Search...

Getting started
Layout
Content

Components
Alerts
Badge
Breadcrumb
Buttons
Button group
Card
Carousel
Collapse
Dropdowns
Forms
Input group
Jumbotron
List group
Media object
Modal
Navs
Navbar
Pagination
Popovers
Progress
Scrollspy
Spinners
Toasts
Tooltips

Utilities
Extend
Migration
About

Collapse

View on GitHub

Toggle the visibility of content across your project with a few classes and our JavaScript plugins.

How it works

The collapse JavaScript plugin is used to show and hide content. Buttons or anchors are used as triggers that are mapped to specific elements you toggle. Collapsing an element will animate the `height` from its current value to `0`. Given how CSS handles animations, you cannot use `padding` on a `.collapse` element. Instead, use the class as an independent wrapping element.

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

Example

Click the buttons below to show and hide another element via class changes:

- `.collapse` hides content
- `.collapsing` is applied during transitions
- `.collapse.show` shows content

Generally, we recommend using a button with the `data-target` attribute. While not recommended from a semantic point of view, you can also use a link with the `href` attribute (and a `role="button"`). In both cases, the `data-toggle="collapse"` is required.

Link with href | Button with data-target

Copy

```
<p>
  <a class="btn btn-primary" data-toggle="collapse" href="#collapseExample"
role="button" aria-expanded="false" aria-controls="collapseExample">
    Link with href
  </a>
  <button class="btn btn-primary" type="button" data-toggle="collapse" data-
target="#collapseExample" aria-expanded="false" aria-controls="collapseExample">
    Button with data-target
  </button>
</p>
<div class="collapse" id="collapseExample">
  <div class="card card-body">
    Some placeholder content for the collapse component. This panel is hidden by
default but revealed when the user activates the relevant trigger.
  </div>
</div>
```

Horizontal

The collapse plugin also supports horizontal collapsing. Add the `.width` modifier class to transition the `width` instead of `height` and set a `width` on the immediate child element. Feel free to write your own custom Sass, use inline styles, or use our width utilities.

> Please note that while the example below has a `min-height` set to avoid excessive repaints in our docs, this is not explicitly required. **Only the `width` on the child element is required.**

Toggle width collapse

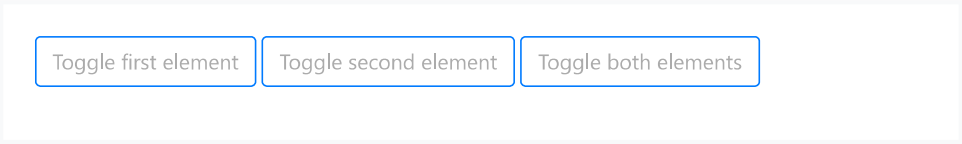
Copy

```
<p>
  <button class="btn btn-primary" type="button" data-toggle="collapse" data-
target="#collapseWidthExample" aria-expanded="false" aria-
controls="collapseWidthExample">
    Toggle width collapse
  </button>
</p>
<div style="min-height: 120px;">
  <div class="collapse width" id="collapseWidthExample">
    <div class="card card-body" style="width: 320px;">
      This is some placeholder content for a horizontal collapse. It's hidden by
default and shown when triggered.
    </div>
  </div>
</div>
```

Multiple targets

A `<button>` or `<a>` can show and hide multiple elements by referencing them with a JQuery selector in its `href` or `data-target` attribute. Multiple `<button>` or `<a>` can show and hide an element if they each reference it with their `href` or `data-target`

