v2.7.2 (5.2)

Alerts

Provide contextual feedback messages for typical user actions with the handful of available and flexible alert messages.

Examples

Alerts are available for any length of text, as well as an optional dismiss button. For proper styling, use one of the eight `variant`s.

This is a primary alert—check it out!

This is a secondary alert—check it out!

This is a success alert—check it out!

This is a danger alert—check it out!

This is a warning alert—check it out!

This is a info alert—check it out!

This is a light alert—check it out!

This is a dark alert—check it out!

Copy

```
import Alert from 'react-bootstrap/Alert';

function BasicExample() {
  return (
    <>
      {[
        'primary',
        'secondary',
        'success',
        'danger',
        'warning',
        'info',
        'light',
        'dark',
      ].map((variant) => (
        <Alert key={variant} variant={variant}>
          This is a {variant} alert—check it out!
        </Alert>
      ))}
    </>
  );
}

export default BasicExample;
```

Conveying meaning to assistive technologies

Using color to add meaning only provides a visual indication, which will not be conveyed to users of assistive technologies – such as screen readers. Ensure that information denoted by the color is either obvious from the content itself (e.g. the visible text), or is included through alternative means, such as additional text hidden with the `.visually-hidden` class.

Links

For links, use the `<Alert.Link>` component to provide matching colored links within any alert.

This is a primary alert with **an example link**. Give it a click if you like.

This is a secondary alert with **an example link**. Give it a click if you like.

This is a success alert with **an example link**. Give it a click if you like.

This is a danger alert with **an example link**. Give it a click if you like.

This is a warning alert with **an example link**. Give it a click if you like.

This is a info alert with **an example link**. Give it a click if you like.

This is a light alert with **an example link**. Give it a click if you like.

This is a dark alert with **an example link**. Give it a click if you like.

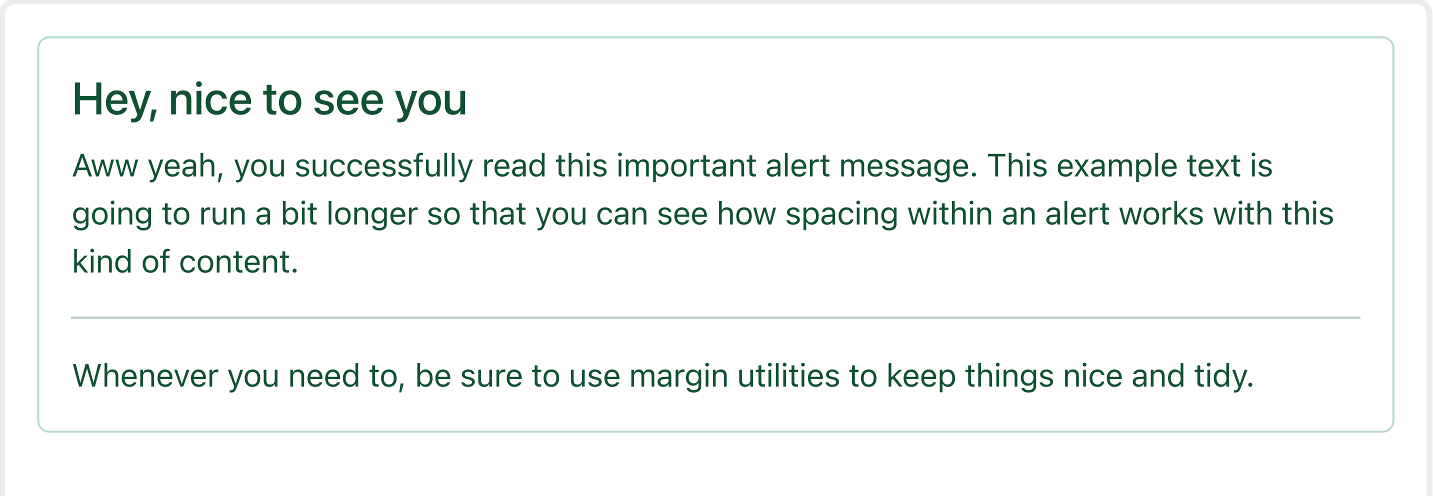
Copy

```
import Alert from 'react-bootstrap/Alert';

function LinksExample() {
  return (
    <>
      {[
        'primary',
        'secondary',
        'success',
        'danger',
        'warning',
        'info',
        'light',
        'dark',
      ].map((variant) => (
        <Alert key={variant} variant={variant}>
```

```
          This is a {variant} alert with{' '}
          <Alert.Link href="#">an example link</Alert.Link>. Give it a click if
          you like.
        </Alert>
      ))}
    </>
  );
}

export default LinksExample;
```

