

Les Médecins Maîtres-Toile

[Accueil](#) ▶ [Boîte à Outils](#) ▶ [Syndication de contenu](#) ▶ Des fils RSS 1.0, RSS 2.0 et Atom 1.0 pour votre site SPIP

Publié le : 18 mars 2006

**Publication antérieure :
14 septembre 2005**



Imprimer cet article

**Auteur :
Alain Sichel †**



Des fils RSS 1.0, RSS 2.0 et Atom 1.0 pour votre site SPIP

Cet article fait suite à [Abandonnons RSS 0.91 !](#), après avoir réalisé des fils RSS 2.0, je vous propose de réaliser des fichiers de syndication dans différents formats (RSS 1.0, RSS 2.0 ou Atom 1.0) [[1](#)].

Sommaire

- [Pour quoi faire](#)
- [Les principes retenus](#)
- [Un fil RSS 2.0](#)
- [Un fil RSS 1.0](#)
- [Un fil Atom 1.0](#)
- [Et les brèves](#)
- [Installation](#)
- [Adaptation des fils](#)
- [Cerise sur le gâteau](#)
- [Conclusion provisoire](#)

Pour quoi faire ?

Le format RSS 0.91 est dépassé : on ne peut pas placer toutes les informations voulues, et beaucoup sont limitées (voir [Abandonnons RSS 0.91 !](#)).

Le format RSS 2.0 est une évolution de 0.9x, il est plus souple mais garde des limitations (en particulier le nombre de caractères de certains éléments est limité), c'est dommage sinon ce serait une bonne évolution.

Le format RSS 1.0 a été mis au point par un autre groupe de travail que RSS 0.9x et 2.0, il est basé sur RDF (format du w3c pour le web sémantique), il est plus souple et ses possibilités peuvent être étendues par l'ajout de modules. Bien que RSS ne soit pas initialement prévu pour échanger des documents, mais plutôt des méta données des documents, RSS 1.0 peut permettre de présenter des articles entiers.

Le Atom est un autre format de documents XML. La version antérieure Atom 0.3 est obsolète, mais Atom 1.0 est intéressant.

Au total il est bien difficile de choisir entre RSS 1.0, RSS 2.0 et Atom 1.0, qui ont chacun des avantages [2]. Pour ceux qui veulent utiliser ces standards, cet article vous présentera des exemples des différents squelettes pour que vous puissiez choisir. Tous les fils présentés ici respectent leur format [3].

Les principes retenus

Cet article reprend les apports d'autres contributions [4] mais propose aussi d'autres possibilités [5] :

► comme pour l'article précédent, je vous propose d'indiquer des informations complémentaires sur votre site,

- soit directement dans le squelette (il n'y aura des difficultés que s'il y a des caractères accentués),
- soit avec des mots clés (c'est la solution choisie pour les exemples de fichiers à télécharger en bas de cet article) pour *webmestre*, *page contact webmestre*, *description (channel et image)* et *copyright* ; pour cela créez un groupe de mots clés particulier [6] : **ELEMENTS_FIL** réservé aux administrateurs, en ayant veillé à ce qu'il ne soit pas proposé pour les articles et brèves, et écrivez 5 mots clés : *Copyright*, *Webmaster*, *uri_Webmaster*, *Description_image* et *Description_site* en utilisant **Texte explicatif** pour noter l'information souhaitée ; il faut alors mettre dans le fil (à adapter pour chaque mot clé) :

```
<BOUCLE_motcle_Copyright(MOTS){titre="Copyright"}>
[<copyright>(#TEXTE|texte_backend)</copyright>]</BOUCLE_motcle_Copyright>
```

- soit en utilisant le fichier **local_fr.php3** [7], par exemple en notant :

```
'copyright' => 'Copyright Médecins Maîtres-Toile francophones',
```

il faudra alors mettre dans backend.html :

```
<copyright><:copyright:></copyright>
```

► on indique :

