Getting started Components

About
Population
Constructor
Ajax
Animation
Loops
Visibility
Effects
Substitute functions
Attributes
Html, text and value
Css and classes
Position and size
Manipulation
DataSet

Utils

Base

Containers
Grid system
Typography
Tables
Forms
Buttons
Images
Figures
Lists

Form component Checkbox

File input Input Input material Keypad Rating Radio Select Slider Double Slider Double Slider Spinner Switch Tag input Textarea

App bar Bottom navigation Bottom sheet Menu Ribbon Menu Side bar Side navigation

Accordion
Badge
Carousel
Cards
Cube
Counter
Charms
Chat
Donut
Image compare
Image magnifiel
Gravatar

List
ListView
Master
NavView
Panels
Progress & Activity
Streamer
Stepper
Solitter

Table
Tiles
TreeView

Wizard

Tabs

Dialogs
Info box
Hints
Notify system
Popovers
Toasts
Windows

Date & time
Calendar
Calendar picker
Date picker
Time picker
Countdown
Formatting date
Media

Table component



Create interactive table with sorting, filtering and pagination features.

Create table

New in 4.2.8, updated in 4.2.23

If you need to create interactive table with ordering, filtering, searching and pagination features, you can use Metro 4 table component. To create table, add attribute data-role="table" to your HTML table element and setup, if need, additional options with special attributes. You can create table and for your own HTML table with predefined data and for the table with remote data definition.

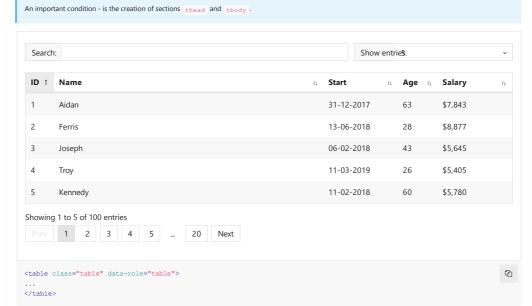


Table options

You can use different options to define table behavior. This options you can use in tag definition.

Option	Data-*	Default	Desc			
check	data-check	false	Enable checkbox or radio for each row			
checkType	data-check-type	checkbox	Draw checkbox or radio for each row			
checkStyle	data-check-style	1	Set style for checkbox or radio, 1 or 2			
checkName	data-check-name	null	Set name for checkbox or radio			
checkColIndex	data-check-col-index	0	Data column index from 0 for checkbox or radio value			
checkStoreKey	data-store-key	TABLE:\$1:KEYS	Key for storage (used window.storage), where check value are store \$1 will be replaced with table id.			
rownum	data-rownum	false	Show/hide rows numbers			
rownumTitle	data-rownum-title	#	Title for rownum head cell			
filters	data-filters		Table filters definition, comma separated value with functions name			
filtersOperator	data-filters-operator	and	Defined how filters combined, value and or or			
source	data-source	null	Link to external data			
searchMinLength	data-search-min-length	1	Min length for begin searching			
searchThreshold	data-search-threshold	500	The threshold in ms to begin searching, while user press keys for search phrase and the interval between key press less than this value, searching not started			
searchFields	data-search-fields		Comma-separated value with column names, where will be search			
showRowsSteps	data-show-rows-steps	true	Show/hide rows steps box			
showSearch	data-show-search	true	Show/hide search field			
showTableInfo	data-show-table-info	true	Show/hide table info block			
showPagination	data-show-pagination	true	Show/hide table pagination			
paginationShortMode	data-pagination-short-mode	true	Show table pagination in short mode			
showActivity	data-show-activity	true	Show activity for table operations			
muteTable	data-mute-table	true	Mute table for table operations			
rows	data-rows	10	How many rows will be displayed on page, if this value is -1, all row will be displayed			

