Manipuler du XML avec PHP

Hyacinthe MENIET

3 août 2019

Je vais indiquer dans ce document, comment manipuler très facilement du XML avec PHP. Pour y parvenir je vais m'appuyer sur l'API SimpleXML introduite avec PHP 7. Pour des traitements plus pointus, PHP 7 dispose d'une API DOM conforme aux standards niveau 2 de DOM. Je me concentre sur SimpleXML.

1. Pré-requis

- Vous êtes familier de PHP et de sa syntaxe objet.
- Vous êtes familier de XML, notamment de XPath et XSLT.
- Vous disposez d'un serveur Apache 2.4 minimum.
- Vous disposez d'un serveur Mariadb 10 minimum (ou MySQL 5) pour la partie SQL.
- Vous disposez de PHP 7 minimum.

2. Initialisation des données

Dans la suite du document vous allez créer un fichier XML à partir de données contenues dans MySQL. Pour cela créez la table suivante :

```
CREATE TABLE individu (
individu_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
prenom VARCHAR(45) NULL,
nom VARCHAR(20) NULL,
PRIMARY KEY(individu_id)
);
```

C'est une banale table contenant les noms et prénoms d'individus. Remplissez-la avec les données suivantes :

```
INSERT INTO `individu` (`individu_id`, `prenom`, `nom`) VALUES
    (1, 'Hyacinthe', 'MENIET');
INSERT INTO `individu` (`individu_id`, `prenom`, `nom`) VALUES
    (2, 'William', 'HEWLETT');
INSERT INTO `individu` (`individu_id`, `prenom`, `nom`) VALUES
    (3, 'David', 'PACKARD');
INSERT INTO `individu` (`individu_id`, `prenom`, `nom`) VALUES
    (4, 'Scott', 'MCNEALY');
```

```
INSERT INTO `individu` (`individu_id`, `prenom`, `nom`) VALUES
    (5, 'Andy', 'BECHTOLSHEIM');
```

3. Créer un objet SimpleXMLElement

Vous pouvez créer un objet `SimpleXMLElement` à partir d'une chaîne de caractère. Vous pouvez la soumettre directement ou la construire à partir de données contenues dans la base de données :

```
$xmlstr = '<?xml version="1.0" encoding="iso-8859-1"?>'."
    textbackslash n"
.'<individus>'."textbackslash n";
$db = mysql_connect('localhost', 'login', 'mdp');
mysql_select_db('mytest',$db);
$indReq = 'SELECT individu_id, prenom, nom FROM individu';
$indResult = mysql_query($indReq);
while ($ind = mysql_fetch_object($indResult)) {
  $xmlstr .= '<individu id="'.$ind->individu_id.'">'
  .'<prenom>'.htmlentities($ind->prenom).'</prenom>'
  .'<nom>'.htmlentities($ind->nom).'</nom></individu>'."
     textbackslash n";
}
$xmlstr .= '</individus>'."textbackslash n";
mysql_close();
$individus = simplexml_load_string($xmlstr);
```

L'objet *« individus »* est de type `SimpleXMLElement` et contient votre fichier XML. Vous pouvez le sauver dans un le fichier *« copie.xml »* grâce à la ligne suivante :

```
$individus->asXML('copie.xml');
```

Si vous disposez déjà d'un fichier XML et que vous souhaitez juste instancier un objet `SimpleXMLElement` à partir de ce fichier, il suffit de faire :

```
if (!file_exists('individus.xml')) {
        exit('Impossible de lire individus.xml.');
}
$individus = simplexml_load_file('individus.xml');
```

4. Manipuler le XML

4.1 Afficher le fichier XML

A ce stade vous avez votre objet *« individus »* qui contient votre XML. Vous pouvez l'afficher ainsi :

```
header("Content-type: text/xml");
echo $individus->asXML();
```

4.2 Accéder aux champs et attributs

Si vous souhaitez afficher le prénom du second individu :

```
echo $individus->individu[1]->prenom;
```

Pour avoir la valeur de l'attribut *« id »* du troisième individu :

```
echo $individus->individu[2]['id'];
```

Pour afficher tous les noms :

```
foreach($individus->individu as $courant) {
  echo $courant->nom.'<br>';
}
```

4.3 Modifier des champs et attributs

Aussi aisément que vous avez affiché champs et attributs vous pouvez les modifier :

```
$individus->individu[1]->prenom = htmlentities('Hello');
$individus->individu[2]['id'] = 10;
```

Vous pouvez même créer de nouveaux champs et de nouveaux attributs. Cependant avec SimpleXML je vous conseille de vous restreindre aux attributs. Pour les champs utilisez DOM. Ajoutez un attribut *« ancien »* au second individu se résume à :

```
$individus->individu[2]['ancien'] = 'William';
```

5. Utilisation avancée

5.1 Utiliser Xpath

Xpath est à XML ce que SQL est aux bases de données. Toute proportion gardée bien-sûr. C'est à dire qu'il vous permet d'interroger un document XML et de n'extraire que certaines informations. Pour avoir tous les prénoms :

```
$prenoms = $individus->xpath('/individus/individu/prenom');
foreach($prenoms as $prenom) {
  echo $prenom.'<br>';
}
```

5.2 Etendre SimpleXMLElement

Comme tout langage orienté objet qui se respecte, PHP autorise la spécialisation d'une classe. Vous pouvez donc spécialiser `SimpleXMLElement` de façon à l'adapter au mieux à votre fichier XML. Créez le la classe MyXMLElement (**renommez le fichier en .php**) ainsi :

```
class MyXMLElement extends SimpleXMLElement {
function listePrenoms() {
return $this->xpath('/individus/individu/prenom');
}
}
```

Cette classe se contente de définir une méthode *« listePrenoms »* qui retourne la liste des prénoms. En supposant que vous ayez sauvegardé votre fichier XML dans individus.xml, vous pouvez instancier un nouvel objet s'appuyant sur cette classe ainsi :

```
include('MyXMLElement.php');
if (!file_exists('individus.xml')) {
        exit('Impossible de lire individus.xml.');
}
$special = simplexml_load_file('individus.xml','MyXMLElement');
```

Et vous pouvez jouer avec :

```
$prenoms = $special->listePrenoms();
foreach($prenoms as $prenom) {
  echo $prenom.'<br>';
}
```

5.3 Utiliser XSLT

PHP 7 est livré avec la classe XSLTProcessor qui permet de travailler de manière simplifié avec XSLT. Nous allons transformer notre XML en un fichier HTML. Je fournis un fichier XSL pour ça, mais rien ne vous empêche de faire le votre. Il ne vous reste plus qu'à générer le HTML :

```
$xslt = new XSLTProcessor();
$xslt->importStylesheet(DOMDocument::load('individus.xsl'));
$html = $xslt->transformToDoc($individus);
$html->formatOutput = true;
echo $html->saveXML();
```

Ceux qui connaissent l'API DOM ont remarqué que j'utilise cette dernière et non SimpleXML. En effet la ligne :

```
$html = $xslt->transformToDoc($individus);
```

Convertit l'objet *« individus »* de type `SimpleXMLElement` en un objet *« html »* de type `DOMDocument`.