

Search...

Getting started

Components

- M4Q
- About
- Population
- Constructor
- Ajax
- Animation
- Loops
- Visibility
- Effects
- Subtree functions
- Attributes
- Html, text and value
- Css and classes
- Position and size
- Manipulation
- DataSet
- Events
- Utils
- Base
- Containers
- Grid system
- Typography
- Tables
- Forms
- Buttons
- Images
- Figures
- Lists
- Form components
- Checkbox
- File input
- Input
- Input material
- Keypad
- Rating
- Radio
- Select
- Slider
- Double Slider
- Spinner
- Switch
- Tag input
- Textarea

- Menus
- App bar
- Bottom navigation
- Bottom sheet
- Menu
- Ribbon Menu
- Side bar
- Side navigation
- Controls
- Accordion
- Badge
- Carousel
- Cards
- Cube
- Counter
- Charms
- Chat
- Donut
- Image compare
- Image magnifier
- Gravatar
- List
- ListView
- Master
- NavView
- Panels
- Progress & Activity
- Streamer
- Stepper
- Splitter
- Tabs
- Tabs material
- Table**
- Tiles
- TreeView
- Wizard

- Information
- Dialogs
- Info box
- Hints
- Notify system
- Popovers
- Toasts
- Windows

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

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.

An important condition - is the creation of sections `thead` and `tbody`.

Search:

Show entries: ▾

ID	Name	Start	Age	Salary
1	Aidan	31-12-2017	63	\$7,843
2	Ferris	13-06-2018	28	\$8,877
3	Joseph	06-02-2018	43	\$5,645
4	Troy	11-03-2019	26	\$5,405
5	Kennedy	11-02-2018	60	\$5,780

Showing 1 to 5 of 100 entries

Prev

1

2

3

4

5

...

20

Next

```
<table class="table" data-role="table">
...
</table>
```

Table options

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

Option	Data-*	Default	Desc
<code>check</code>	<code>data-check</code>	false	Enable checkbox or radio for each row
<code>checkType</code>	<code>data-check-type</code>	checkbox	Draw checkbox or radio for each row
<code>checkStyle</code>	<code>data-check-style</code>	1	Set style for checkbox or radio, 1 or 2
<code>checkName</code>	<code>data-check-name</code>	null	Set name for checkbox or radio
<code>checkColIndex</code>	<code>data-check-col-index</code>	0	Data column index from 0 for checkbox or radio value
<code>checkStoreKey</code>	<code>data-store-key</code>	TABLE:\$1:KEYS	Key for storage (used window.storage), where check value are stored. \$1 will be replaced with table id.
<code>rownum</code>	<code>data-rownum</code>	false	Show/hide rows numbers
<code>rownumTitle</code>	<code>data-rownum-title</code>	#	Title for rownum head cell
<code>filters</code>	<code>data-filters</code>		Table filters definition, comma separated value with functions names
<code>filtersOperator</code>	<code>data-filters-operator</code>	and	Defined how filters combined, value <code>and</code> or <code>or</code>
<code>source</code>	<code>data-source</code>	null	Link to external data
<code>searchMinLength</code>	<code>data-search-min-length</code>	1	Min length for begin searching
<code>searchThreshold</code>	<code>data-search-threshold</code>	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
<code>searchFields</code>	<code>data-search-fields</code>		Comma-separated value with column names, where will be search
<code>showRowsSteps</code>	<code>data-show-rows-steps</code>	true	Show/hide rows steps box
<code>showSearch</code>	<code>data-show-search</code>	true	Show/hide search field
<code>showTableInfo</code>	<code>data-show-table-info</code>	true	Show/hide table info block
<code>showPagination</code>	<code>data-show-pagination</code>	true	Show/hide table pagination
<code>paginationShortMode</code>	<code>data-pagination-short-mode</code>	true	Show table pagination in short mode
<code>showActivity</code>	<code>data-show-activity</code>	true	Show activity for table operations
<code>muteTable</code>	<code>data-mute-table</code>	true	Mute table for table operations
<code>rows</code>	<code>data-rows</code>	10	How many rows will be displayed on page, if this value is -1, all rows will be displayed

