

API Serveur Jamespot.pro

V2.0

Documentation version 2.0.29

Table des matières

Table des matières	2
Introduction	4
Principes	5
1.1. Paramètres obligatoires	6
1.2. Exemples d'appel	8
1.3. Format de retour	9
Référence de l'API	10
1.4. User	11
1.4.1. Get	11
1.4.2. GetByField	12
1.4.3. Create	12
1.4.4. Update	15
1.4.5. List	15
1.4.6. Delete	16
1.4.7. Activity	16
1.4.8. Notify	17
1.4.9. Contacts	17
1.4.10. Spots	18
1.5. Spot	19
1.5.1. Get	19
1.5.2. GetByField	19
1.5.3. Create	19
1.5.4. Update	21
1.5.5. List	21
1.5.6. SetUserLevel	22
1.5.7. GetUserLevel	22
1.5.8. Members	22
1.5.9. Articles	23
1.5.10. Delete	23
1.5.11. Activity	24
1.5.12. Notify	24
1.6. Article	25
1.6.1. Get	25
1.6.2. GetByField	25
1.6.3. Create	25
1.6.4. Update	27
1.6.5. List	28
1.6.6. Delete	28
1.6.7. UploadFile	29
1.6.8. SetImage	29
1.6.9. DeleteFile	30
1.6.10. Comments	30
1.6.11. AddComment	30

1.6.12.	DeleteComment	30
1.7.	Link.....	31
1.7.1.	Create	31
1.7.2.	List.....	31
1.7.3.	Delete	31
1.8.	Global.....	32
1.8.1.	ActivityStream.....	32
1.8.2.	Traduction.....	32
1.8.3.	Profiles.....	32
1.8.4.	Version.....	33
1.8.5.	UserSynchro	33
1.8.6.	Search.....	33
1.8.7.	Fields	34
1.9.	Taxonomie	35
1.9.1.	Get.....	35
1.9.2.	Create	35
1.9.3.	addConcept.....	35
1.9.4.	Import.....	36
1.10.	Communauté.....	37
1.10.1.	Get.....	37
1.10.2.	List.....	37
1.11.	Catégorie de groupe	38
1.11.1.	Insert	38
1.11.2.	GetByLabel.....	38
1.11.3.	GetCategories.....	38
1.11.4.	Rename.....	38
1.11.5.	Delete	39

Introduction

La plateforme Jamespot.pro permet aux organisations de se doter d'un réseau social privatif, puis de partager et commenter l'information qui correspond à leurs centres d'intérêts, de créer, puis d'élargir une communauté de personnes autour de ces sujets.

Ce document décrit l'API de Jamespot.pro. Cette API permet d'effectuer les principales opérations de lecture/écriture. Elle propose un ensemble de fonctions regroupées par grand type d'objets :

Objet	Fonctions
User	create() get() - getByField() contacts() – spots() update() list() delete() activity() - notify()
Spot	create() get()- getByField() update() list() delete() setUserLevel() - getUserLevel() members() - articles() activity() - notify()
Article	create() get()- getByField() update() list() delete() uploadFile(), deleteFile(), setImage() comments(), addComment(), deleteComment()
Link	create(), list(), delete()
Global	version(), translation() profiles() userSynchro()

Principes

L'ensemble des fonctions d'API est appelé à travers un point d'accès unique :

`https://[plateforme].jamespot.pro/api/api.php?<paramList>`

La liste de paramètres <paramList> permet de définir les caractéristiques de l'appel demandé à l'API.

Le tableau suivant résume les paramètres potentiels de cette liste.

Les paramètres et les valeurs sont sensibles à la casse.

1.1. Paramètres obligatoires

Objet	Fonctions
object	Description : Objet sur lequel porte l'appel de fonction d'API Alias : o Exemples de valeurs : spot, article, user, spot_user
function	Description : Fonction demandée sur l'objet Alias : f Exemples de valeurs : get, create, delete , update
version	Description : Version de l'API utilisée Alias : v Exemple : 2.0 Valeur par défaut : 1.0 ATTENTION : pour utiliser l'API en version 2.0, il faut absolument fournir ce paramètre.

