

Sheet1

	Point d'accès	request body	type de réponse	fonction
POST	/api/auth/signup	{email: string, password: string}	{message: string}	Hachage MDP(bcrypt) et email
POST	/api/auth/login	{email: string, password: string}	{use_id: string, token: string}	Vérification du user
POST	/api/auth/recover	{email: string, password: string}	{message: string}	Envoi du mdp dans la boîte mail si seulement est dans la base de données et compte actif. Le user peut réactiver son compte par cette requête, si le compte a été désactivé auparavant.
PUT	/api/user/setting/password	{password: string}	{message: string}	Hachage du MDP (bcrypt)
PUT	/api/user/setting/avatar	{avatar: string}	{message: string}	Changement de l'avatar par défaut
PUT	/api/user/setting/account	{email: string, password: string, active: false}	{message: string}	Envoi d'un email à l'utilisateur. Active =false. Désactive son propre compte.
GET	/api/user/:id		Affiche 1 utilisateur uniquement	Renvoie l'utilisateur correspondant à l'id fourni.
POST	/api/post	{title: string, image:string, text: text}	{message: string}	Analyse, les données envoyées. Initialise les différents type de réactions aux post à 0.
PUT	/api/post/:id	{title: string, image:string, text: text}	{message: string}	Modifie les données nécessaires du post.
GET	/api/post		Array of post with user name, comments, likes	Envoie les post sous forme de fil d'actualité.
GET	/api/post/:id		single post with user name	Affiche un seul post correspondant à l'id avec le user name
POST	/api/post/:id/like	{user_id: integer, post_like: integer}	{message: string}	L'utilisateur à le choix des likes. Si il en choisi un, celui si sera augmenté de 1. Si il reclique dessus celui est annulé. Il devra choisir son type de réaction au survol.
POST	/api/comment/:postId	{text: text}	user comment with name	Enregistre le text du commentaire. Affiche le nom du propriétaire du post. Initialise les différents type de réactions aux post à 0

Sheet1

POST	/api/comment/:commentId/like	{user_id: integer, comment_like: integer}	{message: string}	L'utilisateur à le choix des likes. Si il en choisi un, celui si sera augmenté de 1. Si il reclique dessus celui est annulé. Il devra choisir son type de réaction au survol.
GET	/api/user/admin/userlist		{user_id: integer, fristname: string, lastname: string, email:string, password:string, avatar: string, admin: boolean, active: boolean, createdAt: datetime, updatedAt: datetime}	Affche la liste complète des utilisateurs ainsi que les informations complètes contenu dans la table users.
PUT	/api/user/admin/user/:id/account	{fristname: string, lastname: string, email:string, password:string, avatar: string, admin: boolean, active: boolean}	{message: string}	Renvoie un message dans l'application que le compte concerné est soit supprimé ou modifié. Préviens par mail l'administrateur ainsi que l'utilisateur concernés.
PUT	/api/user/admin/post/:id/modify	{title: string, image:string, text: text}	{message: string}	Modère les post en supprimant ou modifiant une partie du contenu.
DELETE	/api/user/admin/post/:id/delete		{message: string}	Supprime définitivement un post de la base de données avec l'id fourni.