Jumpstart XML Transformation

XML File System

Ein File System ist als Tree aufgebaut und ist damit auch im Format XML darstellbar. Mit dem Tool xfs.exe können wir selber ein XML File vom File System erstellenDer folgende Befehl erstellt ein XML File vom Pfad c:\xslkurs mit allen Dateien:xfs -path c:\xslkurs*.* -out xfsdump.xml -formatDas folgende Listing zeigt das erstellte XML File:<?xml version="1.0" encoding="UTF-8"?>
<filesystem>
 <head>
 <meta name="path">C:\xslkurs\jumpstart\</meta>
 <meta name="search">c:\xslkurs*.*</meta>
 <meta name="drive">c:</meta>
 <meta name="subdir">false</meta>
 </head>
 <body>
 <folder name="." path="c:/xslkurs/." createTime="2022-08-15T21:51:45+01:00" accessTime="2022-08-19T18:59:35+01:00" writeTime="2022-08-19T18:59:35+01:00"></folder>
 <folder name=".." path="c:/xslkurs/.." createTime="2022-08-15T21:51:45+01:00" accessTime="2022-08-19T18:59:35+01:00" writeTime="2022-08-19T18:59:35+01:00"></folder>
 <folder name=".metadata" path="c:/xslkurs/.metadata" createTime="2022-08-15T21:51:45+01:00" accessTime="2022-08-19T18:59:35+01:00" writeTime="2022-08-19T18:59:35+01:00"></folder>
 <folder name="hello" path="c:/xslkurs/hello" createTime="2022-08-15T21:51:45+01:00" accessTime="2022-08-19T18:59:35+01:00" writeTime="2022-08-19T18:59:35+01:00"></folder>
 <folder name="jumpstart" path="c:/xslkurs/jumpstart" createTime="2022-08-15T21:51:45+01:00" accessTime="2022-08-19T18:59:35+01:00" writeTime="2022-08-19T18:59:35+01:00"></folder>
 <folder name="note" path="c:/xslkurs/note" createTime="2022-08-15T21:51:45+01:00" accessTime="2022-08-19T18:59:35+01:00" writeTime="2022-08-19T18:59:35+01:00"></folder>
 <folder name="saxon" path="c:/xslkurs/saxon" createTime="2022-08-15T21:51:45+01:00" accessTime="2022-08-19T18:59:35+01:00" writeTime="2022-08-19T18:59:35+01:00"></folder>
 </body>
</filesystem>Falls Sie das xfs.exe Programm nicht verwenden dürfen, kopieren Sie das Dokument aus dem Listing.Erstellen Sie in der Eclipse IDE das Projekt "jumpstart" und die Datei jumpstart.xml. Erstellen Sie mit xfs.exe einen File System XML Dump oder kopieren Sie das obige Listing.

XSLT for-each Schleife

Eine for-each Schleife ist eine Schleife, die einen bestimmten Wert für eine bestimmte Variable durchläuft. Beim ersten Ansatz verwenden die for-each Schleife und traversieren alle Folders und Files in eine Text Datei:<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:output method="text"/>
 <xsl:template match="filesystem">
 <xsl:for-each select="body/folder">
 folder <xsl:value-of select="@name"/>
 </xsl:for-each>
 <xsl:for-each select="body/file">
 file <xsl:value-of select="@name"/>
 </xsl:for-each>
 </xsl:template>
</xsl:stylesheet>Erstellen Sie im Verzeichnis

"jumpstart" die Datei "jumpstart1.xsl" mit dem obigen Code.Transformieren Sie die Datei "jumpstart.xml" mit der Datei "jumpstart1.xsl" und speichern Sie das Ergebnis in der Datei "jumpstart1.txt" ab:java -jar ..\saxon\saxon-he-11.4.jar -s:jumpstart.xml -xsl:jumpstart1.xsl -o:jumpstart1.txtAnalysieren Sie das Script und das Resultat.

XSLT Template

XSL Templates bilden ein zentrales Konzept für die Programmierung von XSLT Scripts. Templates sind eigentlich Funktionen, die indirekt über Apply Anweisungen angewendet auf die XML Daten aufgerufen werden. Das Template Matching definiert via XPath, welche XML Daten auf das Template anwendbar sind. Das Listing mit dem Template Ansatz:<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:output method="text"/>
 <xsl:template match="filesystem">
 <xsl:apply-templates select="body/folder"/>
 <xsl:apply-templates select="body/file"/>
 </xsl:template>
 <xsl:template match="folder">
 folder <xsl:value-of select="@name"/>
 </xsl:template>
 <xsl:template match="file">
 file <xsl:value-of select="@name"/>
 </xsl:template>
</xsl:stylesheet>Erstellen Sie im Verzeichnis "jumpstart" die Datei "jumpstart2.xsl" mit dem obigen Code.Transformieren Sie die Datei "jumpstart.xml" mit der Datei "jumpstart2.xsl" und speichern Sie das Ergebnis in der Datei "jumpstart2.txt" ab:java -jar ..\saxon\saxon-he-11.4.jar -s:jumpstart.xml -xsl:jumpstart2.xsl -o:jumpstart2.txtAnalysieren Sie das Script und das Resultat.

