

Intégration Frontend Angular avec Backend Spring Boot

Contexte du projet :

Vous avez développé l'application MusicStream (frontend Angular) pour la gestion de musique locale. L'objectif maintenant est de créer le backend REST API correspondant en utilisant Spring Boot avec les mêmes technologies et patterns adoptés jusqu'à maintenant

Objectifs du Projet

1. Import et Structure

- Importer le projet Angular MusicStream dans un nouveau repository
- Créer une architecture monorepo ou séparée (frontend + backend)
- Organiser la structure : /frontend et /backend

2. Connexion Backend → Frontend

- Configurer CORS dans Spring Boot pour autoriser les requêtes Angular
- Adapter les endpoints REST pour correspondre aux besoins du frontend

3. Services Angular → API REST

- Créer des services Angular pour consommer l'API Spring Boot
- Remplacer le stockage local (IndexedDB) par des appels API REST et une base de données de votre choix

4. Fonctionnalités Croisées

- Adapter les fonctionnalités Angular aux endpoints Spring Boot

Architecture Technique

1. Frontend (Angular)

- Garder les Services nécessaires
- Ajouter :
NgRx pour le State Management (Store, Actions, Effects, Selectors)
Interceptors et Guards (si pertinents)

2. Backend (Spring Boot)

- CORS Configuration : autoriser localhost:4200
- Endpoints nécessaires
- Couches nécessaires avec la bonne démarche habituelle
- Tâches de devops : CI/CD, Docker (backEnd et frontEnd), ...
- Tests unitaires côté backEnd et frontEnd

3. Base de Données

- Choix libre : SQL ou NoSQL

Bonus (Optionnel)

- Implémenter la logique ADMIN et USER avec une page d'authentification
- Intégration API de lyrics pour afficher les paroles d'une chanson en cours

Livrables :

- Lien GitHub pour le code source
- Lien Jira
- Diagramme de classe
- Fichier README.md

Modalités pédagogiques :

- Projet individuel
- Durée du projet : 10 jours
- du 19/01/26 à 30/01/26
- Deadline d'envoi des rendus : 30/01/26 avant minuit

Critères de performance

- L'intégration frontend/backend doit être fonctionnelle avec communication API REST
- L'architecture Angular avec NgRx, services et lazy loading doit être respectée
- L'architecture Spring Boot en couches (Controller, Service, Repository, DTO) doit être respectée
- Les endpoints REST doivent être opérationnels avec gestion d'erreurs HTTP
- Le code doit être propre et respecter les standards TypeScript/Angular et Java/Spring Boot
- La configuration CORS et Docker doit être correctement implémentée