

`XML – XSL en PHP`

- Tawfiq CADI TAZI
- tawfiq.caditazi@etskirsch.fr

Introduction au XML

- Qu'est-ce que le XML
- Spécifications et bénéfices du XML
- Les règles du XML
- Utilisation du XML

Qu'est ce que le XML

- XML = Extensible Markup Language
- Langage de balise permettant de décrire la donnée de manière standardisée
- Standard de représentation et de transport de donnée
- Pour utiliser des données représentée en XML il faut utiliser un « parser »
- L'utilisateur est libre de définir ses propres marqueurs
- XML est recommandé par la W3C dès février 1998

Avantages du XML

- Un standard recommandé par la W3C
- Grâce aux balises un fichier XML est plus facilement lisible qu'un fichier CSV

Règles de bases

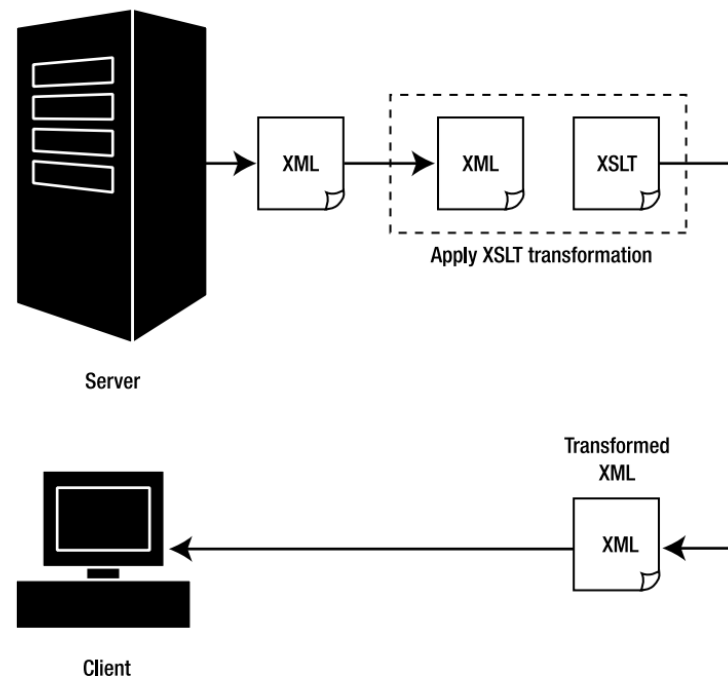
- XML est sensible à la casse <client> est différent de <Client> qui est différent de <CLIENT>
- Un document XML doit avoir un seul élément racine
- Toute balise ouverte doit être fermée
- Les début et fin de balises doivent être correctement imbriquées
- Les attributs doivent être entre des guillemets

Parser un document XML

- Pour exploiter un document XML il faut pouvoir le parcourir ou le parser.
- Il existe plusieurs manière de parser un document XML
 - Les parsers basés sur DOM (Document Objet Model)
 - Les parsers basés sur SAX (Simple API for XML)

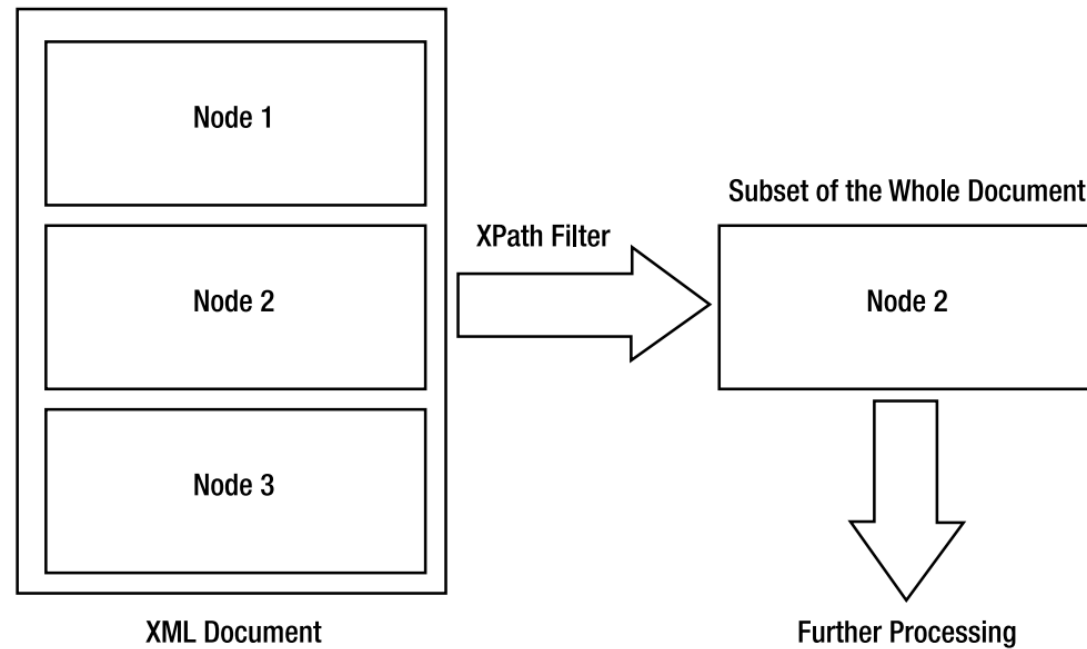
XSLT Transformation d'un fichier XML

- XSLT Permet de transformer un fichier XML
- XSLT peut servir par exemple
 - Pour transformer un fichier XML en HTML