Table of contents

3

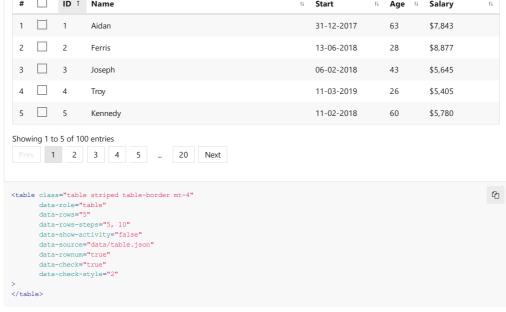
Table
Create table
Table options
Columns
Data cells
Sorting
Table search
Filtering
Pagination
Rows counter
Table info
Table inspector
Export data
External data
Events
Methods
Customize
Custom wrappers

Horizontal scroll

Video player Audio player Collapse Color module Draggable Dropdown Form validator Hotkeys Micro templates Ripple Storage Session storage Sorter Touch and swipe

Utilities Animations Additional

Option	Data-*	Default	Desc				
rowsSteps	data-rows-steps	10, 25, 50, 100	Steps fo rows select box				
staticView	data-static-view	false	If true, table always displayed as defined				
viewSaveMode	data-view-save-mode	client	Where table view will be saved, values: client, server				
viewSavePath	data-view-save-path	TABLE:\$1:OPTIONS	Key for client-side, href for server-side. \$1 will be replaced with table id				
decimalSeparator	data-decimal-separator	. (dot)	Decimal separator for numeric values. Used for right sorting				
thousandSeparator	data-thousand-separator	, (comma)	Thousand separator for numeric values. Used for right sorting				
tableRowsCountTitle	data-table-rows-count-title	Show entries:	Title for rows steps box				
tableSearchTitle	data-table-search-title	Search:	Title for search input				
tableInfoTitle	data-table-info-title	Showing \$1 to \$2 of \$3 entries	Title for table info block				
paginationPrevTitle	data-pagination-prev-title	Prev	Title pagination prev button				
paginationNextTitle	data-pagination-next-title	Next	Title pagination next button				
allRecordsTitle	data-all-records-title	All	Title all records in rows steps block				
nspectorTitle	data-inspector-title	Inspector	Title for table inspector window				
activityType	data-activity-type	cycle	Activity type for long operations				
activityStyle	data-activity-style	color	Activity style for long operations				
activityTimeout	data-activity-timeout	100	Timeout between activity show and table operation start				
cellWrapper	data-cell-wrapper	true	If true, cell data will be placed in div with class cell-wrapper				
norizontalScroll	data-horizontal-scroll	false	Helper to create table horizontal scrolling for wide table				
Search: Show entries: 5							
# ID ↑ I	Name		8 Start 8 Age 8 Salary				
	Aidan		,				
_			31-12-2017 63 \$7,843				
	Ferris		13-06-2018 28 \$8,877				
3 3 J	loseph		06-02-2018 43 \$5,645				
4 4 1	Ггоу		11-03-2019 26 \$5,405				



Columns definition

For right works component, you must right define columns. Columns must be defined in thead section. You can use any attributes to define its:

Attribute	Desc
data-name	Column name
data-sortable	Sortable column or not. Values: true , false
data-sort-dir	Initial sorting direction. Values: asc , desc
data-format	Data format helper for right sorting. Values: string (default), date , number , int , float , money
data-format-mask	If you data is date, and date not in ECMAScript date format, you must define data mask (see Extensions for string.toDate in this article)
data-size	If can define size for column.
data-show	If can define show or hide column.
data-cls	If can define additional classes for column heading cell.
data-cls-column	If can define additional classes for column data cells.

4

Currently, you can define only one row in head section

Data definition

Data must be defined in $\frac{1}{1}$ section. If you want to add classes to data cell, use $\frac{1}{1}$ attribute $\frac{1}{1}$ attribute $\frac{1}{1}$ attribute $\frac{1}{1}$ section.

