



SPIPSERVICE

documentation technique

SpipService est un outil permettant d'actualiser un site Spip depuis n'importe quel environnement. Ce plugin permet à un service tiers de gérer le contenu du système via des services web.

"SpipService est une interface entre vous et le reste du monde..."

TABLE DES MATIÈRES

La communication avec spipservice (protocole http)

Services Web - Lecture

▪ getauteurloggedin	4
▪ search	4
▪ getbystatut	4
▪ getbyauteur	5
▪ getchildren	5
▪ getrubriquedata	6
▪ getarticledata	6
▪ getbrevedata	7
▪ getdocumentdata	7
▪ getarticledocuments	7
▪ getbrevedocuments	8
▪ getrubriquedocuments	8

Services Web - Utilitaires

▪ authentication.....	9
▪ clearcache.....	9
▪ clearcacheimage	9

Services Web - Ecriture

▪ setrubrique	10
▪ setarticle.....	11
▪ setbreve	12
▪ setdocument	12
▪ adddocumentrubrique	13
▪ adddocumentarticle.....	13
▪ adddocumentbreve	14
▪ addlogoarticle.....	14
▪ addlogobreve	15
▪ addlogorubrique	15
▪ institutearticle	16
▪ institutebreve.....	16
▪ deletearticle.....	17
▪ deleterubrique	17
▪ deletedocumentarticle	17
▪ deletedocumentbreve	18
▪ deletedocumentrubrique	18
▪ deletelogoarticle	19
▪ deletelogobreve	19
▪ deletelogorubrique	19

Surcharger/définir des services web

▪ monserviceperso.....	21
------------------------	----

Utiliser un format de donnée particulier

▪ monserviceavecformatperso	22
-----------------------------------	----

LA COMMUNICATION AVEC SPIPSERVICE (PROTOCOLE HTTP)

La communication se fera grâce au protocole HTTP par envoi de requête en méthode GET/POST.

L'URL de la requête est connue par le client, elle s'adressera à un site pourvu du plugin SpipService.

SpipService servira chaque demande de façon totalement décorélée (isolation). Il n'est pas possible de demander plusieurs services dans une même requête, de même qu'il n'est pas possible d'envoyer plusieurs requête en considérant qu'elles seront traitées en respectant l'ordre d'envoi.

La requête HTTP devra être authentifiée par Spip, elle devra donc comporter l'entête suivante (Authentication Basic HTTP) :

Authorization : Basic [login:mdp]

① [login:mdp] devra être encodé en base64 (pour plus d'informations, regardez sur Internet le principe d'authentification HTTP Basic)

① [login:mdp] sont ceux du rédacteur de Spip qui utilise le service web

La requête HTTP sera toujours composée de 3 paramètres (clé:valeur):

- 'format'
- 'service'
- 'data'

Le paramètre 'format' pourra valoir l'une des 2 valeurs suivantes :

- 'xml'
- 'json'

① (si le format inconnu, les services natifs disfonctionneront!)

Le paramètre 'service' contiendra le nom du service web demandé

Le paramètre 'data' contiendra un ensemble de données, il est expliqué plus bas qu'elles sont les valeurs attendues pour chaque service. Il existe 2 formats de données supportés nativement

- XML
- JSON

① Vous l'aurez compris, la valeur du premier paramètre indique à SpipService quel sera le format de données à utiliser pour interpréter le troisième paramètre.

Les valeurs de retour seront au format spécifié par 'format' (sauf si le format est inconnu).

Ce document contient l'exhaustivité des services web proposés par SpipService. Pour chaque service, vous trouverez l'ensemble des valeurs attendues ainsi que l'URL à appeler.