- la rubrique dans laquelle se trouve l'article (ou brève) avec `<dc:subject>` pour RSS 1.0 et `<category>` pour RSS 2.0 et Atom 1.0
 - la date de mise à jour du fil (`<pubDate>` `<dc:date>` ou `<updated>` selon le format) en prenant la date du dernier article (ou brève) publié
 - les informations concernant le logo du site à partir de `#LOGO_SITE_SPIP` [8]
-

► le même fichier de syndication donne une présentation courte (limitée à #INTRODUCTION) ou complète (article entier, chapeau, notes, post scriptum, mots clés, documents joints...), mais cette possibilité est un choix article par article, selon le souhait de son rédacteur, je trouve en effet incorrect de décider à la place de chaque auteur ; le contenu "riche" apparaît quand le mot clé "Syndiquer tout le texte" est choisi pour l'article [9],

► pour `<content:encoded>`, l'objectif est de présenter le contenu complet de l'article, la solution choisie devant permettre un affichage correct par les lecteurs, la contrainte étant le respect du standard du fil. Pour cela, il y a 2 grands types de solutions :

- Une solution universelle qui consiste à placer tout le contenu codé brut dans une déclaration `<![CDATA[;` Avantages : compact, simplicité et lisibilité ; Problème potentiel : certains éléments HTML peuvent être considérés non valides (tel OBJECT et EMBED).
- Une solution proprement « *spipienne* » utilisée dans la [version en développement \(1.9\)](#) consiste à filtrer le contenu à l'aide du filtre **texte_backend** qui a depuis longtemps fait ses preuves ; Avantage : validité d'encodage XML assuré (les valideurs ne peuvent pas détecté l'invalidité d'élément html interdits, il s'agit donc en fait d'une triche) ; Désavantage : plus lourd, moins lisible, il faut appliquer le filtre à chaque balise SPIP.

C'est la deuxième solution qui a été choisie ici.

► le logo de l'article est inclu dans `<description>` mais aussi dans `<content:encoded>` car la plupart des lecteurs prennent en compte l'un ou l'autre, mais rarement les deux [10],

► les exemples de squelettes contenus dans Fils_7.zip nécessitent quelques adaptations au site, notamment pour `<image>` et `<icon>` pour Atom 1.0. Il y a aussi une version Fils_pour_les_nuls.zip écrite pour pouvoir fonctionner telle quelle, dès le transfert sur le serveur, afin de faciliter leur utilisation par des débutants non habitués à modifier le code d'un squelette. Pour permettre cela, j'ai utilisé les boucles pour qu'en l'absence d'un élément, il n'y ait pas d'affichage. Par exemple si #LOGO_SITE_SPIP n'est pas défini, tous les éléments de `<image>` seront absents. J'ai mis aussi pour cela un peu de PHP, par exemple dans les fichiers Atom 1.0 un script vérifie la présence de favicon.ico sur le site, si c'est le cas `<icon>` sera affiché.

A noter que la déclaration du type de fichier est `Content-type: application/rss+xml` pour RSS 1.0 et 2.0 (et `Content-type: application/atom+xml` pour Atom 1.0) car c'est un type MIME plus précis que `Content-type: text/xml`, mais si vous préférez cette dernière déclaration vous pouvez la remettre dans les squelettes, notamment pendant la phase de mise au point, cela facilite l'affichage des fils dans les navigateurs.

En raison de l'utilisation des filtres **texte_backend** et **liens_absolus** ces squelettes nécessitent la version 1.8.2 de SPIP.

