

개요

XML 파일을 로드 후 XML 데이터를 파싱 하여 구조적인 데이터 형태로 변환 하는 기능을 제공한다. XML 데이터 파싱은 웹서비스(Web Service), 전자문서, 데이터교환과 같은 응용 어플리케이션에서 사용할 수 있다.

본 기능은 전자정부 표준프레임워크 공통컴포넌트 요소기술 내에 구성되어 있다.

- XML데이터파싱
 - 개요
 - 설명
 - 환경설정
 - 사용방법
 - 참고자료

설명

- ① XML 데이터를 파싱하여 구조적인 데이터 형태로 변환 하는 기능

관련소스

유형	대상소스명	설명	비고
Service	egovframework.com.utl.sim.service.EgovXMLDoc.java	XML파싱/조립 요소기술 클래스	
JSP	WEB_INF/jsp/egovframework/cmm/utl/EgovXMLDoc.jsp	테스트 페이지	

메소드

결과값	메소드명	설명	내용
boolean	creatSchemaToClass(String xml, String ja)	클래스 생성	XML스키마 구조를 읽어 이에 맞는 자바클래스 생성
SndngMailDocument	getXMLToClass(String file)	XML파싱	XML파일을 파싱하여 메일발송 클래스(임의)에 내용을 담아 반환
boolean	getClassToXML(SndngMailDocument mailDoc, String file)	XML조립	메일발송(임의) 객체의 데이터를 XML파일로 저장
Document	getXMLDocument(String xml)	XML파싱	XML 파일을 파싱하여 데이터를 조작할 수 있는 Document 객체를 반환
Element	getRootElement(Document document)	ROOT 이동	Document의 최상의 Element로 이동
Element	insertElement(Document document, Element rt, String id)	노드생성	하위에 새로운 Element를 생성
Element	insertElement(Document document, Element rt, String id, String text)	문자열 생성	하위에 문자열을 가지는 새로운 Element를 생성
Text	insertText(Document document, Element rt, String text)	문자열 추가	하위에 문자열을 추가
Element	getParentNode(Element current)	상위노드반환	마지막으로 입력되었거나 참조된 XML Node의 상위 Element를 리턴
boolean	getXMLFile(Document document, String file)	XML조립	Document 객체를 XML파일로 저장

Input

- XML스키마 파일: String 타입의 절대경로를 포함한 XML스키마 파일명 (예, /user/com/test/mail.xsd)
- 생성 JAR 경로: String 타입의 절대경로를 포함한 JAR 파일명 (예, /user/com/test/mail.jar)
- XML데이터 파일: String 타입의 절대경로를 포함한 XML데이터 파일명 (예, /user/com/test/mail_data.xml)
- Validation 체크: [요소기술_validation_체크](#)

Output

- Boolean 타입 true / false

환경설정

- 스키마 컴파일을 위한 xmlbeans 라이브러리 다운로드

 <http://xmlbeans.apache.org/index.html> (소개)  <http://ftp.apache.kr.org/xmlbeans/binaries/> (다운로드)

- 스키마 컴파일을 위한 시스템변수 및 PATH 설정

```
( 윈도우 명령어 실행창에서 )
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;
```

- 자바클래스 생성을 위한 XML 스키마 파일의 형태

```
<?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="originlFileList" type="xs:string" />
        <xs:element name="streFileList" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

- 스키마 파일 컴파일 명령어 실행

```
scomp -out target.jar source.xsd
```

source.xsd : 위의 스키마 파일 형태를 갖춘 XML

target.jar : 스키마 컴파일을 통해 생성되는 JAR

-out(src) : JAR 로 생성하는 경우 out 옵션, 소스로 생성하는 경우 src 옵션

- 데이터를 담은 XML 파일의 형태

```
<sndngMail>
  <dsptchPerson>test1@mopas.go.kr</dsptchPerson>
  <recptnPerson>test2@mopas.go.kr</recptnPerson>
  <sj>테스트 메일 입니다.</sj>
  <emailCn>내용입니다.</emailCn>
  <sndngResultCode>R</sndngResultCode>
</sndngMail>
```

사용방법

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

// 스키마 파일을 읽어 자바 클래스 생성
// 결과물은 스키마 구조와 일치하는 클래스가 생성된 jar 파일
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);
}

// 메일정보를 담은 XML 파일의 데이터를 읽어 메일 객체로 파싱한다.
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.발신자: " + mailElement.getDspthPerson() + "<br>";
    resultStr += "2.수신자: " + mailElement.getRecptnPerson() + "<br>";
    resultStr += "3.제목: " + mailElement.getSj() + "<br>";
    resultStr += "4.내용: " + mailElement.getEmailCn() + "<br>";
    resultStr += "5.발신상태: " + mailElement.getSndngResultCode() + "<br>";
}
```

XML 스키마 파일을 통해 생성된 JAR 파일을 사용하고자 하는 시스템에 추가한 뒤, 이 클래스를 이용하여 XML 데이터를 파싱하고 조립할 수 있도록 구현하였다. 요소기술 테스트를 위해서 현재 임의로 메일정보를 담은 클래스를 XML스키마를 통하여 생성하였으며 이를 가지고 XML파일을 클래스로, 클래스의 데이터를 XML파일로 조립/파싱함을 보여주도록 구성하였다.

참고자료

- XML Beans 참조: [marshalling_unmarshalling](#)