Modal

View on GitHub

Use Bootstrap's JavaScript modal plugin to add dialogs to your site for lightboxes, user notifications, or completely custom content.

How it works

Before getting started with Bootstrap's modal component, be sure to read the following as our menu options have recently changed.

- Modals are built with HTML, CSS, and JavaScript. They're positioned over everything else in the document and remove scroll from the `<body>` so that modal content scrolls instead.
- Clicking on the modal "backdrop" will automatically close the modal.
- Bootstrap only supports one modal window at a time. Nested modals aren't supported as we believe them to be poor user experiences.
- Modals use `position: fixed`, which can sometimes be a bit particular about its rendering. Whenever possible, place your modal HTML in a top-level position to avoid potential interference from other elements. You'll likely run into issues when nesting a `.modal` within another fixed element.
- Once again, due to `position: fixed`, there are some caveats with using modals on mobile devices. See our browser support docs for details.
- Due to how HTML5 defines its semantics, the `autofocus` HTML attribute has no effect in Bootstrap modals. To achieve the same effect, use some custom JavaScript:

Copy

```
$('#myModal').on('shown.bs.modal', function () {
  $('#myInput').trigger('focus')
})
```

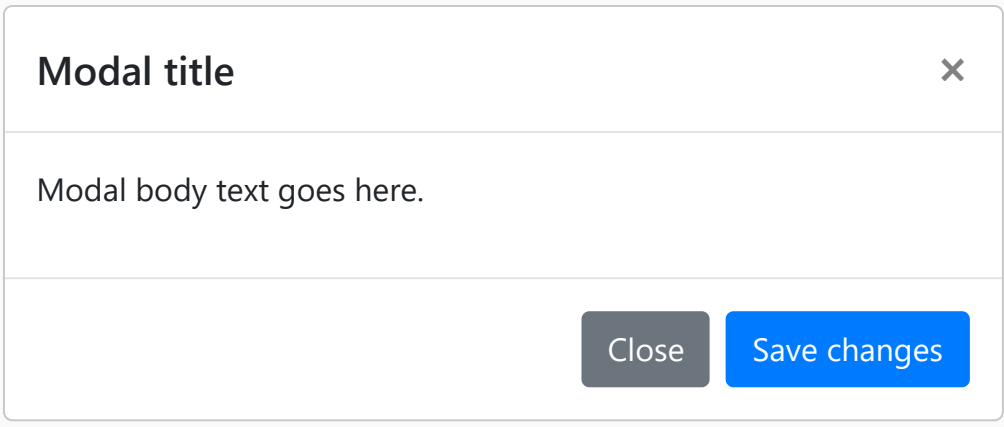
> The animation effect of this component is dependent on the `prefers-reduced-motion` media query. See the reduced motion section of our accessibility documentation.

Keep reading for demos and usage guidelines.

Examples

Modal components

Below is a *static* modal example (meaning its `position` and `display` have been overridden). Included are the modal header, modal body (required for `padding`), and modal footer (optional). We ask that you include modal headers with dismiss actions whenever possible, or provide another explicit dismiss action.



Copy

```
<div class="modal" tabindex="-1">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title">Modal title</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        <p>Modal body text goes here.</p>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
        <button type="button" class="btn btn-primary">Save changes</button>
      </div>
    </div>
  </div>
</div>
```

Live demo

Toggle a working modal demo by clicking the button below. It will slide down and fade in from the top of the page.

Launch demo modal

Copy

```
<!-- Button trigger modal -->
<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModal">
  Launch demo modal
</button>

<!-- Modal -->
<div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel"
aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Modal title</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        ...
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
        <button type="button" class="btn btn-primary">Save changes</button>
      </div>
    </div>
  </div>
</div>
```

Static backdrop

When backdrop is set to static, the modal will not close when clicking outside it. Click the button below to try it.

Launch static backdrop modal

Copy

```
<!-- Button trigger modal -->
<button type="button" class="btn btn-primary" data-toggle="modal" data-
target="#staticBackdrop">
  Launch static backdrop modal
</button>

<!-- Modal -->
<div class="modal fade" id="staticBackdrop" data-backdrop="static" data-keyboard="false"
tabindex="-1" aria-labelledby="staticBackdropLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="staticBackdropLabel">Modal title</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        ...
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
        <button type="button" class="btn btn-primary">Understood</button>
      </div>
    </div>
  </div>
</div>
```

Scrolling long content

When modals become too long for the user's viewport or device, they scroll independent of the page itself. Try the demo below to see what we mean.

Launch demo modal

You can also create a scrollable modal that allows scroll the modal body by adding `.modal-dialog-scrollable` to `.modal-dialog`.