Pour authentifier les appels d'API, on peut au choix :

- Passer dans l'entete d'appel les valeurs
X-Com-Jamespot-Module-Name : Application ID
X-Com-Jamespot-Module-Key : Valeur de la clé associée
- Ajouter des arguments dans l'appel

Objet	Fonctions
module	Description : Application ID déclaré dans la plateforme. Alias : m Exemples de valeur : EXT-applicationName

Objet	Fonctions
<p>date</p>	<p>Description : Date limite de validité de l'appel. YYYY-MM-DDThh:mm:ssTZD</p> <p>Alias : d</p> <p>Exemples de valeur : 2010-10-26T11:00:00Z OU 2010-10-26T12:00:00+01:00</p> <p>Php generation : date('Y-m-d\TH:i:sP) ;</p>
<p>key</p>	<p>Description : Clé de sécurisation de l'appel. La valeur est obtenue par la fonction suivante : md5(applicationID '-' date '-' cléModule)</p> <p>Alias : k</p> <p>Exemples de valeur : Pour md5(EXT-applicationName-2009-11-05T13:15:30-cledesecurite) , k= 3d210275380b3431b6accfaeadbb8612</p>

1.2. Exemples d'appel

Les paramètres peuvent être fournis en GET ou en POST.

Exemple GET :

Pour obtenir la description json du spot, dont l'idspot est 3, pour la plateforme 'foo' on appelle l'URL :

[https://foo.jamespot.pro/api/api.php?k= d41d8cd98f00b204e9800998ecf8427e&d=2010-07-07T00:00:00&m=EXT-application&o=spot&f=get&idspot=3](https://foo.jamespot.pro/api/api.php?k=d41d8cd98f00b204e9800998ecf8427e&d=2010-07-07T00:00:00&m=EXT-application&o=spot&f=get&idspot=3)

1.3. Format de retour

La valeur JSON de retour est toujours un objet comportant trois champs RC (Returned Code), STATS (Statistiques), VAL (Valeur).

RC est un objet contenant :

- Le code de retour entier : Lors d'une exécution correcte, le code de retour est 0. Dans les autres cas, il est différent de 0.
- Le message d'erreur. Ce message est vide lors d'une exécution correcte.

STATS est un objet contenant

- TIME Le temps d'exécution
- Et d'autres paramètres optionnels qui dépendent de la fonction API

VAL est l'objet retourné par l'exécution de la fonction.

Exemples

Retour correct	<pre>{ "RC":{"CODE":0,"MSG":""}, "STATS":{"TIME":74}, "VAL":{"idSpot":"1234",[...]}}</pre>
Retour erreur	<pre>{ "RC":{"CODE":10004,"MSG":"Missing parameter:IdSpot"}, "STATS":{"TIME":0}, "VAL":"" }</pre>

Référence de l'API

L'API est organisée par objets sur lesquels porte la fonction à exécuter.

Les objets possibles sont :

- user
- spot
- article
- global

Pour chaque objet, on décrit ensuite les fonctions disponibles, leurs paramètres d'entrée, le traitement associé, le paramètre retourné.

1.4. User

1.4.1. Get

Paramètre	Valeur
object	user
function	get
Optionnels	
idUser	<idUser> Alias : idUser Exemples de valeur 15
mail	<mail> Description : Mail de l'utilisateur (utilisé pour ce connecter à son compte dans l'interface utilisateur). Exemple de valeur : username@live.com

L'un au moins des paramètres idUser/mail doit être fourni.

Cette fonction retourne des informations sur l'utilisateur dont le mail ou l'identifiant est passé en paramètre.

