Documentation Technique – Application de Gestion de Projets

Cette documentation s'adresse aux développeurs et administrateurs de l'application.

Sommaire

---

1. Architecture

- **Backend**: Java Spring Boot (API REST)
- **Frontend**: Vue.js (SPA)
- **Base de données**: MySQL
- **Gestion des emails**: SMTP (Gmail)

↑ Retour au sommaire

---

2. Configuration

Le fichier principal de configuration se trouve dans:
`api/src/main/resources/application.properties`

Exemple de configuration mail:

```
spring.mail.host=smtp.gmail.com
spring.mail.port=587
spring.mail.username=xxx@gmail.com
spring.mail.password=xxxx xxxx xxxx xxxx
spring.mail.protocol=smtp
spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.starttls.enable=true
```

↑ Retour au sommaire

3. Base de données

- **URL** : `jdbc:mysql://localhost:3306/collabspace`
- **Utilisateur** : `anne`
- **Mot de passe** : `1234`
- **Dialecte** : MySQL8

↑ Retour au sommaire

4. Principales entités

- **Projet** : id, nom, description, dateDebut, dateFin, chefDeProjet, statut, budget, client, membres
- **Tache** : id, titre, description, dateDebut, dateFin, statut, priorite, responsable, projetId
- **Materiel** : id, nom, type, description, etat, projetAffecte
- **Depense** : id, libelle, description, montant, dateDepense, type, projetId

↑ Retour au sommaire

5. Envoi d'emails

- Utilise Spring Boot Starter Mail.
- Les emails sont envoyés lors de l'assignation de tâches, de rappels, etc.
- Les paramètres SMTP sont à configurer dans `application.properties`.

↑ Retour au sommaire

6. Pagination

La pagination est utilisée pour lister les projets, tâches, matériels et dépenses.

- **Backend** : Utilise les paramètres `page` et `size` dans les endpoints REST (exemple : `/api/projets?page=0&size=10`).
- **Frontend** : Affichage paginé avec boutons "Page suivante", "Page précédente" et sélection du nombre d'éléments par page.
- **Exemple d'implémentation Spring Boot** :

```
@GetMapping("/projets")
public Page<Projet> getProjets(@RequestParam int page, @RequestParam int size) {
    return projetService.findAll(PageRequest.of(page, size));
}
```

- **Exemple d'implémentation Vue.js** :

```
// Appel API paginé
axios.get('/api/projets?page=0&size=10')
```

↑ Retour au sommaire

---

7. Démarrage du projet

1. Cloner le dépôt :
   `git clone <url-du-repo>`
2. Démarrer la base de données MySQL.
3. Lancer le backend :
   `cd api`
   `./mvnw spring-boot:run`
4. Lancer le frontend :
   `cd app`
   `npm install`
   `npm run serve`

↑ Retour au sommaire

---

8. Déploiement

- Prévoir un serveur avec Java 17+, Node.js, et MySQL.
- Adapter les variables d'environnement et le fichier `application.properties` pour la production.
- Utiliser un reverse proxy (Nginx, Apache) pour sécuriser l'accès.

↑ Retour au sommaire

---

9. Tests

- **Backend** : tests unitaires et d'intégration avec JUnit.
- **Frontend** : tests unitaires avec Jest ou Vue Test Utils.
- Lancer les tests backend :
  `./mvnw test`
- Lancer les tests frontend :
  `npm run test`

↑ Retour au sommaire

---

10. Support

Pour toute question technique, contactez l'équipe de développement à dev@votreentreprise.com.