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

Media
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

Table of contents

List

Create a list with sorting, filtering, searching and pagination features.

About

New in 4.2.8

If you need to add sorting, filtering, pagination and searching features to the list, you can use `list` component. To create this `list`, you must add attribute `data-role="list"` to your element whose sub-elements need to be sorted and set required additional options with special attributes.

Sort by

Price ↑ Price ↓ Author ↑ Author ↓ Name ↑ Name ↓

Paints By:

When is the wedding
Gogen
$ 300M

Players in cards
Sezann
$ 250M

Three Studies of Lucian Freud
Bacon
$ 142M

Woman III
Kuning
$ 137M

```
<script>
function sortList(col, dir) {
    $('#paintings').data('list').sorting(col,dir,true)
}
</script>

<p class="text-center h3 text-light">Sort by</p>
<div class="d-flex flex-justify-center flex-wrap m-2">
    <button class="button m-1" onclick="sortList('painting-price','asc')">Price <span class="mif-arrow-up"></span></button>
    <button class="button m-1" onclick="sortList('painting-price','desc')">Price <span class="mif-arrow-down"></span></button>
    <button class="button m-1" onclick="sortList('painting-author','asc')">Author <span class="mif-arrow-up"></span></button>
    <button class="button m-1" onclick="sortList('painting-author','desc')">Author <span class="mif-arrow-down"></span></button>
    <button class="button m-1" onclick="sortList('painting-name','asc')">Name <span class="mif-arrow-up"></span></button>
    <button class="button m-1" onclick="sortList('painting-name','desc')">Name <span class="mif-arrow-down"></span></button>
    <div class="cell-md-4 mt-4">
        <select data-role="select" data-filter="false" data-prepend="Paints By:" data-on-change="$('#paintings').data('list').filter(this.value)">
            <option value="" selected>All</option>
            <option value="gogen">Gogen</option>
            <option value="sezann">Sezann</option>
            <option value="bacon">Bacon</option>
            <option value="klimt">Klimt</option>
            <option value="munk">Munk</option>
            <option value="picasso">Picasso</option>
        </select>
    </div>
</div>

<ul id="paintings"
    data-role="list"
    data-sort-class="painting-price"
    data-sort-dir="desc"
    data-cls-list="unstyled-list row flex-justify-center mt-4"
    data-cls-list-item="cell-sm-6 cell-md-4"
>
    <li>
        <figure class="text-center">
            <div class="img-container thumbnail">
                <img src="images/paintings/gogen.jpg" alt="Gogen, When is the wedding">
            </div>
            <figcaption class="painting-name">When is the wedding</figcaption>
            <figcaption class="painting-author text-bold">Gogen</figcaption>
            <figcaption class="painting-price" data-format="money">$ 300M</figcaption>
        </figure>
    </li>
    <li>
        <figure class="text-center">
            <div class="img-container thumbnail">
                <img src="images/paintings/sezann.jpg" alt="Sezann, Players in cards">
            </div>
            <figcaption class="painting-name">Players in cards</figcaption>
            <figcaption class="painting-author text-bold">Sezann</figcaption>
            <figcaption class="painting-price" data-format="money">$ 250M</figcaption>
        </figure>
    </li>
    <li>
        <figure class="text-center">
            <div class="img-container thumbnail">
                <img src="images/paintings/kuning.jpg" alt="Kunning, Woman III">
            </div>
            <figcaption class="painting-name">Woman III</figcaption>
            <figcaption class="painting-author text-bold">Bacon</figcaption>
            <figcaption class="painting-price" data-format="money">$ 137M</figcaption>
        </figure>
    </li>
    <li>
        <figure class="text-center">
            <div class="img-container thumbnail">
                <img src="images/paintings/klimt.jpg" alt="Klimt, Gold Adele">
            </div>
            <figcaption class="painting-name">Gold Adele</figcaption>
            <figcaption class="painting-author text-bold">Klimt</figcaption>
            <figcaption class="painting-price" data-format="money">$ 135M</figcaption>
        </figure>
    </li>
    <li>
        <figure class="text-center">
            <div class="img-container thumbnail">
                <img src="images/paintings/munk.jpg" alt="Munk, The Scream">
            </div>
            <figcaption class="painting-name">The Scream</figcaption>
            <figcaption class="painting-author text-bold">Munk</figcaption>
            <figcaption class="painting-price" data-format="money">$ 119M</figcaption>
        </figure>
    </li>
    <li>
        <figure class="text-center">
            <div class="img-container thumbnail">
                <img src="images/paintings/picasso.jpg" alt="Picasso, Nude, green leaves and bust">
            </div>
            <figcaption class="painting-name">Nude, green leaves and bust</figcaption>
            <figcaption class="painting-author text-bold">Picasso</figcaption>
            <figcaption class="painting-price" data-format="money">$ 106M</figcaption>
        </figure>
    </li>
</ul>
```