On peut avoir par exemple :

```
{ 'RC' : '[...]', 'STATS' : '[...]',  
  'VAL' : {idUser : 1,  
          mail : 'user1@mail.com', ...  
          pseudo : 'user1'  
          img : 'http://imgUser1.png'  
        }  
}
```

1.4.2. GetByField

Paramètre	Valeur
object	user
function	getByField
name	Nom du champ sur laquelle porte la requete.
value	Valeur du champ demandé Exemple de valeur : username@live.com

Cette fonction retourne des informations sur l'utilisateur dont le champ 'name' a la valeur 'value'. Si plusieurs utilisateurs peuvent répondre à cette contrainte, l'utilisateur avec l'idUser le plus élevé est retourné.

1.4.3. Create

Paramètre	Valeur
object	user
function	create
Mail	<mail> Mail de l'utilisateur à créer
Password	<pwd> Mot de passe de l'utilisateur à créer
Pseudo	<pseudo> Pseudo de l'utilisateur à créer
Image	<image> Fichier image attaché
_image_base64	<_image_base64> Fichier image encodé en base64
Language	<language> Langue de l'utilisateur, format TLD en deux lettres Exemples de valeurs : en pour anglais, fr pour français
Country	<pays> Pays de l'utilisateur Exemples de valeurs : en, fr, es
Role	<role> Role de l'utilisateur Exemples de valeur : Administrator, User, External
Optionnels	
Champs de l'import CSV	Tous les champs utilisables dans l'import CSV sont utilisables avec les même noms de colonnes, y compris <i>Profile</i> , ou <i>property_xxx</i> pour fournir des propriétés additionnelles de l'utilisateur.
activity	<booléen> Génère l'activité de création dans le wall
welcomeMail	<booléen> Envoie un mail à l'utilisateur avec ces informations de compte
connectUsers	<booléen> Ajoute tous les utilisateurs comme contact
property_xxx	<valeur de la propriété> Ajoute la propriété « xxx », avec la valeur associée, à l'utilisateur créé.

Communities	<Liste idCommunity> Liste d'IdCommunity séparés par des virgules Ajoute l'utilisateur à la/aux communautés renseignées Exemples de valeur : « 5 » ou « 6,7,8 »
--------------------	---

Cette fonction crée un nouvel utilisateur.

Si le pseudo ou le mail existent déjà, l'API retourne une erreur.

Le retour est équivalent à user.get()

1.4.4. Update

Paramètre	Valeur
object	user
function	update
Mail	<mail> Mail de l'utilisateur à modifier
Optionnels	
Champs de l'import CSV	Tous les champs de l'import CSV sont utilisables avec les mêmes noms de colonnes.

L'update ne fournit que les paramètres à modifier.

Le retour est équivalent à user.get().

1.4.5. List

Paramètre	Valeur
object	user
function	list
Optionnels	
name	<name> Exemple de valeur : mail
id	<id> id de l'utilisateur à partir duquel la limitation s'applique. Les id des utilisateurs retournés seront strictement supérieurs. Exemple de valeur : 15
types[]	<array> Liste des types utilisateurs à retourner. Exemple : types[]=user&types[]=user1
limit	<limit> Nombre d'éléments à retourner (default 50)

Retourne une liste d'id utilisateur + champ définis par l'argument « name » s'il est fournis.

Exemple :

```
{ 'RC' : [...], 'STATS' : [...],  
  'VAL' : [  
    {"idUser": "1", "pseudo": "user1"},  
    {"idUser": "2", "pseudo": "user2"}  
  ]  
}
```

```
]
}
```

1.4.6. Delete

Paramètre	Valeur
object	user
function	delete
Optionnels	
idUser ou Mail	<idUser>

Efface un utilisateur, ses spots, ses articles et ses commentaires.

Retourne seulement le code d'erreur standard, avec code = 0 lorsque l'exécution est correcte.

1.4.7. Activity

Paramètre	Valeur
object	user
function	activity
idUser	<idUser>
wording ou act_resource	<wording> chaines de l'activité directement <act_ressource> appel d'une resourceKey dans la langue de l'utilisateur.
Optionnels	
json_params	<json> Paramètres a insérer dans la ressource
act_url	url du lien retour vers la page émetteur de l'activité.
act_title	Titre du lien

Ajoute une activité liée à l'utilisateur, et distribuée au followers.

