

ESGI – Module API Rest pour l'administration système

PROJET FIN DE MODULE

Mastere Systèmes – Réseaux et Cloud Computing

API Rest pour l'administration système

(Intervenant professeur : Morgan Thibert)

Date de Rendu

Vos travaux devront être transmis pour le 28 février 2025 au plus tard.

Vos rendus devront être transmis via un lien type wetransfer sur l'adresse mail: mthibert@myges.fr.

Une confirmation de réception vous sera envoyé une fois le projet téléchargé.

Synopsis

Le projet final vise à mettre en pratique les concepts et les compétences acquises durant le module API Rest en créant une Api fonctionnelle en langage php. L'accent sera mis sur la bonne mise en place des opérations CRUD, le schéma de connexion à la base de données sqlite fournie, la gestion de l'authentification, la documentation, et le respect des pratiques REST.

Projet Fictif : Contexte



Le ZEvent est un événement caritatif annuel de streaming en direct, réunissant des créateurs de contenu et ayant pour objectif de lever des sommes considérables en faveur de causes humanitaires.

Vous avez été mandaté en exclusivité par Zérator lui même pour préparer le Zevent 2025, et vous allez être en charge de créer une API Rest dédiée à la gestion des dons pour le Zevent. Cette API devra offrir des fonctionnalités permettant la collecte, le suivi et la gestion efficace des contributions financières destinées à soutenir les initiatives caritatives de l'évènement.

Une équipe de développement (dont vous ne faites pas partie) s'occupe de son côté de réaliser un client web (programme) qui sera à la disposition de chaque streamer, qui permettra d'interagir avec l'API Rest que vous allez créer.

Votre rôle à vous est de créer l'API qui fera le lien entre le client web (sur lequel se connecteront les streamers durant l'évènement), et la base de données qui contient les données relatives aux dons.

Base de données :

Une base de données de test vous est fournie sous forme d'un fichier sqlite.

Cette base de données contient 2 tables :

- **une table « streamers »** qui contient les 36 streamers qui seront présents à l'évènement. Elle est composée des champs suivants :

- *id (int) : identifiant unique du streamer*
- *pseudonym (text) : nom du streameur*
- *api_key (text) : clé_api fournie au streamer qui sera automatiquement intégrée dans le client web développé par l'autre équipe de développement.*
- *roles (text) : un rôle associé au streameur. Les rôles possibles sont soit aucun, soit « ADMIN », soit « SUPERADMIN ».*

- **une table « donations »** qui contient les dons effectués par les viewers. Elle est composée des champs suivants :

- *id (int) : identifiant unique du don*
- *pseudonym_donator (text) : Le pseudonyme du viewer qui a réalisé le don.*
- *donation (int) : Le montant du don en euros*
- *streamer_id (int) : La référence au streamer pour l'ajout du don à la cagnotte.*
- *message (text) : Le message du viewer correspondant à son don.*
- *donation_time (text) : la date au format ISO 8601 à l'instant du don.*

Votre Mission :

Réaliser en langage PHP natif (sans framework) une API qui respecte autant que possible les principes de l'architecture REST. Implémentez les routes CRUD définies ci-dessous pour les ressources « **Donations** » et « **Streamers** », tout en respectant les contraintes suivantes :

1. Les endpoints suivants doivent être publics (sans authentification) :

- *GET /api/streamers* : renvoi la collection correspondant à la ressource « Streamer »
- *GET /api/streamers/{id}* : renvoi les informations d'un streamer.
- *GET /api/donations* : renvoi la collection correspondant à la ressource « Donations ».
- *GET /api/donations?streamer={id}* : renvoi la collection de Dons concernant un streamer spécifique.
- *GET /api/donations/{id}* : renvoi les information d'un don.

2. Implémentez de même les endpoints suivants via une authentification par api key passée en header de la requête :

- *POST /api/donations*
- *DELETE /api/donations/{id}*
- *PATCH /api/donations/{id}*

3. Seuls les streameurs avec le rôle SUPERADMIN sont autorisés à créer, modifier ou supprimer une donation.

4. Les streameurs avec le rôle ADMIN sont autorisés uniquement à modifier le message d'une donation, les autres champs ne sont modifiables qu'avec le rôle SUPERADMIN

5. Les clés api des streamers ne doivent jamais être retournées dans les réponses.

Très important : Votre API devra être expliquée et documentée de la façon la plus complète possible via un export de collection Postman.

Recommandations :

- Respectez bien les codes de retour correspondants en fonction de votre réponse.
- Rendez votre API robuste en incluant des tests sur des requêtes malformées ou invalides. Indiquez dans votre documentation les valeurs attendues.
- Sauvegardez bien après chaque modification de requête dans Postman et vérifiez bien vos exports de documentations Postman.
- N'hésitez pas à utiliser toutes les ressources en lignes sur la création d'API Rest pour vous aider dans votre tâche et résoudre vos différents problèmes.

Livrables attendus :

- Un dossier zippé (zip, rar, tar.gz ou autre) nommé "<nom>_<prenom>_projet_esgi_api_rest" contenant votre code, ainsi que votre export de collection Postman (documentation).
- Vos rendus devront être transmis via un lien type wetransfer sur l'adresse mail: **mthibert@myges.fr**. Une confirmation de réception vous sera envoyée une fois le projet téléchargé.

Notation :

Vous serez évalué sur les points suivants :

- Qualité de votre implémentation.
- L'organisation de votre code.
- Votre documentation.