

API "Let's Play" Documentation

Présentation

Le TP1 "API Let's Play" consiste à développer une API pour une application de jeux en ligne. Les étudiants devront mettre en place l'authentification, la gestion des utilisateurs, la gestion du pointage et les opérations sur l'utilisateur actuel. L'évaluation portera sur la mise en place des routes, l'application des bonnes pratiques de développement, l'utilisation de l'architecture MVC-R et un code propre. La matière vue en classe devra être respectée, avec une attention particulière à l'application des concepts et techniques enseignés. Les données seront stockées dans une liste, ce qui entraînera leur réinitialisation à chaque redémarrage du serveur.

Consignes :

- Ce TP se fait en équipe de deux.
- La date de remise sera communiquée sur Omnivox. Les retards sont acceptés, mais entraîneront une pénalité de 10 points par journée de retard.
- Pour la remise, assurez-vous de supprimer le dossier "node_modules" et le fichier "package-lock.json" de votre projet.
- Seule une personne de l'équipe doit effectuer la remise via Omnivox.
- Vous devez également fournir un rapport décrivant la contribution de chaque membre de l'équipe dans le TP1.
- Dans le rapport, attribuez des points à chaque membre, avec un total de 100 points. Par exemple : Jonathan a travaillé sur 40 points du projet, Jacob sur 30 points et Julie sur 30 points. Veuillez spécifier clairement la répartition des points.
- Notez que je ne corrigerai pas le TP si je n'ai pas reçu de rapport de votre part.
- Je prendrai en compte la répartition des points dans l'évaluation du TP pour chacun.

Authentification

Route de connexion

- Endpoint: **POST /auth/login**
- Description: Cette route permet à un utilisateur de se connecter en fournissant ses informations d'identification.
- Paramètres de la requête :
 - email (string, obligatoire) : Adresse e-mail de l'utilisateur.
 - password (string, obligatoire) : Mot de passe de l'utilisateur.
- Réponse réussie :
 - Statut : 200 OK
 - Corps de la réponse : Token d'authentification pour l'utilisateur connecté.
- Réponses d'erreur possibles :
 - 400 Bad Request : Si des paramètres sont manquants ou invalides.
 - 401 Unauthorized : Si l'adresse e-mail ou le mot de passe est incorrect.

Route d'inscription

- Endpoint: **POST /auth/register**
- Description: Cette route permet à un nouvel utilisateur de s'inscrire en fournissant ses informations personnelles.
- Paramètres de la requête :
 - email (string, obligatoire) : Adresse e-mail de l'utilisateur.
 - password (string, obligatoire) : Mot de passe de l'utilisateur.
 - username (string, obligatoire) : Nom de l'utilisateur.
- Réponse réussie :
 - Statut : 201 Created
 - Corps de la réponse : Token d'authentification pour l'utilisateur nouvellement inscrit.
- Réponses d'erreur possibles :
 - 400 Bad Request : Si des paramètres sont manquants ou invalides.
 - 409 Conflict : Si l'adresse e-mail est déjà utilisée par un autre utilisateur.

Gestion des utilisateurs

Liste des utilisateurs

- Endpoint: **GET /users**
- Description: Cette route permet d'obtenir la liste de tous les utilisateurs sauf le ou les administrateurs enregistrés dans le système.
- Paramètres de la requête :
 - **Authorization** (string, obligatoire) : Token d'authentification de l'utilisateur.
- Réponse réussie :
 - Statut : 200 OK
 - Corps de la réponse : Liste des utilisateurs sauf l'administrateur.
- Réponses d'erreur possibles :
 - 401 Unauthorized : Si l'utilisateur n'est pas authentifié ou si le token est invalide

Modifier un utilisateur

- Endpoint: **PUT /users/:id**
- Description: Cette route permet de modifier les informations d'un utilisateur spécifié par son identifiant. Seuls les utilisateurs avec un accès administrateur peuvent accéder à cette route.
- Paramètres de la requête :
 - id (string, obligatoire) : Identifiant de l'utilisateur à modifier.
 - username (string, facultatif) : Nouveau nom de l'utilisateur.
 - email (string, facultatif) : Nouvelle adresse e-mail de l'utilisateur.
 - **Authorization** (string, obligatoire) : Token d'authentification de l'utilisateur.
- Réponse réussie :
 - Statut : 200 OK
 - Corps de la réponse : Utilisateur modifié.
- Réponses d'erreur possibles :
 - 401 Unauthorized : Si l'utilisateur n'est pas authentifié, si le token est invalide ou n'a pas les droits d'accès administrateur.
 - 404 Not Found : Si l'utilisateur spécifié n'existe pas.
 - 409 Conflict : Si l'adresse e-mail est déjà utilisée par un autre utilisateur.

Supprimer un utilisateur

- Endpoint: **DELETE /users/:id**
- Description: Cette route permet de supprimer un utilisateur spécifié par son identifiant. Seuls les utilisateurs avec un accès administrateur peuvent accéder à cette route.
- Paramètres de la requête :
 - id (string, obligatoire) : Identifiant de l'utilisateur à supprimer.
 - **Authorization** (string, obligatoire) : Token d'authentification de l'utilisateur.
- Réponse réussie :
 - Statut : 204 No Content
- Réponses d'erreur possibles :
 - 401 Unauthorized : Si l'utilisateur n'est pas authentifié ou n'a pas les droits d'accès administrateur.
 - 404 Not Found : Si l'utilisateur spécifié n'existe pas.