1.4.8. Notify

Paramètre	Valeur
object	user
function	notify
idUser	<idUser> émetteur
idUserTo	<idUser> receveur
Ressources	<json> Json contenant 3 entrées (valeur = ressourceKey): -subject => sujet de l'email -content => contenu du mail -contentDelayed => contenu du mail différé

Envoie un email à un utilisateur.

Les contenus sont au format HTML.

Suivant la configuration d'alerte de l'utilisateur, la valeur 'content' ou 'contentDelayed' sera utilisée :

- si l'utilisateur a configuré une alerte temps réel, on utilise la valeur 'content'
- si l'utilisateur a configuré une alerte différée, tous les journalière ou hebdomadaire, c'est 'contentDelayed' qui sera utilisé.

1.4.9. Contacts

Paramètre	Valeur
object	user
function	contacts
idUser	<idUser>
Optionnels	
limit	<limit> Nombre maximal de contacts par page. 50 par défaut
page	<page> Numéro de la page à retourner

Retourne la liste des contacts de l'utilisateur.

1.4.10. Spots

Paramètre	Valeur
object	user
function	spots
idUser	<idUser>
Optionnels	
limit	<limit> Nombre maximal de spots par page. 50 par défaut
page	<page> Numéro de la page à retourner

Retourne la liste des spots de l'utilisateur, avec pour chaque spot, son niveau de participation dans le spot.

1.5. Spot

1.5.1. Get

Paramètre	Valeur
object	spot
function	get
idSpot	<idspot> (entier) Identifiant du spot.

Retourne la description d'un spot : IdSpot, Titre, Edito, Image, Skin, Public.

1.5.2. GetByField

Paramètre	Valeur
object	spot
function	getByField
name	Nom du champ sur laquelle porte la requete.
value	Valeur du champ demandé Exemple de valeur : username@live.com

Cette fonction retourne des informations sur le spot dont le champ 'name' a la valeur 'value'. Si plusieurs spots peuvent répondre à cette contrainte, le spot avec l'idSpot le plus élevé est retourné.

1.5.3. Create

Paramètre	Valeur
object	spot
function	create
idUser	<idUser> : Le créateur du Spot
category	<category> Catégorie du spot. Si la plateforme n'utilise pas les catégories, il faut utiliser la valeur 1.
title	<Titre du spot>
Optionnels	
public	Boolean (0, 1) Par défaut : 1
edito	<Texte de l'edito>
lang	<language> (sur 2 caractères) Par défaut on utilisera la langue de l'utilisateur

Crée un nouveau spot dans la catégorie, avec le titre donné, la configuration de skin par défaut. Le 'user' est le créateur du spot.

Retourne un objet identique à spot.get avec getArticles=0 et getMembers=0 .

1.5.4. Update

Paramètre	Valeur
object	spot
function	update
idSpot	<idSpot>
Optionnels	
category	<category>
title	<Titre du spot>
public	Boolean (0, 1) Par défaut : 1
idUser	<idUser>
edito	<Texte de l'édito>

Met à jour le spot.

Retourne un objet identique à spot.get avec getArticles=0 et getMembers=0.

1.5.5. List

Paramètre	Valeur
object	spot
function	list
Optionnels	
name	<name> Exemple de valeur : Description
id	<id> Description : id du spot à partir duquel la limitation s'applique. (Les id retournés seront strictement supérieur) Exemple de valeur 15
types[]	<array> Liste des types spot à retourner Exemple : types[]=spot&types[]=thinkTank
limit	<limit> Nombre d'élément à retourner (default 50)

Retourne une liste d'id spot + champ définis par l'argument « name » s'il est fournis.

Voir exemple de retour sur la méthode user list.

1.5.6. SetUserLevel

Paramètre	Valeur
object	spot
function	setUserLevel
idSpot	<idSpot>
idUser	<idUser>
level	<Level> Le "level" est un entier : CREATOR=0; ADMINISTRATOR=1; CONTRIBUTOR=2; SUBSCRIBER=3; VIEWER=4; VISITOR=5;

Ajoute un utilisateur au Spot.

