XML Data Parsing

Summary

It provides a function to perform XML data parsing after loading the XML file and to convert them into structured data types. XML data parsing can be used in applications such as Web Service, Electronic Document and Data Swapping.

This function is included in the Egovframework common component element technology.

Description

① Function to perform XML data parsing and convert them into structured data types

Related Sources

Type	*Corresponding Sources*	*Description*	*Remarks*
Service	egovframework.com.utl.sim.service.EgovXMLDoc.java	An element technology class on XML parsing/assembly	
JSP	WEB_INF/jsp/egovframework/cmm/utl/EgovXMLDoc.jsp	A test page	

Method

Output	*Method*	*Description*	*Details*
boolean	creatSchemaToClass(String xml, String ja)	Creating a class	Creating an appropriate JAVA class by reading the XML schema structure
SndngMailDocument	getXMLToClass(String file)	XML parsing	Returning the content in the e-mail sending class (temporary) by parsing the XML file
boolean	getClassToXML(SndngMailDocument mailDoc, String file)	XML assembly	Saving the data of e- mail sending(temporary) object as the XML file
Document	getXMLDocument(String xml)	XML parsing	Returning the document object that can manipulate data by parsing the XML file
Element	getRootElement(Document document)	Moving ROOT	Moving to the highest element of the document
Element	insertElement(Document document, Element rt, String id)	Creating a node	Creating a lower new element
Element	insertElement(Document document, Element rt, String id, String text)	Creating a string	Creating a lower new element that has the string
Text	insertText(Document document, Element rt, String text)	Adding a string	Adding a lower string
Element	getParentNode(Element current)	Returning an upper node	Returning the last inserted or referenced XML node upper element
boolean	getXMLFile(Document document, String file)	XML assembly	Saving the document object as the XML file

Input

- XML Schema File: File name that includes a string- type absolute path (ex. /user/com/test/mail.xsd)
- Created JAR Path: JAR file name that includes a string- type absolute path (ex. /user/com/test/mail.jar)
- XML Data File: XML data file name that includes a string- type absolute path (ex. /user/com/test/mail_data.xml)
- Validation Check: Element technology validation check

Output

- Boolean- type true / false

Environmental Settings

- Downloading the xmlbeans library for schema compiling

http://xmlbeans.apache.org/index.html (intro) http://ftp.apache- kr.org/xmlbeans/binaries/ (download)

- Setting system variables and path for schema compiling

(From the Windows command execution window)
set XMLBEANS_HOME=C:\ xmlbeans- 2.0.0
set XMLBEANS_LIB=%XMLBEANS_HOME%\ lib
set JAVA_HOME=C:\ Progra~ 1\ Java\ jdk1.5.0
set PATH=%XMLBEANS_HOME%\ bin;%JAVA_HOME%\ bin;

- XML schema file type to create the JAVA class

```
<?xml version="1.0" encoding="UTF- 8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
        schemaLocation="http://www.w3.org/2001/xml.xsd" />
  <xs:element name="sndngMail">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="mssageId"                type="xs:string"/>
        <xs:element name="dsptchPerson"            type="xs:string"/>
        <xs:element name="recptnPerson"            type="xs:string"/>
        <xs:element name="sj"                      type="xs:string"/>
        <xs:element name="emailCn"           type="xs:string"/>
        <xs:element name="sndngResultCode"type="xs:string"/>
        <xs:element name="orignlFileList"    type="xs:string"/>
        <xs:element name="streFileList"            type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

- Executing the schema file compile command

scomp - out target.jar source.xsd

source.xsd: XML that has the schema file type as mentioned above
target.jar: JAR created via schema compiling
- out(src): If created with JAR, it is the out option; and with source, it is the src option

- XML file type that contains data

```
<sndngMail>
    <dsptchPerson>test1@mopas.go.kr</dsptchPerson>
    <recptnPerson>test2@mopas.go.kr</recptnPerson>
    <sj> This is a test e- mail.</sj>
    <emailCn> This is an e- mail content.</emailCn>
    <sndngResultCode>R</sndngResultCode>
</sndngMail>
```

Manual

```
import egovframework.com.utl.sim.service.EgovXMLDoc;

// creating JAVA class by reading the schema file
// result is the jar file created with a class that has the same schema structure
String file = request.getParameter("file");
String jar = request.getParameter("jar");
boolean result = false;
if (file != null && file.length() > 0
        && jar != null && jar.length() > 0) {
        result = EgovXMLDoc.creatSchemaToClass(file, jar);
}

// e- mail object parsing by reading the XML file data containing e- mail information
String file = request.getParameter("file");
String resultStr = "";
if (file != null && file.length() > 0) {
        SndngMailDocument mailDoc = EgovXMLDoc.getXMLToClass(file);
        SndngMailDocument.SndngMail mailElement = mailDoc.getSndngMail();
        resultStr += "1. sender: " + mailElement.getDsptchPerson() + "<br>";
        resultStr += "2. receiver: " + mailElement.getRecptnPerson() + "<br>";
        resultStr += "3. title: " + mailElement.getSj() + "<br>";
        resultStr += "4. content: " + mailElement.getEmailCn() + "<br>";
        resultStr += "5. Send status: " + mailElement.getSndngResultCode() + "<br>";
}
```

After adding the JAR file created using the XML schema file into the system, it is implemented to enable data parsing/assembly using this class. For element technology testing, XML schema is used to create a class that temporary contains e- mail information. This is applied to configure assembly/parsing of the XML file into the class, and the class data into the XML file.

References

- Reference to XML Beans: marshalling_unmarshalling