

Récupérer et traiter avec XSL

Avec ce chapitre sur le traitement du XSL dans les applications AJAX, nous sommes à la limite du fonctionnement des navigateurs actuels, et les solutions qu'ils proposent sont souvent spécifiques et donc hautement incompatibles.

Nous avons étudié, au chapitre Introduction au XSL, la liaison d'une feuille de style XSL (fichier .xsl) dans un document XML (fichier .xml).

Celle-ci se réalisait par une ligne de code dans le fichier XML. Pour rappel :

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="xsldemo.xsl"?>
<compilation>
<mp3>
<titre>Sarbacane</titre>
<artiste>Francis Cabrel</artiste>
<date>1990</date>
</mp3>
Etc.
```

Ce système, bien que très puissant, ne prend cependant en charge que l'affichage du document XML dans un navigateur, dont la vocation première est, rappelons-le, d'afficher du HTML ou du XHTML.

Avec le modèle AJAX, il faut avoir la possibilité d'accéder aux éléments du fichier XML et XSL. Accessoirement, il faut aussi pouvoir changer, de façon dynamique (par le JavaScript), la feuille de style associée au document XML.

Il a fallu donc mettre en place une technique différente de celle habituellement utilisée. C'est ce que nous allons aborder dans l'étude de ce sous-chapitre.

Nous allons utiliser comme fichier XML (xml.xml) le code suivant, déjà donné en exemple au chapitre Introduction au XSL :

```
<?xml version="1.0" encoding="UTF-8"?>
<compilation>
<mp3>
<titre>Sarbacane</titre>
<artiste>Francis Cabrel</artiste>
<date>1990</date>
</mp3>
<mp3>
<titre>Nickel</titre>
<artiste>Alain Souchon</artiste>
<date>1991</date>
</mp3>
<mp3>
<titre>Sheller en solitaire</titre>
<artiste>William Sheller</artiste>
<date>1992</date>
</mp3>
<mp3>
<titre>Caché derrière</titre>
<artiste>Laurent Voulzy</artiste>
```

```

<date>1993</date>
</mp3>
<mp3>
<titre>Rio Grande</titre>
<artiste>Eddy Mitchell</artiste>
<date>1994</date>
</mp3>
<mp3>
<titre>Samedi soir sur la Terre</titre>
<artiste>Francis Cabrel</artiste>
<date>1995</date>
</mp3>
<mp3>
<titre>Défoule sentimentale</titre>
<artiste>Alain Souchon</artiste>
<date>1996</date>
</mp3>
</compilation>


```

Et comme fichier de feuille de style XSL (fichier xsl.xsl) :

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method='html' version='1.0' encoding='UTF-8'
indent='yes' />
<xsl:template match="/">
<html>
<head>
<title>XSL</title>
</head>
<body>
<table border="1" cellpadding="3" style="border-collapse:
collapse">
<tr style="background: rgb(195,215,235)">
<td>Année</td>
<td>Album</td>
<td>Artiste</td>
</tr>
<xsl:for-each select="compilation/mp3">
<tr>
<td><xsl:value-of select="date"/></td>
<td><xsl:value-of select="titre"/></td>
<td><xsl:value-of select="artiste"/></td>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

 Il faut remarquer qu'il n'y a pas de liaison entre le fichier XML et le fichier XSL.

Le fichier HTML est inchangé.

```
<!DOCTYPE html>
<html lang="fr">
<head>
<title>Le AJAX</title>
<meta charset="UTF-8">
<style>
#transform { background-color: #9cf;
              padding: 10px;
              text-align: center;}
</style>
</head>
<body>
<h2>Les victoires de la musique</h2>
<div id="transform">
<form>
<input type="button" value="Afficher le palmarès"
onclick="transform(xml, xsl, 'transform')">
</form>
</div>
</body>
</html>
```

Le script :

```
<script>
function loadXML(url) {
var xhr;
if (window.ActiveXObject) {
xhr = new ActiveXObject("Microsoft.XMLHTTP");
}
else {
xhr = new XMLHttpRequest();
}
xhr.open("GET", url, false);
try {xhr.responseType = "msxml-document"} catch(err) {} // pour IE11
xhr.send("");
return xhr.responseXML;
}
```

La fonction `loadXML()` crée un objet `XMLHttpRequest` (`xhr`). Les méthodes `open()` et `send()` sont utilisées pour envoyer une requête au serveur. Le fichier retourné est sous la forme de données XML (`responseXML`).

