

Documentation Technique de l'API

Cette API, réalisée grâce à ExpressJS, expose plusieurs routes pour gérer des utilisateurs et permettre l'upload de fichiers. Certaines routes sont accessibles publiquement, d'autres nécessitent un token.

1. GET /api/test

- Description : Vérification du fonctionnement du serveur.
- Méthode : GET
- En-têtes : Aucun requis
- Body : Aucun
- Authentification : Aucune
- Réponse : Renvoie la chaîne "hello".

2. POST /api/register

- Description : Création d'un nouvel utilisateur.
- Méthode : POST
- En-têtes :
 - Content-Type : application/json
- Body :

```
{
  "nom": "Nom",
  "prenom": "Prénom",
  "email": "exemple@domaine.com",
  "pseudo": "Pseudo",
  "password": "MotDePasse"
}
```
- Authentification : Aucune
- Réponse :
 - "OK" si la création s'est bien déroulée.
 - "KO" en cas d'erreur.

3. POST /api/login

- Description : Authentification d'un utilisateur et génération d'un token JWT.
- Méthode : POST
- En-têtes :
 - Content-Type: application/json
- Body

```
{
  "email": "exemple@domaine.com",
  "password": "MotDePasse"
}
```
- Authentification : Aucune
- Réponse :
 - En cas de succès, renvoie un token JWT.

- En cas d'échec, renvoie un message d'erreur.

4. GET /api/profil

- Description : Récupération du profil de l'utilisateur connecté.
- Méthode : GET
- En-têtes :
 - Authorization : Bearer <token JWT>
- Body : Aucun
- Authentification : Requête (token JWT).
- Réponse : Objet JSON contenant les informations de l'utilisateur (nom, prénom, email, pseudo...).

5. DELETE /api/rm/id

- Description : Supprime un utilisateur en fonction de son identifiant.
- Méthode : DELETE
- Paramètres d'URL :
 - id : identifiant de l'utilisateur à supprimer.
- En-têtes :
 - Authorization : Bearer <token JWT>.
- Body : Aucun
- Authentification : Requête
- Réponse : Message confirmant la suppression ou signalant une erreur.

6. PUT /api/users/ban/id

- Description : Bloque un utilisateur en modifiant son attribut isBanned à true.
- Méthode : PUT
- Paramètres d'URL :
 - id : identifiant de l'utilisateur à bloquer.
- En-têtes :
 - Authorization : Bearer <token JWT>
- Body : Aucun
- Authentification : Requête
- Réponse : Message confirmant que l'utilisateur a été bloqué.

7. POST /api/add-file

- Description : Permet d'uploader un fichier sur le serveur.
- Méthode : POST
- En-têtes :
 - Authorization : Bearer <token JWT>
 - Le header Content-Type est géré automatiquement lors de l'envoi en multipart/form-data.
- Body :
 - Type : form-data
 - Champ attendu : file
- Authentification : Requête
- Réponse : Message confirmant l'upload ou retournant des détails sur le fichier.