SERVICES WEB - LECTURE

■ getauteurloggedin

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Récupérer les données de l'utilisateur connecté (dont l'identifiant et le mot de passe sont indiqués dans l'entête 'Authorization' de la requête HTTP)

@ service = getauteurloggedin
@ format = xml/json
@ data =

■ search

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Récupérer les articles, brèves, rubriques répondant à la recherche

@ service = search
@ format = xml/json
@ data =

```
<root>
  <search>[valeur]</search>
  <types>rubrique, breve, article</types>
</root>
OU
{"search":"[valeur]","types":"rubrique, breve, article"}
```

① *search* : chaîne de caractères (au moins 3 caractères)

① *types* : permet de filtrer les types retournés (aucun type revient à demander tous les types pris en charge)

■ getbystatut

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Récupérer les articles, brèves ayant le statut spécifié

@ service = getbystatut
@ format = xml/json
@ data =

```
<root>
  <statut>[prepa/prop/publie/refuse/poubelle]</statut>
  <types>breve, article</types>
</root>
OU
{"statut":"[prepa/prop/publie/refuse/poubelle]","types":"breve, article"}
```

① *statut* : statut demandé

① *types* : permet de filtrer les types retournés (aucun type revient à demander tous les types pris en charge)

■ getbyauteur

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Récupérer les articles de l'auteur spécifié

@ service = **getbyauteur**

@ format = **xml/json**

@ data =

```
<root>
  <auteur>[valeur]</auteur>
  <types>article, rubrique</types>
</root>
OU
{"auteur":"[valeur]","types":"article, rubrique"}
```

① *auteur* : nom de l'auteur

① *types* : permet de filtrer les types retournés (aucun type revient à demander tous les types pris en charge)

■ getchildren

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Récupérer les enfants d'un noeud

@ service = **getchildren**

@ format = **xml/json**

@ data =

```
<root>
  <id_parent>[valeur]</id_parent>
  <recurse>[true/false]</recurse>
  <documents>[true/false]</documents>
</root>
OU
{"id_parent":"[valeur]","recurse":"[true/false]","documents":"[true/false]"}
```

① *id_parent* : identifiant de la rubrique dont vous voulez récupérer les enfants (0 pour la racine du site)

① *recurse* : permet de ne récupérer que les enfants n-1 ou tout l'arbre depuis id_parent

① *documents* : permet d'intégrer les documents des éléments dans le resultat ou non (considéré comme false si vide)

■ getrubriquedata

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Récupérer les données d'une rubrique

@ service = getrubriquedata

@ format = xml/json

@ data =

```
<root>
  <id>[valeur]</id>
  <documents>[true/false]</documents>
</root>
OU
{"id":"[valeur]","documents":"[true/false]"}
```

① *id* : identifiant de la rubrique

① *documents* : intégrer les documents dans le resultat ou non

■ getarticledata

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Récupérer les données d'un article

@ service = getarticledata

@ format = xml/json

@ data =

```
<root>
  <id>[valeur]</id>
  <documents>[true/false]</documents>
</root>
OU
{"id":"[valeur]","documents":"[true/false]"}
```

① *id* : identifiant de l'article

① *documents* : intégrer les documents dans le resultat ou non

■ getbrevedata

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Récupérer les données d'une brève

@ service = **getbrevedata**

@ format = **xml/json**

@ data =

```
<root>
  <id>[valeur]</id>
  <documents>[true/false]</documents>
</root>
OU
{"id":"[valeur]","documents":"[true/false]"}
```

① id : *identifiant de la brève*

① documents : *intégrer les documents dans le resultat ou non*

■ getdocumentdata

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Récupérer les données d'un document

@ service = **getdocumentdata**

@ format = **xml/json**

@ data =

```
<root>
  <id>[valeur]</id>
</root>
OU
{"id":"[valeur]"}
```

① id : *identifiant du document*

■ getarticledocuments

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Récupérer les documents d'un article

@ service = **getarticledocuments**

@ format = **xml/json**

@ data =

```
<root>
  <id>[valeur]</id>
</root>
OU
{"id":"[valeur]"}
```

① id : *identifiant de l'article*

■ getbrevedocuments

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Récupérer les documents d'une brève