Create list

To create interactive list, add attribute `data-role="list"` to your html list element.

```
<ul data-role="list">
    ...
</ul>
```

External data

You can specify a external data for your list. External data must have a valid specified json structure. To set it, use attribute `data-source`.

```
<ul data-role="list" data-set="data/paintings.json"></ul>
```

External data structure

External data must have `header` block and block with `data`. Header must contains template for list items. Link to example json file.

```
{
  "template": "..."
  "data": [...]
}
```

Sorting

By default list sorter use `full inner text` of list item (tags are cleared) to sorting items. You can `clarify` this with set sub-element `class name` with attribute `data-sort-class="..."`.

To set sort direction use attribute `data-sort-dir="..."` . This attribute must be set into one of two values: `asc` or `desc` . By default, this attribute have a `asc` as value.

```
<ul data-role="list">
    <li><span class="num">3</span> item</li>
    <li><span class="num">1</span> item</li>
    <li><span class="num">2</span> item</li>
</ul>
```

Clarify

```
<ul data-role="list" data-sort-class="num">
    <li><span class="num">3</span> item</li>
    <li><span class="num">1</span> item</li>
    <li><span class="num">2</span> item</li>
</ul>
```

For right sorting specified fields (ex: formatted money, date), you can add attribute `data-format` to sorted items. You can use next formats to a field: `number` , `int` , `float` , `date` and `money` .

> This feature works in pair with a `data-sort-class="..."` attribute.

```
<ul id="paintings"
    data-role="list"
    data-sort-class="painting-price"
    data-sort-dir="desc"
>
    <li>
        <figure class="text-center">
            <div class="img-container thumbnail">
                <img src="images/paintings/gogen.jpg" alt="Gogen, When is the wedding">
            </div>
            <figcaption class="painting-name">When is the wedding</figcaption>
            <figcaption class="painting-author text-bold">Gogen</figcaption>
            <figcaption class="painting-price" data-format="money">$ 300M</figcaption>
        </figure>
    </li>
    ...
</ul>
```

You can change attributes `data-sort-class` and `data-sort-dir` at runtime to change list sorting.

Filtering

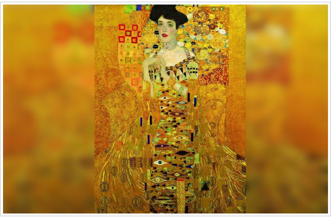
When you are use `list` component, you can set filters on list items. You can specify filtering functions when defining a component or at runtime, using special methods.

Specify filters at defining component

To set filters at defining component, you must use attributes `data-filter` and `data-filters` . Attribute `data-filter` set one filter, attribute `data-filters` set the number of filters. Value for the `data-filter` must be valid function name or javascript code, and for `data-filters` - comma separated functions name list.

Each filter receive a single parameter - `list HTML element` . Filter function must return `true` or `false` .

Three Studies of Lucian Freud
Bacon
$ 142M



Gold Adele
Klimt
$ 135M

```
<ul data-role="list"
    data-sort-class="painting-price"
    data-filter-class="painting-author"
    data-sort-dir="desc"
    data-cls-list="unstyled-list row flex-justify-center mt-4"
    data-cls-list-item="cell-sm-6 cell-md-4"
    data-source="data/paintings.json"
    data-filters="authorFilter, moneyFilter"
></ul>
<script>
    function authorFilter(item){
        var author = item.getElementsByClassName('painting-author')[0].textContent;
        return author.indexOf("Klimt") > -1 || author.indexOf("Bacon") > -1
    }
    function moneyFilter(item){
        return Metro.utils.parseMoney(item.getElementsByClassName('painting-price')[0].textContent) > 120;
    }
</script>
```

Searching

The `list` component support searching on the list items. The `searching` is a special case of filtration. To enable this function, you must set attribute `showSearch` to `true` (this case enable built in search input) and, if your need, set your one custom search wrapper with attribute `data-search-wrapper` .