Un fil RSS 2.0

L'article [précédent](#) proposait déjà la réalisation de fichiers RSS 2.0 *backend.html* pour les articles et *backend-breves.html* pour les brèves. Ce dernier est complet si vous souhaitez un fil pour vos brèves au format RSS 2.0, mais pour les articles, je vous propose de reprendre *backend.html* et de le modifier (on va l'appeler *rss2.html*) en ajoutant les éléments supplémentaires pour présenter l'article complet :

```
<BOUCLE_autorisation(ARTICLES){id_article}{titre_mot="Syndiquer tout le
texte"}><content:encoded>[&lt;img
src="#URL_SITE_SPIP/(#LOGO_ARTICLE|extraire_attribut{src})" align="right"
hspace="4" vspace="4"&gt;
][&lt;div class='rss_surtitre'&gt;(#SURTITRE|texte_backend)&lt;/div&gt;
][&lt;div class='rss_soustitre'&gt;(#SOUSTITRE|texte_backend)&lt;/div&gt;
```

```

] [&lt;div class='rss_chapo'&gt; (#CHAPO|liens_absolus|texte_backend) &lt;/div&gt;
] [&lt;div class='rss_texte'&gt; (#TEXTE|liens_absolus|texte_backend) &lt;/div&gt;
] [&lt;hr /&gt;
&lt;div class='rss_notes'&gt; (#NOTES|liens_absolus|texte_backend) &lt;/div&gt;
] [&lt;div class='rss_ps'&gt; (#PS|liens_absolus|texte_backend) &lt;/div&gt;]
<BOUCLE_mots_mf(MOTS){id_article}{", "}>
  [&lt;a href="[ (#URL_MOT|url_absolue)]"
rel="tag"&gt; (#TITRE|texte_backend) &lt;/a&gt;] </BOUCLE_mots_mf>
</content:encoded>
<BOUCLE_podcast(DOCUMENTS){id_article}{mode=document}{doublons}{0,1}>[ <enclosure
url="( #URL_DOCUMENT|url_absolue|unique)" [ length="( #TAILLE)" [
type="( #MIME_TYPE)" />] </BOUCLE_podcast></BOUCLE_autorisation>

```

La <BOUCLE_autorisation> n'autorise l'affichage que si le mot clé *"Syndiquer tout le texte"* a été choisi. A noter pour ce format la limitation des documents joints à un seul (par le filtre *"unique"*) afin que le fichier soit valide.

► les éléments ajoutés à *backend.html* (voir) sont également ajoutés ici :

- <dc:creator> pour le webmestre
- <copyright>
- <generator>
- <width> et <height> de l'image
- <description> de l'image

► le format RSS 2.0 permet d'ajouter :

- <category> qui permet de préciser la rubrique de l'article :

```

<BOUCLE_rubrique(RUBRIQUES){id_rubrique}><category
domain="#URL_SITE_SPIP/#URL_RUBRIQUE">
[ (#TITRE|supprimer_numero|texte_backend)] </category></BOUCLE_rubrique>

```

on peut aussi l'écrire sous une forme plus simple, sans l'URL de la rubrique :

```

<BOUCLE_rubrique(RUBRIQUES){id_rubrique}><category>
[ (#TITRE|supprimer_numero|texte_backend)] </category></BOUCLE_rubrique>

```

Un fil RSS 1.0

Le format est différent du RSS 2.0 [11], mais on retrouve les mêmes informations avec parfois des noms différents pour les éléments :

- <dc:rights> indique le *Copyright*,

- <dc:date> la date de mise à jour du fil,
- <dc:description> la description de l'image,
- <dc:subject> la rubrique de l'article :

```
<BOUCLE_rubriques (RUBRIQUES) {id_rubrique}><dc:subject>
  <rdf:Description>
    <taxo:topic rdf:resource="#URL_SITE_SPIP/#URL_RUBRIQUE">
      <rdf:value>[(#TITRE|supprimer_numero|texte_backend)]</rdf:value>
    </taxo:topic>
  </rdf:Description>
</dc:subject></BOUCLE_rubriques>
```

ou si vous ne souhaitez pas indiquer l'URL de la rubrique, tout simplement :

```
<BOUCLE_rubriques (RUBRIQUES) {id_rubrique}>
<dc:subject>[(#TITRE|supprimer_numero|texte_backend)]</dc:subject>
</BOUCLE_rubriques>
```