Retourne seulement le code d'erreur standard, avec code = 0 lorsque l'exécution est correcte.

1.5.7. GetUserLevel

Paramètre	Valeur
object	spot
function	getUserLevel
idSpot	<idSpot>
idUser	<idUser>

Retourne le rôle d'un utilisateur dans un Spot.

1.5.8. Members

Paramètre	Valeur
object	spot
function	members
idSpot	<idSpot>

Retourne la liste des utilisateurs membres d'un spot, avec leur rôle.

1.5.9. Articles

Paramètre	Valeur
Object	spot
function	articles
idSpot	<idSpot>
Optionnels	
limit	<limit> Nombre d'éléments à retourner (default 50)
id	<id> id de l'article à partir duquel la limitation s'applique. Les id des articles retournés seront strictement supérieurs. Exemple de valeur : 15

Retourne la liste des articles (tous types confondus) d'un spot, par ordre d'idArticle. La valeur de dateOrder sert à classer les articles dans le spot.

1.5.10. Delete

Paramètre	Valeur
object	spot
function	delete
idSpot	<idSpot>

Efface un spot.

Retourne seulement le code d'erreur standard, avec code = 0 lorsque l'exécution est correcte.

1.5.11. Activity

Paramètre	Valeur
object	spot
function	activity
idUser	<idUser>
idSpot	<idSpot>
wording ou act_resource	<wording> chaines de l'activité directement <act_resource> appel d'une resourceKey dans la langue de l'utilisateur.
Optionnels	
json_params	<json> Paramètres a insérer dans la ressource
act_url	url du lien vers l'émetteur
act_title	Titre du lien

Ajoute une activité associée au spot, et distribuée suivant la logique du spot.

1.5.12. Notify

Paramètre	Valeur
object	spot
function	notify
idUser	<idUser>
idSpot	<idSpot>
ressources	<json> Json contenant 3 entrées (valeur = resourceKey): -subject => sujet de l'email -content => contenu du mail -contentDelayed => contenu du mail différé

Envoie un email à tous les membres d'un spot.

1.6. Article

1.6.1. Get

Paramètre	Valeur
object	article
function	get
idArticle	<idarticle> entier

Retourne l'ensemble des champs de l'article et des attributs custom, et des fichiers attachés.

1.6.2. GetByField

Paramètre	Valeur
object	article
function	getByField
name	Nom du champ sur laquelle porte la requête.
value	Valeur du champ demandé Exemples de valeur : username@live.com
type	Type de la valeur du champ demandé. Les valeurs possibles sont : <i>name, bigint, tinyint, text, longtext, float, date, timestamp</i>

Cette fonction retourne des informations sur l'article dont le champ 'name' a la valeur 'value'. Si plusieurs articles peuvent répondre à cette contrainte, l'article avec l'idArticle le plus élevé est retourné.

1.6.3. Create

Paramètre	Valeur
object	article
function	create
idUser ou mail	<idUser> ou <mail> de l'utilisateur
publishTo	<uris> Exemple de valeur : spot/1, spot/2, user/2
type	<Type de l'article> « mpArticle » pour les types par défaut.
title	<Titre de l'article>
Optionnels	
description	<contenu de l'article>
url	<url>
edito	<Edito de l'article>
image	<string> : « url »
UrlImage	<img_url>
tags	<tags> de l'article séparés par des virgules
Sendalert	<string> « on » Envoie un mail à tous les membres de l'audience.
xxx	Où 'xxx' est le nom de l'attribut supplémentaire, dépendant du type d'article.

Ajoute un article pour l'audience spécifiée dans le publishTo. Le idUser en devient l'auteur. Retourne l'équivalent de Article.get().

1.6.4. Update

Paramètre	Valeur
object	article
function	update
idArticle	<idarticle> entier.
Optionnels	
title	<Titre de l'article>
text	<contenu de l'article>
idUser	<idUser>
idSpot	<idSpot>
url	<url>
tags	<tags>
xxx	Où 'xxx' est le nom de l'attribut supplémentaire, dépendant du type d'article.