XSLT xsl:if/xsl:choose

Mit XSLT sind einfache Entscheidungen mit xsl:if und Mehrfachentscheidungen mit xsl:choose/xsl:when/xsl:otherweise programmierbar:<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:output method="text"/>
 <xsl:template match="filesystem">
 <xsl:apply-templates select="body/folder"/>
 <xsl:apply-templates select="body/file"/>
 </xsl:template>
 <xsl:template match="folder">
 <xsl:choose>
 <xsl:when test="@name = '.'">
 this folder
 </xsl:when>
 <xsl:when test="@name = '..'">
 parent folder
 </xsl:when>
 <xsl:otherwise>
 folder <xsl:value-of select="@name"/>
 </xsl:otherwise>
 </xsl:choose>
 </xsl:template>
 <xsl:template match="file">
 file <xsl:value-of select="@name"/>
 </xsl:template>
</xsl:stylesheet>Erstellen Sie im Verzeichnis "jumpstart" die Datei "jumpstart3.xsl" mit dem obigen Code.Transformieren Sie die Datei "jumpstart.xml" mit der Datei "jumpstart3.xsl" und speichern Sie das Ergebnis in der Datei "jumpstart3.txt" ab:java -jar ..\saxon\saxon-he-11.4.jar -s:jumpstart.xml -xsl:jumpstart3.xsl -o:jumpstart3.txtAnalysieren Sie das Script und das Resultat.

XSLT select if-then-else

Mit XSLT 2.0 sind if-then-else Ausdrücke innerhalb select Statements programmierbar:<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:output method="text"/>
 <xsl:template match="filesystem">
 <xsl:apply-templates

select="body/folder"/>
  <xsl:apply-templates select="body/file"/>
  </xsl:template>
 <xsl:template match="folder">
  <xsl:value-of select="if (@name = '.') then '
this folder' else (if (@name = '..') then '
parent folder' else concat('
folder ', @name))"/>
  </xsl:template>
  <xsl:template match="file">
  file <xsl:value-of select="@name"/>
 </xsl:template>
</xsl:stylesheet>Erstellen Sie im Verzeichnis "jumpstart" die Datei "jumpstart4.xsl" mit dem obigen Code.Transformieren Sie die Datei "jumpstart.xml" mit der Datei "jumpstart4.xsl" und speichern Sie das Ergebnis in der Datei "jumpstart4.txt" ab:java -jar ..\saxon\saxon-he-11.4.jar -s:jumpstart.xml -xsl:jumpstart4.xsl -o:jumpstart4.txtAnalysieren Sie das Script und das Resultat.

XSLT sort

Das folgende XML Dokument enthält 2 weitere Dateien test.xsl und test.xml unterschiedlicher Grösse (size):<?xml version="1.0" encoding="iso-8859-1"?>
<filesystem>
 <head>
  <meta name="path">C:\xslkurs\jumpstart\</meta>
  <meta name="search">c:\xslkurs*.*</meta>
  <meta name="drive">c:</meta>
  <meta name="subdir">false</meta>
  </head>
 <body>
  <folder name="." path="c:/xslkurs/." createTime="2022-08-15T21:51:45+01:00" accessTime="2022-08-20T15:13:06+01:00" writeTime="2022-08-20T15:13:06+01:00"></folder>
 <folder name=".." path="c:/xslkurs/.." createTime="2022-08-15T21:51:45+01:00" accessTime="2022-08-20T15:13:06+01:00" writeTime="2022-08-20T15:13:06+01:00"></folder>
 <folder name=".metadata" path="c:/xslkurs/.metadata" createTime="2022-08-15T21:51:45+01:00" accessTime="2022-08-20T15:13:06+01:00" writeTime="2022-08-20T15:13:06+01:00"></folder>
 <folder name="hello" path="c:/xslkurs/hello" createTime="2022-08-15T21:51:45+01:00" accessTime="2022-08-20T15:13:06+01:00" writeTime="2022-08-20T15:13:06+01:00"></folder>
 <folder name="jumpstart" path="c:/xslkurs/jumpstart" createTime="2022-08-15T21:51:45+01:00" accessTime="2022-08-20T15:13:06+01:00" writeTime="2022-08-20T15:13:06+01:00"></folder>
 <folder name="note" path="c:/xslkurs/note" createTime="2022-08-15T21:51:45+01:00" accessTime="2022-08-20T15:13:06+01:00" writeTime="2022-08-20T15:13:06+01:00"></folder>
 <file name="readme.txt" path="c:/xslkurs/readme.txt" mimetype="text/plain" size="1024" createTime="2022-08-15T21:51:45+01:00" accessTime="2022-08-20T15:13:06+01:00" writeTime="2022-08-20T15:13:06+01:00"/>
  <file name="test.xsl" path="c:/xslkurs/readme.txt" mimetype="text/plain" size="452" createTime="2022-08-15T21:51:45+01:00" accessTime="2022-08-20T15:13:06+01:00" writeTime="2022-08-20T15:13:06+01:00"/>
  <file name="test.xml" path="c:/xslkurs/readme.txt" mimetype="text/plain" size="12455" createTime="2022-08-15T21:51:45+01:00" accessTime="2022-08-20T15:13:06+01:00" writeTime="2022-08-20T15:13:06+01:00"/>
  <folder