Pour certains éléments, faute de mieux, j'ai utilisé des balises Dublin Core (le fil est donc valide, mais les balises sont un peu détournées de leur fonction initiale) :

- <dc:source> le générateur (ici SPIP),
- la largeur et la hauteur de l'image sont indiqués dans <dc:format> :

```
<dc:format>width="largeur_en_pixels" height="hauteur_en_pixels"</dc:format>
```

Un fil Atom 1.0

Pour le format Atom 1.0 [12] :

► la déclaration est différente, et comprend 2 éléments <link>, l'un vers le site, l'autre vers le fil, et un élément <id> :

```
<feed xmlns="http://www.w3.org/2005/Atom"
  xml:lang="#LANG">
  <link rel="alternate" type="text/html" href="#URL_SITE_SPIP/"
  [title="( #NOM_SITE_SPIP|texte_backend)"] />
  <link rel="self" href="#URL_SITE_SPIP/#SELF" />
  <id>#URL_SITE_SPIP/</id>
```

► certains éléments changent de nom :

- <subtitle> présente la "Description" du fil,
-

- `<rights>` le "Copyright",
- `<author><name>` pour le webmestre,

► de nouveaux éléments sont acceptés :

- `<icon>` pour l'URL de favicon.ico,
- `<logo>` pour l'URL du logo,
- `<category>` indique la rubrique qui contient l'article présenté avec mention de l'URL de la rubrique :

```
<BOUCLE_rubrique(RUBRIQUES){id_rubrique}><category
scheme="#URL_SITE_SPIP/#URL_RUBRIQUE"
term="[(#TITRE|supprimer_numero|texte_backend)]" /></BOUCLE_rubrique>
```

► le fil doit comporter `<published>` (la date de publication) et `<updated>` (la date de mise à jour) ; pour les articles, on regarde si `#DATE_REDAC` existe [13] (date de première publication), si oui il sera indiqué pour `<published>`, si non on prend `#DATE` (date de mise en ligne) :

```
<?php $date_redac = "#DATE_REDAC";
if ($date_redac) { echo
"    <published>[(#DATE_REDAC|date_iso)]</published>\n"; }
else { echo "    <published>[(#DATE|date_iso)]</published>\n"; } ?>
    <updated>[(#DATE|date_iso)]</updated>
```

Et les brèves ?

Si vous souhaitez présenter vos brèves par un fil plus moderne, l'article précédent vous proposait déjà un fichier RSS 2.0. ; je vous propose maintenant un fichier au format RSS 1.0 et un autre au format Atom 1.0. Ces fichiers ne demandent pas d'explication particulière, c'est juste une adaptation à la présentation des brèves. A noter l'élément vide `<author><name></name></author>` pour *atom1-brevés* pour éviter le refus du fichier par les validateurs.

Installation

Les différents exemples sont regroupés dans **Fils_7.zip** et **Fils_pour_les_nuls_3.zip** [14] téléchargeables en bas de cette page :

- *rss2.html* et *rss2.php3* pour le format RSS 2.0,
- *rss1.html* et *rss1.php3* pour le format RSS 1.0,
- *atom1.html* et *atom1.php3* pour le format Atom 1.0,
- *rss1-brevés.html* et *rss1-brevés.php3* pour les brèves au format RSS 1.0,
- *atom1-brevés.html* et *atom1-brevés.php3* pour les brèves au format Atom 1.0.
- *mes_fonctions.php3* pour les filtres nécessaires pour réaliser des différents fils [15].

Téléchargez à la racine de votre site les couples (.html et .php3) de fichiers correspondants aux formats souhaités. S'il n'y a pas à la racine de votre site de fichier *mes_fonctions.php3* transférez celui présent dans *Fils_3.zip*, sinon, il faut ouvrir le fichier existant et y copier les fonctions du fichier *mes_fonctions.php3* de cet article.

