

Projet 3I017 - Fiche service

15 avril 2019

Table des matières

1. Services utilisateur	1
2. Service ami	6
3. Services message	8
4. Services commentaire	10

1. Services utilisateur

Table 1: Fiche *login*

Nom du <i>Web service</i>	Login
URL du <i>Web service</i>	<i>/Login</i>
Description du <i>Web service</i>	Connexion de l'utilisateur vers le site
Paramètre(s) du <i>Web service</i>	Le login de l'utilisateur et son mot de passe
Format de sortie	<i>{"status" : "ok", "key" : clé_connexion}</i> ou <i>{"status" : "ko", "message" : message_erreur, "code" : data_erreur}</i>
Exemple de sortie	<i>{"status" : "ok", "key" : "5Gb4PH6K..."}</i>
Erreurs possibles	Paramètre(s) manquant, login non présent dans la base de données, mauvais mot de passe, erreur <i>JSON</i> , erreur base de données <i>MySQL</i>
Avancement du <i>Web service</i>	Fini
Classe <i>JAVA</i> en rapport avec le <i>Web service</i>	<i>Servlet Login.java</i>
Informations additionnelles	

Table 2: Fiche *logout*

Nom du <i>Web service</i>	Logout
URL du <i>Web service</i>	<i>Logout</i>
Description du <i>Web service</i>	Déconnexion de l'utilisateur
Paramètre(s) du <i>Web service</i>	La clé de connexion de l'utilisateur
Format de sortie	<code>{"status" : "ok"} ou {"status" : "ko", "message" : message_erreur, "Errorcode" : data_erreur}</code>
Exemple de sortie	<code>{"User already connected.":1007,"Output":"Service Accepted","key":"W3hmt4ki4oSmy5xYSfgrJCurVgBG1xG"}</code>
Erreurs possibles	Paramètre(s) manquant, utilisateur non connecté, erreur <i>JSON</i> , erreur base de données <i>MySQL</i>
Avancement du <i>Web service</i>	Fini
Classe <i>JAVA</i> en rapport avec le <i>Web service</i>	<i>Servlet Logout.java</i>
Informations additionnelles	

Table 3: Fiche *createUser*

Nom du <i>Web service</i>	CreateUser
URL du <i>Web service</i>	<i>CreateUser</i>
Description du <i>Web service</i>	Ajoute un utilisateur à la base de données <i>MySQL</i>
Paramètre(s) du <i>Web service</i>	Nom, Prénom de l'utilisateur, un login et un mot de passe Son adresse e-mail
Format de sortie	<code>{"status" : "ok"} ou {"status" : "ko", "message" : message_erreur, "Errorcode" : data_erreur}</code>

Exemple de sortie	<code>{"user has been added successfully":1,"Output":"Service Accepted","hTSV4H7ONrS6XOTIeAcBa0Lstzfk1HM5":1}</code>
Erreurs possibles	Paramètre(s) manquant, login déjà présent dans la base de données, erreur <i>JSON</i> , erreur base de données <i>MySQL</i>
Avancement du <i>Web service</i>	Fini
Classe <i>JAVA</i> en rapport avec le <i>Web service</i>	<i>Servlet CreatUser.java</i>
Informations additionnelles	(<i>'e' oublié dans creat x</i>))

Table 5: Fiche *PasswordRecovery*

Nom du <i>Web service</i>	PasswordRecovery
URL du <i>Web service</i>	<i>/PasswordRecovery</i>
Description du <i>Web service</i>	Envoie un mail à l'utilisateur avec un nouveau mot de passe.
Paramètre(s) du <i>Web service</i>	L'adresse mail
Format de sortie	<code>{"status" : "ok"}</code> ou <code>{"status" : "ko", "message" : message_erreur, "code" : code_erreur}</code>
Exemple de sortie	<code>{"Message": "MAIL does not exist.", "ErrorCode": 1002}</code>
Erreurs possibles	Paramètre(s) manquant, les deux mails sont différents, erreur <i>JSON</i> , erreur base de données <i>MySQL</i>
Avancement du <i>Web service</i>	Fini
Classe <i>JAVA</i> en rapport avec le <i>Web service</i>	<i>Servlet PasswordRecovery.java</i>
Informations additionnelles	Le Service check seulement si le mail et dans la BD et non si réellement exist.