Modifie les champs fournis pour l'article demandé. L'un au moins des champs Title, Content, IdSpot, IdUser, Url ou Img doit être fourni.

Retourne l'équivalent de `article.get()`.

1.6.5. List

Paramètre	Valeur
object	article
function	list
Optionels	
id	<id> id de l'article à partir duquel la limitation s'applique. Les id des articles retournés seront strictement supérieurs. Exemple de valeur : 15
limit	<limit> Nombre d'éléments à retourner (défaut 50)
idUser	<idUser> Auteur des articles
idSpot	<idSpot> Spot contenant les articles
type	<type> Type des articles
itemFormat	<format> Format de retour des articles de la liste. Valeurs possibles : 'header', 'article' Par défaut : header

Retourne une liste d'articles formatés selon 'itemFormat', ordonnés par idArticle ascendant, qui respectent les conditions fournies.

1.6.6. Delete

Paramètre	Valeur
object	article
function	delete
idArticle	<idarticle>

Efface l'article demandé.

Retourne seulement le code d'erreur standard, avec code = 0 lorsque l'exécution est correcte.

1.6.9. DeleteFile

Paramètre	Valeur
object	article
function	deleteFile
idFile	<idafile>

Supprime le fichier idFile.

1.6.10. Comments

Paramètre	Valeur
Object	article
function	comments
idArticle	<IdArticle>

Retourne la liste des commentaires de l'article.

1.6.11. AddComment

Paramètre	Valeur
Object	article
function	addComment
idUser	<idUser>
ou	ou
mail	<mail> de l'utilisateur
idArticle	<IdArticle>
text	Texte du commentaire à ajouter.

Ajoute un commentaire à l'article demandé pour cet utilisateur. Retourne la structure de commentaire avec l'identifiant du commentaire.

1.6.12. DeleteComment

Paramètre	Valeur
object	Article
function	deleteComment
idComment	Identifiant du commentaire à supprimer.

Efface le commentaire

1.7. Link

1.7.1. Create

Paramètre	Valeur
object	link
function	create
type	<Type du lien>
srcUri	URI de l'objet source du lien
targetUri	URI de l'objet cible du lien
Optionnels	
value	Valeur texte associée

Ajoute un lien typé, depuis la source, vers la cible. Ces uris peuvent référencer aussi bien des contenus, des utilisateurs, ou des spots.

1.7.2. List

Paramètre	Valeur
object	link
function	list
Optionnels	
type	<Type du lien>
srcUri	URI de l'objet source du lien
targetUri	URI de l'objet cible du lien

Retourne la liste des liens filtrés suivant les contraintes de type, de source ou de cible. Il faut au minimum fournir la source ou la cible des liens.

1.7.3. Delete

Paramètre	Valeur
object	link
function	delete
type	<Type du lien>
idLink	Identifiant du lien

Suppression du lien.

1.8. Global

1.8.1. ActivityStream

Paramètre	Valeur
Object	global
Function	activityStream
startId	<id> Identifiant de la première activité à retourner
Limit	<limit> Nombre d'activités à retourner

Retourne le flux d'activités de la plateforme, e démarrant à startId, contenant jusqu'à 'limit' activités.

1.8.2. Traduction

Paramètre	Valeur
object	global
function	translation
Lang	<language> Exemple de valeurs : en, fr, pl
stringArray	<json> Tableau json de resourceKey Exemple de valeurs : ["GLOBAL_Spot", "GLOBAL_Search"]

Retourne les traductions des chaines demandées sur la plateforme.

1.8.3. Profiles

Paramètre	Valeur
Object	global
function	profiles

Retourne la liste des profils disponibles sur la plateforme. Utile pour la création des utilisateurs.

1.8.4. Version

Paramètre	Valeur
Object	global
function	version

Retourne la version de l'API utilisée.
Utile pour tester votre connection et vos paramétrages.