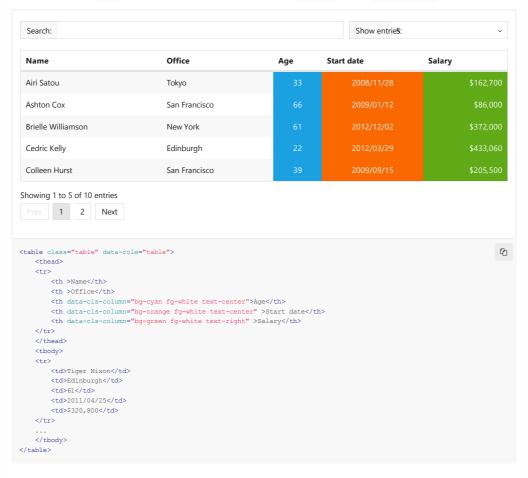
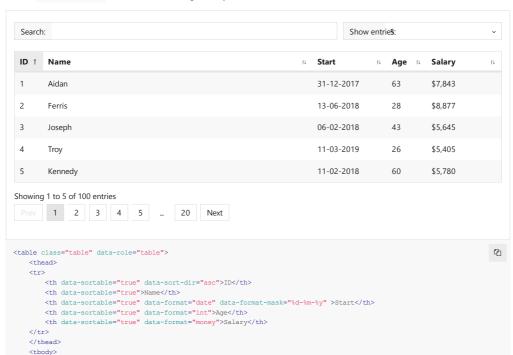


Table sorting

To use sorting, you must define columns, who can be sorted. To create sortable column, add attribute data-sortable="true" to head cell. If you need to sort on this column after table init, add attribute data-sort-dir="asc|desc" for initializing sorting direction.

If data for sortable column is a not string, you must use attribute data-format. If data is date and date not in ECMAScript format, you must add attribute data-format—mask. More about covert string to date you can read in this article



```
        >1

        >2
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
        >1
```

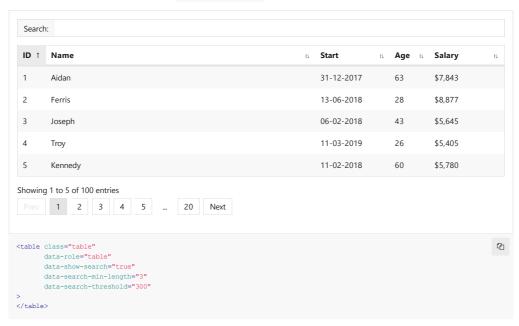
Table search

Table component support interactive search. To activate search function you must:

Enable search input field with attribute data-show-search="true" (enabled by default)

Set min string length for search with attribute data-search-min-length (default: 1)

Setup threshold for start search with attribute data-search-threshold (default: 500)



By default, all fields are searched. You can define the fields to be searched for. To define it use attribute data-search-fields. Fields list is must be a comma separated string with field names, defined in head cell attribute data-name.

```
4
<table class="table"
   data-role="table"
   data-show-search="true"
   data-search-min-length="3"
   data-search-threshold="300"
   data-search-fields="name, age">
  <thead>
    ID
    Name
    Start
Age

    Salary
  </thead>
  1
    Tiger Nixon
    Edinburgh
    25/04/2011
    61
    $320,800
```

Filtering

Table components contains API to create custom filters for data. These filters works before search and is connected with it by the and operator if search is not empty. These filters are user-defined functions and can be applied in two ways: with attribute data-filters and with table api method addFilter. In both cases, the functions must be defined before they are used.

Use attribute

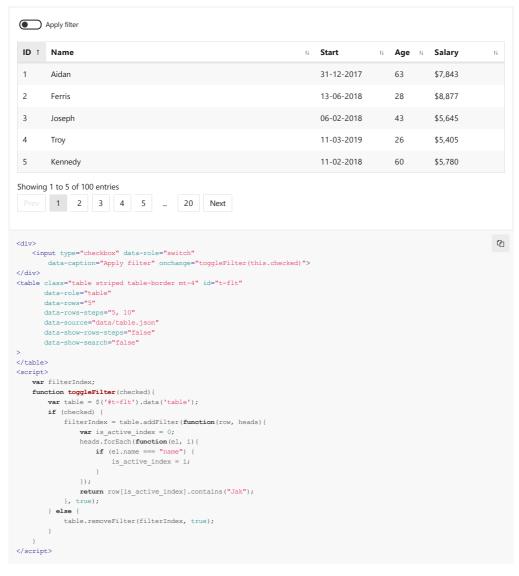


```
function myFilter(row, heads) {
    var is_active_index = 0;
    heads.forEach(function(el, i) {
        if (el.name === "name") {
            is_active_index = i;
        }
     });
    return row[is_active_index].contains("Jak");
    }

</script>

...
```

