Creating a Main Menu

Summary

It provides a function to display the server data received from the Client in a menu form.

It shall have functions to add, update and delete the menu. This service component is available when a tree menu is to be displayed on a screen.

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

Description

① Check whether the functions to add, update and delete menus are available (True/False)
① Generate a tree menu using the saved information

Related Sources

Type	***Corresponding Sources***	***Description***	***Remarks***
Service	egovframework.com.utl.service.EgovMenuGov.java	The element technology class of a main menu	Creating a menu file
JSP	WEB_INF/jsp/egovframework/cmm/utl/EgovMenuGov.jsp	A test page	

Method

Output	***Method***	***Description***	***Details***
Boolean	parsFileByMenuChar(String parFile, String parChar, int parField)	A screen that shows the menu table form	With the data received, it is to divide them into a menu field form in accordance with the separator and the number of fields
Boolean	setDataByDATFile(String parFile, String[] menuIDArray, String[] menuNameArray, String[] menuLevelArray, String[] menuURLArray)	Data of the main menu screen is converted into a DAT file	With the data received, it is to generate a server data file in accordance with the separator and the number of fields (update & delete)

Input

- Menu Conversion File: A directory that includes a String-type absolute path(ex. /user/com/test/file1.dat)
- Menu ID: Figures represented in decimal numbers (ex. 1,10, 100,1000 ...)
- Menu Name: A String-type menu name (ex, Menu Management, Main Menu, ...)
- Upper Menu ID: Figures represented in decimal numbers (ex. 1,10, 100,1000 ...)
- URL: A String-type URL path. (ex. 'http://kr.yahoo.com/', ...)
- Validation Check: Element Technology _validation_check

Output

- Boolean-type true / false

Environmental Settings

A directory authorized to read and write the server data (file format) with the relevant account must be designated.

Manual

- It imports the DAT file format data from the server and displays them on a screen in accordance with the field type and the number of fields.

```
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='add line'>+</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='delete line'>- </a></td>
<td width=16 ></td>
</tr>
}

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

// It is a file; if it exists, parsing starts.
if (file.exists() && file.isFile()) {
    list = EgovFileTool.parsFileByChar(parFile, parChar, parField);
} else{
    list = new Vector();
}
```

- The data of a menu management screen is read in array, generating DAT files.

```
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();
```

References

N/A