Additional content

Alerts can contain whatever content you like. Headers, paragraphs, dividers, go crazy.



Hey, nice to see you

Aww yeah, you successfully read this important alert message. This example text is going to run a bit longer so that you can see how spacing within an alert works with this kind of content.

Whenever you need to, be sure to use margin utilities to keep things nice and tidy.

```
import Alert from 'react-bootstrap/Alert';

function AdditionalContentExample() {
  return (
    <Alert variant="success">
      <Alert.Heading>Hey, nice to see you</Alert.Heading>
      <p>
        Aww yeah, you successfully read this important alert message. This
        example text is going to run a bit longer so that you can see how
        spacing within an alert works with this kind of content.
      </p>
      <hr />
      <p className="mb-0">
        Whenever you need to, be sure to use margin utilities to keep things
        nice and tidy.
      </p>
    </Alert>
  );
}

export default AdditionalContentExample;
```

Dismissing

Add the `dismissible` prop to add a functioning dismiss button to the Alert.

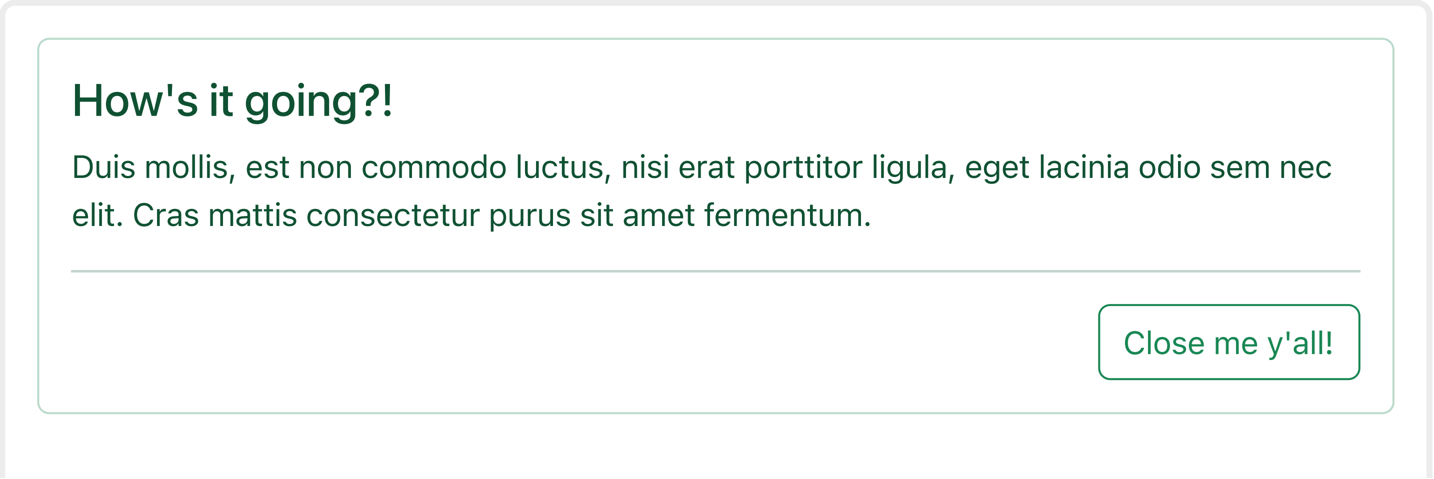
Oh snap! You got an error!

