

AP XML 2 : créer une « base de données » de films en XML

1. XML : fichier de données

Un fichier XML est un fichier au format texte qui contient des données formatées avec des balises gérées par le créateur du fichier.

Vous pouvez donc écrire un fichier XML avec un éditeur quelconque, mais il est préférable d'utiliser un parseur. Un parseur est un éditeur spécifique qui va vous permettre :

- de visualiser les balises plus clairement (généralement avec une autre couleur)
- de valider le fichier (en testant la syntaxe)
- d'accepter différents formats en rapport avec le XML (DTD, XSL...)

Suivant les parseurs, les possibilités sont plus ou moins étendues.

Pour ce TP, vous allez travailler avec un parseur freeware : Cooktop. Récupérez le sur SIONAS ou sur le site <http://xmlcooktop.com/>

Pour les utilisateurs de mac ou linux, réalisez l'activité avec une VM Windows.

a) Création d'un fichier XML

Un fichier XML présente des informations entourées de balises. Chaque information est obligatoirement entourée d'une balise ouvrante et d'une balise fermante.

Le fichier XML possède obligatoirement une balise racine, qui peut contenir elle même une à plusieurs balises, etc...

Commençons par un premier exemple :

→ Ouvrez Cooktop. Dans le menu File, prenez New puis XML Cooktop et OK. Placez vous dans le premier onglet (source xml) de la fenetre qui vient de s'ouvrir et tapez le texte suivant :

```
<films>
  <film>
    <titre>Le seigneur des anneaux</titre>
    <genre>Aventure</genre>
    <annee>2001</annee>
    <duree>180</duree>
    <acteurs>
      <personne>Ian McKellen</personne>
      <personne>Liv Tyler</personne>
    </acteurs>
  </film>
</films>
```

Remarquez le principe : vous avez en quelque sorte un fichier de films (la balise racine est "films") qui contient des films (pour le moment un seul) délimité par la balise "film". Pour chaque film, on a un titre, un genre, une année, une durée et des acteurs. La présentation (décalages et retour à la ligne) n'a pas d'incidence sur l'interprétation du fichier, mais permet de clarifier et simplifier la lecture.

→ Sauvez votre fichier xml (avec le nom films.xml) en faisant File/Save As (ne sauvez que le format xml, annulez les 2 autres sauvegardes proposées).


b) Edition d'un fichier XML

Vous allez visualiser dans le navigateur le résultat de ce fichier xml, brut, sans transformation.

→ Ouvrez le navigateur WEB et faites un "drag and drop" de votre fichier vers la fenêtre le navigateur WEB. Cette fois, votre fichier apparaît mais avec un formatage un peu particulier, que vous connaissez déjà : certaines balises ouvrantes sont précédées du signe "-" sur lequel vous pouvez cliquer pour fermer ou ouvrir le contenu d'une balise. Essayez. Fermez le navigateur.

c) Validation d'un fichier XML

Le parseur est capable de contrôler si les balises sont bien placées et s'il n'en manque pas.

→ Enlevez une des balises fermantes puis cliquez sur le bouton Validate 

Normalement, une erreur doit s'afficher vous précisant à partir de quel endroit les choses ne sont plus normales.

→ Rajoutez la balise précédemment effacée et recliquez sur Validate. Cette fois cela devrait bien se passer (aucun message).

Exercices :

- 1) Complétez le fichier XML en rajoutant 2 autres films.
- 2) Validez le fichier.
- 3) Editez le fichier XML sous le navigateur WEB et vérifiez que tout fonctionne correctement.

2. DTD (Document Type Definition) : fichier de structure

Le fichier DTD est un fichier de description, qui permet de décrire à quoi doit ressembler le fichier XML.

Ce fichier n'est pas obligatoire mais, quand il est présent, permet de contrôler le contenu d'un fichier XML.

Il est possible d'inclure le code DTD directement dans le code XML, ou de faire un fichier séparé et ensuite de l'inclure à la bonne place dans le fichier XML. Nous allons utiliser la seconde méthode.

→ Dans le menu, prenez File, New et cette fois choisissez un fichier DTD puis. Tapez le code suivant

```
<!ELEMENT films (film+)>
<!ELEMENT film (titre, genre, annee, duree, acteurs)>
<!ELEMENT titre (#PCDATA)>
<!ELEMENT genre (#PCDATA)>
<!ELEMENT annee (#PCDATA)>
<!ELEMENT duree (#PCDATA)>
<!ELEMENT acteurs (personne+)>
<!ELEMENT personne (#PCDATA)>
```

→ Sauvez sous le nom films.dtd (attention le fichier doit être placé dans le même répertoire que le fichier xml)

Cela mérite quelques explications.

Chaque ligne permet de décrire un élément (il n'y a pas que les éléments qui peuvent être décrits, mais pour le moment nous ne verrons qu'eux).