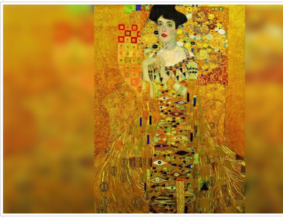
Built in search

Search:

When is the wedding
Gogen
$ 300M

Players in cards
Sezann
$ 250M

Three Studies of Lucian Freud
Bacon
$ 142M



Gold Adele
Klimt
$ 135M

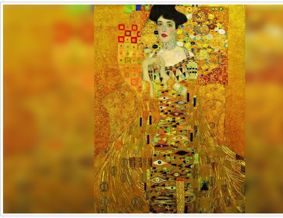
```
<ul data-role="list"
    data-sort-class="painting-price"
    data-filter-class="painting-author"
    data-sort-dir="desc"
    data-cls-list="unstyled-list row flex-justify-center mt-4"
    data-cls-list-item="cell-sm-6 cell-md-4"
    data-source="data/paintings.json"
    data-show-search="true"
></ul>
```

Your one search wrapper

When is the wedding
Gogen
$ 300M

Players in cards
Sezann
$ 250M

Three Studies of Lucian Freud
Bacon
$ 142M



Gold Adele
Klimt
$ 135M

Search:

```
<ul data-role="list"
    data-sort-class="painting-price"
    data-filter-class="painting-author"
    data-sort-dir="desc"
    data-cls-list="unstyled-list row flex-justify-center mt-4"
    data-cls-list-item="cell-sm-6 cell-md-4"
    data-source="data/paintings.json"
    data-search-wrapper=".my-search-wrapper"
    data-show-search="true"
></ul>
<div class="my-search-wrapper"></div>
```

Pagination

The `list` component also support `pagination` to display part of items at one time. To enable this function, you must set attribute `showPagination` to `true` (this case enable built pagination) and, if your need, set your one custom pagination wrapper with attribute `data-pagination-wrapper` . Also you can enable `item count block` with attribute `data-items-steps="true"` And you can enable `list items count info block` with attribute `data-show-info="true"` .

Show entries: 3

When is the wedding

Gogen

$ 300M

Players in cards

Sezann

$ 250M

Three Studies of Lucian Freud

Bacon

$ 142M

Showing 1 to 3 of 7 entries

Prev 1 2 3 Next

```
<ul data-role="list"
    data-sort-class="painting-price"
    data-filter-class="painting-author"
    data-sort-dir="desc"
    data-cls-list="unstyled-list row flex-justify-center mt-4"
    data-cls-list-item="cell-sm-6 cell-md-4"
    data-source="data/paintings.json"
    data-show-pagination="true"
    data-items="3"
    data-items-steps="all, 3, 5"
    data-show-items-steps="true"
    data-show-list-info="true"
></ul>
```

Options

You can setup behavior of your list with any options. To these options you can use special attributes.

