Search docs... `Ctrl + /` v5.0 ▾
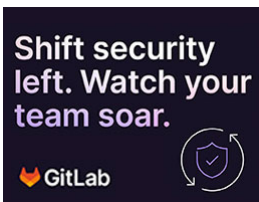
View on GitHub

Scrollspy

Automatically update Bootstrap navigation or list group components based on scroll position to indicate which link is currently active in the viewport.

On this page

How it works
Example in navbar
Example with nested nav
Example with list-group
Usage
   Via data attributes
   Via JavaScript
   Methods
      refresh
      dispose
      getInstance
      getOrCreateInstance
   Options
   Events

How it works

Scrollspy has a few requirements to function properly:

- It must be used on a Bootstrap [nav component](#) or [list group](#).
- Scrollspy requires `position: relative;` on the element you're spying on, usually the `<body>`.
- Anchors (`<a>`) are required and must point to an element with that `id`.

When successfully implemented, your nav or list group will update accordingly, moving the `.active` class from one item to the next based on their associated targets.

Scrollable containers and keyboard access

If you're making a scrollable container (other than the `<body>`), be sure to have a `height` set and `overflow-y: scroll;` applied to it—alongside a `tabindex="0"` to ensure keyboard access.

Example in navbar

Scroll the area below the navbar and watch the active class change. The dropdown items will be highlighted as well.

Navbar   First   Second   Dropdown ▾

First heading

This is some placeholder content for the scrollspy page. Note that as you scroll down the page, the appropriate navigation link is highlighted. It's repeated throughout the component example. We keep adding some more example copy here to emphasize the scrolling and highlighting.

Second heading

This is some placeholder content for the scrollspy page. Note that as you scroll down the

```
<nav id="navbar-example2" class="navbar navbar-light bg-light px-3">
  <a class="navbar-brand" href="#">Navbar</a>
  <ul class="nav nav-pills">
    <li class="nav-item">
      <a class="nav-link" href="#scrollspyHeading1">First</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#scrollspyHeading2">Second</a>
    </li>
```

```
    <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle" data-bs-toggle="dropdown" href="#" role="b
      <ul class="dropdown-menu">
        <li><a class="dropdown-item" href="#scrollspyHeading3">Third</a></li>
        <li><a class="dropdown-item" href="#scrollspyHeading4">Fourth</a></li>
        <li><hr class="dropdown-divider"></li>
        <li><a class="dropdown-item" href="#scrollspyHeading5">Fifth</a></li>
      </ul>
    </li>
  </ul>
</nav>
<div data-bs-spy="scroll" data-bs-target="#navbar-example2" data-bs-offset="0" class
  <h4 id="scrollspyHeading1">First heading</h4>
  <p>...</p>
  <h4 id="scrollspyHeading2">Second heading</h4>
  <p>...</p>
  <h4 id="scrollspyHeading3">Third heading</h4>
  <p>...</p>
  <h4 id="scrollspyHeading4">Fourth heading</h4>
  <p>...</p>
  <h4 id="scrollspyHeading5">Fifth heading</h4>
  <p>...</p>
```

Example with nested nav

Scrollspy also works with nested `.nav`s. If a nested `.nav` is `.active`, its parents will also be `.active`. Scroll the area next to the navbar and watch the active class change.

Navbar

Item 1

Item 1-1

Item 1-2

Item 2

Item 3

Item 3-1

Item 1

This is some placeholder content for the scrollspy page. Note that as you scroll down the page, the appropriate navigation link is highlighted. It's repeated throughout the component example. We keep adding some more example copy here to emphasize the scrolling and highlighting.

Item 1-1

This is some placeholder content for the scrollspy page. Note that as you scroll down the page, the appropriate navigation link is highlighted. It's repeated throughout the component example. We keep adding some more example copy here to emphasize the scrolling and

Item 3-2

```html
<nav id="navbar-example3" class="navbar navbar-light bg-light flex-column align-item
  <a class="navbar-brand" href="#">Navbar</a>
  <nav class="nav nav-pills flex-column">
    <a class="nav-link" href="#item-1">Item 1</a>
    <nav class="nav nav-pills flex-column">
      <a class="nav-link ms-3 my-1" href="#item-1-1">Item 1-1</a>
      <a class="nav-link ms-3 my-1" href="#item-1-2">Item 1-2</a>
    </nav>
    <a class="nav-link" href="#item-2">Item 2</a>
    <a class="nav-link" href="#item-3">Item 3</a>
    <nav class="nav nav-pills flex-column">
      <a class="nav-link ms-3 my-1" href="#item-3-1">Item 3-1</a>
      <a class="nav-link ms-3 my-1" href="#item-3-2">Item 3-2</a>
    </nav>
  </nav>
</nav>

<div data-bs-spy="scroll" data-bs-target="#navbar-example3" data-bs-offset="0" tabin
  <h4 id="item-1">Item 1</h4>
  <p>...</p>
  <h5 id="item-1-1">Item 1-1</h5>
  <p>...</p>
  <h5 id="item-1-2">Item 1-2</h5>
  <p>...</p>
  <h4 id="item-2">Item 2</h4>
  <p>...</p>
  <h4 id="item-3">Item 3</h4>
  <p>...</p>
  <h5 id="item-3-1">Item 3-1</h5>
  <p>...</p>
  <h5 id="item-3-2">Item 3-2</h5>
  <p>...</p>
</div>
```

Example with list-group

Scrollspy also works with `.list-group`s. Scroll the area next to the list group and watch the active class change.