@ service = getbrevedocuments

@ format = xml/json

@ data =

```
<root>
  <id>[valeur]</id>
</root>
OU
{"id":"[valeur]"}
```

① id : *identifiant de la brève*

■ getrubriquedocuments

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Récupérer les documents d'une rubrique

@ service = getrubriquedocuments

@ format = xml/json

@ data =

```
<root>
  <id>[valeur]</id>
</root>
OU
{"id":"[valeur]"}
```

① id : *identifiant de la rubrique*

SERVICES WEB - UTILITAIRES

■ authentication

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Valider de l'authentification

@ service = authentication
@ format = xml/json
@ data =

① *Note : ce service retourne "true" si la requête est bien parvenue au plugin SpipService, donc en passant l'authentification de Spip. Si elle échoue, soit elle renvoie le code HTML de la page d'authentification de Spip soit un message indiquant que le plugin n'existe pas.*

■ clearcache

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Vider le cache de Spip (`_DIR_SQELS`)

@ service = clearcache
@ format = xml/json
@ data =

■ clearcacheimage

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Vider le cache image de Spip (`_DIR_VAR` et `_DIR_CACHE`)

@ service = clearcacheimage
@ format = xml/json
@ data =

SERVICES WEB - ECRITURE

■ setrubrique

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Créer/modifier une rubrique

@ service = setrubrique

@ format = xml/json

@ data =

```
<root>
  <id>[valeur]</id>
  <values>
    <id_parent>[value]</id_parent>
    <titre>[value]</titre>
    <descriptif>[value]</descriptif>
    <texte>[value]</texte>
  </values>
</root>
OU
{"id":"[valeur]","values":{"id_parent":"[valeur]","titre":"[valeur]","descriptif":"[valeur]","texte":"[valeur]"}}
```

① id : identifiant de la rubrique (création si vide ou si vaut 0, modification sinon)

① id_parent : identifiant de la rubrique parente (ne pas ajouter ce champs pour une rubrique racine) Note : vous pouvez donc par ce biais changer le parent

① titre : c'est explicite!

① descriptif : c'est explicite!

① texte : c'est explicite!

■ setarticle

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Créer/modifier un article

@ service = setarticle

@ format = xml/json

@ data =

```
<root>
  <id>[valeur]</id>
  <values>
    <id_parent>[value]</id_parent>
    <surtitre>[value]</surtitre>
    <titre>[value]</titre>
    <soustitre>[value]</soustitre>
    <descriptif>[value]</descriptif>
    <chapo>[value]</chapo>
    <texte>[value]</texte>
  </values>
</root>
```

OU

```
{"id":"[valeur]","values":{"id_parent":"[valeur]","surtitre":"[valeur]","titre":"[valeur]","sous
titre":"[valeur]","descriptif":"[valeur]","texte":"[valeur]","chapo":"[valeur]"}}
```

① id : identifiant de l'article (création si vide ou si vaut 0, modification sinon)

① id_parent : identifiant de la rubrique parente - Note : vous pouvez donc par ce biais changer le parent

① surtitre : c'est explicite!

① titre : c'est explicite!

① soustitre : c'est explicite!

① descriptif : c'est explicite!

① chapo : c'est explicite!

① texte : c'est explicite!

■ setbreve

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Créer/modifier une brève

@ service = setbreve

@ format = xml/json

@ data =

```
<root>
  <id>[valeur]</id>
  <values>
    <id_parent>[value]</id_parent>
    <titre>[value]</titre>
    <texte>[value]</texte>
    <lien_titre>[value]</lien_titre>
    <lien_url>[value]</lien_url>
  </values>
</root>
OU
{"id":"[valeur]","values":{"id_parent":"[valeur]","titre":"[valeur]","texte":"[valeur]","lien_
titre":"[valeur]","lien_url":"[valeur]"}}
```

① id : identifiant de la brève (création si vide ou si vaut 0, modification sinon)

① id_parent : identifiant de la rubrique parente - Note : vous pouvez donc par ce biais changer le parent

① titre : c'est explicite!