Adaptation des fils

Pour utiliser complètement les possibilités de ces fichiers, il faut créer des mots clés comme expliqué plus haut, ainsi :

- ▶ *Webmaster* indiquera le nom du Webmaster,
- ▶ *uri_Webmaster* la page de présentation du Webmaster (ce qui permet un contact grâce au formulaire de cette page sans diffusion de l'e-mail),
- ▶ *Description_site* la phrase de description du site,
- ▶ *Copyright* la mention Copyright,
- ▶ *Description_image* la phrase de description du logo du site


Pour un site multilingue, la solution utilisant le fichier **local_fr.php3** me parait plus adaptée.

Et pour ceux qui n'ont pas peur de mettre les mains dans le cambouis (c'est-à-dire de modifier le squelette), vous pouvez simplifier, par exemple :

- ▶ les mots clés n'ont d'intérêt que si on utilise des caractères accentués qui seront incorrectement affichés si on les écrit directement dans le squelette, mais si le webmaster s'appelle Alain Lambert et sa page de présentation http://www.monsite.org/auteur.php3?id_auteur=1 autant écrire directement ces informations qui ne changent pas dans le squelette ;
- ▶ pour un fil Atom 1.0, si vous savez qu'une image *favicon.ico* existe sur votre site, vous pouvez supprimer le script qui vérifie son existence...

Cerise sur le gateau

Que peut-on faire de plus ?

- ▶ **Valorisation sur votre site** : Comme indiqué dans [Abandonnons RSS 0.91 !](#) vous pouvez indiquer la présence de fils dans les navigateurs modernes avec le symbole :  et permettre ainsi aux visiteurs d'ajouter un marque page dynamique qui indiquera vos derniers articles.

Afin que vos fichiers de syndication soient signalés par cette image, il faut indiquer dans le code de la page entre `<head>` et `</head>` la balise par exemple :

```
<link rel="alternate" type="application/rss+xml" title="<:syndiquer_site:>"
href="rss2.php3">
```

Vous pouvez également ajouter pour la page *rubrique.html* toujours entre `<head>` et `</head>` (l'exemple ici est avec un fichier Atom 1.0) :

```
<BOUCLE_rubrique(RUBRIQUES) {id_rubrique}>
<link rel="alternate" type="application/atom+xml"
title="<:syndiquer_rubrique:>" href="atom1.php3?id_rubrique=#ID_RUBRIQUE">
</BOUCLE_rubrique>
```

Pour le format RSS 1.0, il vaut mieux noter `type="application/rdf+xml"`

► **Réécrire les URL** : si vous préférez une autre extension que .php3 c'est possible : on peut adapter le principe de [La réécriture des URL "à la volée"](#), en mettant dans un fichier `.htaccess` à la racine du site contenant par exemple (ici l'extension choisie est .xml) :

```
RewriteEngine on
RewriteRule ^backend.xml$ /backend.php3 [L]
RewriteRule ^backend-breves.xml$ /backend-breves.php3 [L]
RewriteRule ^rss1.xml$ /rss1.php3 [L]
RewriteRule ^rss2.xml$ /rss2.php3 [L]
RewriteRule ^atom1.xml$ /atom1.php3 [L]
RewriteRule ^rss1-breves.xml$ /rss1-breves.php3 [L]
```

ou plus simplement :

```
RewriteEngine on
RewriteRule ^(backend|rss|atom)([0-2]*)(-breves)*\.xml$ /$1$2$3.php3 [L]
```

Conclusion provisoire

J'ai essayé de tirer le meilleur parti des formats actuels, mais j'ai eu des difficultés avec les validateurs [16] et des limitations en raison de ce que j'ai pu comprendre d'une documentation toujours en anglais.

Cet article n'est qu'une proposition pour ceux qui veulent des fils modernes sur leur site SPIP, il en est à sa 4^e modification profonde et il sera mis à jour en fonction des suggestions d'amélioration et de l'évolution des formats.