2. Service ami

Table 10: Fiche *addFriend*

Nom du <i>Web service</i>	AddFriend
URL du <i>Web service</i>	<i>/AddFriend</i>
Description du <i>Web service</i>	L'utilisateur s'abonne à un autre utilisateur et va pouvoir par conséquence voir ses twists.
Paramètre(s) du <i>Web service</i>	L'id de l'utilisateur, l'id de l'ami et la clé de connexion
Format de sortie	<i>{"status" : "ok"} ou {"status" : "ko", "message" : message_erreur, "code" : code_erreur}</i>
Exemple de sortie	<i>{"user1: just followed user2":1,"Output":"Service Accepted"}</i>
Erreurs possibles	Paramètre(s) manquant, utilisateur non connecté, l'utilisateur ou l'abonné n'existent pas dans la base de données <i>MySQL</i> , erreur <i>JSON</i> , erreur base de données <i>MySQL</i>
Avancement du <i>Web service</i>	Pas fonctionnel
Classe <i>JAVA</i> en rapport avec le <i>Web service</i>	<i>Servlet AddFriend.java</i>
Informations additionnelles	Le service marche coté serveur mais ne fonctionne pas coté client.

Table 11: Fiche *removeFriend*

Nom du <i>Web service</i>	removeFriend
URL du <i>Web service</i>	<i>/RemoveFriend</i>
Description du <i>Web service</i>	<i>L'utilisateur se désabonne d'un utilisateur existant dans liste abonnés (dans la base MySql)</i>
Paramètre(s) du <i>Web service</i>	L'id de l'utilisateur, l'id de l'ami et la clé de connexion

Format de sortie	<code>{"status" : "ok"} ou {"status" : "ko", "message" : message_erreur, "code" : code_erreur}</code>
Exemple de sortie	<code>{"user1 has unfollowed user2":1,"Output":"Service Accepted"}</code>
Erreurs possibles	Paramètre(s) manquant, utilisateur non connecté, l'utilisateur ou l'abonné n'existent pas dans la base de données <i>MySQL</i> , erreur <i>JSON</i> , erreur base de données <i>MySQL</i>
Avancement du <i>Web service</i>	Fini
Classe <i>JAVA</i> en rapport avec le <i>Web service</i>	<i>Servlet RemoveFriend.java</i>
Informations additionnelles	

Table 12: Fiche *listFriend*

Nom du <i>Web service</i>	ListFollowers
URL du <i>Web service</i>	/ListFollowers
Description du <i>Web service</i>	Liste de tous les abonnés
Paramètre(s) du <i>Web service</i>	L'id de l'utilisateur, et la clé de connexion
Format de sortie	<ul style="list-style-type: none"> Followers: Array(3) <ul style="list-style-type: none"> 0: {id: 8, user: "user0"} 1: {id: 10, user: "user1"} 2: {id: 11, user: "user2"}
Exemple de sortie	{ "Followers": [{ "id": 8, "user": "chikow" }, { "id": 10, "user": "Delpax" }, { "id": 11, "user": "Delpaxll" }] }
Erreurs possibles	Paramètre(s) manquant, utilisateur non connecté, erreur <i>JSON</i> , erreur base de données <i>MySQL</i>
Avancement du <i>Web service</i>	Fini
Classe <i>JAVA</i> en rapport avec le <i>Web service</i>	<i>Servlet ListFriend.java</i>
Informations additionnelles	

Table 13: Fiche *searchUser*

Nom du <i>Web service</i>	searchFriend
URL du <i>Web service</i>	/searchFriend
Description du <i>Web service</i>	Effectue une recherche d'abonné
Paramètre(s) du <i>Web service</i>	la clé de connexion, login d'abonné
Format de sortie	{ "status" : "ok", "list" : [{ "user_id" : user_login }, ...] } ou { "status" : "ko", "message" : message_erreur, "code" : code_erreur }

Exemple de sortie	<code>{{"Message":"User not connected","ErrorCode":10000}}</code>
Erreurs possibles	Paramètre(s) manquant, utilisateur non connecté, erreur <i>JSON</i> , erreur base de données <i>MySQL</i>
Avancement du <i>Web service</i>	Pas implémenté
Classe <i>JAVA</i> en rapport avec le <i>Web service</i>	<i>Servlet SearchFriend.java</i>
Informations additionnelles	Peut être améliorer avec Map Reduce