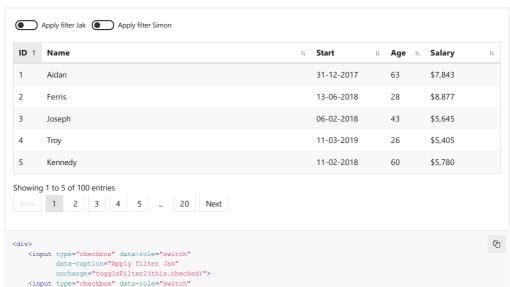
Use table API



Filters operator

data-caption="Apply filter Simon"

By default, all filters applied with operator $_{\mathtt{and}}$. You can set this to operator $_{\mathtt{or}}$ with attribute $_{\mathtt{data-filters-operator="or"}}$.



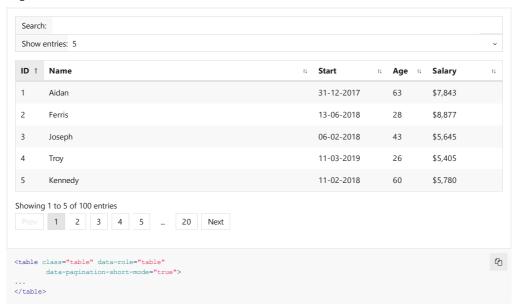
```
toggleFilter3(this.checked)">
</div>
data-rows="5"
       data-rows-steps="5, 10"
       data-show-rows-steps="false"
data-show-search="false"
       data-filters-operator="or"
var filterIndex2, filterIndex3;
    function toggleFilter2 (checked) {
        var table = $('#t-flt-2').data('table');
        if (checked) {
            filterIndex2 = table.addFilter(function(row, heads){
                var is_active_index = 0;
heads.forEach(function(el, i){
                   if (el.name === "name") {
   is_active_index = i;
                return row[is_active_index].contains("Jak");
            }, true);
        } else {
            table.removeFilter(filterIndex2, true);
    function toggleFilter3 (checked) {
        var table = $('#t-flt-2').data('table');
        if (checked) {
            filterIndex3 = table.addFilter(function(row, heads){
                var is_active_index = 0;
heads.forEach(function(el, i){
                        is_active_index = i;
                return row[is_active_index].contains("Simon");
            }, true);
        } else {
            table.removeFilter(filterIndex3, true);
</script>
```

Filters works before search and is connected with it by the and operator if search is not empty.

Pagination

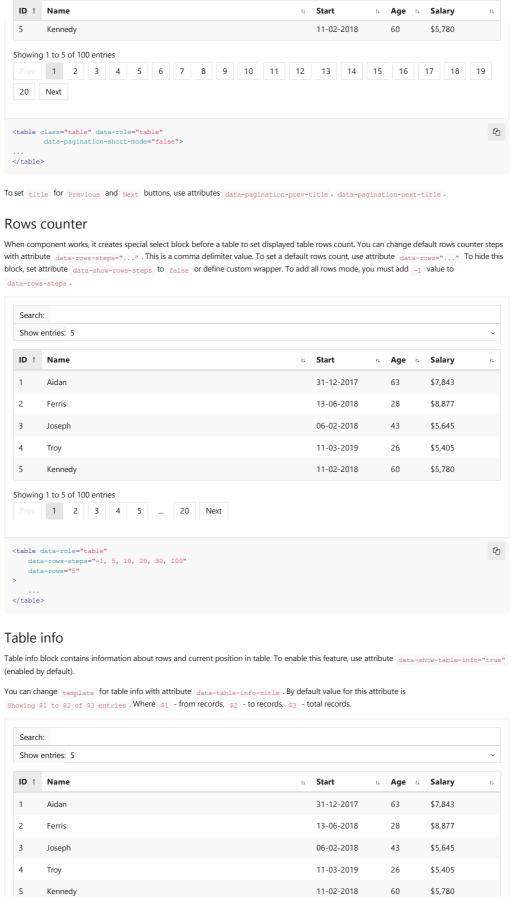
To use pagination option, this options must be anabled with attribute data-show-pagination="true" (enabled by default). By default, used pagination short mode. If you need show all pages links in pagination, you can set it with attribute data-pagination-short-mode="false" .