[1] Cet article a été publié le 14/09/05, il a ensuite été modifié le 29/12/05, le 5/01/06, le 10/01/06 et le 23/01/06. Le fichier à télécharger tient compte de ces modifications. Merci à [Nicolas Krebs](#) pour ses conseils pour améliorer cet article.

[2] Pour voir d'autres avis sur la question : [Syndication avec RSS & Atom](#), [La syndication de contenus](#), [FichiersDeSyndication](#) (aspects techniques des fils pour SPIP) et [RSS 2.0 and Atom 1.0, Compared](#).

[3] Pour vérifier, voyez les validateurs : [W3C Feed Validation Service](#) et [Feed Validator](#).

[4] En particulier les articles de la rubrique [Syndication](#) de [Spip-Contrib](#), mais aussi la [version de développement actuelle](#) (1.9).

[5] Je reprends dans certains commentaires des conseils d'[André Vincent](#).

[6] J'adapte ici l'astuce présentée par Yves Grenier dans [Un fil RSS 2.0 et un fil ATOM 0.3](#) pour le "Sous_titre_site".

[7] Dans les fichiers de langue de SPIP, dans `/ecrire/lang/` ; si vous n'avez pas l'habitude d'utiliser **local_fr.php3**, voyez la page [Internationalisation du squelette](#) qui donne des explications et un exemple.

[8] Si vous n'avez pas l'habitude de `#LOGO_SITE_SPIP` voyez cette [note](#).

[9] Si par contre pour votre site,
▶ vous voulez ce contenu "riche" pour tous les articles, il suffit de supprimer la boucle

```
<BOUCLE_autorisation(ARTICLES){id_article}{titre_mot="Syndiquer tout le texte"}>
```

et plus bas

```
</BOUCLE_autorisation>
```

▶ vous ne voulez pas du contenu "riche" pour aucun article, il suffit de supprimer la boucle ainsi que tout le code qu'elle contient.

[10] • La plupart des lecteurs de nouvelles (NewsReaders) prennent en compte seulement le contenu « riche » de `<content:encoded>`, mis dans `<![CDATA[` et ignore le contenu de `<description>`, sauf pour les bulles dans certains cas,
• la plupart des CMS (dont SPIP, Drupal et Mambo) prennent en compte seulement le contenu « propre » de `<description>`, nettoyé par le filtre `texte_backend` et ignorent le contenu « riche » de `<content:encoded>`,
• dans le cas spécifique de SPIP, en l'absence de `<description>` il prend en compte le contenu « riche » de `<content:encoded>` à la place (du moins jusqu'à récemment).

[11] Informations sur le format RSS 1.0 : [Construction d'un fil RSS 1.0](#), [Introduction à RSS 1.0](#), [RDF Site Summary \(RSS\) 1.0](#), les modules d'extension [Modules](#), [Modules : Dublin Core](#), [Modules : Syndication](#) et [Modules : Content](#).

[12] Informations sur le format Atom 1.0 [The Atom Syndication Format 1.0](#) et [AtomEnabled](#) et l'article de Nicolas Krebs : [Squelette atom 1.0](#) (décembre 2005).

[13] Voir [La gestion des dates](#).

[14] Version destinée aux débutants souhaitant que ça marche dès l'installation, mais vous feriez mieux d'utiliser la version **Fils_7.zip** avec l'aide de quelqu'un qui a l'habitude d'adapter un squelette.

[15] Toutes ces fonctions proviennent de l'article d'[Yves Grenier](#).

[16] Ceci est un problème spécifique à mon hébergement (mutualisé OVH).

Documents joints à l'article




Fils_7.zip



Fils_pour_les_nuls_3.zip



 [Imprimer cet article](#)

Copyright Médecins Maîtres-Toile francophones
[Espace membres](#) - [Administration](#) - [Crédits](#)
