

Mes Tâches

Rapport du Projet Todo Fullstack 2025

Angular + Spring Boot + PostgreSQL

Résumé

Application Todo fullstack avec design moderne (Material v20), authentification JWT, filtrage, recherche, tri, drag & drop, édition inline et actions groupées, le tout filtré par utilisateur.

CHOU Naima — ILIA2

November 30, 2025

Vue d'ensemble

Objectif: fournir une application de tâches attractive et professionnelle, avec une architecture robuste et une UX moderne, responsive, et un thème violet cohérent en light/dark.

Backend (Spring Boot, Security JWT, JPA/Hibernate, PostgreSQL)

- Entités: Todo, User (relation ManyToOne).
- Sécurité: JWT stateless; CORS pour `http://localhost:4200`.
- Accès filtré par utilisateur authentifié.

Frontend (Angular Standalone, Material v20, CDK Drag & Drop)

- Pages: Login (auth), Todos (liste, filtres, DnD, édition).
- Intercepteur JWT; thèmes personnalisés light/dark (violet).

Fonctionnalités Implémentées

Authentification

- Inscription et connexion avec JWT; stockage du token côté client.
- Intercepteur HTTP: injection du header **Authorization**, redirection auto vers Login si 401/403.

Gestion des Todos

- CRUD complet: création, édition inline, suppression.
- Classement par ordre, priorité, date de création.

Filtrage, Recherche, Tri

- Filtre `completed=true|false` via paramètres.
- Recherche par titre (`titleContains`), tri par `orderIndex/createdAt/priority`, direction `asc|desc`.
- Segmented control (tous/actifs/complétés) et KPIs (Total, Actifs, Complétés).

Drag & Drop et Réordonnement

- CDK DnD avec poignée; persistance via PUT `/api/todos/reorder`.

Actions Groupées

- **Tout compléter** (POST `/api/todos/complete-all`).
- **Supprimer les complétés** retiré de l'UI, endpoint maintenu.

Design & Thème

- Thème Material v20 personnalisé: **primary** violet, **tertiary** vert (harmonie light/dark).
- AppBar avec titre **Mes Tâches** grand, dégradé violet, bouton **Déconnexion**, bascule de thème.
- Overrides CSS pour `mat-card`, `mat-form-field`, `mat-select`, icônes, boutons outlined.

Principales APIs

Méthode	Route	Description
POST	/api/auth/register	Inscription, renvoie token
POST	/api/auth/login	Connexion, renvoie token
GET	/api/todos	Liste (completed, titleContains, sortBy, direction)
POST	/api/todos	Créer un todo lié à l'utilisateur connecté
PUT	/api/todos/{id}	Mettre à jour (titre, complété, priorité)
DELETE	/api/todos/{id}	Supprimer
POST	/api/todos/complete-all	Compléter tous les todos de l'utilisateur
PUT	/api/todos/reorder	Réordonner selon la liste d'IDs

Sécurité

- JWT HS256; clé robuste (support Base64, fallback sécurisé).
- Filtrage global par utilisateur; endpoints sécurisés via Spring Security.

Problèmes Résolus

- WeakKeyException JWT: support Base64 et fallback HS256.
- order_index null PostgreSQL: `Integer` au lieu de `int`.
- Erreurs Angular de template: KPIs via signaux `computed`.

Démarrage & Intégration

Backend

- `./mvnw.cmd spring-boot:run`
- Base URL: `http://localhost:8080/`

Frontend

- `npm run start`
- App: `http://localhost:4200/`

Association des anciens todos

```
update todos set user_id = (select id from users where username = 'votre_username')
where user_id is null;
```

Pistes Futures

- Menu utilisateur (avatar, profil, déconnexion).
- Dialog de confirmation pour “Tout compléter”.
- Palette Material ajustée en dark.
- Page “Rapport” imprimable pour export PDF.