Table of contents

- 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
- Tools
- 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
<code>rowsSteps</code>	<code>data-rows-steps</code>	10, 25, 50, 100	Steps fo rows select box
<code>staticView</code>	<code>data-static-view</code>	false	If true, table always displayed as defined
<code>viewSaveMode</code>	<code>data-view-save-mode</code>	client	Where table view will be saved, values: client, server
<code>viewSavePath</code>	<code>data-view-save-path</code>	TABLE:\$1:OPTIONS	Key for client-side, href for server-side. \$1 will be replaced with table id
<code>decimalSeparator</code>	<code>data-decimal-separator</code>	, (dot)	Decimal separator for numeric values. Used for right sorting
<code>thousandSeparator</code>	<code>data-thousand-separator</code>	, (comma)	Thousand separator for numeric values. Used for right sorting
<code>tableRowsCountTitle</code>	<code>data-table-rows-count-title</code>	Show entries:	Title for rows steps box
<code>tableSearchTitle</code>	<code>data-table-search-title</code>	Search:	Title for search input
<code>tableInfoTitle</code>	<code>data-table-info-title</code>	Showing \$1 to \$2 of \$3 entries	Title for table info block
<code>paginationPrevTitle</code>	<code>data-pagination-prev-title</code>	Prev	Title pagination prev button
<code>paginationNextTitle</code>	<code>data-pagination-next-title</code>	Next	Title pagination next button
<code>allRecordsTitle</code>	<code>data-all-records-title</code>	All	Title all records in rows steps block
<code>inspectorTitle</code>	<code>data-inspector-title</code>	Inspector	Title for table inspector window
<code>activityType</code>	<code>data-activity-type</code>	cycle	Activity type for long operations
<code>activityStyle</code>	<code>data-activity-style</code>	color	Activity style for long operations
<code>activityTimeout</code>	<code>data-activity-timeout</code>	100	Timeout between activity show and table operation start
<code>cellWrapper</code>	<code>data-cell-wrapper</code>	true	If true, cell data will be placed in div with class <code>cell-wrapper</code>
<code>horizontalScroll</code>	<code>data-horizontal-scroll</code>	false	Helper to create table horizontal scrolling for wide table

Search:

Show entries: 5

#	<input type="checkbox"/>	ID ↑	Name	Start	Age	Salary
1	<input type="checkbox"/>	1	Aidan	31-12-2017	63	\$7,843
2	<input type="checkbox"/>	2	Ferris	13-06-2018	28	\$8,877
3	<input type="checkbox"/>	3	Joseph	06-02-2018	43	\$5,645
4	<input type="checkbox"/>	4	Troy	11-03-2019	26	\$5,405
5	<input type="checkbox"/>	5	Kennedy	11-02-2018	60	\$5,780

Showing 1 to 5 of 100 entries

Prev

1

2

3

4

5

...

20

Next

```
<table class="table striped table-border mt-4" data-role="table" data-rows="5" data-rows-steps="5, 10" data-show-activity="false" data-source="data/table.json" data-rownum="true" data-check="true" data-check-style="2">
  >
</table>
```

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

```
<table class="table" data-role="table">
```

```
<thead>
<tr>
  <th data-sortable="true" data-sort-dir="asc">Name</th>
  <th data-sortable="true">Office</th>
  <th data-sortable="true" data-format="number">Age</th>
  <th data-sortable="true" data-format="date">Start date</th>
  <th data-sortable="true" data-show="false" data-format="money">Salary</th>
</tr>
</thead>
...
</table>
```

Currently, you can define only one row in head section.

Data definition

Data must be defined in `tbody` section. If you want to add classes to data cell, use `head cell` attribute `data-cls-column`.