1.8.5. UserSynchro

Paramètre	Valeur
object	global
function	userSynchro
publishTo	<uri> Exemple de valeur : spot/5
title	<titre de l'article>
idUser	<idUser> Créateur de la demande de synchronisation
upload	<file> Fichier CSV contenant les utilisateurs à synchroniser. Pour plus d'informations, voir la documentation de la fonction <code>article.uploadFile()</code>

Retourne la fiche de synchronisation, avec son identifiant, et contenant le log de traitement de la demande de synchronisation.

1.8.6. Search

Paramètre	Valeur
object	global
function	search
adminsearch	<true false> Paramètre facultatif. Contexte de sécurité*
Mail idUser	< adresse email identifiant numérique > Paramètre facultatif. Contexte de sécurité*
query	< requête solr > Paramètre obligatoire

Retourne le résultat d'une requête Solr en fonction du contexte paramétré. Pour plus d'information : <http://lucene.apache.org/solr/>.

1.8.7. Fields

Paramètre	Valeur
object	global
function	fields
type	<string> Paramètre obligatoire. Type de l'objet dont on souhaite obtenir les champs

Retourne les champs d'un objet dont le type est donné.

1.9. Taxonomie

1.9.1. Get

Paramètre	Valeur
object	taxonomy
function	get
uri	<uri> Identifiant de la taxonomie

Retourne la description d'une taxonomie : uri, type, title, dateCreation, mandatory.

1.9.2. Create

Paramètre	Valeur
object	taxonomy
function	create
idUser	<idUser>
title	<titre de la taxonomie>
type	<type> type de la taxonomie : taxonomyOpen taxonomyClose taxonomySkos
mandatory	<0 1> Caractère obligatoire ou non de la taxonomie0

Création d'une nouvelle taxonomie.

1.9.3. addConcept

Paramètre	Valeur
object	taxonomy
function	addConcept
idUser	<idUser>
uri	<uri> Identifiant de la taxonomie
title	<titre de la taxonomie>

Ajoute un concept pour une taxonomie existante.

1.9.4. Import

Paramètre	Valeur
object	taxonomy
function	import
idUser	<idUser>
uri	<uri> Identifiant de la taxonomie
file	<file> Fichier .rdf contenant l'arbre à transférer. Pour plus d'informations, voir la documentation de la fonction <code>article.uploadFile()</code>

1.10. Communauté

1.10.1. Get

Paramètre	Valeur
object	community
function	get
idCommunity	<idCommunity> Exemple de valeur 15

Retourne les informations relatives à la communauté répondant à l'identifiant numérique fourni : 'id', 'title', 'description', 'status'.

1.10.2. List

Paramètre	Valeur
object	community
function	list
[technical]	<0 1 null>

Retourne les données relatives à un ensemble de communautés défini selon le paramètre optionnel 'technical' toutes, techniques ou non techniques.

1.11. Catégorie de groupe

1.11.1. Insert

Paramètre	Valeur
object	category
function	insert
label	< label> Exemple de valeur « Catégorie n°1 »

Insert une catégorie de groupe dont le label est précisé via le champ dédié. Retourne l'identifiant numérique de la catégorie créée.

1.11.2. GetByLabel

Paramètre	Valeur
object	category
function	getByLabel
label	< label> Exemple de valeur « Catégorie n°1 »

Retourne l'identifiant numérique de la catégorie trouvée.

1.11.3. GetCategories

Paramètre	Valeur
object	category
function	getCategories

Retourne la liste de l'ensemble des catégories de groupe de la plateforme.

1.11.4. Rename

Paramètre	Valeur
object	category
function	rename
Id	<id> identifiant numérique de la catégorie Exemple de valeur 15
label	< label> nouveau label Exemple de valeur « Catégorie renommée »

Retourne la liste de l'ensemble des catégories de groupe de la plateforme, dont la catégorie renommée.

1.11.5. Delete

Paramètre	Valeur
object	category
function	rename
Id	<id> identifiant numérique de la catégorie Exemple de valeur 15

Retourne la liste de l'ensemble des catégories de groupe de la plateforme, à l'exception de la catégorie supprimée.