# Utilisation de SimpleXML avec le PHP

SimpleXML est une des deux nouvelles extensions de PHP 5 consacrées à XML. L'autre est DOM, qui a évolué de domxml. Contrairement à cette dernière, SimpleXML est une approche nouvelle du traitement des fichiers XML. Elle met en avant la simplicité d'utilisation.

SimpleXML aborde le problème des fichiers XML simples. Prenez un fichier XML comme celui-ci :

<?xml version="1.0" encoding="utf-8"?>

<bibliotheque>

<style id="roman">

<livre>

<titre>La fortune des Rougon</titre>

<auteur>Emile Zola</auteur>

</livre>

<livre>

<titre>Hernani</titre>

<auteur>Victor Hugo</auteur>

</livre>

</style>

<style id="fiction">

<livre>

<titre>Le seigneur des anneaux</titre>

<auteur>J.R.R. Tolkien</auteur>

</livre>

</style>

</bibliotheque>

Le document est éminemment lisible par un humain ainsi que par une machine. Sa structure est claire et facile à comprendre. Le fichier XML est stocké sous le nom de 'livres.xml'. Pour le traiter avec SimpleXML, il suffit de le charger avec ces quelques lignes.

<?php

$bibliotheque = simplexml\_load\_file('livres.xml'); //(1)

foreach ($bibliotheque->style as $style) { //(2)

print "\nType {$style['id']} <br />\n"; //(3)

foreach ($style->livre as $livre) { //(4)

print "Titre : {$livre->titre} <br />\n"; //(5)

print "Auteur : {$livre->auteur} <br />\n"; //(6)

}

}

?>

(1) Lecture du fichier XML stockage,

Il serait difficile de faire plus simple. La première ligne charge le contenu du fichier livres.xml, dans la variable objet $bibliotheque. Le fichier est lu en entier. Si le document n'est pas compatible XML, alors les erreurs d'analyse sont affichées dans la page, et $bibliothèque reçoit le booléen false.

Alternativement, SimpleXML vous propose la fonction simplexml\_load\_string(). Cette dernière fonctionne comme simple\_load\_file(), mais attend le contenu XML dans une chaîne de caractères. Le traitement appliqué est le même que dans le cas des fichiers. Ainsi, si vos données XML sont déjà en mémoire, il n'y a pas besoin de les écrire sur le disque pour les traiter. C'est le cas si vous avez obtenu ces données depuis une base de données, ou via une ressource réseau.

Accès aux données XML

Une fois le fichier XML correctement lu, on retrouve la même syntaxe puissante qu'avec les outils de conversions XML vers des tableaux PHP. SimpleXML simplifie considérablement l'accès aux données en jouant sur sa nature de ressource PHP : elle est à la fois un tableau et un objet. La ressource retournée représente simplement la racine de notre document XML. Il ne reste plus qu'à accéder aux attributs et aux sous-balises.

(2)

Le contenu d'une balise est scindé en sous-balises. Le contenu de chaque balise est affecté à un membre de la balise parente, sous le même nom que la balise. Si plusieurs sous-balises ont le même nom, elles sont rangées en tableau, et indexées numériquement. C'est ce qui arrive à notre balise 'style' ici. Il y a deux balises styles, et l'objet $bibliotheque contient un membre appelé 'style', qui est un tableau. Nous pouvons le passer à l'instruction foreach(), et passer en revue toutes les balises 'style'.

Chaque balise 'style' contient deux balises distinctes : auteur et titre. Comme elles sont uniques dans chaque balise style, il suffit de les appeler par leur nom, et les données contenues ont été enregistrées comme valeur, sous forme de chaîne de caractères.

Accès aux attributs

(3) Les attributs sont aussi décodés et placés dans un tableau particulier. C'est en fait la même variable que l'objet que l'on utilise pour accéder aux différentes sous-balises. Ainsi :

$style->livre nous renvoie la sous-balise livre de balise style.

$style['livre'] nous renverrait la valeur de l'attribut livre de la balise 'style', s'il existe.

SimpleXML utilise donc les deux notations simultanément : tableau et objet. C'est très pratique pour identifier rapidement les données que vous manipulez, et pour leur donner des espaces de noms distincts.

Pour être complet, SimpleXML dispose de deux méthodes qui retournent la liste complète des attributs et des sous-balises et de leurs valeurs respectives. Il s'agit des méthodes attributes() et children(). On les utilise comme ceci :

Liste des attributs

<?php

$string = <<<XML

<a>

<foo attribut1="un" attribut2="deux">1</foo>

</a>

XML;

$xml = simplexml\_load\_string($string);

foreach($xml->foo[0]->attributes() as $a => $b) {

echo $a, '="', $b, "\"\n";

}

?>

Ce script va afficher :

attribut1="un"

attribut2="deux"

Liste des sous-balises

<?php

$string = <<<XML

<a>

<b>1</b>

<b>2</b>

<b>3</b>

<b>4</b>

</a>

XML;

$xml = simplexml\_load\_string($string);

foreach($xml->children() as $a => $b) {

echo $a, '="', $b, "\"\n";

}

?>

Ce script va afficher :

b="1"

b="2"

b="3"

b="4"