Search:

Show entries: ▾

Name	Office	Age	Start date	Salary
Airi Satou	Tokyo	33	2008/11/28	\$162,700
Ashton Cox	San Francisco	66	2009/01/12	\$86,000
Brielle Williamson	New York	61	2012/12/02	\$372,000
Cedric Kelly	Edinburgh	22	2012/03/29	\$433,060
Colleen Hurst	San Francisco	39	2009/09/15	\$205,500

Showing 1 to 5 of 10 entries

Prev

1

2

Next

```
<table class="table" data-role="table">
  <thead>
    <tr>
      <th >Name</th>
      <th >Office</th>
      <th data-cls-column="bg-cyan fg-white text-center">Age</th>
      <th data-cls-column="bg-orange fg-white text-center" >Start date</th>
      <th data-cls-column="bg-green fg-white text-right" >Salary</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Tiger Nixon</td>
      <td>Edinburgh</td>
      <td>61</td>
      <td>2011/04/25</td>
      <td>$320,800</td>
    </tr>
    ...
  </tbody>
</table>
```

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](#)

Search:

Show entries: ▾

ID ↑	Name	Start	Age	Salary
1	Aidan	31-12-2017	63	\$7,843
2	Ferris	13-06-2018	28	\$8,877
3	Joseph	06-02-2018	43	\$5,645
4	Troy	11-03-2019	26	\$5,405
5	Kennedy	11-02-2018	60	\$5,780

Showing 1 to 5 of 100 entries

Prev

1

2

3

4

5

...

20

Next

```
<table class="table" data-role="table">
  <thead>
    <tr>
      <th data-sortable="true" data-sort-dir="asc">ID</th>
      <th data-sortable="true">Name</th>
      <th data-sortable="true" data-format="date" data-format-mask="%d-%m-%y" >Start</th>
      <th data-sortable="true" data-format="int">Age</th>
      <th data-sortable="true" data-format="money">Salary</th>
    </tr>
  </thead>
  <tbody>
```

```
<tr>
  <td>1</td>
  <td>Tiger Nixon</td>
  <td>Edinburgh</td>
  <td>25/04/2011</td>
  <td>61</td>
  <td>$320,800</td>
</tr>
...
</tbody>
</table>
```

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)

Search:

ID ↑	Name	Start	Age	Salary
1	Aidan	31-12-2017	63	\$7,843
2	Ferris	13-06-2018	28	\$8,877
3	Joseph	06-02-2018	43	\$5,645
4	Troy	11-03-2019	26	\$5,405
5	Kennedy	11-02-2018	60	\$5,780

Showing 1 to 5 of 100 entries

Prev

1

2

3

4

5

...

20

Next

```
<table class="table"
  data-role="table"
  data-show-search="true"
  data-search-min-length="3"
  data-search-threshold="300">
  >
</table>
```

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`.

```
<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>
    <tr>
      <th data-name="id" >ID</th>
      <th data-name="name" >Name</th>
      <th data-name="start" >Start</th>
      <th data-name="age" >Age</th>
      <th data-name="salary" >Salary</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Tiger Nixon</td>
      <td>Edinburgh</td>
      <td>25/04/2011</td>
      <td>61</td>
      <td>$320,800</td>
    </tr>
    ...
  </tbody>
</table>
```

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

Search:

ID ↑	Name	Start	Age	Salary
8	Jakeem	29-06-2018	18	\$6,084

Showing 1 to 1 of 1 entries

Prev

1

Next

```
<script>
    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>

<table class="table"
    data-role="table"
    data-filters="myFilter"
>
...
</table>
```

Use table API

☒ Apply filter

ID ↑	Name	Start	Age	Salary
1	Aidan	31-12-2017	63	\$7,843
2	Ferris	13-06-2018	28	\$8,877
3	Joseph	06-02-2018	43	\$5,645
4	Troy	11-03-2019	26	\$5,405
5	Kennedy	11-02-2018	60	\$5,780

Showing 1 to 5 of 100 entries

Prev

1

2

3

4

5

...

20

Next

<div>

<input type="checkbox" data-role="switch" data-caption="Apply filter" onchange="toggleFilter(this.checked)">

</div>

<table class="table striped table-border mt-4" id="t-flt"

data-role="table"

data-rows="5"

data-rows-steps="5, 10"

data-source="data/table.json"

data-show-rows-steps="false"

data-show-search="false"

>

</table>

<script>

var filterIndex;

function toggleFilter (checked){

var table = \$('#t-flt').data('table');

if (checked) {

filterIndex = 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(filterIndex, true);

}

}

</script>

Filters operator

By default, all filters applied with operator `and`. You can set this to operator `or` with attribute `data-filters-operator="or"`.

