Scroller example Server-side processing (5,000,000 rows)

DataTables' server-side processing mode is a feature that naturally fits in with Scroller perfectly. Server-side processing can be used to show large data sets, with the server being used to do the data processing, and Scroller optimising the display of the data in a scrolling viewport.

When using server-side processing, Scroller will wait a small amount of time to allow the scrolling to finish before requesting more data from the server (200mS by default). This prevents you from DoSing your own server!

This example shows Scroller using server-side processing mode and 5 million rows. **Important** This particular example uses [ajax](http://datatables.net/reference/option/ajax) as a function to *fake* the data to show Scroller's ability to show large data sets. It does not have a real database behind it! You would normally not use [ajax](http://datatables.net/reference/option/ajax) as a function to generate data, but rather as a url for where to fetch the real data!

In this example we also enable the [scroller.loadingIndicator](http://datatables.net/reference/option/scroller.loadingIndicator) option of Scroller to show the end user what is happening when they scroll passed the currently loaded data.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | First name | Last name | ZIP / Post code | Country |

* 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( { serverSide: true, ordering: false, searching: false, ajax: function ( data, callback, settings ) { var out = []; for ( var i=data.start, ien=data.start+data.length ; i<ien ; i++ ) { out.push( [ i+'-1', i+'-2', i+'-3', i+'-4', i+'-5' ] ); } setTimeout( function () { callback( { draw: data.draw, data: out, recordsTotal: 5000000, recordsFiltered: 5000000 } ); }, 50 ); }, scrollY: 200, scroller: { loadingIndicator: true } } ); } );

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

* [//code.jquery.com/jquery-1.12.3.min.js](http://code.jquery.com/jquery-1.12.3.min.js)
* [../../../../media/js/jquery.dataTables.js](http://docs.google.com/media/js/jquery.dataTables.js)
* [../../js/dataTables.scroller.js](http://docs.google.com/js/dataTables.scroller.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/scroller.dataTables.css](http://docs.google.com/css/scroller.dataTables.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

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

* [Basic initialisation](http://docs.google.com/simple.html)
* [State saving](http://docs.google.com/state_saving.html)
* [Client-side data source (50,000 rows)](http://docs.google.com/large_js_source.html)
* [Server-side processing (5,000,000 rows)](http://docs.google.com/server-side_processing.html)
* [API](http://docs.google.com/api_scrolling.html)
* [FixedColumns integration](http://docs.google.com/fixedColumns.html)
* [Select integration](http://docs.google.com/select.html)

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

* [Bootstrap](http://docs.google.com/styling/bootstrap.html)
* [Bootstrap 4](http://docs.google.com/styling/bootstrap4.html)
* [Foundation](http://docs.google.com/styling/foundation.html)
* [Semantic UI](http://docs.google.com/styling/semanticui.html)
* [jQuery UI](http://docs.google.com/styling/jqueryui.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 [extensions](http://www.datatables.net/extensions) 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) © 2007-2016

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