Item 1	**Item 1**
Item 2	This is some placeholder content for the scrollspy page. Note that as you scroll down the page, the appropriate navigation link is highlighted. It's repeated throughout the component example. We keep adding some more example copy here to emphasize the scrolling and highlighting.
Item 3	
Item 4	

```html
<div id="list-example" class="list-group">
  <a class="list-group-item list-group-item-action" href="#list-item-1">Item 1</a>
  <a class="list-group-item list-group-item-action" href="#list-item-2">Item 2</a>
  <a class="list-group-item list-group-item-action" href="#list-item-3">Item 3</a>
  <a class="list-group-item list-group-item-action" href="#list-item-4">Item 4</a>
</div>
<div data-bs-spy="scroll" data-bs-target="#list-example" data-bs-offset="0" class="s
  <h4 id="list-item-1">Item 1</h4>
  <p>...</p>
  <h4 id="list-item-2">Item 2</h4>
  <p>...</p>
  <h4 id="list-item-3">Item 3</h4>
  <p>...</p>
  <h4 id="list-item-4">Item 4</h4>
  <p>...</p>
</div>
```

Usage

Via data attributes

To easily add scrollspy behavior to your topbar navigation, add `data-bs-spy="scroll"` to the element you want to spy on (most typically this would be the `<body>`). Then add the `data-bs-target` attribute with the ID or class of the parent element of any Bootstrap `.nav` component.

```css
body {
  position: relative;
}
```

```
<body data-bs-spy="scroll" data-bs-target="#navbar-example">
  ...
  <div id="navbar-example">
    <ul class="nav nav-tabs" role="tablist">
      ...
    </ul>
  </div>
  ...
</body>
```

Via JavaScript

After adding `position: relative;` in your CSS, call the scrollspy via JavaScript:

```
var scrollSpy = new bootstrap.ScrollSpy(document.body, {
  target: '#navbar-example'
})
```

Resolvable ID targets required

Navbar links must have resolvable id targets. For example, a `<a href="#home">home</a>` must correspond to something in the DOM like `<div id="home"></div>`.

Non-visible target elements ignored

Target elements that are not visible will be ignored and their corresponding nav items will never be highlighted.

Methods

refresh

When using scrollspy in conjunction with adding or removing of elements from the DOM, you'll need to call the refresh method like so:

```
var dataSpyList = [].slice.call(document.querySelectorAll('[data-bs-spy="scroll"]'))
dataSpyList.forEach(function (dataSpyEl) {
  bootstrap.ScrollSpy.getInstance(dataSpyEl)
    .refresh()
```

```
    })
```

dispose

Destroys an element's scrollspy. (Removes stored data on the DOM element)

getInstance

Static method which allows you to get the scrollspy instance associated with a DOM element

```
var scrollSpyContentEl = document.getElementById('content')
var scrollSpy = bootstrap.ScrollSpy.getInstance(scrollSpyContentEl) // Returns a Boo
```

getOrCreateInstance

Static method which allows you to get the scrollspy instance associated with a DOM element, or create a new one in case it wasn't initialised

```
var scrollSpyContentEl = document.getElementById('content')
var scrollSpy = bootstrap.ScrollSpy.getOrCreateInstance(scrollSpyContentEl) // Retur
```
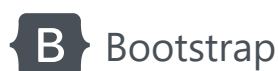
Options

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

Name	Type	Default	Description
offset	number	10	Pixels to offset from top when calculating position of scroll.
method	string	auto	Finds which section the spied element is in. `auto` will choose the best method to get scroll coordinates. `offset` will use the [Element.getBoundingClientRect()](#) method to get scroll coordinates. `position` will use the [HTMLElement.offsetTop](#) and [HTMLElement.offsetLeft](#) properties to get scroll coordinates.
target	string | jQuery object | DOM element		Specifies element to apply Scrollspy plugin.

Events

Event type	Description
`activate.bs.scrollspy`	This event fires on the scroll element whenever a new item becomes activated by the scrollspy.

```
var firstScrollSpyEl = document.querySelector('[data-bs-spy="scroll"]')
firstScrollSpyEl.addEventListener('activate.bs.scrollspy', function () {
  // do something...
})
```

B Bootstrap

Designed and built with all the love in the world by the Bootstrap team with the help of our contributors.

Code licensed MIT, docs CC BY 3.0.

Currently v5.0.2.

Analytics by Fathom.

Links

Home

Docs

Examples

Themes

Blog

Swag Store

Guides

Getting started

Starter template

Webpack

Parcel

Projects

Bootstrap 5

Bootstrap 4

Icons

RFS

Community

Issues

Discussions

Corporate sponsors

Open Collective