gdMessages

# Introduction

## Installation / Loading the Code

You must first get gdMessages’ code loaded onto the page before using it to display a message.

First, download gdMessages’ code from the downloads page. After unzipping, find the files gdMessages.js and gdMessages.css.

Then, either download the JavaScript files for gdMessages’ dependencies: jQuery, FontAwesome and Bootstrap v4.

Then, write something like this in the <head> of your page:

<link rel='stylesheet' href='css/bootstrap.min.css’ />

<link rel='stylesheet' href='css/fontawesome.min.css’ />

<link rel='stylesheet' href='css/gdMessages.css' />

<script src='lib/js/jquery.min.js'></script>

<script src="js/gdMessages.js"></script>

Or use CDN links such as these:

<link rel="stylesheet" href=<https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css> integrity="sha384‑Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<link rel="stylesheet" href=<https://use.fontawesome.com/releases/v5.3.1/css/all.css> integrity="sha384‑mzrmE5qonljUremFsqc01SB46JvROS7bZs3IO2EmfFsd15uHvIt+Y8vEf7N7fWAU" crossorigin="anonymous" />

<link rel='stylesheet' href='css/gdMessages.css' />

<script src=[https://code.jquery.com/jquery‑3.3.1.js](https://code.jquery.com/jquery3.3.1.js) integrity="sha256‑2Kok7MbOyxpgUVvAk/HJ2jigOSYS2auK4Pfzbm7uH60=" crossorigin="anonymous"></script>

<script src="js/gdMessages.js"></script>

# Message Types

## Notifications

### Options

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Default** | **Description** |
| title | {string} | null | Sets a title for the notification. Refer to className below. |
| msg | {string} | null | Sets the notification message. |
| animation | {boolean} | true | Determines if the display of the notification will be animated. |
| className | {‘info’ || ‘success’ || ‘warning’ || ‘error’} | info | Sets the “title” to {‘Information’ || ‘Success!’ || ‘Warning!’ || ‘Error!’} if one has not been supplied.  Also determines the class to be used to format the notification message and select the icon to be displayed. |
| width | {‘auto’ || px || %} | ‘300px’ | Sets the width of the notification. |
| position | {‘top right’ ||  ‘top left’ || ‘top centre’ || ‘bottom right’ || ‘bottom left’ ||  ‘bottom centre’ || ‘centre’} | ‘bottom left’ | Sets the notification display position. |
| closeAfter | {object} || boolean | {object} | Set to false if you want the notification to stay open until it is clicked |
| closeAfter.time | {number} | 3 | Sets the number of seconds after which the notification will close. Minimum = 1. |
| closeAfter.resetOnHover | {Boolean} | true | Resets the time when the cursor is over the notification. |
| afterClose | {function} | null | Provides a function to be called after the notification is closed. |

### Sample Usage

#### Minimal

gdMessages.notify({

msg: 'Record has been updated',

className: 'success'

});

#### Common

gdMessages.notify({

msg: 'Record has been updated',

className: 'info',

width: "auto",

position: 'bottom centre'

});

