ ■■■■ Soutenance.java  
  ■ ■ ■■■■ MembreJury.java  
  ■ ■ ■■■■ Prerequis.java (Embeddable)  
  ■ ■■■■ enums/StatutSoutenance.java, RoleJury.java  
  ■ ■■■■ controllers/SoutenanceController.java  
  ■■■■ document-service/  
  ■ ■■■■ services/PdfGeneratorService.java  
  ■■■■ notification-service/  
  ■■■■ consumers/SoutenanceEventConsumer.java
```

## 5. Modèles de Données - Alignement Backend/Frontend

**IMPORTANT** : Le frontend doit utiliser les mêmes noms de propriétés que le backend. Voici le mapping correct :

Backend (Java)	Frontend (TypeScript)	Notes
titreThese	titreThese	PAS sujetThese
membresJury	membresJury	PAS jury
doctorantInfo (UserDTO)	doctorantInfo	PAS doctorantNom
directeurInfo (UserDTO)	directeurInfo	PAS directeurNom
MembreJury.role	role	PAS roleJury
MembreJury.rapportSoumis	rapportSoumis	PAS rapportRecu
MembreJury.avisFavorable	avisFavorable	PAS rapportFavorable
Prerequis.prerequisValides	prerequisValides	Boolean
cheminManuscrit	cheminManuscrit	Chemin vers le PDF
cheminRapportAntiPlagiat	cheminRapportAntiPlagiat	Chemin vers le PDF

### Modèle Soutenance (TypeScript) :

```
interface Soutenance {  
  id: number;  
  doctorantId: number;  
  directeurId: number;  
  titreThese: string;  
  statut: StatutSoutenance;  
  prerequis?: Prerequis;  
  membresJury: MembreJury[];  
  dateSoutenance?: string;  
  heureSoutenance?: string;  
  lieuSoutenance?: string;  
  cheminManuscrit?: string;  
  cheminRapportAntiPlagiat?: string;  
  doctorantInfo?: UserDTO;  
  directeurInfo?: UserDTO;  
  createdAt: string;  
}
```

## 6. Workflow des Soutenances (CDC)

### Étapes du processus :

Étape	Statut	Acteur	Action
1	BROUILLON	Doctorant	Prépare sa demande (titre, manuscrit, rapport anti-plagiat)
2	SOU MIS	Doctorant	Soumet la demande
3	PREREQUIS_VALIDES	Admin	Valide les prérequis (2 publi, 2 conf, 200h)
4	JURY_PROPOSE	Directeur	Propose la composition du jury
5	PLANIFIEE	Admin	Valide le jury, fixe date/lieu
6	AUTORISEE	Admin	Autorise la soutenance
7	TERMINEE	Admin	Enregistre le résultat (mention)

### Prérequis obligatoires (CDC) :

- ✓ 2 Publications dans des revues Q1/Q2
- ✓ 2 Participations à des conférences
- ✓ 200 heures de formation
- ✓ Durée  $\leq 3$  ans (ou dérogation approuvée)
- ✓ Upload du manuscrit (PDF)
- ✓ Upload du rapport anti-plagiat (PDF)

### Composition du Jury :

Rôle	Nombre	Critères
Président	1	PES, externe à l'établissement
Rapporteurs	2	Externes, spécialistes du domaine
Examineurs	2-3	Internes ou externes
Directeur	1	Participe mais ne vote pas



## 7. Pages Implémentées

### Pages Admin :

Page	Route	Statut	Description
Dashboard	/admin/dashboard	■ Fait	Tableau de bord admin
Utilisateurs	/admin/users	■ Fait	Gestion des utilisateurs
Dérogations	/admin/derogations	■ Fait	Gestion des dérogations
Liste Soutenances	/admin/soutenances	■ Fait	Liste avec filtres (onglets)
Détail Soutenance	/admin/soutenances/:id	■ Fait	Détails + actions

### Fonctionnalités de la page Liste Soutenances :

- **5 onglets de filtrage** : Toutes, À examiner (SOU MIS), Jury à valider (JURY\_PROPOSE), Planifiées, Terminées
- **Cartes avec infos** : Doctorant, Directeur, Titre thèse, Prérequis
- **Documents** : Icônes cliquables pour télécharger Manuscrit et Rapport Anti-Plagiat
- **Actions** : Bouton 'Voir détails' navigue vers /admin/soutenances/:id
- **Recherche** : Filtre par nom ou sujet

## 8. Travail Restant à Faire

### Actions non implémentées :

Fonctionnalité	Priorité	Notes
Implémenter validerPrerequis()	Haute	Appeler le service backend
Implémenter rejeterDemande()	Haute	Avec motif de rejet
Modal planification soutenance	Haute	Date, heure, lieu
Modal assignation directeur	Moyenne	Lors de validation admin
Génération PV (PDF)	Moyenne	Appeler document-service
Génération Attestation (PDF)	Moyenne	Appeler document-service
Page campagnes	Basse	Composant à créer
Page candidatures	Basse	Composant à créer

### Endpoints backend à connecter :

- GET /soutenances/{id} - Récupérer une soutenance
- PUT /soutenances/{id}/valider-prerequis - Valider prérequis
- PUT /soutenances/{id}/rejeter - Rejeter une demande
- PUT /soutenances/{id}/planifier - Planifier (date, lieu)
- PUT /soutenances/{id}/autoriser - Autoriser
- PUT /soutenances/{id}/terminer - Marquer terminée
- POST /documents/pv - Générer PV
- POST /documents/attestation - Générer attestation

## 9. Fichiers Clés et Leur Emplacement

Fichier	Emplacement
soutenance.model.ts	src/app/core/models/
soutenance.service.ts	src/app/core/services/
soutenance-list.component.ts	src/app/features/admin/soutenance-list/
soutenance-detail.component.ts	src/app/features/admin/soutenance-detail/
admin.routes.ts	src/app/features/admin/
Soutenance.java	soutenance-service/src/.../entities/
MembreJury.java	soutenance-service/src/.../entities/
Prerequis.java	soutenance-service/src/.../entities/
SoutenanceController.java	soutenance-service/src/.../controllers/

## 10. Prompt de Continuation

Copie ce prompt dans un nouveau chat pour continuer le travail :

Je travaille sur un projet "Portail Doctorat" (Angular 17 + Spring Boot 3 Microservices).

CONTEXTE IMPORTANT :

- Le CDC (Cahier des Charges) est dans le fichier PORTAIL\_DOCTORAT\_SPECIFICATION.md
- Le backend utilise ces noms de propriétés :
  - \* titreThese (pas sujetThese)
  - \* membresJury (pas jury)
  - \* doctorantInfo/directeurInfo (pas doctorantNom/directeurNom)
  - \* MembreJury.role (pas roleJury)
  - \* cheminManuscrit, cheminRapportAntiPlagiat pour les documents

FICHIERS DÉJÀ IMPLÉMENTÉS :

- soutenance.model.ts (modèle aligné avec backend)
- soutenance.service.ts (avec getSoutenanceById, validerPrerequis, etc.)
- soutenance-list.component.ts (liste avec 5 onglets de filtrage)
- soutenance-detail.component.ts (page de détails)
- admin.routes.ts (routes /admin/soutenances et /admin/soutenances/:id)

CE QUI RESTE À FAIRE :

1. Implémenter les actions réelles (validerPrerequis, rejeter, planifier, etc.)
2. Créer les modals pour planification et assignation directeur
3. Connecter la génération de PDF (document-service)
4. Créer les pages manquantes (campagnes, candidatures)

Peux-tu m'aider à continuer ?