Option	Data-*	Default	Desc
`sortClass`	`data-sort-class`		Class for sort source element. If you set this attribute, the sorter will use element with this class for sorting
`sortDir`	`data-sort-dir`		Direction for sorting. Must be `asc` or `desc`
`sortInitial`	`data-sort-initial`		When you set this attribute to true, you can reset list to initial order
`filterClass`	`data-filter-class`		Class for search source element. If you set this attribute, the searcher will use element with this class for searching
`filter`	`data-filter`		Initial filter function.
`filters`	`data-filters`		Initial filters function.
`filterString`	`data-filter-string`		Initial search value.
`source`	`data-source`		Define external data for list.
`showItemsSteps`	`data-show-items-steps`	false	Show items steps counter. Must be true or false.
`showSearch`	`data-show-search`	false	Show search input. Must be true or false.
`showListInfo`	`data-show-list-info`	false	Show list info block. Must be true or false.
`showPagination`	`data-show-pagination`	false	Show pagination. Must be true or false.
`showAllPages`	`data-show-all-pages`	false	Show all links to all pages in pagination.
`showActivity`	`data-show-activity`	true	Show activity for long operations. Must be true or false.
`muteList`	`data-muteList`	true	Mute list in the time of long operations. Must be true or false.
`items`	`data-items`	-1	How many items must be showed. -1 - all items. For pagination.
`itemsSteps`	`data-items-steps`	all, 10,25,50,100	Steps for items count to displayed. For pagination.
`itemsAllTitle`	`data-items-all-title`	Show all items	Title for all items step.
`listItemsCountTitle`	`data-list-items-count-title`	Show entries:	Title for items counter.
`listSearchTitle`	`data-list-search-title`	Search:	Title for search input.
`listInfoTitle`	`data-list-info-title`	Showing $1 to $2 of $3 entries	Title for info block.
`paginationPrevTitle`	`data-pagination-prev-title`	Prev	Title pagination prev button.
`paginationNextTitle`	`data-pagination-next-title`	Next	Title pagination next button.
`activityType`	`data-activity-type`	cycle	Activity type. More about activity
`activityStyle`	`data-activity-style`	color	Activity style.
`searchWrapper`	`data-search-wrapper`		Selector for search input wrapper.
`rowsWrapper`	`data-rows-wrapper`		Selector for steps select wrapper.
`infoWrapper`	`data-info-wrapper`		Selector for list info wrapper.
`paginationWrapper`	`data-pagination-wrapper`		Selector for list pagination block wrapper.

Events

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

Option	Data-*	Desc
`onDraw(el)`	`data-on-draw`	Event fired when the list was drawn.
`onDrawItem(item)`	`data-on-draw-item`	Event fired when the list item was drawn.
`onSortStart(items)`	`data-on-sort-start`	Event fired before sort started.
`onSortStop(items)`	`data-on-sort-stop`	Event fired after sort.
`onSortItemSwitch(a, b, result)`	`data-on-sort-item-switch`	Event fired when two item switched.
`onSearch(filter, items)`	`data-on-search`	Event fired when user use searching.
`onRowsCountChange(val)`	`data-on-rows-count-change`	Event fired when user change items count to display.
`onDataLoad(res)`	`data-on-data-load`	Event fired before external data loads.
`onDataLoaded(res, data)`	`data-on-data-loaded`	Event fired after external data was loaded.
`onFilterItemAccepted(item)`	`data-on-filter-item-accepted`	Event fired filter return true for item.
`onFilterItemDeclined(item)`	`data-on-filter-item-declined`	Event fired filter return false for item.
`onListCreate(el)`	`data-on-list-create`	Event fired after the list created.

Methods

You can use list methods to interact with it.

- **draw()** - redraw list
- **sorting(sortClass, sortDir, redraw)** - sorting list
- **filter(val)** - programming search on list items

- **loadData(res)** - load items from external json
- **next()** - goto next page
- **prev()** - goto prev page
- **first()** - goto first page
- **last()** - goto last page
- **page(n)** - goto specific page
- **addFilter(func)** - add filter function
- **removeFilter(func_index)** - remove filter function
- **removeFilters()** - remove all filters
- **getFilters()** - get filters
- **getFilterIndex()** - get index for filter who set with attribute `data-filter`
- **getFiltersIndexes()** - get indexes for filters who set with attribute `data-filters`

Load data

```
<div class="mb-4 text-center">
    <button class="button"
        onclick="$('#list-load-data').data('list').loadData('data/paintings.json')">
        Load data
    </button>
</div>

<ul id="list-load-data"
    data-role="list"
    data-sort-class="painting-price"
    data-sort-dir="asc"
    data-cls-list="unstyled-list row flex-justify-center mt-4"
    data-cls-list-item="cell-sm-6 cell-md-4"
></ul>
```

Customize list

You can customize style of list and him elements with next options:

Option	`Data-*`	Desc
clsComponent	`data-cls-component`	Additional class for component