☒ Apply filter Jak

☒ Apply filter Simon

ID ↑	Name	Start	Age	Salary
1	Aidan	31-12-2017	63	\$7,843
2	Ferris	13-06-2018	28	\$8,877
3	Joseph	06-02-2018	43	\$5,645
4	Troy	11-03-2019	26	\$5,405
5	Kennedy	11-02-2018	60	\$5,780

Showing 1 to 5 of 100 entries

Prev

1

2

3

4

5

...

20

Next

<div>

<input type="checkbox" data-role="switch" data-caption="Apply filter Jak" onchange="toggleFilter2(this.checked)">

<input type="checkbox" data-role="switch" data-caption="Apply filter Simon"

```

        onchange="toggleFilter3(this.checked)">
</div>
<table class="table striped table-border mt-4" id="t-flt-2"
  data-role="table"
  data-rows="5"
  data-rows-steps="5, 10"
  data-source="data/table.json"
  data-show-rows-steps="false"
  data-show-search="false"
  data-filters-operator="or"
>
</table>
<script>
    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) {
                    if (el.name === "name") {
                        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 enabled 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

Search:

Show entries: 5

ID ↑	Name	Start	Age	Salary
1	Aidan	31-12-2017	63	\$7,843
2	Ferris	13-06-2018	28	\$8,877
3	Joseph	06-02-2018	43	\$5,645
4	Troy	11-03-2019	26	\$5,405
5	Kennedy	11-02-2018	60	\$5,780

Showing 1 to 5 of 100 entries

Prev

1

2

3

4

5

...

20

Next

```
<table class="table" data-role="table"
  data-pagination-short-mode="true">
...
</table>
```

Pagination all mode

Search:

Show entries: 5

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

ID	1	Name	Start	Age	Salary
5		Kennedy	11-02-2018	60	\$5,780

Showing 1 to 5 of 100 entries

Prev

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

Next

```
<table class="table" data-role="table" data-pagination-short-mode="false">...</table>
```

To set `title` for `Previous` and `Next` buttons, use attributes `data-pagination-prev-title` , `data-pagination-next-title` .

Rows counter

When component works, it creates special select block before a table to set displayed table rows count. You can change default rows counter steps with attribute `data-rows-steps="..."` . This is a comma delimiter value. To set a default rows count, use attribute `data-rows="..."` . To hide this block, set attribute `data-show-rows-steps` to `false` or define custom wrapper. To add all rows mode, you must add `-1` value to `data-rows-steps` .

Search:

Show entries: 5

ID	1	Name	Start	Age	Salary
1		Aidan	31-12-2017	63	\$7,843
2		Ferris	13-06-2018	28	\$8,877
3		Joseph	06-02-2018	43	\$5,645
4		Troy	11-03-2019	26	\$5,405
5		Kennedy	11-02-2018	60	\$5,780

Showing 1 to 5 of 100 entries

Prev

1

2

3

4

5

...

20

Next

```
<table data-role="table" data-rows-steps="-1, 5, 10, 20, 30, 100" data-rows="5">...</table>
```

Table info

Table info block contains information about rows and current position in table. To enable this feature, use attribute `data-show-table-info="true"` (enabled by default).

You can change `template` for table info with attribute `data-table-info-title` . By default value for this attribute is `Showing $1 to $2 of $3 entries` . Where `$1` - from records, `$2` - to records, `$3` - total records.

Search:

Show entries: 5

ID	1	Name	Start	Age	Salary
1		Aidan	31-12-2017	63	\$7,843
2		Ferris	13-06-2018	28	\$8,877
3		Joseph	06-02-2018	43	\$5,645
4		Troy	11-03-2019	26	\$5,405
5		Kennedy	11-02-2018	60	\$5,780

Display from 1 to 5 of 100 records

Prev

1

2

3

4

5

...

20

Next

```
<table class="table" data-role="table" data-table-info-title="Display from $1 to $2 of $3 records">...</table>
```

Table inspector

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

Open Inspector

Search:

Show entries: 5

ID	Name	Start	Age	Salary
1	Aidan	31-12-2017	63	\$7,843
2	Ferris	13-06-2018	28	\$8,877
3	Joseph	06-02-2018	43	\$5,645
4	Troy	11-03-2019	26	\$5,405
5	Kennedy	11-02-2018	60	\$5,780

Showing 1 to 5 of 100 entries

Prev

1

2

3

4

5

...

20

Next

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

