

개요

클라이언트(Client)에서 서버(Server)의 데이터를 받아 메뉴 형태로 보여주는 기능을 제공한다.

메뉴의 추가, 수정, 삭제 기능이 있어야 한다. 이 서비스 컴포넌트는 트리메뉴를 화면에 보여줄 때 사용할 수 있다.
본 기능은 전자정부 표준프레임워크 공통컴포넌트 요소기술 내에 구성되어 있다.

설명

- ① 메뉴의 추가, 수정, 삭제 기능 가능 여부 확인 (True/False)
- ① 저장된 정보를 활용하여 트리메뉴 생성

관련소스

유형	대상소스명	설명	비고
Service	egovframework.com.utl.service.EgovMenuGov.java	메인메뉴 요소기술 클래스	메뉴파일 생성
JSP	WEB_INF/jsp/egovframework/cmm/utl/EgovMenuGov.jsp	테스트 페이지	

메소드

결과값	메소드명	설명	내용
boolean	parsFileByMenuChar(String parFile, String parChar, int parField)	메뉴테이블형태 화면	데이터를 받아 구분값, 필드수에 맞추어 메뉴필드형태로 나눈다
boolean	setDataByDATFile(String parFile, String[] menuIDArray, String[] menuNameArray, String[] menuLevelArray, String[] menuURLArray)	메뉴관리 화면의 데이터를 DAT 파일로 생성	데이터를 받아 구분값, 필드수에 맞추어 서버데이터 파일 생성(수정,삭제)

Input

- 메뉴변환파일 : String 타입의 절대경로를 포함한 디렉토리 (예, /user/com/test/file1.dat)
- 메뉴ID : Number 타입의 숫자(예, 1,10, 100,1000 ...)
- 메뉴명 : String 타입의 메뉴명 (예, 메뉴관리, 메인메뉴, ...)
- 상위메뉴ID : Number 타입의 숫자(예, 1,10, 100,1000 ...)
- URL : String 타입의 URL경로. (예, 'http://kr.yahoo.com/', ...)
- Validation 체크: [요소기술_validation_체크](#)

Output

- Boolean 타입 true / false

환경설정

해당계정으로 서버(Server)의 데이터(File 형태)를 읽고, 쓸수 있는 권한을 가진 디렉토리를 지정되어 있어야함.

사용방법

- 서버에서 DAT 파일형태의 데이터를 읽어와 해당 필드구분과 필드수 만큼 화면에 표현함.

```
import egovframework.com.utl.sim.service.EgovMenuGov;
Vector result1 = EgovMenuGov.parsFileByMenuChar(parFile, parChar, parField);

String str = "";
if(result1.size() > 0){
for (int j = 0; j < result1.size()-1; j++) {
    ArrayList arr = (ArrayList)result1.elementAt(j);
    <tr>
    <td width=80 ><a onclick='fRowAdd(this)' style='cursor:hand;' title='라인추가'>+</a> </td>
    <td width=80 ><input type='text' name='hiddenMenuIDArray' value='<%= (String)arr.get(0) %>' size=10
    maxlength='10'></td>
    <td width=164><input type='text' name='hiddenMenuNameArray' value='<%= (String)arr.get(2) %>' size=20 ></td>
    <td width=80 ><input type='text' name='hiddenMenuLevelArray' value='<%= (String)arr.get(1) %>' size=5
    maxlength='5'></td>
    <td width=360><input type='text' name='hiddenMenuURLArray' value='<%= (String)arr.get(3) %>' size=50 ></td>
    <td width=80 ><a onclick='fRowDelete(this)' style='cursor:hand;' title='라인삭제'>-</a></td>
    </tr>
    }
}
```

```
FileName = parFile.replace('\\', FILE_SEPARATOR).replace('/', FILE_SEPARATOR);
File file = new File(FileName);

// 파일이며, 존재하면 파싱 시작
if (file.exists() && file.isFile()) {
    list = EgovFileTool.parsFileByChar(parFile, parChar, parField);
}else{
    list = new Vector();
}
}
```

- 메뉴관리 화면의 데이터를 배열 형태로 읽어 DAT 파일로 생성

```
import egovframework.com.utl.sim.service.EgovMenuGov;
String parFile = safeGetParameter(request, "file");
String[] menuIDArray = safeGetParameterValues(request, "hiddenMenuIDArray");
String[] menuNameArray = safeGetParameterValues(request, "hiddenMenuNameArray");
String[] menuLevelArray = safeGetParameterValues(request, "hiddenMenuLevelArray");
String[] menuURLArray = safeGetParameterValues(request, "hiddenMenuURLArray");
boolean result2 = EgovMenuGov.setDataByDATFile(parFile, menuIDArray, menuNameArray, menuLevelArray, menuURLArray);
```

```
FileName = parFile.replace('\\', FILE_SEPARATOR).replace('/', FILE_SEPARATOR);
File file = new File(FileName);
BufferedWriter out = new BufferedWriter(new FileWriter(file));

for(int i = 0; i < menuIDArray.length ; i++){
    //nodeId | parentNodeId | nodeName | nodeUrl
    out.write(menuIDArray[i]+"|" +menuLevelArray[i]+"|" +menuNameArray[i]+"|" +menuURLArray[i]+"|");
    out.newLine();
}
success = true;
out.close();
```

참고자료

N/A