attribute

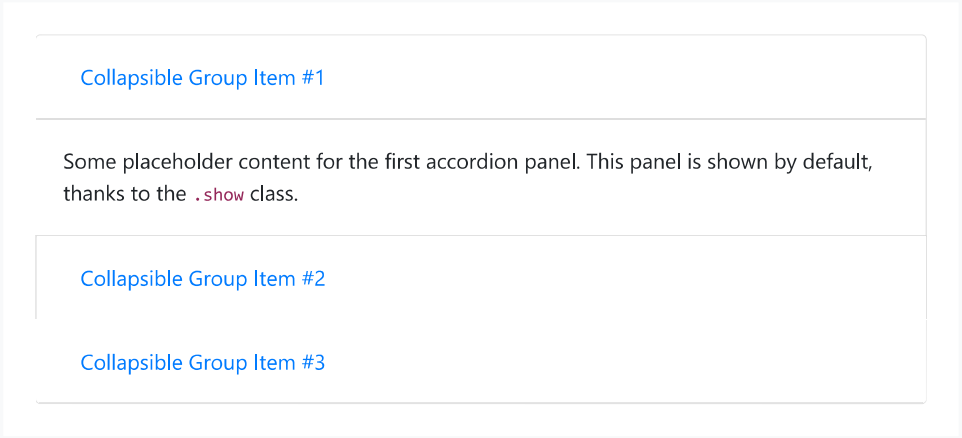


Copy

```
<p>
  <a class="btn btn-primary" data-toggle="collapse" href="#multiCollapseExample1"
role="button" aria-expanded="false" aria-controls="multiCollapseExample1">Toggle first
element</a>
  <button class="btn btn-primary" type="button" data-toggle="collapse" data-
target="#multiCollapseExample2" aria-expanded="false" aria-
controls="multiCollapseExample2">Toggle second element</button>
  <button class="btn btn-primary" type="button" data-toggle="collapse" data-
target=".multi-collapse" aria-expanded="false" aria-controls="multiCollapseExample1
multiCollapseExample2">Toggle both elements</button>
</p>
<div class="row">
  <div class="col">
    <div class="collapse multi-collapse" id="multiCollapseExample1">
      <div class="card card-body">
        Some placeholder content for the first collapse component of this multi-
collapse example. This panel is hidden by default but revealed when the user activates
the relevant trigger.
      </div>
    </div>
  </div>
  <div class="col">
    <div class="collapse multi-collapse" id="multiCollapseExample2">
      <div class="card card-body">
        Some placeholder content for the second collapse component of this multi-
collapse example. This panel is hidden by default but revealed when the user activates
the relevant trigger.
      </div>
    </div>
  </div>
</div>
```

Accordion example

Using the card component, you can extend the default collapse behavior to create an accordion. To properly achieve the accordion style, be sure to use `.accordion` as a wrapper.



Copy

```
<div class="accordion" id="accordionExample">
  <div class="card">
    <div class="card-header" id="headingOne">
      <h2 class="mb-0">
        <button class="btn btn-link btn-block text-left" type="button" data-
toggle="collapse" data-target="#collapseOne" aria-expanded="true" aria-
controls="collapseOne">
          Collapsible Group Item #1
        </button>
      </h2>
    </div>

    <div id="collapseOne" class="collapse show" aria-labelledby="headingOne" data-
parent="#accordionExample">
      <div class="card-body">
        Some placeholder content for the first accordion panel. This panel is shown by
default, thanks to the <code>.show</code> class.
      </div>
    </div>
  </div>
  <div class="card">
    <div class="card-header" id="headingTwo">
      <h2 class="mb-0">
        <button class="btn btn-link btn-block text-left collapsed" type="button" data-
toggle="collapse" data-target="#collapseTwo" aria-expanded="false" aria-
controls="collapseTwo">
          Collapsible Group Item #2
        </button>
      </h2>
    </div>
    <div id="collapseTwo" class="collapse" aria-labelledby="headingTwo" data-
parent="#accordionExample">
      <div class="card-body">
        Some placeholder content for the second accordion panel. This panel is hidden
by default.
      </div>
    </div>
  </div>
  <div class="card">
    <div class="card-header" id="headingThree">
      <h2 class="mb-0">
        <button class="btn btn-link btn-block text-left collapsed" type="button" data-
toggle="collapse" data-target="#collapseThree" aria-expanded="false" aria-
controls="collapseThree">
          Collapsible Group Item #3
        </button>
      </h2>
    </div>
    <div id="collapseThree" class="collapse" aria-labelledby="headingThree" data-
parent="#accordionExample">
      <div class="card-body">
        And lastly, the placeholder content for the third and final accordion panel.
This panel is hidden by default.
      </div>
    </div>
  </div>
</div>
```

