Egov 페이징 커스텀

- description : Egov 페이징 커스텀
- author : 오션
- email : shlim@repia.com
- lastupdate : 2022-09-07 Wed

Egov 페이징 커스텀

1. Java Resources > Libraries > Maven Dependencies > egovframework.rte.ptl.mvc-3.9.0.jar > egovframework.rte.ptl.mvc > tags.ui > pagination> AbstractPaginationRenderer.class 반드시 참조

AbstractPaginationRenderer.class

```
package egovframework.rte.ptl.mvc.tags.ui.pagination;

import java.text.MessageFormat;

/**
 * AbstractPaginationRenderer.java
 * <p/><b>NOTE:</b><pre> 인터페이스 PaginationRenderer의 구현 추상클래스.
 * 기본적인 페이징 기능이 구현되어 있으며, 화면에서 아래와 같이 display 된다.
 *
 * [처음][이전] 1 2 3 4 5 6 7 8 [다음][마지막]
 *
 * 클래스 변수들이 각 element와 매핑이 되는데,
 * firstPageLabel = [처음]
 * previousPageLabel = [이전]
 * currentPageLabel = 현재 페이지 번호
 * otherPageLabel = 현재 페이지를 제외한 페이지 번호
 * nextPageLabel = [다음]
 * lastPageLabel = [마지막]
 *
 * 클래스 변수값을 AbstractPaginationRenderer를 상속받은 하위 클래스에서 주게 되면,
 * 페이징 포맷만 프로젝트 UI에 맞춰 커스터마이징 할 수 있다.
 * 자세한 사항은 개발자 메뉴얼의 개발프레임워크 실행환경/화면처리/MVC/View/Pagination Tag를 참고하라.
 * </pre>
 */
public abstract class AbstractPaginationRenderer implements
PaginationRenderer {
```

```
    protected String firstPageLabel;
    protected String previousPageLabel;
    protected String currentPageLabel;
    protected String otherPageLabel;
    protected String nextPageLabel;
    protected String lastPageLabel;

    public String renderPagination(PaginationInfo paginationInfo, String
jsFunction) {

        StringBuffer strBuff = new StringBuffer();

        int firstPageNo = paginationInfo.getFirstPageNo();
        int firstPageNoOnPageList =
paginationInfo.getFirstPageNoOnPageList();
        int totalPageCount = paginationInfo.getTotalPageCount();
        int pageSize = paginationInfo.getPageSize();
        int lastPageNoOnPageList = paginationInfo.getLastPageNoOnPageList();
        int currentPageNo = paginationInfo.getCurrentPageNo();
        int lastPageNo = paginationInfo.getLastPageNo();

        if (totalPageCount > pageSize) {
            if (firstPageNoOnPageList > pageSize) {
                strBuff.append(MessageFormat.format(firstPageLabel, new
Object[] { jsFunction, Integer.toString(firstPageNo) }));
                strBuff.append(MessageFormat.format(previousPageLabel, new
Object[] { jsFunction, Integer.toString(firstPageNoOnPageList - 1) }));
            } else {
                strBuff.append(MessageFormat.format(firstPageLabel, new
Object[] { jsFunction, Integer.toString(firstPageNo) }));
                strBuff.append(MessageFormat.format(previousPageLabel, new
Object[] { jsFunction, Integer.toString(firstPageNo) }));
            }
        }

        for (int i = firstPageNoOnPageList; i <= lastPageNoOnPageList; i++)
{
            if (i == currentPageNo) {
                strBuff.append(MessageFormat.format(currentPageLabel, new
Object[] { Integer.toString(i) }));
            } else {
                strBuff.append(MessageFormat.format(otherPageLabel, new
Object[] { jsFunction, Integer.toString(i), Integer.toString(i) }));
            }
        }

        if (totalPageCount > pageSize) {
            if (lastPageNoOnPageList < totalPageCount) {
                strBuff.append(MessageFormat.format(nextPageLabel, new
Object[] { jsFunction, Integer.toString(firstPageNoOnPageList + pageSize)
```

```
}));
                strBuff.append(MessageFormat.format(lastPageLabel, new
Object[] { jsFunction, Integer.toString(lastPageNo) }));
            } else {
                strBuff.append(MessageFormat.format(nextPageLabel, new
Object[] { jsFunction, Integer.toString(lastPageNo) }));
                strBuff.append(MessageFormat.format(lastPageLabel, new
Object[] { jsFunction, Integer.toString(lastPageNo) }));
            }
        }
        return strBuff.toString();
    }
}
```

PaginationInfo.class

totalPageCount : 전체 페이지 개수
totalRecordCount : 필수 필드 값, 전체 게시물 개수
recordCountPerPage : 필수 필드 값, 한 페이지 당 표시되는 게시물 개수

```
// 전체 페이지 개수 = ((전체 게시물 개수 -1) / 한 페이지 당 표시되는 게시물 개수) + 1
// totalPageCount = ((totalRecordCount - 1) / recordCountPerPage) + 1;
public int getTotalPageCount() {
    totalPageCount = ((getTotalRecordCount() - 1) / getRecordCountPerPage())
+ 1;
    return totalPageCount;
}
```

firstPageNoOnPageList : 페이징 번호 표시 Bar에 표시되는 첫 번째 페이지 번호
currentPageNo : 필수 필드 값, 페이징 번호 표시 Bar에서 클릭하여 현재 보여지는 페이지 번호
pageSize : 필수 필드 값, 페이징 번호 표시 Bar에 표시되는 페이지 번호들의 개수

```
// 페이징 Bar의 첫 번째 페이지 번호 = ((현재 페이지 번호 - 1) / 페이징 Bar의 번호들의 개수) *
페이징 Bar의 번호들의 개수 + 1;
// firstPageNoOnPageList = ((currentPageNo - 1) / pageSize ) * pageSize + 1;
public int getFirstPageNoOnPageList() {
    firstPageNoOnPageList = ((getCurrentPageNo() - 1) / getPageSize()) *
getPageSize() + 1;
    return firstPageNoOnPageList;
}
```

lastPageNoOnPageList : 페이징 번호 표시 Bar에 표시되는 마지막 페이지 번호

```
public int getLastPageNoOnPageList() {
    lastPageNoOnPageList = getFirstPageNoOnPageList() + getPageSize() - 1;
    if (lastPageNoOnPageList > getTotalPageCount()) {
```

```
            lastPageNoOnPageList = getTotalPageCount();
        }
        return lastPageNoOnPageList;
    }
```

Ref LInk

egovframework:rte:ptl:view:paginationtag-페이징처리

오션, Egov, 페이징, 커스텀

From:
http://125.132.25.164/dokuwiki/ - **문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12**

Permanent link:
http://125.132.25.164/dokuwiki/doku.php?id=wiki:miscellaneous:egov_%ED%8E%98%EC%9D%B4%EC%A7%95_%EC%BB%A4%EC%8A%A4%ED%85%80&rev=1662548269

Last update: **2022/09/07 19:57**