

Web Services

Pour lancer le serveur : npm run serve

Pour lancer les tests: npm run test

I) API User

1) createUser

URL : POST /api/create/user

Description : Création d'un nouvel utilisateur

Paramètres en entrée : login, password, lastname, firstname, mail, birthday

Format de sortie : ok/sortie

Exemple de sortie : { « id », 1 }

Erreurs possibles : user existe déjà, bd injoignable, manque un paramètre.

2) getUser

URL : GET /api/user/ :userid

Description : Retourne l'utilisateur d'id userid

Paramètres en entrée : aucun

Format de sortie : json

Exemple : { « username » : « jean », ... }

Erreur possible : « User inconnu »

3) getAllUser

URL : GET /api/user/all

Description : retourne l'ensemble des utilisateurs enregistrés dans le site

Paramètres en entrée : aucun

Format de sortie : json

Erreur possible : « Aucun user dans la base »

4) deleteUser

URL : DELETE /api/user/ :userid

Description : Supprime l'user d'id userid de la base de données

Paramètres en entrée : userid

Format de sortie : json

Erreur possible : « User inexistant »

5) Login

URL : POST /api/user/login

Description : permet de créer une clé de connexion valide.

Paramètre(s) en entrée (dans le champs body) : login, password

Format de sortie : clé/erreur

Exemple de sortie : { "status" : "403", "message" : "login et/ou le mot de passe invalide(s)" } ou { "status" : "200", "message" : "Login et mot de passe acceptés" }

Erreurs possibles : user inconnu (401), password incorrect (401), erreur interne

6) Logout

URL : DELETE /api/user/userid/logout

Description : permet de fermer une connexion (session) d'un utilisateur

Paramètre(s) en entrée : {}

Format de sortie : ok/erreur

Exemple de sortie : {"status" : "401", "message" : "Utilisateur inconnu"}
ou "status" : "200", "message" : "session
fermée"}

Erreurs possibles : aucun user ne correspond à l'user id en paramètre

II) API Messages

1) createMessage

URL : POST /apim/messages/post

Description : Création d'un nouveau message dans la base

Paramètres en entrée : username, userid, date, text

Format de sortie : json

Exemple de sortie : {«msg_st » : « Message dans la base de données »}

Erreurs possibles : bd injoignable, manque un paramètre.

2) deleteMessages

URL: DELETE /apim/messages/del/:username

Description: Supprime les messages envoyés par un user dans la base

Paramètres : username

Format de sortie: json

Exemple: {"response": "Messages supprimes"}

3) getMessagesFromUser

URL: GET /apim/messages/get/:username

Description: Retourne les messages envoyés par un user dans la base

Paramètres : username

Format de sortie: json contenant la liste des messages envoyés par l'user username

4) getAllMessages

URL: GET /apim/messages/all

Description: Retourne tous les messages envoyés

Paramètres : aucun

Format de sortie: json contenant la liste des messages envoyés