Accessibility

Be sure to add `aria-expanded` to the control element. This attribute explicitly conveys the current state of the collapsible element tied to the control to screen readers and similar assistive technologies. If the collapsible element is closed by default, the attribute on the control element should have a value of `aria-expanded="false"`. If you've set the collapsible element to be open by default using the `show` class, set `aria-expanded="true"` on the control instead. The plugin

will automatically toggle this attribute on the control based on whether or not the collapsible element has been opened or closed (via JavaScript, or because the user triggered another control element also tied to the same collapsible element). If the control element's HTML element is not a button (e.g., an `<a>` or `<div>`), the attribute `role="button"` should be added to the element.

If your control element is targeting a single collapsible element – i.e. the `data-target` attribute is pointing to an `id` selector – you should add the `aria-controls` attribute to the control element, containing the `id` of the collapsible element. Modern screen readers and similar assistive technologies make use of this attribute to provide users with additional shortcuts to navigate directly to the collapsible element itself.

Note that Bootstrap's current implementation does not cover the various keyboard interactions described in the ARIA Authoring Practices Guide accordion pattern - you will need to include these yourself with custom JavaScript.

Usage

The collapse plugin utilizes a few classes to handle the heavy lifting:

- `.collapse` hides the content
- `.collapse.show` shows the content
- `.collapsing` is added when the transition starts, and removed when it finishes

These classes can be found in `_transitions.scss`.

Via data attributes

Just add `data-toggle="collapse"` and a `data-target` to the element to automatically assign control of one or more collapsible elements. The `data-target` attribute accepts a CSS selector to apply the collapse to. Be sure to add the class `collapse` to the collapsible element. If you'd like it to default open, add the additional class `show`.

To add accordion-like group management to a collapsible area, add the data attribute `data-parent="#selector"`. Refer to the demo to see this in action.

Via JavaScript

Enable manually with:

Copy

```
$('.collapse').collapse()
```

Options

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

Name	Type	Default	Description
parent	selector | jQuery object | DOM element	false	If parent is provided, then all collapsible elements under the specified parent will be closed when this collapsible item is shown. (similar to traditional accordion behavior - this is dependent on the `card` class). The attribute has to be set on the target collapsible area.
toggle	boolean	true	Toggles the collapsible element on invocation

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.

`.collapse(options)`

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

Copy

```
$('#myCollapsible').collapse({
  toggle: false
})
```

`.collapse('toggle')`

Toggles a collapsible element to shown or hidden. **Returns to the caller before the collapsible element has actually been shown or hidden** (i.e. before the `shown.bs.collapse` or `hidden.bs.collapse` event occurs).

`.collapse('show')`

Shows a collapsible element. **Returns to the caller before the collapsible element has actually been shown** (i.e. before the `shown.bs.collapse` event occurs).

`.collapse('hide')`

Hides a collapsible element. **Returns to the caller before the collapsible element has actually been hidden** (i.e. before the `hidden.bs.collapse` event occurs).

`.collapse('dispose')`

Destroys an element's collapse.

Events

Bootstrap's collapse class exposes a few events for hooking into collapse functionality.

Event Type	Description
show.bs.collapse	This event fires immediately when the `show` instance method is called.
shown.bs.collapse	This event is fired when a collapse element has been made visible to the user (will wait for CSS transitions to complete).
hide.bs.collapse	This event is fired immediately when the `hide` method has been called.
hidden.bs.collapse	This event is fired when a collapse element has been hidden from the user (will wait for CSS transitions to complete).

Copy

```
$('#myCollapsible').on('hidden.bs.collapse', function () {
  // do something...
})
```