

# **Document de spécification pour l'application : Todo Team**

**//pour la table des matières**

# 1. Contexte et objectifs du projet :

Le but de ce projet est de développer une application mobile de gestion de tâches collaborative, appelée **ToDo Team**. Elle permet à des utilisateurs de s'authentifier, créer/modifier/supprimer des tâches, recevoir des rappels, et synchroniser leurs données via une API sécurisée.

# 2. Description générale :

- **Plateforme visée:** Android et IOS
- **Technologie utilisée:** Flutter (frontend mobile), SpringBoot (backend)
- **Communication:** API REST avec authentification
- **Stockage local** (en mode hors ligne)
- Langue principale français

# 3. Fonctionnalités prévues :

- **Coté mobile:**
  - Authentification (login + token)
  - Affichage de la liste de tâches
  - Ajout/modification/suppression de tâche
  - Notifications (push/locales selon temps dispo)
  - Stockage local léger
- **Coté Backend:**

## 1. Authentification & sécurité

- **Inscription & connexion** des utilisateurs (email / username + mot de passe).
- **Token JWT** pour sécuriser les appels API.
- **Rôles** (par défaut **USER**, possibilité future d'ajouter **ADMIN**).

## 2. Gestion des utilisateurs

Modèle **User** :

id, username, email, password, roles, enabled, createdAt

#### **API UserController :**

GET /user

GET /user/{id}

GET /user/name/{name}

DELETE /user/{id}

POST /user/save

#### **Modèle Task :**

id, title, description, dueDate, reminderDate, priority, status, archived, owner, createdAt, updatedAt

#### **Relations :**

**Task** est liée à un **User** → **@ManyToOne** (un utilisateur a plusieurs tâches, mais chaque tâche appartient à un seul utilisateur).

#### **API TaskController :**

GET /api/tasks

GET /api/tasks/{id}

GET /api/tasks/completed

GET /api/tasks/today

GET /api/tasks/archived

POST /api/tasks

PUT /api/tasks/{id}

DELETE /api/tasks/{id}

PUT /api/tasks/{id}/archive

PUT /api/tasks/{id}/unarchive  
PUT /api/tasks/{id}/duedate?date=...  
POST /api/tasks/auto-archive

## 4. Modèle de données :

- **User:**

id : identifiant unique pour chaque utilisateur  
username  
email  
password  
role : (admin ou bien utilisateur simple)  
createdAt  
updatedAt  
@OneToMany => Tasks

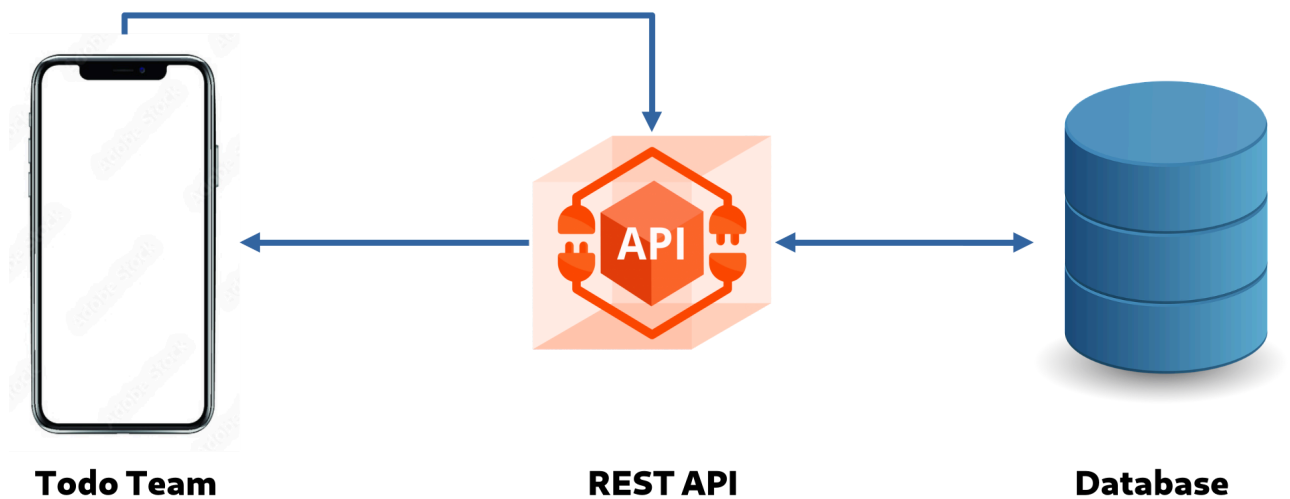
- **Task:**

id  
title  
description  
DueDate  
Completed  
archived  
createdAt  
updatedAt  
@ManyToOne => user

## 5.DB :

La base de données PostgreSQL est déployée dans un conteneur Docker pour faciliter le développement, l'isolation et la portabilité de l'application.

## 5. Architecture de l'application :



- *Diagramme simplifié de fonctionnement de l'appli*

## 6. Contraintes technique :

Flutter version 3.32.0 / Dart version 3.8

Spring Boot

Base de données :Postgres