① texte : c'est explicite!

① lien_titre : titre du lien de la brève

① lien_url : url du lien de la brève

■ setdocument

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Modifier les champs d'un document

@ service = setdocument

@ format = xml/json

@ data =

```
<root>
  <id>[valeur]</id>
  <values>
    <titre>[value]</titre>
    <descriptif>[value]</descriptif>
  </values>
</root>
OU
{"id":"[valeur]","values":{"titre":"[valeur]","descriptif":"[valeur]"}}
```

① id : identifiant du document

① titre : c'est explicite!

① descriptif : c'est explicite!

■ addocumentrubrique

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Ajouter un document à une rubrique

@ service = addocumentrubrique

@ format = xml/json

@ data =

```
<root>
  <id>[valeur]</id>
  <file_name> [valeur] </file_name>
  <binary_base_64>[valeur]</binary_base_64>
</root>
OU
{"id":"[valeur]","file_name":"[valeur]","binary_base_64":"[valeur]"}
```

① *id* : identifiant de la rubrique

① *file_name* : nom du fichier

① *binary_base_64* : binaire du fichier encodé en base64

■ addocumentarticle

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Ajouter un document à un article

@ service = addocumentarticle

@ format = xml/json

@ data =

```
<root>
  <id>[valeur]</id>
  <file_name> [valeur] </file_name>
  <binary_base_64>[valeur]</binary_base_64>
</root>
OU
{"id":"[valeur]","file_name":"[valeur]","binary_base_64":"[valeur]"}
```

① *id* : identifiant de l'article

① *file_name* : nom du fichier

① *binary_base_64* : binaire du fichier encodé en base64

■ adddocumentbreve

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Ajouter un document à une brève

@ service = adddocumentbreve

@ format = xml/json

@ data =

```
<root>
  <id>[valeur]</id>
  <file_name> [valeur] </file_name>
  <binary_base_64>[valeur]</binary_base_64>
</root>
OU
{"id":"[valeur]","file_name":"[valeur]","binary_base_64":"[valeur]"}
```

① id : identifiant de la brève

① file_name : nom du fichier

① binary_base_64 : binaire du fichier encodé en base64

■ addlogoarticle

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Ajouter/remplacer un logo d'article

@ service = addlogoarticle

@ format = xml/json

@ data =

```
<root>
  <id>[valeur]</id>
  <file_name> [valeur] </file_name>
  <binary_base_64>[valeur]</binary_base_64>
</root>
OU
{"id":"[valeur]","file_name":"[valeur]","binary_base_64":"[valeur]"}
```

① id : identifiant de l'article

① file_name : nom du fichier

① binary_base_64 : binaire du fichier encodé en base64

■ addlogobreve

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Ajouter/remplacer un logo de brève

@ service = addlogobreve

@ format = xml/json

@ data =

```
<root>
  <id>[valeur]</id>
  <file_name> [valeur] </file_name>
  <binary_base_64>[valeur]</binary_base_64>
</root>
OU
{"id":"[valeur]","file_name":"[valeur]","binary_base_64":"[valeur]"}
```

① id : identifiant de la brève

① file_name : nom du fichier

① binary_base_64 : binaire du fichier encodé en base64

■ addlogorubrique

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Ajouter/remplacer un logo de rubrique

@ service = addlogorubrique

@ format = xml/json

@ data =

```
<root>
  <id>[valeur]</id>
  <file_name> [valeur] </file_name>
  <binary_base_64>[valeur]</binary_base_64>
</root>
OU
{"id":"[valeur]","file_name":"[valeur]","binary_base_64":"[valeur]"}
```

