Grille de correction détaillée – EC04

Informations générales

- **Épreuve :** EC04 – API sécurisée et documentée
- **Durée :** 4h – Individuelle
- **Projet fil rouge :** RebootCamp – API publique pour défis, scores et badges
- **Nature :** Épreuve pratique (développement, documentation, tests)

Critères d’évaluation et attentes

Critère	Intitulé officiel	Attentes côté correcteur	Points de vigilance
C8.3	Conception API REST sécurisée	Endpoints organisés, authentification JWT, autorisations correctes, gestion des erreurs.	Vérifier respect des statuts HTTP (`200` , `401` , `403` , `422`), absence de failles (ex : accès non protégés).
C8.4	Documentation API	Swagger/OpenAPI (fichier `.yaml` ou `.json`) documentant toutes les routes, exemples de requêtes.	Attention aux docs générées automatiquement mais incomplètes.
C8.5	Tests unitaires API	Tests sur endpoints critiques (authentification, progression, leaderboard). Rapport de couverture fourni.	Vérifier qualité des assertions : pas seulement “200 OK”, mais cohérence des réponses.

Livrables à corriger

- **Code source** : API structurée, endpoints clairs et sécurisés
- **Documentation API** : Swagger/OpenAPI complet et lisible
- **Tests unitaires** : automatisés, rapport de couverture visible
- **README** : instructions de test, exemples de requêtes avec tokens

Barème indicatif (sur 20 points)

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

> **Tolérance :** ±2 points selon la robustesse globale et la maintenabilité du projet.

Points positifs attendus

- Endpoint `GET /users/me` protégé par JWT, retour clair du profil utilisateur
- Endpoint `POST /challenges/:id/complete` validant un défi et mettant à jour la progression
- Endpoint `GET /leaderboard` calculant correctement les scores et affichant le classement
- Gestion centralisée des erreurs (`try/catch` , middleware d'erreurs, messages clairs)
- Documentation Swagger testable immédiatement via Swagger UI ou Postman
- Tests automatisés couvrant l'authentification et les routes critiques, avec taux de couverture satisfaisant

Erreurs fréquentes à surveiller

- Endpoints non protégés ou JWT non implémenté correctement
- Statuts HTTP incorrects (`200` renvoyé même en cas d'erreur)
- Documentation API absente, trop vague ou non testable
- Tests unitaires incomplets ou inexistants
- README sans exemples pratiques (absence de requêtes curl/Postman)
- Routes mal organisées ou non conformes à une logique REST (`/doSomething` au lieu de `/resources/:id`)

Rappel pédagogique

Cette épreuve valide la capacité à :

- Concevoir une **API sécurisée et exploitable en production**
- Structurer le code de manière claire et maintenable
- Produire une **documentation technique utilisable par des partenaires externes**
- Vérifier la fiabilité via des **tests unitaires**

Elle place l'apprenant dans un contexte réaliste de **déploiement d'une API ouverte**, utilisable par une appli mobile ou par des partenaires institutionnels (mairies, écoles, associations).