

Résumé des fonctionnalités du microservice Hackathon

Fonctionnalités déjà implémentées

- CRUD complet pour l'entité Hackathon (Create, Read, Update, Delete)
- Connexion à PostgreSQL avec configuration via fichiers .env et .env.vault
- Gestion des erreurs centralisée avec un package dédié
- Métriques initiales (endpoint /metrics) pour supervision
- Middleware d'authentification Keycloak intégré pour protéger les endpoints
- Swagger partiellement configuré (doc.json généré mais nécessitant corrections dans les commentaires)

Fonctionnalités à compléter

- Finaliser la documentation Swagger avec des commentaires complets pour tous les handlers
- Corriger les erreurs liées aux types non trouvés dans les commentaires Swagger (par ex. echo.HTTPError)
- Implémenter le CRUD pour les entités restantes : Participant, Team, Registration
- Ajouter la validation des entrées avec un middleware (par ex. validation des champs obligatoires)
- Mettre en place les tests unitaires et d'intégration pour chaque handler et service
- Améliorer les métriques avec des compteurs spécifiques (par exemple : nombre d'inscriptions, nombre d'équipes créées)
- Configurer un endpoint /health et /readiness pour supervision et déploiement en production
- Sécuriser les rôles et permissions via Keycloak (admin, user, etc.)
- Compléter le fichier Swagger UI (doc.json) pour tester directement depuis l'interface Swagger
- Écrire un README clair expliquant l'installation, la configuration et l'exécution du projet