```
<!-- $1 in save path will be replaced with table id -->
<table class="table" data-role="table"
      data-view-save-mode="client"
      data-view-save-path="TABLE:$1:KEYS"
>
</table>
```

Server

```
<table class="table" data-role="table"
      data-view-save-mode="server"
      data-view-save-path="//server.com/save_view.php"
>
</table>
```

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', {
  csvDelimiter: "\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`.

Search:

Show entries: 10

ID	Name	Start	Age	Salary
1	Aidan	31-12-2017	63	\$7,843
2	Ferris	13-06-2018	28	\$8,877
3	Joseph	06-02-2018	43	\$5,645
4	Troy	11-03-2019	26	\$5,405
5	Kennedy	11-02-2018	60	\$5,780
6	Cullen	23-09-2017	62	\$5,482
7	Eric	04-01-2019	53	\$6,490
8	Jakeem	29-06-2018	18	\$6,084
9	Kieran	11-04-2019	18	\$5,043
10	Stephen	11-04-2019	54	\$5,895

Showing 1 to 10 of 100 entries

Prev

1

2

3

4

5

...

10

Next

```
<table class="table striped table-border mt-4" data-role="table" data-source="data/table.json">
  >
</table>
```

External data format

To use external source, you must create JSON data in specific format. Example of [JSON file](#).

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

Header

In `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": [
  {
    "name": "id",
    "title": "ID",
    "size": 50,
    "sortable": true,
    "sortDir": "asc",
    "format": "number"
  },
  {
    "name": "name",
    "title": "Name",
    "sortable": true
  },
  {
    "name": "start",
    "title": "Start",
    "sortable": true,
    "size": 150,
    "format": "date",
    "formatMask": "dd-mm-yyyy"
  },
  {
    "name": "age",
    "title": "Age",
    "sortable": true,
    "size": 80
  },
  {
    "name": "salary",
    "title": "Salary",
    "sortable": true,
    "size": 150,
    "format": "money",
    "show": true
  }
],7
```

```
}
}
```

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
<code>onDraw</code>	<code>data-on-draw</code>	Event fired when the table was drawn.
<code>onDrawRow</code>	<code>data-on-draw-row</code>	Event fired when draw row.
<code>onDrawCell</code>	<code>data-on-draw-cell</code>	Event fired when draw cell.
<code>onAppendRow</code>	<code>data-on-append-row</code>	Event fired when row added to table.
<code>onAppendCell</code>	<code>data-on-append-cell</code>	Event fired when cell added to row.
<code>onSortStart</code>	<code>data-on-sort-start</code>	Event fired when sorting process stated.
<code>onSortStop</code>	<code>data-on-sort-stop</code>	Event fired when sorting process stopped.
<code>onSortItemSwitch</code>	<code>data-on-sort-item-switch</code>	Event fired when table rows switched on sorting process.
<code>onSearch(filter, items)</code>	<code>data-on-search</code>	Event fired on search.
<code>onRowCountChange(val)</code>	<code>data-on-rows-count-change</code>	Event fired when user change rows count to display.
<code>onDataLoad(source)</code>	<code>data-on-data-load</code>	Event fired before data will be loaded.
<code>onDataLoaded(source, response)</code>	<code>data-on-data-loaded</code>	Event fired after data loaded.
<code>onDataLoadError(source, xhr, textStatus, errorThrown)</code>	<code>data-on-data-load-error</code>	Event fired when data loads stop with error.
<code>onFilterRowAccepted(row)</code>	<code>data-on-filter-row-accepted</code>	Row accepted by filters.
<code>onFilterRowDeclined(row)</code>	<code>data-on-filter-row-declined</code>	Row declined by filters.
<code>onCheckClick()</code>	<code>data-on-check-click</code>	
<code>onCheckClickAll()</code>	<code>data-on-check-click-all</code>	
<code>onCheckDraw()</code>	<code>data-on-check-draw</code>	
<code>onViewSave()</code>	<code>data-on-view-save</code>	
<code>onViewGet()</code>	<code>data-on-view-get</code>	
<code>onTableCreate(el)</code>	<code>data-on-table-create</code>	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)
deleteItem(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 `search`, `rows counter`, `table info` and `pagination` elements. To set it, use special named attributes: `data-search-wrapper`, `data-rows-wrapper`, `data-info-wrapper`, `data-pagination-wrapper`.

