

EC03 – Rattrapage : Développement back-end “Gestion des ateliers” avec BDD relationnelle + NoSQL

Contexte général

Dans le cadre du projet **SkillHub**, l'épreuve de rattrapage consiste à développer une partie du back-end dédiée à la **gestion des ateliers**.

L'objectif est de manipuler à la fois une base **relationnelle** et une base **NoSQL**, afin de stocker et traiter des informations différentes :

- PostgreSQL pour la gestion des **entités ateliers** (titre, formateur, date, capacité)
- MongoDB pour les **inscriptions et interactions** (logs, historique d'inscription, commentaires)

Le socle technique est fourni : environnement Docker prêt, authentification en place, bases initialisées et documentation fonctionnelle.

L'évaluation porte autant sur l'écriture de code que sur la capacité à justifier les choix techniques.

Livrables attendus

Livrable	Détail
Code source	Développement en POO / MVC avec BDD relationnelle ET NoSQL
Dump SQL et JSON	Extraits de données (PostgreSQL et MongoDB)
Scripts de sauvegarde	<code>backup.sh</code> , <code>restore.sh</code> pour les deux bases

Livrable	Détail
README	Justification des choix techniques, commandes terminal, arbitrages

Modalités d'évaluation

- **Type d'épreuve** : Mise en situation reconstituée sur ordinateur
- **Durée** : 4h
- **Nature** : Épreuve individuelle, sans oral
- **Critères évalués** :
 - C7.1 : Configuration environnement de dev (Docker, terminal)
 - C8.1 : Programmation orientée objet, architecture MVC
 - C9 : Sécurisation, optimisation du serveur
 - C10 : Utilisation de base de données relationnelle (PostgreSQL)
 - C11 : Utilisation de base de données NoSQL (MongoDB)
 - C12 : Justification des choix BDD (SQL vs NoSQL)
 - C13 : Mise en place de la sauvegarde et restauration

Recommandations

- Penser dès le début à la **séparation des responsabilités** (MVC)
- Sécuriser les routes sensibles (authentification JWT, middlewares)
- Utiliser **migrations ou scripts SQL** pour la base relationnelle
- Prévoir des **dumps cohérents** et testables
- Documenter les **choix d'implémentation** dans le README
- Automatiser les sauvegardes/restaurations avec des scripts simples

Exemples d'éléments attendus

- Une entité `Workshop` persistée dans PostgreSQL avec migrations
- Une collection `Registrations` ou `Comments` dans MongoDB
- Une route protégée (`/workshops/:id/registrations`) combinant données SQL + NoSQL
- Un fichier `.env` propre, un `docker-compose.yml` fonctionnel

- Des logs et captures de terminal montrant les commandes exécutées

Nom de dossier attendu

- Une archive ZIP nommée `EC03R_NomPrenom.zip` contenant :
 - Le dossier `src` du projet
 - Le fichier `README.md` avec explications techniques
 - Le dump SQL (`/data/skillhub_workshops.sql`) et le dump Mongo (`/data/mongo_registrations.json`)
 - Les scripts `backup.sh` et `restore.sh`

Rappel pédagogique

Cette épreuve de rattrapage vise à valider les mêmes compétences que l'EC03 initiale :

- Capacité à développer un back-end structuré (POO, MVC)
- Maîtrise de deux types de bases complémentaires (relationnelle et NoSQL)
- Compréhension des enjeux de **sécurité** et de **structuration des données**
- Capacité à justifier ses choix techniques et à automatiser les sauvegardes

Le contexte change (gestion des ateliers au lieu de gestion des utilisateurs), mais le niveau et les objectifs restent identiques.