## Confirmations

### Options

| **Name** | **Type** | **Default** | **Description** |
| --- | --- | --- | --- |
| msg | {string} | ‘Click OK to proceed’ | Sets the confirmation message. |
| animation | {boolean} | true | Determines if the display of the confirmation will be animated. |
| className | {string} | ‘stdConfirm’ | Defines the class to be used to format the confirmation message. |
| width | {‘auto’ || px || %} | ‘400px’ | Sets the width of the confirmation. |
| position | {‘top right’ ||  ‘top left’ || ‘top centre’ || ‘bottom right’ || ‘bottom left’ ||  ‘bottom centre’ || ‘centre’} | ‘bottom left’ | Sets the confirmation display position. |
| backdrop | {boolean} | true | Determines if a backdrop should be displayed making the confirmation box modal. |
| closeAfter | {object} || {boolean} | {object} | Set to false if you do not want the confirmation to automatically close. |
| closeAfter.time | {number} | 10 | Sets the number of seconds after which the confirmation will close. |
| closeAfter.resetOnHover | {Boolean} | true | Resets the time when the cursor is over the confirmation. |
| okBtn | {object} |  |  |
| okBtn.className | {string} | 'btn btn-primary' | Sets the class name(s) to be used for the OK button. |
| okBtn.label | {string} | ‘Ok’ | Sets the label of the OK button |
| cancelBtn | {object} |  |  |
| cancelBtn.className | {string} | 'btn btn-danger' | Sets the class name(s) to be used for the Cancel button. |
| cancelBtn.label | {string} | ‘Cancel’ | Sets the label of the Cancel button |
| otherBtn | {object} || {boolean} | false | Sets an optional user defined button. If ‘false’, a button will not be displayed. It provides a method for alternate post-processing when closing the confirmation. |
| otherBtn.className | {string} | 'btn btn-warning’ | Sets the class name(s) to be used for a user defined button. |
| otherBtn.label | {string} | ‘Other’ | Sets the label of the user defined button |
| beforeClose | {function} | null | Callback function that executes just before the confirmation closes if ‘Ok’ is clicked. This provides the ability to validate data before the confirmation is closed. If you return false you will cancel the ‘Ok’ action. |
| afterClose | {function} | null | Callback function that executes just after the confirmation closes. |

### Sample Usage

#### Minimal

gdMessages.confirm();

#### Common

gdMessages.confirm({

msg: 'Record has been updated',

className: 'stdconfirm',

width: "auto",

position: 'centre',

closeAfter: false,

afterClose: afterCloseFunction

});

Dialogs

### Options

| **Name** | **Type** | **Default** | **Description** |
| --- | --- | --- | --- |
| title | {string} | null | Sets the dialog title. |
| msg | {string} | null | Defines the body of the message. |
| returnSelector | {string} | null | Sets the jQuery selector to be used to return the contents of the msg If required in the beforeClose function.  e.g. “#msgID” |
| animation | {boolean} | true | Determines if the display of the confirmation will be animated. |
| className | {string} | ‘stddialog’ | Defines the class to be used to format the dialog message. |
| width | {‘auto’ || px || %} | ‘400px’ | Sets the width of the dialog. |
| position | {‘top right’ ||  ‘top left’ || ‘top centre’ || ‘bottom right’ || ‘bottom left’ ||  ‘bottom centre’ || ‘centre’} | ‘bottom left’ | Sets the dialog display position. |
| backdrop | {boolean} | true | Determines if a backdrop should be displayed making the dialog box modal. |
| closeAfter | {object} || {boolean} | false | Set the ‘time’ and ‘resetOnHover’ variables if you want the dialog to automatically close. |
| closeAfter.time | {number} | 10 | Sets the number of seconds after which the confirmation will close. |
| closeAfter.resetOnHover | {Boolean} | true | Resets the time when the cursor is over the confirmation. |
| okBtn | {object} |  |  |
| okBtn.className | {string} | 'btn btn-primary' | Sets the class name(s) to be used for the OK button. |
| okBtn.label | {string} | ‘Ok’ | Sets the label of the OK button |
| cancelBtn | {object} |  |  |
| cancelBtn.className | {string} | 'btn btn-danger' | Sets the class name(s) to be used for the Cancel button. |
| cancelBtn.label | {string} | ‘Cancel’ | Sets the label of the Cancel button |
| otherBtn | {object} || {boolean} | false | Sets an optional user defined button. If ‘false’, a button will not be displayed. It provides a method for alternate post-processing when closing the dialog. |
| otherBtn.className | {string} | 'btn btn-warning’ | Sets the class name(s) to be used for a user defined button. |
| otherBtn.label | {string} | ‘Other’ | Sets the label of the user defined button |
| beforeClose | {function} | null | Callback function that executes just before the confirmation closes if ‘Ok’ is clicked. This provides the ability to validate data before the dialog is closed. If you return false you will cancel the ‘Ok’ action. |
| afterClose | {function} | null | Callback function that executes just after the dialog closes. |

### Sample Usage

#### Minimal

gdMessages.dialog();

#### Common

gdMessages.dialog({

msg: 'Record has been updated',

className: 'stdconfirm',

width: "auto",

position: 'centre',

closeAfter: false,

afterClose: afterCloseProcess

});