

개요

일반 구조적 형태의 데이터를 XML 데이터로 조립하여 파일로 저장하는 기능을 제공한다. XML 데이터 조립은 웹서비스(Web Service), 전자문서, 데이터교환과 같은 응용어플리케이션에서 사용할 수 있다.

본 기능은 전자정부 표준프레임워크 공통컴포넌트 요소기술 내에 구성되어 있다.

- XML데이터조립
 - 개요
 - 설명
 - 환경설정
 - 사용방법
 - 참고자료

설명

① 일반 구조적 형태의 데이터를 XML 데이터로 조립하여 파일로 저장하는 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

- 메시지ID: String 타입의 날짜 (예, MAILMSG_000000000001)
- 발신자: String 타입의 날짜 (예, sndng_user)
- 수신자: Integer 타입의 사이즈 (예, test@test.com)
- 제목: Integer 타입의 사이즈 (예, test mail)
- 내용: String 타입의 파일 생성자 (예, This is test mail.)
- 발신상태: String 타입의 파일 생성자 (예, R)
- XML결과 파일: String 타입의 파일 생성자 (예, /user/com/test/mail.result.xml)
- Validation 체크: [요소기술_validation_체크](#)

Output

- Boolean 타입 true / false

환경설정

N/A

사용방법

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

// 메일 클래스(임의)에 정보를 담고 이를 xml파일로 조립한다.
String file = request.getParameter("file");
String dsptchPerson = request.getParameter("dsptchPerson");
String recptnPerson = request.getParameter("recptnPerson");
String sj = request.getParameter("sj");
String emailCn = request.getParameter("emailCn");
String sndngResultCode = request.getParameter("sndngResultCode");

boolean result = false;
if (file != null && file.length() > 0
    && dsptchPerson != null && dsptchPerson.length() > 0
    && recptnPerson != null && recptnPerson.length() > 0
    && sj != null && sj.length() > 0
    && emailCn != null && emailCn.length() > 0
    && sndngResultCode != null && sndngResultCode.length() > 0
) {

    SndngMailDocument mailDoc;
    SndngMailDocument.SndngMail mailElement;
    mailDoc = SndngMailDocument.Factory.newInstance();
    mailElement = mailDoc.addNewSndngMail();
    mailElement.setDsptchPerson(dsptchPerson);
    mailElement.setRecptnPerson(recptnPerson);
    mailElement.setSj(sj);
    mailElement.setEmailCn(emailCn);
    mailElement.setSndngResultCode(sndngResultCode);

    result = EgovXMLDoc.getClassToXML(mailDoc, file);
}
```

요소기술 테스트를 위해서 현재 임의로 메일정보를 담는 클래스를 XML스키마를 통하여 생성하였으며 이를 가지고 XML파일을 클래스로, 클래스의 데이터를 XML파일로 조립/파싱함을 보여주도록 구성하였다.

참고자료

- XML Beans 참조: [marshalling_unmarshalling](#)