

kylefox /
jquery-modal

<> Code

🕒 Issues 54

🔗 Pull requests 17

🎬 Actions

📁 Projects

🛡 Security

📊 Insights

👁

🔗

☆

The simplest possible modal for jQuery

[🔗 jquerymodal.com/](#)

[📄 View license](#)

☆ 2.6k stars 🔗 662 forks 👁 73 watching 🌿 4 Branches 🏷 19 Tags 🔄 Activity

🌐 Public repository

master

4 Branches

19 Tags


Go to file

Go to file

+

Add file

Code



kylefox use cdn in example site

8b8a7ce · last year

<div>examples</div>	use cdn in example site	last year
<div>.gitignore</div>	gulp stuffs	9 years ago
<div>CHANGELOG.md</div>	Bump to 0.9.2	5 years ago
<div>LICENSE</div>	Resolves #30 : Added explicit MIT license	12 years ago
<div>README.md</div>	use cdn in example site	last year
<div>bower.json</div>	Support for jQuery 3.x	8 years ago
<div>close.sketch</div>	retina close icon	7 years ago
<div>compositor.json</div>	Release 0.9.1	7 years ago
<div>gulpfile.js</div>	add minified css	8 years ago
<div>jquery.modal.css</div>	Avoid having to set inline styles to hide modal by...	7 years ago
<div>jquery.modal.js</div>	Bump to 0.9.2	5 years ago
<div>jquery.modal.min.css</div>	Release 0.9.1	7 years ago
<div>jquery.modal.min.js</div>	Bump to 0.9.2	5 years ago
<div>package.json</div>	Bump to 0.9.2	5 years ago

README License

A simple & lightweight method of displaying modal windows with jQuery.

For quick examples and demos, head to jquerymodal.com.

Why another modal plugin?

Most plugins I've found try to do too much, and have specialized ways of handling photo galleries, iframes and video. The resulting HTML & CSS is often bloated and difficult to customize.

By contrast, this plugin handles the two most common scenarios I run into

- displaying an existing DOM element
- loading a page with AJAX

and does so with as little HTML & CSS as possible.

Installation

You can install [jquery-modal](#) with npm:

```
npm install jquery-modal
```

or with [Bower](#):

```
bower install jquery-modal
```

or use the hosted version from [cdnjs](#):

```
<!-- Remember to include jQuery :) -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.0.0/jquery.min.js"></script>

<!-- jQuery Modal -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-modal/0.9.2/jquery.modal.min.js"></script>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/jquery-modal/0.9.2/jquery.modal.min.css" />
```

Using Rails? Check out the [jquery-modal-rails](#) plugin!

jQuery Requirements: As of version 0.3.0, jQuery 1.7 is required. If you're using an earlier version of jQuery you can use the [v.0.2.5 tag](#).

Naming conflict with Bootstrap: Bootstrap's [modal](#) uses the same `$.modal` namespace. If you want to use jquery-modal with Bootstrap, the simplest solution is to manually modify the name of this plugin.

Opening

Method 1: Automatically attaching to links

The simplest approach is to add `rel="modal:open"` to your links and use the `href` attribute to specify what to open in the modal.

Open an existing DOM element by ID:

```
<form id="login-form" class="modal">
  ...
</form>

<a href="#login-form" rel="modal:open">Login</a>
```



Load a remote URL with AJAX:

```
<a href="login.html" rel="modal:open">Login</a>
```



Method 2: Manually

You can manually open a modal by calling the `.modal()` method on the element:

```
<form id="login-form" class="modal">
  ...
</form>
```



```
$('#login-form').modal();
```



You can also invoke `.modal()` directly on links:

```
<a href="#ex5" data-modal>Open a DOM element</a>
<a href="ajax.html" data-modal>Open an AJAX modal</a>
```



```
$( 'a[data-modal]' ).click( function( event ) {
    $( this ).modal();
    return false;
});
```



Compatibility Fallback

You can provide a clean fallback for users who have JavaScript disabled by manually attaching the modal via the `data-modal` attribute. This allows you to write your links pointing to the `href` as normal (fallback) while enabling modals where JavaScript is enabled.

```
<!-- By default link takes user to /login.html -->
<a href="/login.html" data-modal="#login-modal">Login</a>
```



```
<!-- Login modal embedded in page -->
<div id="login-modal" class="modal">
  ...
</div>
```

```
<!-- For browsers with JavaScript, open the modal. -->
<script>
    $(function() {
        $('a[data-modal]').on('click', function() {
            $(this).data('modal').modal();
            return false;
        });
    });
</script>
```

