Grille de correction détaillée – EC04

Informations générales

- **Épreuve :** EC04 – API sécurisée et documentée
- **Durée :** 4h – Individuelle
- **Projet fil rouge :** SkillHub – API publique de la plateforme
- **Technologies imposées :** libre choix (REST ou GraphQL), mais API sécurisée et documentée obligatoire

Critères d'évaluation et attentes

Critère	Intitulé officiel	Attentes côté correcteur	Points de vigilance
C8.3	API REST sécurisée	Vérifier que l'API dispose d'un système d'authentification (JWT ou équivalent), de middlewares de contrôle d'accès, et de routes protégées. Organisation claire : contrôleurs, services, middlewares séparés.	Tester un endpoint protégé sans token → doit renvoyer `401` . Tester avec token invalide → `403` . Vérifier la cohérence des statuts HTTP.
C8.4	Documentation API	Présence d'un fichier Swagger/OpenAPI (`openapi.yaml` ou équivalent) décrivant toutes les routes, paramètres, réponses et codes d'erreur. Documentation exploitable par un outil type Swagger UI.	Documentation absente, incomplète ou non exploitable = gros manque. Vérifier si exemples de requêtes fournis.
C8.5	Tests unitaires API	Vérifier existence de tests automatisés sur les endpoints critiques (`GET /users/me` ,	Couverture faible ou tests trop superficiels = perte de points. Vérifier

Critère	Intitulé officiel	Attentes côté correcteur	Points de vigilance
		`POST /workshops` , etc.). Rapports de couverture fournis.	que les tests passent réellement.

Livrables à corriger

- **Code source :** projet API (`src/`), architecture claire
- **README.md :** instructions de lancement, exemples de requêtes, gestion des tokens
- **Documentation :** fichier Swagger/OpenAPI complet
- **Tests :** code + rapport de tests (couverture visible)

Barème indicatif (sur 20 points)

Axe évalué	Points
API sécurisée (C8.3)	/8
Documentation API (C8.4)	/6
Tests unitaires des endpoints (C8.5)	/6

> **Tolérance :** ±1 point par critère selon la qualité globale et la propreté du code.

Points positifs attendus

- API bien structurée, endpoints clairs et cohérents
- Authentification fonctionnelle avec JWT ou équivalent
- Gestion des erreurs robuste (codes HTTP adaptés, messages explicites)
- Documentation Swagger lisible et complète, exploitable sans aide
- Tests automatisés couvrant les principales routes (création, lecture, sécurisation)
- README précis avec exemples concrets de requêtes (via `curl` , Postman…)

Erreurs fréquentes à surveiller

- Endpoints publics non sécurisés (exposition de données sensibles)
- JWT mal implémenté ou inexploitable
- Documentation Swagger inexistante, trop générique ou obsolète
- Codes HTTP incohérents (`200` pour des erreurs, `500` à répétition)
- Tests absents, incomplets, ou non exécutables
- README incomplet (aucune indication sur comment tester l'API)

Rappel pédagogique

Cette épreuve vérifie que l'apprenant sait :

- Exposer une **API professionnelle et sécurisée**
- Documenter ses endpoints pour une intégration externe
- Mettre en place des **tests unitaires fiables**
- Structurer son code de manière maintenable

Elle prépare directement les étapes suivantes du fil rouge, où l'API sera consommée par le front-end (EC02) ou exploitée dans un environnement cloud et CI/CD (EC05–EC06).