① id : identifiant de la rubrique

① file_name : nom du fichier

① binary_base_64 : binaire du fichier encodé en base64

■ institutearticle

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Modifier le statut/date de publication d'un article

@ service = institutearticle

@ format = xml/json

@ data =

```
<root>
  <id>[valeur]</id>
  <statut>[prepa/prop/publie/refuse/poubelle]</statut>
  <date>[valeur]</date>
</root>
OU
{"id":"[valeur]","statut":[prepa/prop/publie/refuse/poubelle], "date":"[valeur]"}
```

① id : identifiant de l'article

① statut : nouveau statut

① date : date de publication au format AAAA-MM-JJ hh:mm:ss (format stricte)

■ institutebreve

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Modifier le statut/date de publication d'une brève

@ service = institutebreve

@ format = xml/json

@ data =

```
<root>
  <id>[valeur]</id>
  <statut>[prepa/prop/publie/refuse/poubelle]</statut>
  <date>[valeur]</date>
</root>
OU
{"id":"[valeur]","statut":[prepa/prop/publie/refuse/poubelle], "date":"[valeur]"}
```

① id : identifiant de la brève

① statut : nouveau statut

① date : date de publication au format AAAA-MM-JJ hh:mm:ss (format stricte)

■ deletearticle

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Supprimer un article

@ service = deletearticle

@ format = xml/json

@ data =

```
<root>
  <id>[valeur]</id>
</root>
OU
{"id":"[valeur]"}
```

① id : identifiant de l'article

■ deleterubrique

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Supprimer une rubrique

@ service = deleterubrique

@ format = xml/json

@ data =

```
<root>
  <id>[valeur]</id>
</root>
OU
{"id":"[valeur]"}
```

① id : identifiant de la rubrique

■ deletedocumentarticle

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Supprimer un document d'article (Sip3 -> fait simplement une dissociation)

@ service = deletedocumentarticle

@ format = xml/json

@ data =

```
<root>
  <id_document>[valeur]</id_document>
  <id_article>[valeur]</id_article>
</root>
OU
{"id_document":"[valeur]","id_article":"[valeur]"}
```

① id_document : identifiant du document

① id_article : identifiant de l'article

■ deletedocumentbreve

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Supprimer un document de brève (Spip3 -> fait simplement une dissociation)

@ service = deletedocumentbreve

@ format = xml/json

@ data =

```
<root>
  <id_document>[valeur]</id_document>
  <id_breve>[valeur]</id_breve>
</root>
OU
{"id_document":"[valeur]","id_breve":"[valeur]"}
```

① *id_document* : identifiant du document

① *id_breve* : identifiant de la brève

■ deletedocumentrubrique

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Supprimer un document de rubrique (Spip3 -> fait simplement une dissociation)

@ service = deletedocumentrubrique

@ format = xml/json

@ data =

```
<root>
  <id_document>[valeur]</id_document>
  <id_rubrique>[valeur]</id_rubrique>
</root>
OU
{"id_document":"[valeur]","id_rubrique":"[valeur]"}
```

① *id_document* : identifiant du document

① *id_rubrique* : identifiant de la rubrique

■ deletelogoarticle

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Supprimer le logo d'un article

@ service = deletelogoarticle

@ format = xml/json

@ data =

```
<root>
  <id_article>[valeur]</id_article>
</root>
OU
{"id_article":"[valeur]"}
```

① *id_article* : identifiant de l'article

■ deletelogobreve

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Supprimer le logo d'une brève

@ service = deletelogobreve

@ format = xml/json

@ data =

```
<root>
  <id_breve>[valeur]</id_breve>
</root>
OU
{"id_breve":"[valeur]"}
```

① *id_breve* : identifiant de la brève

■ deletelogorubrique

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Supprimer le logo d'une rubrique

@ service = deletelogorubrique

@ format = xml/json

@ data =

```
<root>
  <id_rubrique>[valeur]</id_rubrique>
</root>
OU
{"id_rubrique":"[valeur]"}
```

① *id_rubrique* : identifiant de la rubrique

SURCHARGER/DÉFINIR DES SERVICES WEB

Vous pouvez soit surcharger, soit définir vos propres services web pour des besoins spécifiques. Vous devez dans un premier temps analyser votre besoin pour déterminer si la surcharge/création ne devrait pas être intégrée comme service natif au plugin. Dans ce cas rendez-vous sur la spip-zone et faites évoluer SpipService.

