

Nom du web service :	SignUp
URL	/users/api/signup POST
Description du web service	créer un nouvel utilisateur
Paramètres en entrée	Username, Email, Password
Format de sortie	
Exemples de sortie	<pre>{ "status": 201, "message": "user created succsesfully", "details": "" } { "status": 400, "message": "Email already registered", "details": ""} { "status": 500, other errors }</pre>
Erreurs possibles	email déjà utilisé -> 400, les autres erreurs -> 500
Avancement du service	fini
Classes / Fichiers JS	./routeur/user.js ./schema/user.js
Informations additionnelles	

Nom du web service :	LogIn
URL	/users/api/login POST
Description du web service	permet à un utilisateur de se connecter à un compte personnel
Paramètres en entrée	Email, Password
Format de sortie	JSON
Exemples de sortie	<pre>{ "status": 200, "message": "user connected succsesfully", "details": "" } { "status": 403, "message": "email or password incorrect", "details": ""} { "status": 500, other errors }</pre>
Erreurs possibles	email ou mot de passe incorrect -> 403, les autres erreurs -> 500
Avancement du service	fini
Classes / Fichiers JS	./routeur/user.js ./schema/user.js
Informations additionnelles	

Nom du web service :	Profile
URL	/users/:userId GET
Description du web service	consulter les informations personnelles de l'utilisateur actuel
Paramètres en entrée	
Format de sortie	JSON
Exemples de sortie	<pre>{ "status": 203, "_id": "643c145e9b8f4d7cd0a71b2b", "name": "test", "email": "test@test.com", "password": "test", "__v": 0, "details": "" } { "status": 500, other errors }</pre>
Erreurs possibles	obtention réussie des informations personnelles -> 403, les autres erreurs -> 500
Avancement du service	fini
Classes / Fichiers JS	./routeur/user.js ./schema/user.js
Informations additionnelles	

Nom du web service :	DeleteUser
URL	/users/:userId/api/deleteuser DELETE
Description du web service	supprimer les informations de l'utilisateur
Paramètres en entrée	_id
Format de sortie	JSON
Exemples de sortie	{ "status": 200, "message": "delete user succsesfully", "details": "" } { "status": 500, other errors }
Erreurs possibles	suppression de l'utilisateur réussie -> 403, les autres erreurs -> 500
Avancement du service	fini
Classes / Fichiers JS	./routeur/user.js ./schema/user.js
Informations additionnelles	

Nom du web service :	CreatPost
URL	/users/:userId/post POST
Description du web service	créer un post
Paramètres en entrée	message
Format de sortie	JSON
Exemples de sortie	<pre>{ "status": 201, "message": "post successfully", "details": "" } { "status": 500, other errors }</pre>
Erreurs possibles	partager message réussie -> 201, les autres erreurs -> 500
Avancement du service	fini
Classes / Fichiers JS	./routeur/post.js ./schema/post.js
Informations additionnelles	

Nom du web service :	DeletePost
URL	/users/:userId/posts/:postId/api/deletePost DELETE
Description du web service	supprimer un post
Paramètres en entrée	
Format de sortie	JSON
Exemples de sortie	<pre>{ "status": 200, "message": "delete post successfully", "details": "" } { "status": 403, "message": "can't delete others' post" } { "status": 500, other errors }</pre>
Erreurs possibles	supprimer post réussie -> 200, peut pas supprimer les posts des autre ->403 les autres erreurs -> 500
Avancement du service	fini
Classes / Fichiers JS	./routeur/post.js ./schema/post.js
Informations additionnelles	

Nom du web service :	AddFriend
URL	/users/:userId/api/addFriend POST
Description du web service	ajouter un ami
Paramètres en entrée	_id de ami ajoutant
Format de sortie	JSON
Exemples de sortie	<pre>{ "status": 201, "message": "add firend successfully", "details": "" } { "status": 403, "message": "user does not exist" ou "can't add friend with your own" { "status": 500, other errors }</pre>
Erreurs possibles	ajouter ami réussie -> 201, peut pas ajouter ami avec vous-mêmes ou ce utilisateur n'existe pas ->403 les autres erreurs -> 500
Avancement du service	fini
Classes / Fichiers JS	./routeur/friend.js ./schema/user.js
Informations additionnelles	

Nom du web service :	DeleteFriend
URL	/users/:userId/api/deleteFriend POST
Description du web service	supprimer un ami
Paramètres en entrée	
Format de sortie	JSON
Exemples de sortie	<pre>{ "status": 200, "message": "delete friend successfully", "details": "" } { "status": 403, "message": "do not have the friend relationship"} { "status": 500, other errors }</pre>
Erreurs possibles	ajouter ami réussie -> 200, "ne pas avoir de relation d'amitié ->403 les autres erreurs -> 500
Avancement du service	fini
Classes / Fichiers JS	./routeur/friend.js ./schema/user.js
Informations additionnelles	