```
function afficher() {
xml = loadXML("xml.xml");
xsl = loadXML("xsl.xsl");
// code for Internet Explorer
```

```

if (window.ActiveXObject || xhr.responseType == "msxml-document") {
    exemple = xml.transformNode(xsl);
    document.getElementById("exemple").innerHTML = exemple;
}
// code pour Chrome, Firefox, etc.
else if (document.implementation &&
document.implementation.createDocument){
    xsltProcessor = new XSLTProcessor();
    xsltProcessor.importStylesheet(xsl);
    resultDocument = xsltProcessor.transformToFragment(xml, document);
    document.getElementById("exemple").appendChild(resultDocument);
}
}
</script>

```

La fonction `afficher()` charge d'abord les fichiers XML (`loadXML("xml.xml")`) et XSL (`loadXML("xsl.xsl")`). Ensuite le script teste (if) le navigateur de l'utilisateur.

S'il s'agit d'Internet Explorer (`window.ActiveXObject || xhr.responseType == "msxml-document"`), la méthode `transformNode()` est utilisée pour appliquer la feuille de style XSL au document XML. Le document, ainsi stylé, prend place (`innerHTML`) dans la division identifiée par `exemple`. Pour Firefox et Chrome (`document.implementation && document.implementation.createDocument`), le script crée un nouvel objet `XSLTProcessor` et importe dans celui-ci le fichier XSL.

Le fichier HTML complet :

```

<!DOCTYPE html>
<html lang="fr">
<head>
<title>AJAX</title>
<meta charset="UTF-8">
<script>

function loadXML(url) {
    if (window.ActiveXObject) {
        xhr = new ActiveXObject("Microsoft.XMLHTTP");
    }
    else {
        xhr = new XMLHttpRequest();
    }
    xhr.open("GET", url, false);
    try {xhr.responseType = "msxml-document"} catch(err) {} // pour
    IE11
    xhr.send("");
    return xhr.responseXML;
}

function afficher() {
    xml = loadXML("xml.xml");
    xsl = loadXML("xsl.xsl");
    // code for Internet Explorer
    if (window.ActiveXObject || xhr.responseType == "msxml-document")
    {
        exemple = xml.transformNode(xsl);
        document.getElementById("exemple").innerHTML = exemple;
    }
}

```

```

}
// code pour Chrome, Firefox, etc.
else if (document.implementation &&
document.implementation.createDocument){
xsltProcessor = new XSLTProcessor();
xsltProcessor.importStylesheet(xsl);
resultDocument = xsltProcessor.transformToFragment(xml, document);
document.getElementById("exemple").appendChild(resultDocument);
}
}
</script>
</head>
<body onload="afficher()">
<h2>Les victoires de la musique</h2>
<div id="exemple">
</body>
</html>

```



The screenshot shows a web browser window with the title 'AJAX' and the address bar displaying 'localhost/w3c/'. The page content features a heading 'Les victoires de la musique' followed by a table with three columns: 'Année', 'Album', and 'Artiste'. The table lists the following data:

Année	Album	Artiste
1990	Sarbacane	Francis Cabrel
1991	Nickel	Alain Souchon
1992	Sheller en solitaire	William Sheller
1993	Caché derrière	Laurent Voulzy
1994	Rio Grande	Eddy Mitchell
1995	Samedi soir sur la Terre	Francis Cabrel
1996	Défoule sentimentale	Alain Souchon