Fade Transitions

By default the overlay & window appear instantaneously, but you can enable a fade effect by specifying the `fadeDuration` option.

```
$( 'a.open-modal' ).click( function( event ) {  
    $( this ).modal({  
        fadeDuration: 250  
    });  
    return false;  
});
```



This will fade in the overlay and modal over 250 milliseconds *simultaneously*. If you want the effect of the overlay appearing *before* the window, you can specify the `fadeDelay` option. This indicates at what point during the overlay transition the window transition should begin.

So if you wanted the window to fade in when the overlay's was 80% finished:

```
$( elm ).modal({  
    fadeDuration: 250,  
    fadeDelay: 0.80  
});
```



Or, if you wanted the window to fade in a few moments after the overlay transition has completely finished:

```
$( elm ).modal({  
    fadeDuration: 250,  
    fadeDelay: 1.5  
});
```



The `fadeDelay` option only applies when opening the modal. When closing the modal, both the modal and the overlay fade out simultaneously according to the `fadeDuration` setting.

Fading is the only supported transition.

Closing

Because there can be only one modal active at a single time, there's no need to select which modal to close:

```
$.modal.close();
```



Similar to how links can be automatically bound to open modals, they can be bound to close modals using `rel="modal:close"`:

```
<a href="#close" rel="modal:close">Close window</a>
```



(Note that modals loaded with AJAX are removed from the DOM when closed).

Checking current state

- Use `$.modal.isActive()` to check if a modal is currently being displayed.
- Use `$.modal.getCurrent()` to retrieve a reference to the currently active modal instance, if any.

Options

These are the supported options and their default values:

```
$.modal.defaults = {  
    closeExisting: true,    // Close existing modals. Set this to false if you need to stack multiple modal instances.  
    escapeClose: true,     // Allows the user to close the modal by pressing `ESC`  
    clickClose: true,      // Allows the user to close the modal by clicking the overlay  
    closeText: 'Close',    // Text content for the close <a> tag.  
    closeClass: '',        // Add additional class(es) to the close <a> tag.  
    showClose: true,       // Shows a (X) icon/link in the top-right corner  
    modalClass: "modal",   // CSS class added to the element being displayed in the modal.  
    blockerClass: "modal", // CSS class added to the overlay (blocker).  
  
    // HTML appended to the default spinner during AJAX requests.  
    spinnerHtml: '<div class="rect1"></div><div class="rect2"></div><div class="rect3"></div><div class="rect4"></div>',  
  
    showSpinner: true,      // Enable/disable the default spinner during AJAX requests.  
    fadeDuration: null,     // Number of milliseconds the fade transition takes (null means no transition)
```



```
fadeDelay: 1.0 // Point during the overlay's fade-in that the modal begins to fade in (.5 = 50%, 1.5 = 150%, etc.)  
};
```

Events

The following events are triggered on the modal element at various points in the open/close cycle (see below for AJAX events).

```
$.modal.BEFORE_BLOCK = 'modal:before-block'; // Fires just before the overlay (blocker) appears.  
$.modal.BLOCK = 'modal:block'; // Fires after the overlay (block) is visible.  
$.modal.BEFORE_OPEN = 'modal:before-open'; // Fires just before the modal opens.  
$.modal.OPEN = 'modal:open'; // Fires after the modal has finished opening.  
$.modal.BEFORE_CLOSE = 'modal:before-close'; // Fires when the modal has been requested to close.  
$.modal.CLOSE = 'modal:close'; // Fires when the modal begins closing (including animations).  
$.modal.AFTER_CLOSE = 'modal:after-close'; // Fires after the modal has fully closed (including animations).
```

The first and only argument passed to these event handlers is the `modal` object, which has four properties:

```
modal.$elm; // Original jQuery object upon which modal() was invoked.  
modal.options; // Options passed to the modal.  
modal.$blocker; // The overlay element.  
modal.$anchor; // Anchor element originating the event.
```

So, you could do something like this:

```
$('#purchase-form').on($.modal.BEFORE_CLOSE, function(event, modal) {  
    clear_shopping_cart();  
});
```

AJAX

Basic support

jQuery Modal uses `$.get` for basic AJAX support. A simple spinner will be displayed by default (if you've included `modal.css`) and will have the class `modal-spinner`. If you've set the `modalClass` option, the spinner will be prefixed with that class name instead.