Launch demo modal

Copy

```
<!-- Scrollable modal -->
<div class="modal-dialog modal-dialog-scrollable">
  ...
</div>
```

Vertically centered

Add `.modal-dialog-centered` to `.modal-dialog` to vertically center the modal.

Vertically centered modal | Vertically centered scrollable modal

Copy

```
<!-- Vertically centered modal -->
<div class="modal-dialog modal-dialog-centered">
  ...
</div>

<!-- Vertically centered scrollable modal -->
<div class="modal-dialog modal-dialog-centered modal-dialog-scrollable">
  ...
</div>
```

Tooltips and popovers

Tooltips and popovers can be placed within modals as needed. When modals are closed, any tooltips and popovers within are also automatically dismissed.

Launch demo modal

Copy

```
<div class="modal-body">
  <h5>Popover in a modal</h5>
  <p>This <a href="#" role="button" class="btn btn-secondary popover-test" title="Popover title" data-content="Popover body content is set in this attribute.">button</a> triggers a popover on click.</p>
  <hr>
  <h5>Tooltips in a modal</h5>
  <p><a href="#" class="tooltip-test" title="Tooltip">This link</a> and <a href="#" class="tooltip-test" title="Tooltip">that link</a> have tooltips on hover.</p>
</div>
```

Using the grid

Utilize the Bootstrap grid system within a modal by nesting `.container-fluid` within the `.modal-body`. Then, use the normal grid system classes as you would anywhere else.

Launch demo modal

Copy

```
<div class="modal-body">
  <div class="container-fluid">
    <div class="row">
      <div class="col-md-4">.col-md-4</div>
      <div class="col-md-4 ml-auto">.col-md-4 .ml-auto</div>
    </div>
    <div class="row">
      <div class="col-md-3 ml-auto">.col-md-3 .ml-auto</div>
      <div class="col-md-2 ml-auto">.col-md-2 .ml-auto</div>
    </div>
    <div class="row">
      <div class="col-md-6 ml-auto">.col-md-6 .ml-auto</div>
    </div>
    <div class="row">
      <div class="col-sm-9">
        Level 1: .col-sm-9
        <div class="row">
          <div class="col-8 col-sm-6">
            Level 2: .col-8 .col-sm-6
          </div>
          <div class="col-4 col-sm-6">
            Level 2: .col-4 .col-sm-6
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

Varying modal content

Have a bunch of buttons that all trigger the same modal with slightly different contents? Use `event.relatedTarget` and HTML `data-*` attributes (possibly via jQuery) to vary the contents of the modal depending on which button was clicked.

Below is a live demo followed by example HTML and JavaScript. For more information, read the modal events docs for details on `relatedTarget`.

Open modal for @mdo | Open modal for @fat | Open modal for @getbootstrap

Copy

```html
<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModal"
data-whatever="@mdo">Open modal for @mdo</button>
<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModal"
data-whatever="@fat">Open modal for @fat</button>
<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModal"
data-whatever="@getbootstrap">Open modal for @getbootstrap</button>

<div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel"
aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">New message</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        <form>
          <div class="form-group">
            <label for="recipient-name" class="col-form-label">Recipient:</label>
            <input type="text" class="form-control" id="recipient-name">
          </div>
          <div class="form-group">
            <label for="message-text" class="col-form-label">Message:</label>
            <textarea class="form-control" id="message-text"></textarea>
          </div>
        </form>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
        <button type="button" class="btn btn-primary">Send message</button>
      </div>
    </div>
  </div>
</div>
```

Copy

```js
$('#exampleModal').on('show.bs.modal', function (event) {
  var button = $(event.relatedTarget) // Button that triggered the modal
  var recipient = button.data('whatever') // Extract info from data-* attributes
  // If necessary, you could initiate an AJAX request here (and then do the updating in a
callback).
  // Update the modal's content. We'll use jQuery here, but you could use a data binding
library or other methods instead.
  var modal = $(this)
  modal.find('.modal-title').text('New message to ' + recipient)
  modal.find('.modal-body input').val(recipient)
})
```

Change animation

The `$modal-fade-transform` variable determines the transform state of `.modal-dialog` before the modal fade-in animation, the `$modal-show-transform` variable determines the transform of `.modal-dialog` at the end of the modal fade-in animation.

If you want for example a zoom-in animation, you can set `$modal-fade-transform: scale(.8)`.

Remove animation

For modals that simply appear rather than fade in to view, remove the `.fade` class from your modal markup.

Copy