3. Services Twist

Table 14: Fiche *AddTwist*

Nom du <i>Web service</i>	AddTwist
URL du <i>Web service</i>	<i>/AddTwist</i>
Description du <i>Web service</i>	Ajoute un twist à la base de données <i>MongoDB</i>
Paramètre(s) du <i>Web service</i>	La clé de connexion et le twist à ajouter
Format de sortie	<code>{"status" : "ok", "added_twist" : twist}</code> ou <code>{"status" : "ko", "message" : twist_erreur, "code" : code_erreur}</code>
Exemple de sortie	<code>{"Message":"User not connected","ErrorCode":10000}</code>
Erreurs possibles	Paramètre(s) manquant, utilisateur non connecté, erreur <i>JSON</i> , erreur base de données <i>MongoDB</i>
Avancement du <i>Web service</i>	Fini
Classe <i>JAVA</i> en rapport avec le <i>Web service</i>	<i>Servlet AddTwist.java</i>
Informations additionnelles	Besoin de d'actualiser la page (côté client) pour un re-render de compensent afin d'afficher le twist

Table 15: Fiche *RemoveTwist*

Nom du <i>Web service</i>	RemoveTwist
URL du <i>Web service</i>	<i>/RemoveTwist</i>
Description du <i>Web service</i>	Supprime un twist de la base de donnée <i>MongoDB</i>
Paramètre(s) du <i>Web service</i>	La clé de connexion et le message à supprimer
Format de sortie	<code>{"status" : "ok", "removed_message" : le_message}</code> ou <code>{"status" : "ko", "message" : message_erreur, "code" : code_erreur}</code>

Exemple de sortie	<code>{"Output":"Service Accepted","Comment has been added":1}</code>
Erreurs possibles	Paramètre(s) manquant, utilisateur non connecté, erreur <i>JSON</i> , erreur base de données <i>MongoDB</i>
Avancement du <i>Web service</i>	Fini
Classe <i>JAVA</i> en rapport avec le <i>Web service</i>	<i>Servlet RemoveTwist.java</i>
Informations additionnelles	Ça beug de temps en temps coté client

Table 16: Fiche *ListTwist*

Nom du <i>Web service</i>	ListTwist
URL du <i>Web service</i>	<i>/ListTwist</i>
Description du <i>Web service</i>	Liste tous les twists d'un utilisateur
Paramètre(s) du <i>Web service</i>	La clé de connexion et le login de l'utilisateur, nombre de twists(static) , trie (boolean)
Format de sortie	<code>{"status" : "ok", "list_message" : [{"twist1" : "hello"},...]} ou {"status" : "ko", "message" : message_erreur, "code" : code_erreur}</code>
Exemple de sortie	<code>{"Output":"Service Accepted","My_twists":[{"date":"Tue Apr 16 11:41:55 CEST 2019","comments":[],"id":"5cb5a3636d7bb05485887b67","login":"neel","content":"hellooooo"}]}</code>
Erreurs possibles	Paramètre(s) manquant, utilisateur non connecté, erreur JSON, erreur base de données MongoDB
Avancement du <i>Web service</i>	Fini (composent appelé sur la page profile)
Classe <i>JAVA</i> en rapport avec le <i>Web service</i>	<i>Servlet ListTwist.java</i>
Informations additionnelles	Une fonctionnalité de choix du nombre de twists à afficher était ajoutée côté serveur mais pas implémentée côté client

Table 17: Fiche *ListTwist utilisateurs suivis*

Nom du <i>Web service</i>	searchMessage
URL du <i>Web service</i>	<i>/searchMessage</i>
Description du <i>Web service</i>	Liste tous les twists des utilisateurs pour qui l'utilisateur est abonné
Paramètre(s) du <i>Web service</i>	La clé de connexion et le login de l'utilisateur, nombre de twists(static) , trie (boolean)