name="saxon" path="c:/xslkurs/saxon"
createTime="2022-08-15T21:51:45+01:00"
accessTime="2022-08-20T15:13:06+01:00"
writeTime="2022-08-20T15:13:06+01:00"></folder>

</body>
</filesystem>Erstetzen Sie die Datei
"jumpstart.xml" mit dem obigen Content.Das nächste XSLT Script sortiert die Dateien
nach der Grösse absteigend:<?xml version="1.0"
encoding="UTF-8"?>
<xsl:stylesheet version="2.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:output
method="text"/>
 <xsl:template
match="filesystem">
 <xsl:apply-templates
select="body/folder"/>
 <xsl:apply-templates
select="body/file">
 <xsl:sort select="@size"
data-type="number" order="descending"/>

</xsl:apply-templates>
 </xsl:template>

<xsl:template match="folder">
 <xsl:choose>

<xsl:when test="@name = '.'">
 this folder

</xsl:when>
 <xsl:when test="@name =
'..'">
 parent folder
 </xsl:when>

<xsl:otherwise>
 folder <xsl:value-of
select="@name"/>
 </xsl:otherwise>

</xsl:choose>
 </xsl:template>
 <xsl:template
match="file">
 file <xsl:value-of select="@name"/>
[<xsl:value-of select="@size"/>]

</xsl:template>
</xsl:stylesheet>Erstellen Sie im Verzeichnis
"jumpstart" die Datei "jumpstart5.xsl" mit dem obigen Code.Transformieren Sie
die Datei "jumpstart.xml" mit der Datei "jumpstart5.xsl" und speichern Sie das
Ergebnis in der Datei "jumpstart5.txt" ab:java -jar ..\saxon\saxon-he-11.4.jar
-s:jumpstart.xml -xsl:jumpstart5.xsl -o:jumpstart5.txtAnalysieren Sie das Script und das Resultat.

XSLT Variablen

XSLT Variablen bieten die Möglichkeit, Werte wiederzuverwenden. Das folgende Script zeigt dies
auf, als Ausgabe Format HTML:<?xml version="1.0"
encoding="UTF-8"?>
<xsl:stylesheet version="2.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:template
match="filesystem">
 <xsl:variable
name="title">filesystem [<xsl:value-of select="head/meta[@name =
'path']"/>]</xsl:variable>
 <html>

<head>
 <title><xsl:value-of
select="$title"/></title>
 </head>

<body>
 <table border="1">

<tr>
 <td colspan="2"><xsl:value-of
select="$title"/></td>
 </tr>

<xsl:for-each select="body/*">
 <tr>

<td><xsl:value-of
select="local-name()"/></td>

<td><xsl:value-of
select="@name"/></td>
 </tr>

</xsl:for-each>
 </table>

</body>
 </html>

</xsl:template>
</xsl:stylesheet>Erstellen Sie im Verzeichnis
"jumpstart" die Datei "jumpstart6.xsl" mit dem obigen Code.Transformieren Sie
die Datei "jumpstart.xml" mit der Datei "jumpstart6.xsl" und speichern Sie das
Ergebnis in der Datei "jumpstart6.html" ab:java -jar ..\saxon\saxon-he-11.4.jar
-s:jumpstart.xml -xsl:jumpstart6.xsl -o:jumpstart6.htmlAnalysieren Sie das Script und das Resultat.

Filesystem Rekursiv

Eine rekusrive Version des Filesystem finden Sie unter folgendem XML Dokument
xfsdump.rekursiv.xml.

Kontakt

Simtech AG
Finkenweg 23
3110 Münsingen
Schweiz

Impressum

Das Copyright für sämtliche Inhalte dieser Website liegt bei Simtech AG, Schweiz.
Beachten Sie auch unsere Hinweise zum Urheberrecht, Datenschutz und Haftungsauschluss.
Jeder Hinweis auf Fehler nehmen wir gerne entgegen.

Copyright

2024 Simtech AG, All rights reserved, Powered by stack.ch written in Golang by Daniel Schmutz

https://www.simtech-ag.ch/Education Kurse Jumpstart XML Transformation