Change this and that and try again. Duis mollis, est non commodo luctus, nisi erat porttitor ligula, eget lacinia odio sem nec elit. Cras mattis consectetur purus sit amet fermentum.

Copy

```
import React, { useState } from 'react';
import Alert from 'react-bootstrap/Alert';
import Button from 'react-bootstrap/Button';

function AlertDismissibleExample() {
  const [show, setShow] = useState(true);

  if (show) {
    return (
      <Alert variant="danger" onClose={() => setShow(false)} dismissible>
        <Alert.Heading>Oh snap! You got an error!</Alert.Heading>
        <p>
          Change this and that and try again. Duis mollis, est non commodo
          luctus, nisi erat porttitor ligula, eget lacinia odio sem nec elit.
          Cras mattis consectetur purus sit amet fermentum.
        </p>
      </Alert>
    );
  }
  return <Button onClick={() => setShow(true)}>Show Alert</Button>;
}

render(<AlertDismissibleExample />);
```

You can also control the visual state directly which is great if you want to build more complicated alerts.



How's it going?!

Duis mollis, est non commodo luctus, nisi erat porttitor ligula, eget lacinia odio sem nec elit. Cras mattis consectetur purus sit amet fermentum.

```
import React, { useState } from 'react';
import Alert from 'react-bootstrap/Alert';
import Button from 'react-bootstrap/Button';

function AlertDismissible() {
  const [show, setShow] = useState(true);

  return (
    <>
      <Alert show={show} variant="success">
        <Alert.Heading>How's it going?!</Alert.Heading>
        <p>
          Duis mollis, est non commodo luctus, nisi erat porttitor ligula, eget
          lacinia odio sem nec elit. Cras mattis consectetur purus sit amet
          fermentum.
        </p>
        <hr />
        <div className="d-flex justify-content-end">
          <Button onClick={() => setShow(false)} variant="outline-success">
            Close me y'all!
          </Button>
        </div>
      </Alert>

      {!show && <Button onClick={() => setShow(true)}>Show Alert</Button>}
    </>
  );
}

render(<AlertDismissible />);
```

API

Alert

```
import Alert from 'react-bootstrap/Alert'
```

Name	Type	Default	Description
closeLabel	string	`'Close alert'`	Sets the text for alert close button.
closeVariant	`'white'`		Sets the variant for close button.
dismissible	boolean		Renders a properly aligned dismiss button, as well as adding extra horizontal padding to the Alert.
onClose	function		*controls `show`* Callback fired when alert is closed.
show	boolean	`true`	*controlled by: `onClose`, initial prop: `defaultShow`* Controls the visual state of the Alert.
transition	boolean | elementType	`Fade`	Animate the alert dismissal. Defaults to using `<Fade>` animation or use `false` to disable. A custom `react-transition-group` Transition can also be provided.
variant	'primary' | 'secondary' | 'success' | 'danger' | 'warning' | 'info' | 'dark' | 'light'	`'primary'`	The Alert visual variant
bsPrefix	string	`'alert'`	Change the underlying component CSS base class name and modifier class names prefix. **This is an escape**

Name	Type	Default	Description
			hatch for working with heavily customized bootstrap css.

Alert.Heading

```
import Alert from 'react-bootstrap/Alert'
```

Name	Type	Default	Description
as	elementType	`<DivStyledAsH4>`	You can use a custom element type for this component.
bsPrefix required	string	`'alert-heading'`	Change the underlying component CSS base class name and modifier class names prefix. **This is an escape hatch** for working with heavily customized bootstrap css.

Alert.Link

```
import Alert from 'react-bootstrap/Alert'
```

Name	Type	Default	Description
as	elementType	`<Anchor>`	You can use a custom element type for this component.

Name	Type	Default	Description
bsPrefix required	string	`'alert-link'`	Change the underlying component CSS base class name and modifier class names prefix. **This is an escape hatch** for working with heavily customized bootstrap css.