


```
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 :

totalRecordCount :

recordCountPerPage :

```

//          = ((          -1) /          ) + 1
// totalPageCount = ((totalRecordCount - 1) / recordCountPerPage) + 1;
public int getTotalPageCount() {
    totalPageCount = ((getTotalRecordCount() - 1) / getRecordCountPerPage())
+ 1;
    return totalPageCount;
}

```

firstPageNoOnPageList :

Bar

currentPageNo :

Bar

pageSize :

Bar

```

//      Bar          = ((          - 1) /          Bar          ) *
//      Bar          + 1;
// firstPageNoOnPageList = ((currentPageNo - 1) / pageSize) * pageSize + 1;
public int getFirstPageNoOnPageList() {
    firstPageNoOnPageList = ((getCurrentPageNo() - 1) / getPageSize()) *
getPageSize() + 1;
    return firstPageNoOnPageList;
}

```

lastPageNoOnPageList :

Bar

```

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

```

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

Ref Link

[egovframework:rte:ptl:view:paginationtag-](#)

, [Egov](#), ,

From: <http://125.132.25.164/dokuwiki/> -

- 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