Format de sortie	<code>{"status" : "ok", "list_message" : [{"twist1" : "hello"}, ...]}</code> ou <code>{"status" : "ko", "message" : message_erreur, "code" : code_erreur}</code>
Exemple de sortie	<code>{"status" : "ok", "list" : [{"message1" : "Comment ça va ?"}, {"message2" : "Comme la dernière fois !"}]}</code>
Erreurs possibles	Paramètre(s) manquant, utilisateur non connecté, erreur <i>JSON</i> , erreur base de données <i>MySQL</i>
Avancement du <i>Web service</i>	<i>Réalisée sur la page d'accueil</i>
Classe <i>JAVA</i> en rapport avec le <i>Web service</i>	<i>Servlet ListTwistFollowers.java</i>
Informations additionnelles	Besoin de rafraichire la page côté client pour avoir la liste des twists à jour

4. Services commentaire

Table 18: Fiche *addComment*

Nom du <i>Web service</i>	<i>addComment</i>
URL du <i>Web service</i>	<i>/addComment</i>
Description du <i>Web service</i>	Ajoute un commentaire de twist à la base de donnée <i>MongoDB</i>
Paramètre(s) du <i>Web service</i>	La clé de connexion, l'id du message auquel le commentaire fait référence, et le contenu du commentaire
Format de sortie	<pre>{"status" : "ok", "added_comment" : le_commentaire} ou {"status" : "ko", "message" : message_erreur, "code" : code_erreur}</pre>
Exemple de sortie	<pre>{"status" : "ok", "comment has been added" : "magnifique photo!"}</pre>
Erreurs possibles	Paramètre(s) manquant, utilisateur non connecté, erreur <i>JSON</i> , erreur base de données <i>MongoDB</i>
Avancement du <i>Web service</i>	Fini
Classe <i>JAVA</i> en rapport avec le <i>Web service</i>	<i>Servlet AddComment.java</i>
Informations additionnelles	

Table 19: Fiche *removeComment*

Nom du <i>Web service</i>	<i>removeComment</i>
URL du <i>Web service</i>	<i>/removeComment</i>
Description du <i>Web service</i>	Supprime un commentaire de la base de données <i>MongoDB</i>
Paramètre(s) du <i>Web service</i>	La clé de connexion et l'id du commentaire à supprimer

Format de sortie	<code>{"status" : "ok", "removed_message" : le_message}</code> ou <code>{"status" : "ko", "message" : message_erreur, "code" : code_erreur}</code>
Exemple de sortie	<code>{"Your comment has been removed" : "1"}</code>
Erreurs possibles	Paramètre(s) manquant, utilisateur non connecté erreur <i>JSON</i> , erreur base de données <i>MongoDB</i>
Avancement du <i>Web service</i>	Fini
Classe <i>JAVA</i> en rapport avec le <i>Web service</i>	<i>Servlet RemoveComment.java</i>
Informations additionnelles	

Table 20: Fiche *ListComment*

Nom du <i>Web service</i>	ListComment
URL du <i>Web service</i>	<i>/ListComment</i>
Description du <i>Web service</i>	Liste tous les commentaires d'un twist
Paramètre(s) du <i>Web service</i>	La clé de connexion et l'id du message pour lequel nous voulons afficher les commentaires
Format de sortie	<ul style="list-style-type: none"> • <code>comments</code>: Array(2) <ul style="list-style-type: none"> ○ 0: <ul style="list-style-type: none"> ▪ <code>author</code>: "user" ▪ <code>content</code>: "hey" ▪ <code>date</code>: "Mon Apr 15 15:28:21 CEST 2019" ▪ <code>__proto__</code>: Object ○ 1: <ul style="list-style-type: none"> ▪ <code>author</code>: "user1" ▪ <code>content</code>: "hey there" ▪ <code>date</code>: "Mon Apr 15 15:28:29 CEST 2019" ▪ <code>__proto__</code>: Object
Exemple de sortie	<code>{"status": "ok", "comments": [{"commentaire1": "t'as raison !"}, {"commentaire2": "je ne suis pas tout a fait de votre avis..."}]}</code>
Erreurs possibles	Paramètre(s) manquant, utilisateur non connecté, erreur <i>JSON</i> , erreur base de données <i>MongoDB</i>
Avancement du <i>Web service</i>	Fini
Classe <i>JAVA</i> en rapport avec le <i>Web service</i>	<i>Servlet ListComment.java</i>
Informations additionnelles	Pas besoin de rafraichir coté client affichage instantané