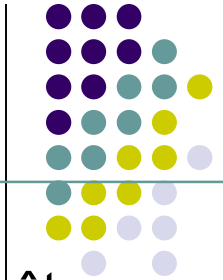
Mise à jour d'un document XML

Manipulation de données XML

El hadji Mamadou NGUER Enseignant chercheur en Informatique à l'UVS

Séquence 4 : Mise à jour d'un document XML



Objectifs spécifiques : A la suite de cette séquence, l'apprenant doit être capable de:

- Décrire comment mettre à jour un document XML dans le serveur
- Mettre à jour un document XML en utilisant les langages XSLT et PHP

elhadjimamadou.nguer@uvs.edu.sn

Séquence 4 : Mise à jour d'un document XML

Plan :

- Introduction
- Le formulaire de modification
- Affichage des données modifiées
- Modification des données
- Conclusion

elhadjimamadou.nguer@uvs.edu.sn

Introduction

- En utilisant XSL, il est possible, à partir d'un navigateur, de modifier les données stockées dans un fichier XML.

- Il s'agit d'utiliser les formulaires pour :

  - d'écrire les valeurs des éléments XML dans des champs de saisie de formulaire HTML.

  - puis de modifier les données depuis le formulaire

  - ensuite d'envoyer les données du formulaire au serveur qui se chargera de mettre à jour le document XML.

elhadjimamadou.nguer@uvs.edu.sn

Le formulaire de modification

Exemple:

On considère le document XML (produit.xml) ci-dessous qu'on voudrait modifier depuis le navigateur.

```
<?xml version="1.0" encoding="UTF-8"?>
<produit>
  <champ id="nomProd">
    <valeur>HAMMER HG2606</valeur>
  </champ>
  <champ id="NumProd">
    <valeur>32456240</valeur>
  </champ>
  <champ id="prix">
    <valeur>3000</valeur>
  </champ>
</produit>
```

Le formulaire de modification

Exemple: Le document XSL (**formproduit.xsl**) ci-dessous contenant le code du formulaire de modification de **produit.xml**.

```
<xsl:template match="/">
  <html><body>
  <form method="post" action="editerprod.php">
  <h2>Formulaire d'edition d'un produit :</h2>
  <table border="0">
    <xsl:for-each select="produit/champ">
    <tr>
      <td><xsl:value-of select="@id"/></td>
      <td><input type="text">
      <xsl:attribute name="id">
         <xsl:value-of select="@id" />
      </xsl:attribute>
      <xsl:attribute name="name">
        <xsl:value-of select="@id" />
      </xsl:attribute>
```

Le formulaire de modification

Exemple: Suite du document XSL (**formproduit.xsl**).

```
      <xsl:attribute name="value">
        <xsl:value-of select="valeur" />
      </xsl:attribute>
      </input>
      </td>
    </tr>
    </xsl:for-each>
  </table>
  <br />
<input type="submit" id="btn_sub" name="btn_sub" value="Submit"/
>
<input type="reset" id="btn_res" name="btn_res" value="Reset" />
  </form>
  </body>
  </html>
</xsl:template>
```

Le formulaire de modification

Exemple: Explication du code du document XSL (**formproduit.xsl**).

- Le fichier XSL ci-dessus boucle à travers les éléments "champ" du fichier XML et crée un champ d'entrée pour chaque élément XML.
- La valeur de l'attribut "id" de l'élément "champ" du document XML est ajouté à la fois dans les attributs "id" et "name" de chaque champ de saisie HTML.
- La valeur de chaque élément "valeur" du document XML est ajouté à l'attribut "value" de chaque champ de saisie HTML.
- Le résultat est un formulaire HTML modifiable contenant les valeurs provenant du fichier XML.
- Ensuite le fichier "**produpdated.xsl**" est utilisé pour afficher sous forme de table HTML statique les données XML mises à jour.

Affichage des données modifiées

Exemple: Le fichier "produpdated.xsl" utilisé pour afficher sous forme de table HTML statique les données XML mises à jour.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html><body>
  <h2>Les données du produit modifié:</h2>
  <table border="1">
    <xsl:for-each select="produit/champ">
    <tr>
      <td><xsl:value-of select="@id" /></td>
      <td><xsl:value-of select="valeur" /></td>
    </tr>
    </xsl:for-each>
  </table>
  </body></html>
</xsl:template>
```

Modification des données

Exemple: Le fichier "editerprod.php"

La page "editerprod.php" contient deux fonctions: la fonction loadFile() charge et transforme le fichier XML à afficher et la fonction updateFile() applique les modifications apportées au fichier XML:

```
<?php
function loadFile($xml, $xsl){
        //Charge le ficher XML
        $xmlDoc = new DOMDocument();
        $xmlDoc->load($xml);
        //Charge le ficher XSL
        $xslDoc = new DOMDocument();
        $xslDoc->load($xsl);
        //Crée le transformateur
        $proc = new XSLTProcessor();
        //Ajoute le xsl dans le transformateur
        $proc->importStyleSheet($xslDoc);
        //Transforme le document XML
        echo $proc->transformToXML($xmlDoc);
}
```

Modification des données

Exemple: Le fichier "editerprod.php" contient deux fonctions: la fonction loadFile() charge et transforme le fichier XML à afficher et la fonction updateFile() applique les modifications apportées au fichier XML:

```
function updateFile($xml){
    $xmlLoad = simplexml_load_file($xml); /*simplexml_load_file
convertit un fichier XML en objet de type SimpleXMLElement*/

    foreach($xmlLoad->children() as $x){
        foreach($_POST as $key=>$value){
            if($key == $x->attributes()){
                $x->value = $value;
            }
        }
    }

    $xmlLoad->asXML($xml); /*Ecrit les données de $xmlLoad dans le
    fichier XML dont le nom est $xml*/
    loadFile($xml,"produpdated.xsl"); /*Afficher les nvelles données*/
}
```

Parcourir l'objet $xmlLoad pour remplacer les données modifiées

Modification des données

Exemple: Le fichier "editerprod.php" (Suite)

```
if($_POST["btn_sub"] == ""){
	loadFile("produit.xml", "formproduit.xsl");
}
else{
	updateFile("produit.xml");
}
?>
```

Remarque:

- La transformation est faite et les modifications apportées au fichier XML sont appliquées au serveur. Cette solution fonctionne dans tous les navigateurs. Le client ne recevra que le code HTML provenant du serveur.
- Pour plus d'information sur les classes PHP de traitement du DOM, voir :
  - **http://php.net/manual/fr/class.domdocument.php**
  - **http://php.net/manual/fr/class.xsltprocessor.php**
  - **http://php.net/manual/fr/class.simplexmlelement.php**

Fin de la séquence 4