

Projet 3I017 - Fiche service

Fevrier 2018

1 Services utilisateur

Table 1: Fiche *login*

Nom du <i>Web service</i>	Login
URL du <i>Web service</i>	<i>projet_3I017/user/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	<code>{"status": "ok", "key": clé_connexion}</code> ou <code>{"status": "ko", "message": message_erreur, "code": code_erreur}</code>
Exemple de sortie	<code>{"status": "ok", "key": "AVKSQM..."}</code>
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>projet_3I017/user/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"}</code> ou <code>{"status": "ko", "message": message_erreur, "code": code_erreur}</code>
Exemple de sortie	<code>{"status": "ko", "message": "Not connected", "code": "1005"}</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>projet_3I017/user/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>	Le login de l'utilisateur, son mot de passe, son prénom, son nom et son adresse e-mail
Format de sortie	<code>{"status": "ok"}</code> ou <code>{"status": "ko", "message": message_erreur, "code": code_erreur}</code>
Exemple de sortie	<code>{"status": "ko", "message": "User already exists", "code": "1001"}</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 createUser.java</i>
Informations additionnelles	

2 Services message

Table 4: Fiche *addMessage*

Nom du <i>Web service</i>	<i>addMessage</i>
URL du <i>Web service</i>	<i>projet_3I017/message/addMessage</i>
Description du <i>Web service</i>	Ajoute un message à la base de données <i>MongoDB</i>
Paramètre(s) du <i>Web service</i>	La clé de connexion et le message à ajouter
Format de sortie	<code>{"status": "ok", "added_message": le_message}</code> ou <code>{"status": "ko", "message": message_erreur, "code": code_erreur}</code>
Exemple de sortie	<code>{"status": "ok", "added_message": "Mon super message !"} </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 addMessage.java</i>
Informations additionnelles	À peaufiner en cas de définition d'une classe <i>Message</i>

Table 5: Fiche *removeMessage*

Nom du <i>Web service</i>	<i>removeMessage</i>
URL du <i>Web service</i>	<i>projet_3I017/message/removeMessage</i>
Description du <i>Web service</i>	Supprime un message de la base de données <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>{"status": "ok", "removed_message": "Mon super message ! pas si super que ca..."}</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 removeMessage.java</i>
Informations additionnelles	À peaufiner en cas de définition d'une classe <i>Message</i>

Table 6: Fiche *listMessage*

Nom du <i>Web service</i>	listMessage
URL du <i>Web service</i>	<i>projet_3I017/message/listMessage</i>
Description du <i>Web service</i>	Liste tous les messages d'un utilisateur
Paramètre(s) du <i>Web service</i>	La clé de connexion et le login de l'utilisateur
Format de sortie	{ "status": "ok", "list_message": [{ "message1": message1 }, { "message2": message2 }, ...] } ou { "status": "ko", "message": message_erreur, "code": code_erreur }
Exemple de sortie	{ "status": "ok", "list_message": [{ "message1": "De nouveau mon super message !" }, { "message2": "Vive Jason Statham !" }] }
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 listMessage.java</i>
Informations additionnelles	À peaufiner en cas de définition d'une classe <i>Message</i>

3 Service ami

Table 7: Fiche *addFriend*

Nom du <i>Web service</i>	addFriend
URL du <i>Web service</i>	<i>projet_3I017/friend/addFriend</i>
Description du <i>Web service</i>	Ajoute une relation d'amitié dans la base de données <i>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	{ "status": "ok" } ou { "status": "ko", "message": message_erreur, "code": code_erreur }
Exemple de sortie	{ "status": "ok" }
Erreurs possibles	Paramètre(s) manquant, utilisateur non connecté, l'utilisateur ou l'ami 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 addFriend.java</i>
Informations additionnelles	

Table 8: Fiche *removeFriend*

Nom du <i>Web service</i>	removeFriend
URL du <i>Web service</i>	<i>projet_3I017/friend/removeFriend</i>
Description du <i>Web service</i>	Supprime une relation d'amitié de la base de données <i>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"}</code> ou <code>{"status": "ko", "message": message_erreur, "code": code_erreur}</code>
Exemple de sortie	<code>{"status": "ko", "message": "Missing parameters", "code": "-1"}</code>
Erreurs possibles	Paramètre(s) manquant, utilisateur non connecté, l'utilisateur ou l'ami 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 9: Fiche *listFriend*

Nom du <i>Web service</i>	listFriend
URL du <i>Web service</i>	<i>projet_3I017/friend/listFriend</i>
Description du <i>Web service</i>	Liste tous les amis d'un utilisateur
Paramètre(s) du <i>Web service</i>	L'id de l'utilisateur, et la clé de connexion
Format de sortie	<code>{"status": "ok", "list": [{"ami1": nom_prenom_ami1}, {"ami2": nom_prenom_ami2}, ...]}</code> ou <code>{"status": "ko", "message": message_erreur, "code": code_erreur}</code>
Exemple de sortie	<code>{"status": "ok", "list": [{"ami1": "Jason Statham"}, {"ami2": "Brian Basco"}]}</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 listFriend.java</i>
Informations additionnelles	À peaufiner en cas de définition d'une classe <i>Personne</i>