Si le besoin est spécifique à votre structure de données, un objet éditorial particulier par exemple, suivez les étapes suivantes.

- Créez dans **votre plugin** un dossier **inc/** si ce n'est pas déjà le cas.
- Ajoutez dans ce dossier un fichier PHP, selon la nomenclature stricte suivante :

```
spipservice_[service_name].php
```

① *[service_name]* est le nom du service que vous indequerez dans votre requête HTTP (paramètre 'service')

- Créez dans ce fichier une fonction selon la nomenclature stricte suivante :

```
inc_spipservice_[service_name]($format, $service, $data){  
    // traitement...  
    return array(vos valeurs de retour);  
}
```

① *[service_name]* doit être le même que celui utilisé pour le nommage du fichier

① *\$format* : vous indique le format

① *\$service* : le nom du service web demandé (vaudra forcément *[service_name]*)

① *\$data* : un tableau associatif contenant les valeurs du paramètre 'data' de votre requête

① la valeur retournée par votre fonction doit être un tableau associatif (si le format demandé est le XML et que le tableau associatif doit avoir plusieurs clés identiques avec des valeurs différentes (liste d'articles par exemple), voir la fonction `arrayIntoXml($arr)` dans `spip_service/inc/spip_service_utils.php`)

Pour surcharger les services existants, procédez de la même manière en nommant `[service_name]` avec le nom d'un des services natifs présentés plus haut. Rédefinissez le corps de la fonction, c'est le votre qui sera appelé!

Pour appeler votre nouveau service, faites comme pour les services natifs, exemple :

■ monserviceperso

url : <http://www.votrespip.com/ecrire/?action=spipservice>

Fait des trucs de folie...

@ service = monserviceperso

@ format = xml/json

@ data =

```
<root>
  <une_valeur>[valeur]</une_valeur>
  <une_autre_valeur>
    <une_valeur_dans_autre_valeur>[valeur]</une_valeur_dans_autre_valeur>
  </une_autre_valeur>
</root>
OU
{"une_valeur":"[valeur]", "une_autre_valeur":{"une_valeur_dans_autre_valeur":[valeur]}}
```

① dans votre fonction `inc_spipservice_monserviceperso($format, $service, $data){}`, la variable `$data` passée vaudra :

```
array("une_valeur"=>[valeur], "une_autre_valeur"=>array("une_valeur_dans_autre_valeur"=>[valeur]))
```

UTILISER UN FORMAT DE DONNÉE PARTICULIER

Vous pouvez utiliser un format de données autre que XML/JSON. Dans ce cas, la variable `$data` passée en paramètre de la fonction du service web vaudra exactement la valeur du paramètre 'data' de la requête.

De la même manière, la valeur de retour du service web sera exactement la valeur retournée par la fonction du service web.

Exemple :

■ monserviceavecformatperso

`url : http://www.votrespip.com/ecrire/?action=spipservice`

Fait pas grand chose en fait!

@ service = monserviceavecformatperso

@ format = monformat

@ data = #id=12#texte=mevoila

① dans votre fonction `inc_spipservice_monserviceavecformatperso($format, $service, $data){}`, la variable `$data` passée vaudra :

```
#id=12#texte=mevoila
```

① admetons l'implémentation suivante de la fonction de votre service web

```
function inc_spipservice_monserviceavecformatperso($format, $service, $data){  
    return "@ok";  
}
```

① le service web retournera la valeur suivante :

```
@ok
```