Un élément comporte un nom et une description. Cette description peut être #PCDATA, #CDATA (pour les types simples), une liste d'autres éléments, un élément dupliqué (avec +) etc...

Dans l'exemple, on voit bien que "films" est constitué de plusieurs éléments de type "film". Chaque film est constitué d'une liste d'éléments distincts (titre; genre...). Puis, chaque élément de cette liste est ensuite décrit.

Maintenant il faut lier ce fichier de description au fichier XML.

→ Retournez dans votre fichier XML, créez une ligne vierge au tout début et placez le curseur sur cette ligne. Prenez dans les menus "Code bits"/DTD/Doctype/system. Une ligne de code s'est automatiquement insérée. A la place de l'étoile dans *.dtd, mettez "films" pour obtenir films.dtd. Remplacez aussi "top_element" par "films" car il faut mettre ici le nom de la racine.

→ Sauvez et validez. Normalement rien ne doit se passer (NB : la validation a bien marché)

→ Pour le plaisir, on va tester si les contrôles se font bien. On ne va pas supprimer une balise fermante, cette fois, mais une balise complète : supprimez la ligne complète année pour un des 3 films puis validez. Normalement cela provoque une erreur car il attend la balise année d'après le fichier DTD. D'ailleurs le message d'erreur précise la balise manquante. Remettez la ligne précédemment effacée. Il n'y aurait pas le lien avec le fichier DTD, aucune erreur ne serait apparue car l'absence d'une balise complète ne peut être détectée.

Exercices :

- 4) Pour chaque film, rajoutez (dans le fichier XML) le réalisateur.
- 5) Modifiez le fichier DTD en conséquence.
- 6) Validez les fichiers et testez. Sous le navigateur WEB.

3. XSL : fichier de formatage et de recherche (transformation en HTML)

Pour le moment, les fichiers étudiés permettent de stocker des informations et éventuellement de décrire leur structure.

Le but d'un fichier XML est bien sûr d'être affiché. Le fichier XSL a pour rôle de formater les données du fichier XML en vue d'une intégration dans une page HTML. Cette intégration ne concerne pas forcément tout le fichier xml mais uniquement les parties à afficher.

Le parseur permet d'intégrer facilement les lignes de code qui gèrent la structure d'un fichier XSL :

→ Sur le fichier xml (dans Cooktop), cliquez sur l'onglet "stylesheet (XSL)". Dans le menu "Code bits", sélectionnez "XSL HTML Template". Normalement du code s'est automatiquement intégré dans la page. C'est la structure d'un fichier xsl. Enregistrez juste le xsl sous le nom films.xsl

Voici quelques explications en ce qui concerne ce code :

```
<?xml version="1.0"?>
```

Cette ligne n'est pas obligatoire : elle permet juste de préciser la version xml utilise.

```
<xsl:stylesheet version="1.0" xmlns:xsl =  
"http://www.w3.org/1999/XSL/Transform">
```

Cette ligne permet de préciser la version xsl ainsi que l'origine du nom de préfixe donné à toutes les variables, pour éviter les confusions.

```
<xsl:output method = "html" encoding="Windows-1252" />
```

Cette ligne est très importante et permet de dire que le but est d'obtenir du html.

```
<xsl:template match="/">
```

Par défaut le positionnement se fait sur la racine.

```
<html>  
  <title></title>  
  <body></body>  
</html>
```

On retrouve les balises classiques du html (qui ne sont d'ailleurs pas obligatoire).

```
</xsl:template>
```

```
</xsl:stylesheet>
```

Fermeture des balises

Dans un premier temps, le but va être d'afficher simplement les titres des films.

→ Entre les balises "body", insérez le code suivant :

```
<xsl:for-each select="film">  
  Titre du film : <xsl:value-of select = "titre"/><br/>  
</xsl:for-each>
```

Ce code va permettre de boucler sur chaque film, et afficher "Titre du film :" suivi du titre récupéré dans le fichier xml.

→ Vous devez aussi modifier le match dans la balise template. Pour le moment, il est à "/" et vous devez le positionner sur les films, donc : "films".

→ Sauvez et validez le fichier XSL.

→ Pour tester le résultat, vous pouvez le faire directement dans le parseur (cliquez sur le bouton XSLT).

Exercices :

7) Créez un autre fichier XSL qui va permettre d'afficher, pour chaque film, son titre, son genre et les acteurs.

8) Validez et testez ce nouveau fichier.

4. Création d'une base de données

Nous avons vu comment fonctionne un fichier xml qui correspond à une table dans une base de données classique. Voyons comment enregistrer une base complète.

Il suffit de se positionner à un niveau au dessus. Donc, dans le même fichier xml, il est possible de faire cohabiter toutes les occurrences des "tables" de la base de données. Pour cela, il faut inclure une nouvelle notion, celle de lien, et donc d'identifiant.

