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

Toast

Toasts are used for system messaging. They also display at the bottom of the screen, but may not be swiped off-screen. Metro 4 provides simple methods to create toasts.

Setup toast

You can setup options for toasts with function `init`. The following options are available:

Option	Default	Desc
`callback`	Metro.noop	Callback for execute code after toast showing
`timeout`	METRO_TIMEOUT	Time for toast shown, by default 2000
`distance`	20	Distance from window side (top or bottom) in pixels, by default 20px
`showTop`	false	When true, toast show on top of window
`clsToast`		Additional classes for toast

```
Metro.toast.init({
    showTop: true,
    distance: 60
}).create("This is a toast");
```

Create toast

To create `toast` you must call method `Metro.toast.create`. Method contains next parameters:

- **message**: Toast message
- **callback**: Callback function.Executed after toast hided
- **timeout**: Time to show toast
- **cls**: Classes to customize toast

```
Metro.toast.create(message, callback, timeout, cls);
```

Quick example

Default toast | Toast with a callback | Toast timeout | Toast custom class

```
<button class="button primary" onclick="runToast()">Default toast</button>
<button class="button secondary" onclick="runToast('callback')">Toast with a callback</button>
<button class="button info" onclick="runToast('timeout')">Toast timeout</button>
<button class="button success" onclick="runToast('class')">Toast custom class</button>

<script>
    function runToast(mode) {
        var toast = Metro.toast.create;
        switch (mode) {
            case 'callback': toast("This is a toast with callback", function(){
                alert('Toast callback executed!');
            }); break;
            case 'timeout': toast("This is a toast with timeout 5s", null, 5000); break;
            case 'class': toast("This is a toast with custom class", null, 5000, "bg-green fg-white"); break;
            default: toast("This is default toast");
        }
    }
</script>
```

Callback

If you need execute code after toast showing, add `callback function` to call toast.

```
Metro.toast.create("Toast message", function(){
    ...callback function...
});
```

Timeout

Want to show the toast a certain time? Add a `timeout` parameter.

```
Metro.toast.create("Toast message", null, 5000);
```

Predefined classes

New in 4.2.10

You can use predefined classes for your `toasts`. This list of classes include: `.primary`, `.secondary`, `.success`, `.alert`, `.warning`, `.yellow`, `.info` and `.light` classes.

Primary | Secondary | Success | Alert | Warning | Yellow | Info | Light

```
Metro.toast.create("Toast message", null, null, "info");
```

Custom toast

If you want to display a toast in your own style, add a `cls` parameter.

```
Metro.toast.create("Toast message", null, null, cls);
```

Additional options

You can sets additional options. Options is a object with next properties:

```
options: {
    callback: Metro.noop,
    timeout: METRO_TIMEOUT,
    distance: 20,
    showTop: false,
    clsToast: ""
}
```

Show on top

```
var options = {
    showTop: true,
    timeout: 3000
}
Metro.toast.create("Toast message", null, null, null, options);
```