Search:

Show entries:5

Showing 1 to 5 of 100 entries

Prev

1

2

3

4

5

...

20

Next

ID ↑	Name	Start	Age	Salary
1	Aidan	31-12-2017	63	\$7,843
2	Ferris	13-06-2018	28	\$8,877
3	Joseph	06-02-2018	43	\$5,645
4	Troy	11-03-2019	26	\$5,405
5	Kennedy	11-02-2018	60	\$5,780

<div class="row mb-2">

```
<div class="cell-md-8 my-search-wrapper"></div>
<div class="cell-md-4 my-rows-wrapper"></div>
</div>
<p class="h3 text-center my-info-wrapper"></p>
<div class="d-flex flex-justify-center my-pagination-wrapper"></div>
<table class="table table-border striped"
  data-role="table"
  data-rows-steps="5, 10"
  data-rows="5"
  data-info-wrapper=".my-info-wrapper"
  data-pagination-wrapper=".my-pagination-wrapper"
  data-search-wrapper=".my-search-wrapper"
  data-rows-wrapper=".my-rows-wrapper"
  data-source="data/table.json">
</table>
<div class="d-flex flex-justify-center my-pagination-wrapper"></div>
```

Horizontal scroll

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

Search:

Show entries: 5

ID	Name	Age	Email	Phone	Country	City	Zip	Address	Credit c
1	Owen H. Burris	51	mollis.dui.in@nonsapientmolestie.org	(137) 281-9285	Maldives	Burnie	21277	746-6581 Ut Av.	4929 587160
2	Kane G. Craft	61	dolor.Nulla.semper@non.com	(440) 131-9263	El Salvador	Paisley	33222-441	583-7014 Lectus, Road	4024007
3	Ann Barnett	65	id.erat@egestasblanditNam.org	(936) 755-5080	Sierra Leone	Gorinchem	2237 KB	7117 Non Rd.	4996748
4	Zeph Ballard	28	Morbi.accumsan@arcu.ca	(716) 948-5721	Uruguay	Robelmont	61940	P.O. Box 698, 1109 Elit, Av.	448524685676
5	Kyle Moreno	52	Donec.feugiat@Duis.edu	(909) 900-6539	Jersey	Terragnolo	20805	962-7900 Nonummy Street	453 967865

Showing 1 to 5 of 100 entries

Prev

1

2

3

4

5

...

20

Next

<table class="table table-border striped" data-role="table" data-rows-steps="-1, 5, 10" data-rows="5" data-horizontal-scroll="true" data-source="data/table-wide.json"></table>

You can stop horizontal scrolling with attribute `data-horizontal-scroll-stop` .

Search:

Show entries: 5

ID	Name	Age	Email	Phone	Country	City	Zip	Address	Credit c
1	Owen H. Burris	51	mollis.dui.in@nonsapientmolestie.org	(137) 281-9285	Maldives	Burnie	21277	746-6581 Ut Av.	4929 587160
2	Kane G. Craft	61	dolor.Nulla.semper@non.com	(440) 131-9263	El Salvador	Paisley	33222-441	583-7014 Lectus, Road	4024007
3	Ann Barnett	65	id.erat@egestasblanditNam.org	(936) 755-5080	Sierra Leone	Gorinchem	2237 KB	7117 Non Rd.	4996748
4	Zeph Ballard	28	Morbi.accumsan@arcu.ca	(716) 948-5721	Uruguay	Robelmont	61940	P.O. Box 698, 1109 Elit, Av.	448524685676
5	Kyle Moreno	52	Donec.feugiat@Duis.edu	(909) 900-6539	Jersey	Terragnolo	20805	962-7900 Nonummy Street	453 967865

Showing 1 to 5 of 100 entries

Prev

1

2

3

4

5

...

20

Next

<table class="table table-border striped" data-role="table" data-rows-steps="-1, 5, 10" data-rows="5"

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