You can add text or additional HTML to the spinner with the `spinnerHtml` option, or disable the spinner entirely by setting `showSpinner: false`.

The default spinner is from the excellent [SpinKit](#) by [Tobias Ahlin](#) 🍌

Events

The following events are triggered when AJAX modals are requested.

```
$.modal.AJAX_SEND = 'modal:ajax:send';  
$.modal.AJAX_SUCCESS = 'modal:ajax:success';  
$.modal.AJAX_FAIL = 'modal:ajax:fail';  
$.modal.AJAX_COMPLETE = 'modal:ajax:complete';
```

The handlers receive no arguments. The events are triggered on the `<a>` element which initiated the AJAX modal.

More advanced AJAX handling

It's a good idea to provide more robust AJAX handling -- error handling, in particular. Instead of accommodating the myriad [\\$.ajax options](#) jQuery provides, jquery-modal makes it possible to directly modify the AJAX request itself.

Simply bypass the default AJAX handling (i.e.: don't use `rel="modal"`)

```
<a href="ajax.html" rel="ajax:modal">Click me!</a>
```

and make your AJAX request in the link's click handler. Note that you need to manually append the new HTML/modal in the `success` callback:

```
$('#a[rel="ajax:modal"]').click(function(event) {  
    $.ajax({
```

```
url: $(this).attr('href'),

success: function(newHTML, textStatus, jqXHR) {
    $(newHTML).appendTo('body').modal();
},

error: function(jqXHR, textStatus, errorThrown) {
    // Handle AJAX errors
}

// More AJAX customization goes here.

});

return false;
});
```

Note that the AJAX response must be wrapped in a div with class `modal` when using the second (manual) method.

Bugs & Feature Requests

Found a bug? MEH!

Just kidding. Please [create an issue](#) and include a publicly-accessible demonstration of the bug. [Dropbox](#) or [JSFiddle](#) work well for demonstrating reproducible bugs, but you can use anything as long as it's publicly accessible. Your issue is much more likely to be resolved/merged if it includes a fix & pull request.

Have an idea that improves `jquery-modal`? Awesome! Please fork this repository, implement your idea (including documentation, if necessary), and submit a pull request.

I don't use this library as frequently as I used to, so if you want to see a fix/improvement you're best off submitting a pull request. Bugs without a test case and/or improvements without a pull request will be shown no mercy and closed!

Contributing

Maintainers Wanted

maintainers **wanted**

This library became more popular and active than I ever expected, and unfortunately I don't have time to maintain it myself.

If you are interested in helping me maintain this library, please let me know — **I would love your help!**

[Read more about becoming a maintainer »](#)

*I'd especially like people who would be excited about working towards a brand new **jQuery Modal 2.0**. See my [Proposal for jQuery Modal 2.0](#) for more details & discussion.*

How to contribute

I welcome improvements to this plugin, particularly with:

- Performance improvements
- Making the code as concise/efficient as possible
- Bug fixes & browser compatibility

Please fork and send pull requests, or create an [issue](#). Keep in mind the spirit of this plugin is **minimalism** so I'm very picky about adding *new* features.

Tips for development/contributing

- Make sure dependencies are installed: `npm install`
- After modifying `jquery.modal.js` and/or `jquery.modal.css`, you can optionally regenerate the minified files with `gulp min` and `gulp css` respectively.
- Make sure you have updated documentation (`README.md` and/or `examples/index.html`) if necessary. **Pull requests without documentation updates will be rejected.**
- Maintainers should increment version numbers and run `gulp changes` when cutting a new release.

- Maintainers should increment version numbers and run `gulp changelog` when cutting a new release.

Support

Please post a question on [StackOverflow](#). Commercial support by email is also available — please contact kylefox@gmail.com for rates. Unfortunately I am unable to provide free email support.

License (MIT)

jQuery Modal is distributed under the [MIT License](<http://opensource.org/licenses/mit-license.php>):

Copyright (c) 2012 Kyle Fox



Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Releases

🏷 19 tags

Packages

No packages published

Used by 3k



Contributors 28



[+ 14 contributors](#)

Deployments 4

✅ [github-pages](#) 7 years ago

[+ 3 deployments](#)

Languages

JavaScript 70.4% CSS 29.6%