Pagination short mode



Pagination all mode

Search							
Show 6	entries: 5						
ID ↑	Name	11	Start N		Age ₦	Salary	ŧ
1	Aidan		31-12-2017		63	\$7,843	
2	Ferris		13-06-2018	i	28	\$8,877	
3	Joseph		06-02-2018		43	\$5,645	
4	Troy		11-03-2019		26	\$5,405	



Display from 1 to 5 of 100 records Prev 1 2 3 4 5 ... 20 Next 4 data-table-info-title="Display from \$1 to \$2 of \$3 records"

Table inspector

 $Component\ table\ contains\ inside\ a\ \ {\tt table\ inspector}\ \ to\ manipulate\ columns\ in\ visual\ mode.\ To\ open\ \ {\tt inspector}\ ,\ you\ must\ call\ API\ method$ openInspector() or toggleInspector(), and to close it - closeInspector(); .

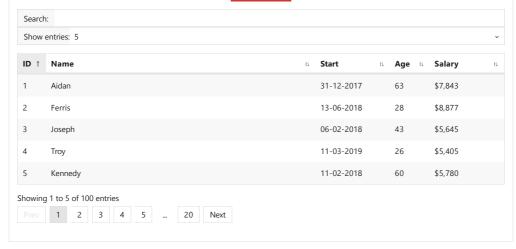


Table view

For each table, a view is created, which stores information about the current state of the table. All changes that the user makes in the table inspector are saved in this view.

You can setup view options with special attributes:

```
data-static-view="true|false" - use static view or not data-view-save-mode="client|server" - how view is stored data-view-save-path="storage_key|url" - where view is stored
```

Client

Server

View format

Object view have next format:

```
{
    0: {index: 0, index-view: 0, show: true, size: 50}
    1: {index: 1, index-view: 1, show: true, size: ""}
    2: {index: 2, index-view: 2, show: true, size: 150}
    3: {index: 3, index-view: 3, show: true, size: 80}
    4: {index: 4, index-view: 4, show: true, size: 150}
    ...
}
```

Where index - field index in original data, index-view - field index in displayed table, show - field displayed or not, size - field size.

Export table data to CSV

You can exporting table data into CSV file. To export data, you must use API method export(to, mode, filename, options) .

```
var table = $(el).data('table');
table.export('CSV', 'all', 'table-export.csv', {
    csvNewLine: "\t",
    csvNewLine: "\r\n",
    includeHeader: true
});
```

Export to...

Currently supported only CSV target.

Export modes

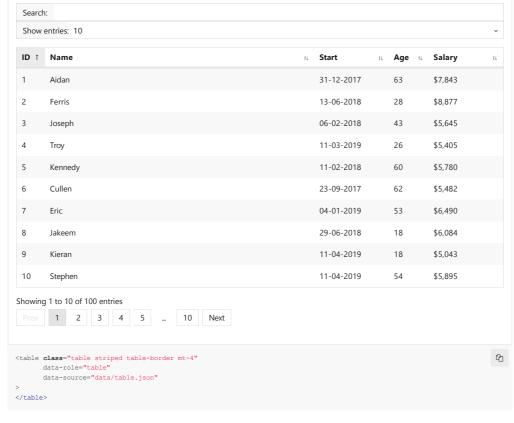
- all all table data
- all-filtered all with applied filters and search
- checked checked rows
- view rows, displayed in current page

Export options

- csvDelimiter default \t
- csvNewLine default \r\n
- includeHeader default true

External data

You can create table from external JSON data. To create it, use attribute data-source .