XPATH

- XPath est un moyen de sélectionner des noeuds d'un documents XML grâce à un ensemble de fonctions



Manipuler un document XML en utilisant le DOM

- Quand utiliser le DOM
 - DOM permet d'écrire et de lire dans un document XML, avez-vous besoin de lire et d'écrire ?
 - DOM charge l'ensemble du document en mémoire, quel est la taille du document XML ?
 - DOM permet d'accéder directement à tout les nœuds comme il est chargé en mémoire, est-ce qu'un accès séquentiel serait suffisant ?

Quand utiliser le DOM

- Lorsque vous voulez lire et écrire dans un document XML
- Un accès séquentiel n'est pas suffisant
- Vous avez assez de mémoire pour charger tout le document

Utilisation du Parser DOM

- Voir fichier exemple : Employees.xml
 - Ouverture d'un document XML chargé en mémoire et traiter comme un arbre
 - Ouverture par URL
 - Ouverture par Stream
 - Ouverture par chaîne de caractères
- Chargement sous forme d'arbre

Retrouver un élément spécifique

- `GetElementByTagName()`
- `GetElementById()`
- `SelectNodes()` et `SelectSingleNode()` avec XPath

Xpath Selection

Expression	Description
/	Selection des enfants direct du nœud courant
//	Sélection des descendant du nœud courant
.	Sélection du nœud courant
..	Sélection du parent du noeud courant
@	Selection des attributs

Xpath Selection Examples

Path Expression	Result
bookstore	Selects all nodes with the name "bookstore"
/bookstore	Selects the root element bookstore Note: If the path starts with a slash (/) it always represents an absolute path to an element!
bookstore/book	Selects all book elements that are children of bookstore
//book	Selects all book elements no matter where they are in the document
bookstore//book	Selects all book elements that are descendant of the bookstore element, no matter where they are under the bookstore element
//@lang	Selects all attributes that are named lang

XPath Predicates

Path Expression	Result
<code>/bookstore/book[1]</code>	Selects the first book element that is the child of the bookstore element
<code>/bookstore/book[last()]</code>	Selects the last book element that is the child of the bookstore element
<code>/bookstore/book[last()-1]</code>	Selects the last but one book element that is the child of the bookstore element
<code>/bookstore/book[position()<3]</code>	Selects the first two book elements that are children of the bookstore element
<code>//title[@lang]</code>	Selects all the title elements that have an attribute named lang
<code>//title[@lang='eng']</code>	Selects all the title elements that have an attribute named lang with a value of 'eng'
<code>/bookstore/book[price>35.00]</code>	Selects all the book elements of the bookstore element that have a price element with a value greater than 35.00
<code>/bookstore/book[price>35.00]/title</code>	Selects all the title elements of the book elements of the bookstore element that have a price element with a value greater than 35.00

Xpath WildCards

Wildcard	Description
*	Matches any element node
@*	Matches any attribute node
node()	Matches any node of any kind

Path Expression	Result
/bookstore/*	Selects all the child nodes of the bookstore element
//*	Selects all elements in the document
//title[@*]	Selects all title elements which have any attribute

Xpath Operators

Operator	Description	Example
	Computes two node-sets	//book //cd
+	Addition	6 + 4
-	Subtraction	6 - 4
*	Multiplication	6 * 4
div	Division	8 div 4
=	Equal	price=9.80
!=	Not equal	price!=9.80
<	Less than	price<9.80
<=	Less than or equal to	price<=9.80
>	Greater than	price>9.80
>=	Greater than or equal to	price>=9.80
or	or	price=9.80 or price=9.70
and	and	price>9.00 and price<9.90
mod	Modulus (division remainder)	5 mod 2

XPath fonctions sur les noeuds

Fonction	Description
last()	Returns the number of nodes in the current node set
position()	Returns the index of the context node in the current node set
count()	Returns the total number of nodes in the given node set
id()	Returns a node set containing nodes with an ID attribute matching the specified value
name()	Returns the fully qualified name of the specified node
text()	Returns the text of the specified node

XPath fonctions de texte

Fonction	Description
concat()	Returns a concatenated string
starts-with()	Returns true if the string starts with the specified letters
contains()	Returns true if the string contains the specified string
substring()	Returns part of the specified string
string-length()	Returns the number of characters in the string
translate()	Replaces characters from a string with the specified characters

XPath fonctions mathématiques

Fonction	Description
number()	Converts the specified string to its equivalent number
sum()	Returns the sum of numbers
floor()	Returns a number rounded down to the next integer
ceiling()	Returns a number rounded up to the next integer
round()	Returns a number rounded to the nearest integer
number()	Converts the specified string to its equivalent number