```html
<div class="modal" tabindex="-1" aria-labelledby="..." aria-hidden="true">
  ...
</div>
```

Dynamic heights

If the height of a modal changes while it is open, you should call `$('#myModal').modal('handleUpdate')` to readjust the modal's position in case a scrollbar appears.

Accessibility

Be sure to add `aria-labelledby="..."`, referencing the modal title, to `.modal`. Additionally, you may give a description of your modal dialog with `aria-describedby` on `.modal`. Note that you don't need to add `role="dialog"` since we already add it via JavaScript.

Embedding YouTube videos

Embedding YouTube videos in modals requires additional JavaScript not in Bootstrap to automatically stop playback and more. See this helpful Stack Overflow post for more information.

Optional sizes

Modals have three optional sizes, available via modifier classes to be placed on a `.modal-dialog`. These sizes kick in at certain breakpoints to avoid horizontal scrollbars on narrower viewports.

Size	Class	Modal max-width
Small	`.modal-sm`	`300px`
Default	None	`500px`
Large	`.modal-lg`	`800px`
Extra large	`.modal-xl`	`1140px`

Our default modal without modifier class constitutes the "medium" size modal.

Extra large modal Large modal Small modal

Copy

```
<div class="modal-dialog modal-xl">...</div>
<div class="modal-dialog modal-lg">...</div>
<div class="modal-dialog modal-sm">...</div>
```

Usage

The modal plugin toggles your hidden content on demand, via data attributes or JavaScript. It also adds `.modal-open` to the `<body>` to override default scrolling behavior and generates a `.modal-backdrop` to provide a click area for dismissing shown modals when clicking outside the modal.

Via data attributes

Activate a modal without writing JavaScript. Set `data-toggle="modal"` on a controller element, like a button, along with a `data-target="#foo"` or `href="#foo"` to target a specific modal to toggle.

Copy

```
<button type="button" data-toggle="modal" data-target="#myModal">Launch modal</button>
```

Via JavaScript

Call a modal with id `myModal` with a single line of JavaScript:

Copy

```
$('#myModal').modal(options)
```

Options

Options can be passed via data attributes or JavaScript. For data attributes, append the option name to `data-`, as in `data-backdrop=""`.

Name	Type	Default	Description
backdrop	boolean or the string `'static'`	true	Includes a modal-backdrop element. Alternatively, specify `static` for a backdrop which doesn't close the modal on click.
keyboard	boolean	true	Closes the modal when escape key is pressed
focus	boolean	true	Puts the focus on the modal when initialized.
show	boolean	true	Shows the modal when initialized.

Methods

> **Asynchronous methods and transitions**
>
> All API methods are **asynchronous** and start a **transition**. They return to the caller as soon as the transition is started but **before it ends**. In addition, a method call on a **transitioning component will be ignored**.
>
> See our JavaScript documentation for more information.

`.modal(options)`

Activates your content as a modal. Accepts an optional options `object`.

```
$('#myModal').modal({
  keyboard: false
})
```

`.modal('toggle')`

Manually toggles a modal. **Returns to the caller before the modal has actually been shown or hidden** (i.e. before the `shown.bs.modal` or `hidden.bs.modal` event occurs).

```
$('#myModal').modal('toggle')
```

`.modal('show')`

Manually opens a modal. **Returns to the caller before the modal has actually been shown** (i.e. before the `shown.bs.modal` event occurs).

```
$('#myModal').modal('show')
```

`.modal('hide')`

Manually hides a modal. **Returns to the caller before the modal has actually been hidden** (i.e. before the `hidden.bs.modal` event occurs).

```
$('#myModal').modal('hide')
```

`.modal('handleUpdate')`

Manually readjust the modal's position if the height of a modal changes while it is open (i.e. in case a scrollbar appears).

```
$('#myModal').modal('handleUpdate')
```

Copy

`.modal('dispose')`

Destroys an element's modal.

Events

Bootstrap's modal class exposes a few events for hooking into modal functionality. All modal events are fired at the modal itself (i.e. at the `<div class="modal">`).

Event Type	Description
show.bs.modal	This event fires immediately when the `show` instance method is called. If caused by a click, the clicked element is available as the `relatedTarget` property of the event.
shown.bs.modal	This event is fired when the modal has been made visible to the user (will wait for CSS transitions to complete). If caused by a click, the clicked element is available as the `relatedTarget` property of the event.
hide.bs.modal	This event is fired immediately when the `hide` instance method has been called.
hidden.bs.modal	This event is fired when the modal has finished being hidden from the user (will wait for CSS transitions to complete).
hidePrevented.bs.modal	This event is fired when the modal is shown, its backdrop is `static` and a click outside the modal or an escape key press is performed with the keyboard option or `data-keyboard` set to `false`.

```
$('#myModal').on('hidden.bs.modal', function (e) {
  // do something...
})
```

Copy