External data format

To use external source, you must create JSON data in specific format. Example of JSON file.

```
{
    header: [...],
    data: [...]
}
```

Header

In $_{\rm header}$, you must describe information about all table columns. Each item can contains the next data:

```
name - column name
sortable - sortable or not
sortDir - sort direction asc or desc
format - column format
cls - classes for head cell
clsColumn - classes for data cell
size - column size
show - show or hide column
template - template for cell data. Use this.value to manipulate with cell value
thousandSeparator - thousand separator for numbers
decimalSeparator - decimal separator for numbers
```

See this article for details about micro templating in Metro 4.

```
"header": [
                                                                                                                                          4
"name": "id",
   "title": "ID",
   "size": 50,
   "sortable": true,
"sortDir": "asc",
    "format": "number'
   "name": "name",
"title": "Name",
   "sortable": true
 },7
 "sortable": true,
   "format": "date",
    "formatMask": "dd-mm-yyyy"
   "title": "Age",
"sortable": true,
    "size": 80
   "title": "Salary",
"sortable": true,
    "format": "money",
    "show": true
```

```
1
```

Data

In data section you must put simple text or html data.

```
"data": [
[
1,
    "Aidan",
    "31-12-2017",
    63,
    "$7,843"
],
[
2,
    "Ferris",
    "13-06-2018",
    28,
    "$8,877"
],
[
[
3,
    "Joseph",
    "06-02-2018",
    43,
    "$5,645"
],
```

Events

When table works, it generates a series of events that you can use to interact with the component.

Option	Data-*	Desc
onDraw	data-on-draw	Event fired when the table was drawn.
onDrawRow	data-on-draw-row	Event fired when draw row.
onDrawCell	data-on-draw-cell	Event fired when draw cell.
onAppendRow	data-on-append-row	Event fired when row added to table.
onAppendCel1	data-on-append-cell	Event fired when cell added to row.
onSortStart	data-on-sort-start	Event fired when sorting process stated.
onSortStop	data-on-sort-stop	Event fired when sorting process stopped.
onSortItemSwitch	data-on-sort-item-switch	Event fired when table rows switched on sorting process.
onSearch(filter, items)	data-on-search	Event fired on search.
onRowsCountChange(val)	data-on-rows-count-change	Event fired when user change rows count to display.
onDataLoad(source)	data-on-data-load	Event fired before data will be loaded.
onDataLoaded(source, response)	data-on-data-loaded	Event fired after data loaded.
onDataLoadError(source, xhr, textStatus, errorThrown)	data-on-data-load-error	Event fired when data loads stop with error.
onFilterRowAccepted(row)	data-on-filter-row-accepted	Row accepted by filters.
onFilterRowDeclined(row)	data-on-filter-row-declined	Row declined by filters.
onCheckClick()	data-on-check-click	
onCheckClickAll()	data-on-check-click-all	
onCheckDraw()	data-on-check-draw	
onViewSave()	data-on-view-save	
onViewGet()	data-on-view-get	
onTableCreate(e1)	data-on-table-create	Event fired after the table created.

Methods

You can use table API methods to interact with it.

```
draw()
sorting(dir)
search(val)
loadData(source, review)
reload(review)
next()
prev()
first()
last()
page(num)
addFilter(func, redraw)
removeFilter(func_index)
removeFilters()
getItems()
```

getHeads() getFilteredItems() getSelectedItems() getStoredKeys() clearSelected() clearSelected(redraw) getFilters() getFiltersIndexes() openInspector() closeInspector() toggleInspector() resetView() export(to, mode, filename, options) $deleteltem(field_index, val)$ deleteItemByName(field_name, val) deleteItem(field_index, function(val){return true|false}) $deleteItemByName(field_name,\ function(val)\{return\ true | false\})$

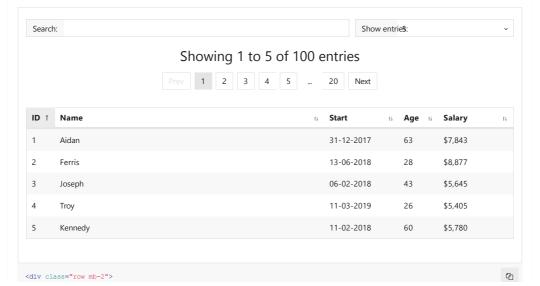
Customize

You can customize style of table and its elements with next options:

Option	Data-*	Default	Description
clsComponent	data-cls-component	Additional classes for table component	
clsTableContainer	data-cls-table-container	Additional for table container	
clsTable	data-cls-table	Additional class for table	
clsHead	data-cls-head	Additional class for table head	
clsHeadRow	data-cls-head-row	Additional class for table head row	
clsHeadCell	data-cls-head-cell	Additional class for table head cell	
clsBody	data-cls-body	Additional class for table body	
clsBodyRow	data-cls-body-row	Additional class for table body row	
clsBodyCell	data-cls-body-cell	Additional class for table body cell	
clsFooter	data-cls-footer	Additional class for table footer	
clsFooterRow	data-cls-footer-row	Additional class for table footer row	
clsFooterCell	data-cls-footer-cell	Additional class for table footer cell	
clsPagination	data-cls-pagination	Additional class for table pagination	
clsTableTop	data-cls-table-top	Additional class for table top block	
clsTableBottom	data-cls-table-bottom	Additional class for table bottom block	
clsTableInfo	data-cls-table-info	Additional class for table info block	
clsTablePagination	data-cls-table-pagination	Additional class for table pagination block	
clsRowsCount	data-cls-rows-count	Additional class for rows count block	
clsSearch	data-cls-search	Additional class for search block	
clsRow	data-cls-row	Additional class for each data row	
clsRowEven	data-cls-row-even	Additional class for even data rows	
clsRowOdd	data-cls-row-odd	Additional class for odd data rows	

Custom elements wrappers

You can set custom wrappers for <code>search</code>, <code>rows</code> counter, <code>table</code> info and <code>pagination</code> elements. To set it, use special named attributes: data-search-wrapper, data-rows-wrapper, data-info-wrapper, data-pagination-wrapper.

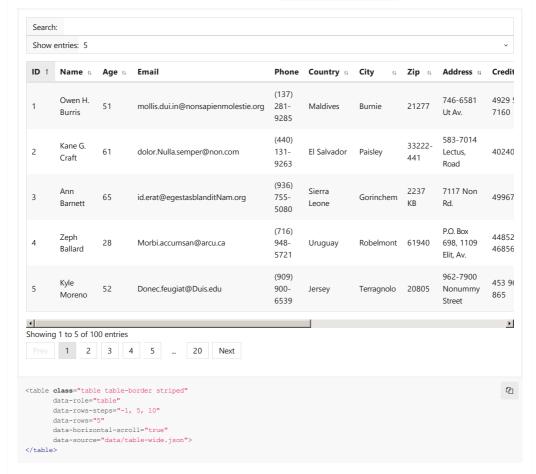


```
<div class="cell-md-8 my-search-wrapper"></div>
<div class="cell-md-4 my-rows-wrapper"></div>
</div>

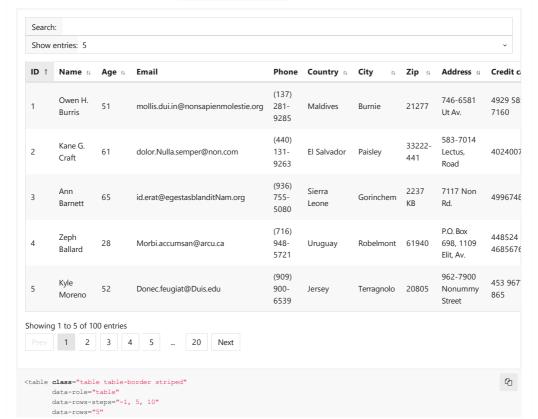
</pr
```

Horizontal scroll

You can enable horizontal scrolling for wide table. To enable it, use attribute data-horizontal-scroll="true".



You can stop horizontal scrolling with attribute ${\tt data-horizontal-scroll-stop}$.



data-horizontal-scroll="true"
data-horizontal-scroll-stop="lg"
data-source="data/table-wide.json">