Gestion des points

Incrémenter le pointage d'un utilisateur

- Endpoint: **PUT /game/wins**
- Description: Cette route permet d'incrémenter le pointage d'un utilisateur.
- Paramètres de la requête :
 - **Authorization** (string, obligatoire) : Token d'authentification de l'utilisateur.
- Réponse réussie :
 - Statut : 200 OK
 - Corps de la réponse : Pointage mis à jour de l'utilisateur.
- Réponses d'erreur possibles :
 - 401 Unauthorized : Si l'utilisateur n'est pas authentifié ou si le token est invalide.

Décrémenter le pointage d'un utilisateur

- Endpoint: **PUT /game/lost**
- Description: Cette route permet de décrémenter le pointage d'un utilisateur.
- Paramètres de la requête :
 - **Authorization** (string, obligatoire) : Token d'authentification de l'utilisateur.
- Réponse réussie :
 - Statut : 200 OK
 - Corps de la réponse : Pointage mis à jour de l'utilisateur.
- Réponses d'erreur possibles :
 - 401 Unauthorized : Si l'utilisateur n'est pas authentifié ou si le token est invalide.

Utilisateur actuel

Obtenir les informations de l'utilisateur actuel

- Endpoint: **GET /me**
- Description: Cette route permet d'obtenir les informations de l'utilisateur actuellement authentifié.
- Paramètres de la requête :
 - **Authorization** (string, obligatoire) : Token d'authentification de l'utilisateur.
- Réponse réussie :
 - Statut : 200 OK
 - Corps de la réponse : Informations de l'utilisateur.
- Réponses d'erreur possibles :
 - 401 Unauthorized : Si l'utilisateur n'est pas authentifié ou si le token est invalide.

Modifier les informations de l'utilisateur actuel

- Endpoint: **PUT /me**
- Description: Cette route permet de modifier les informations de l'utilisateur actuellement authentifié.
- Paramètres de la requête :
 - username (string, facultatif) : Nouveau nom de l'utilisateur.
 - email (string, facultatif) : Nouvelle adresse e-mail de l'utilisateur.
 - **Authorization** (string, obligatoire) : Token d'authentification de l'utilisateur.
- Réponse réussie :
 - Statut : 200 OK
 - Corps de la réponse : Utilisateur modifié.
- Réponses d'erreur possibles :
 - 401 Unauthorized : Si l'utilisateur n'est pas authentifié ou si le token est invalide.
 - 409 Conflict : Si l'adresse e-mail est déjà utilisée par un autre utilisateur.

Supprimer l'utilisateur actuel

- Endpoint: **DELETE /me**
- Description: Cette route permet de supprimer l'utilisateur actuellement authentifié.
- Paramètres de la requête :
 - **Authorization** (string, obligatoire) : Token d'authentification de l'utilisateur.
- Réponse réussie :
 - Statut : 204 No Content
- Réponses d'erreur possibles :
 - 401 Unauthorized : Si l'utilisateur n'est pas authentifié ou si le token est invalide.

Remettre à zéro le pointage de l'utilisateur

- Endpoint: **PUT /me/reset-score**
- Description: Cette route permet à l'utilisateur actuel de remettre à zéro son pointage.
- Paramètres de la requête :
 - **Authorization** (string, obligatoire) : Token d'authentification de l'utilisateur.
- Réponse réussie :
 - Statut : 200 OK
 - Corps de la réponse : Message indiquant que le pointage a été remis à zéro.
- Réponses d'erreur possibles :
 - 401 Unauthorized : Si l'utilisateur n'est pas authentifié ou si le token est invalide.

Répartition des points

Authentification : 20 points

- Connexion : 10 points
- Inscription : 10 points

Gestion des utilisateurs : 25 points

- Liste des utilisateurs : 5 points
- Modification d'un utilisateur : 10 points
- Suppression d'un utilisateur : 10 points

Gestion du pointage : 20 points

- Incrémenter le pointage d'un utilisateur : 10 points
- Décrémenter le pointage d'un utilisateur : 10 points

Utilisateur actuel : 35 points

- Obtenir les informations de l'utilisateur actuel : 5 points
- Modifier les informations de l'utilisateur actuel : 10 points
- Supprimer l'utilisateur actuel : 10 points
- Remettre à zéro le pointage de l'utilisateur : 10 points

Respect de la matière vue en classe : 20 points

- Utilisation correcte des concepts et des techniques abordés en classe : 10 points
- Application appropriée des bonnes pratiques de développement : 10 points

Utilisation de MVC-R : 15 points

- Utilisation correcte de l'architecture MVC-R (Modèle, Vue, Contrôleur, Routes) : 10 points
- Séparation adéquate des responsabilités entre les différentes composantes du MVC-R : 5 points

Code propre : 15 points

- Respect des conventions de nommage et de la structure du code : 5 points
- Organisation logique du code (indentation, espacement, etc.) : 5 points
- Absence de duplications de code inutiles : 5 points

Commentaires : 10 points

- Présence de commentaires significatifs pour expliquer le code : 5 points
- Commentaires de documentation pour les fonctions, classes, etc. : 5 points

Total : 160 points