ColReorder example Server-side processing

Server-side processing can be exceptionally useful in DataTables when dealing with massive data sets, and ColReorder works with this as would be expected.

It is recommend that you use object based data with server-side processing and ColReorder, as this provides easily understandable mapping between the the columns and the data relation on the server, otherwise you need to work out array indexes on each call!

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Position | Office | Extn. | Start date | Salary |
| Name | Position | Office | Extn. | Start date | Salary |

* Javascript
* HTML
* CSS
* Ajax
* Server-side script

The Javascript shown below is used to initialise the table shown in this example:

$(document).ready(function() { $('#example').dataTable( { dom: 'Rlfrtip', processing: true, serverSide: true, ajax: "../../../examples/server\_side/scripts/objects.php", columns: [ { data: "first\_name" }, { data: "last\_name" }, { data: "position" }, { data: "office" }, { data: "start\_date" }, { data: "salary" } ] } ); } );

In addition to the above code, the following Javascript library files are loaded for use in this example:

* [../../../media/js/jquery.js](http://docs.google.com/media/js/jquery.js)
* [../../../media/js/jquery.dataTables.js](http://docs.google.com/media/js/jquery.dataTables.js)
* [../js/dataTables.colReorder.js](http://docs.google.com/js/dataTables.colReorder.js)

The HTML shown below is the raw HTML table element, before it has been enhanced by DataTables:

This example uses a little bit of additional CSS beyond what is loaded from the library files (below), in order to correctly display the table. The additional CSS used is shown below:

The following CSS library files are loaded for use in this example to provide the styling of the table:

* [../../../media/css/jquery.dataTables.css](http://docs.google.com/media/css/jquery.dataTables.css)
* [../css/dataTables.colReorder.css](http://docs.google.com/css/dataTables.colReorder.css)

This table loads data by Ajax. The latest data that has been loaded is shown below. This data will update automatically as any additional data is loaded.

The script used to perform the server-side processing for this table is shown below. Please note that this is just an example script using PHP. Server-side processing scripts can be written in any language, using [the protocol described in the DataTables documentation](http://datatables.net/manual/server-side).

Other examples

[**Examples**](http://docs.google.com/index.html)

* [Basic initialisation](http://docs.google.com/simple.html)
* [Initialisation using `new`](http://docs.google.com/new_init.html)
* [Alternative insert styling](http://docs.google.com/alt_insert.html)
* [Realtime updating](http://docs.google.com/realtime.html)
* [State saving](http://docs.google.com/state_save.html)
* [Scrolling table](http://docs.google.com/scrolling.html)
* [Predefined column ordering](http://docs.google.com/predefined.html)
* [Reset ordering API](http://docs.google.com/reset.html)
* [ColVis integration](http://docs.google.com/colvis.html)
* [FixedColumns integration](http://docs.google.com/fixedcolumns.html)
* [FixedHeader integration](http://docs.google.com/fixedheader.html)
* [jQuery UI styling](http://docs.google.com/jqueryui.html)
* [Individual column filtering](http://docs.google.com/col_filter.html)
* [Server-side processing](http://docs.google.com/server_side.html)

Please refer to the [DataTables documentation](http://www.datatables.net) for full information about its API properties and methods.

Additionally, there are a wide range of [extras](http://www.datatables.net/extras) and [plug-ins](http://www.datatables.net/plug-ins) which extend the capabilities of DataTables.

DataTables designed and created by [SpryMedia Ltd](http://www.sprymedia.co.uk) © 2007-2015

DataTables is licensed under the [MIT license](http://www.datatables.net/mit).