Il est possible d'ajouter des informations de type identifiant sur les éléments d'un fichier xml. Un identifiant est un attribut rattaché à un élément. Voyons le fonctionnement :

- Faites une copie du fichier xml que vous avez créé et appelez le "basefilms.xml".
- Ouvrez ce fichier xml sous Cooktop. Créez une racine au dessus de <films> et appelez la <basefilms>. Pensez à insérer, à la fin du fichier, la balise fermante correspondante.
- Donc basefilms contient en quelque sorte la "table" films. On va lui insérer la "table" artistes. En dessous de la balise <basefilms>, créez les balises <artistes></artistes> et, entre ces balises, insérez des artistes (mettez en 3 ou 4, avec un nom et une catégorie, du genre "acteur" ou "réalisateur").
- Faites une copie de votre ancien fichier dtd et renommez le en basefilms.dtd. Apportez les modifications nécessaires.
- Dans le fichier xml, pensez aussi à changer le nom du fichier dtd.
- Validez et testez l'ensemble.
- Dans le fichier DTD, rajoutez en fin de fichier la ligne suivante :

```
<!ATTLIST artiste codartiste ID #REQUIRED>
```

Cela signifie que vous ajoutez un attribut à l'élément artiste. Un attribut permet d'apporter une information complémentaire à un élément. Cet attribut s'appelle codartiste et il est de type ID (identifiant) et obligatoire.

Maintenant il faut modifier le fichier xml en conséquence :

- Dans le fichier xml, chaque ligne contenant l'entête de l'élément artiste et qui devait normalement être écrite de la façon suivante :

```
<artiste>
```

doit être modifié de la façon suivante :

```
<artiste codartiste = "A">
```

Bien sûr ne mettez pas "A" pour chaque artiste, mais un code différent...

(utilisez des lettres et non des chiffres dans vos codes)

- Validez à nouveau votre fichier xml.

L'identifiant est créé pour les artistes. Essayons de modifier dans la "table" films les informations correspondantes :

- Dans le fichier dtd, rajoutez la ligne suivante :

```
<!ATTLIST realisateur codartiste IDREF #REQUIRED>
```

Ceci indique que le réalisateur possède en fait une clé étrangère (IDREF et non ID qui est une clé primaire) sur codartiste, qui a déjà été défini.

- Dans le fichier xml, modifiez toutes les lignes "realisateur" pour les remplacer par des lignes du genre :

```
<realisateur codartiste = "A"/>
```

A vous de mettre à chaque fois le code de la clé étrangère qui correspond à la clé primaire de l'artiste concerné.

Vous remarquez le signe / avant le > qui signifie que la balise se ferme. C'est un raccourci d'écriture qui évite d'écrire la balise fermante, puisqu'on a aucune information à marquer dans l'élément réalisateur.

→ Validez le fichier xml pour voir s'il n'y a pas d'erreur.

→ Créez un fichier xsl qui permet de présenter, pour chaque film, le titre et le nom du réalisateur. Pour cela, modifiez la boucle (ce n'est plus "film" mais "films/film"). L'affichage du titre ne change pas. En revanche, pour le réalisateur, vous devez d'abord créer une variable intermédiaire qui stocke le code du réalisateur :

```
<xsl:variable name="val" select="realisateur/@codartiste"></xsl:variable>
```

Puis il faut chercher le nom de l'artiste par rapport au code :

```
realisateur : <xsl:value-of select = "/basefilms/artistes/artiste[@codartiste=$val]/nom"/><br/>
```

→ Sauvez et testez.

Annexe 1 : XML

Le XML est sensible à la casse (<nom> est différent de <Nom>).

Le commentaire en XML :

```
<!-- commentaire -->
```

Tout ce qui n'est pas balise est du texte. Attention, dans le texte, 5 caractères sont interdits et doivent être remplacés :

```
<      à remplacer par &lt;
>      à remplacer par &gt;
&      à remplacer par &amp;
"      à remplacer par &quot;
'      à remplacer par &apos;
```

Annexe 2 : DTD

Structure d'un fichier DTD :

```
<!ELEMENT nom_racine (nom_element+)>
<!ELEMENT nom_element (contenu_element)>
<!ELEMENT nom_element1 (contenu_element1)>
<!ELEMENT nom_element2 (contenu_element2)>
...
```

Contenu élément :

#PCDATA	données textuelles qui seront traitées par le parseur
#CDATA	données textuelles qui ne seront pas traitées par le parseur
EMPTY	signifie que l'élément ne peut rien contenir
ANY	signifie que l'élément peut contenir n'importe quoi
nom_element	l'élément est du type d'un autre élément
nom_element+	l'élément contient un ou plusieurs éléments de même type
nom_element*	l'élément contient zéro ou plusieurs éléments de même type
nom_element?	l'élément contient zéro ou un éléments de même type
(liste d'éléments)	l'élément contient une liste d'éléments différents