HeaderData - Sort, Paging, Filter.

# Introduction

The HeaderData class is part of Simplisity Server code. It keeps track of variables required for sorting, paging and filtering.

It is designed to build SQL filters and sort [order by] partial statements, that can be used to select and sort different object.

# How it works

# Manual Sorting Example

If a list of records required a manual sort, this can be done by using the HeaderData, although it can be done by other methods. The HeaderData class holds a manual sort variable to track and move a record based on the [SortOrder] field.

## Razor Template code required

### Listing the manual sort button

We place this code on the end of each row in the list. The idea is that once the “rocketmodedit\_activatesortorder” is clicked the server responses by storing the select itemId into the HeaderData field, which is then rendered on the razor template as a param field, ready to eb send back to the server.

@if (articleDataList.Header.SortActivate > 0)

{

if (articleDataList.Header.SortActivate != info.ItemID)

{

<div class="w3-col w3-button simplisity\_click w3-right " s-cmd="rocketmodedit\_setsortorder" s-fields='{"itemid":"@info.ItemID"}' style="width: 64px;">

<i class="far fa-check-circle w3-text-green"></i>

</div>

}

else

{

<div class="w3-col w3-button simplisity\_click w3-right " s-cmd="rocketmodedit\_cancelsortorder" s-fields='{"itemid":"@info.ItemID"}' style="width: 64px;">

<i class="fas fa-ban w3-text-red"></i>

</div>

}

}

else

{

<div class="w3-col w3-button simplisity\_click w3-right " s-cmd="rocketmodedit\_activatesortorder" s-fields='{"itemid":"@info.ItemID"}' style="width: 64px;">

<i class="fas fa-sort"></i>

</div>

}

### Keeping track of selected record

We need to place the select (or not selected) itemId in a param field in the razor template, the easiest way to do this is by using “simplisity\_setParamField”

<script>

$(document).ready(function () {

// return ALL headerData in the headerdata param field.

// The “InfoToCoded()” method is used so we have an encoded value to protect special chars.

simplisity\_setParamField('headerdata','@articleDataList.Header.InfoToCoded()');

});

</script>

## Server-Side Code using HeaderData class

We’ll give examples matching the example client-side code.

First, we get the existing values from the webpage.

// setup headerdata

\_headerData = new HeaderData(\_paramInfo);

Creating a new HeaderData and passing the paramInfo will take the “headerdata” XML from the page, but also any updated fields. In this example we do not update the headerdata from the webpage, we will show an example of passing updated values later in this document.

We have 3 functions to: Activate “rocketmodedit\_activatesortorder” the manual sort, Cancel “rocketmodedit\_cancelsortorder” the manual sort and Move “rocketmodedit\_setsortorder” to do the actual switch if the [SortOrder] value.

public String ActivateSortOrder()

{

var articleDataList = new ArticleDataList(\_postInfo, \_paramInfo, \_editLang, false);

articleDataList.Header = \_headerData;

articleDataList.Header.ActivateItemSort(\_paramInfo.GetXmlPropertyInt("genxml/hidden/itemid"));

return GetArticleList(true);

}

public String CancelSortOrder()

{

var articleDataList = new ArticleDataList(\_postInfo, \_paramInfo, \_editLang, false);

articleDataList.Header = \_headerData;

articleDataList.Header.CancelItemSort();

return GetArticleList(true);

}

public String MoveSortOrder()

{

var articleDataList = new ArticleDataList(\_postInfo, \_paramInfo, \_editLang, false);

var toId = \_paramInfo.GetXmlPropertyInt("genxml/hidden/itemid");

articleDataList.Header = \_headerData;

articleDataList.SortOrderMove(toId);

return GetArticleList(true);

}

# Search on TextBox input