Alert

Alerts display brief messages for the user without interrupting their use of the app.

View as Markdown Feedback Bundle size Source WAI-ARIA Figma Sketch

Introduction

Alerts give users brief and potentially time-sensitive information in an unobtrusive manner.

The Material UI Alert component includes several props for quickly customizing its styles to provide immediate visual cues about its contents.

✓ Here is a gentle confirmation that your action was successful.

Edit in Chat JS TS Collapse code

```
import Alert from '@mui/material/Alert';
import CheckIcon from '@mui/icons-material/Check';

export default function SimpleAlert() {
  return (
    <Alert icon={<CheckIcon fontSize="inherit" />} severity="success">
      Here is a gentle confirmation that your action was successful.
    </Alert>
  );
}
```

ⓘ This component is no longer documented in the Material Design guidelines ↗, but Material UI will continue to support it.

Usage

A key trait of the alert pattern is that it should not interrupt the user's experience ↗ of the app. Alerts should not be confused with alert *dialogs* (ARIA ↗), which *are* intended to interrupt the user to obtain a response. Use the Material UI Dialog component if you need this behavior.

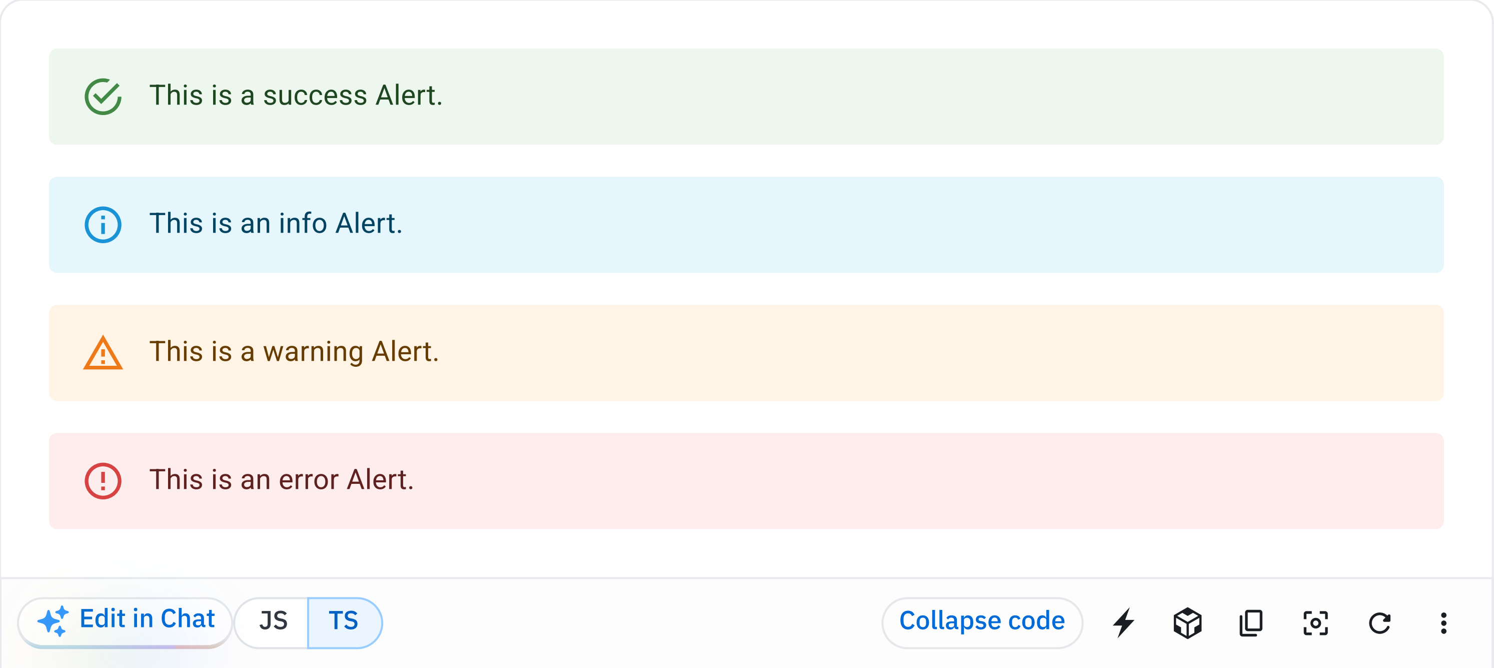
Basics

```
import Alert from '@mui/material/Alert';
```

The Alert component wraps around its content, and stretches to fill its enclosing container.

Severity

The `severity` prop accepts four values representing different states—`success` (the default), `info`, `warning`, and `error`—with corresponding icon and color combinations for each:



This is a success Alert.

This is an info Alert.

This is a warning Alert.

This is an error Alert.

```
import Alert from '@mui/material/Alert';
import Stack from '@mui/material/Stack';

export default function BasicAlerts() {
  return (
    <Stack sx={{ width: '100%' }} spacing={2}>
      <Alert severity="success">This is a success Alert.</Alert>
      <Alert severity="info">This is an info Alert.</Alert>
      <Alert severity="warning">This is a warning Alert.</Alert>
      <Alert severity="error">This is an error Alert.</Alert>
    </Stack>
  );
}
```

Variants

The Alert component comes with two alternative style options—`filled` and `outlined`—which you can set using the `variant` prop.

Filled

This is a filled success Alert.

This is a filled info Alert.

This is a filled warning Alert.

This is a filled error Alert.

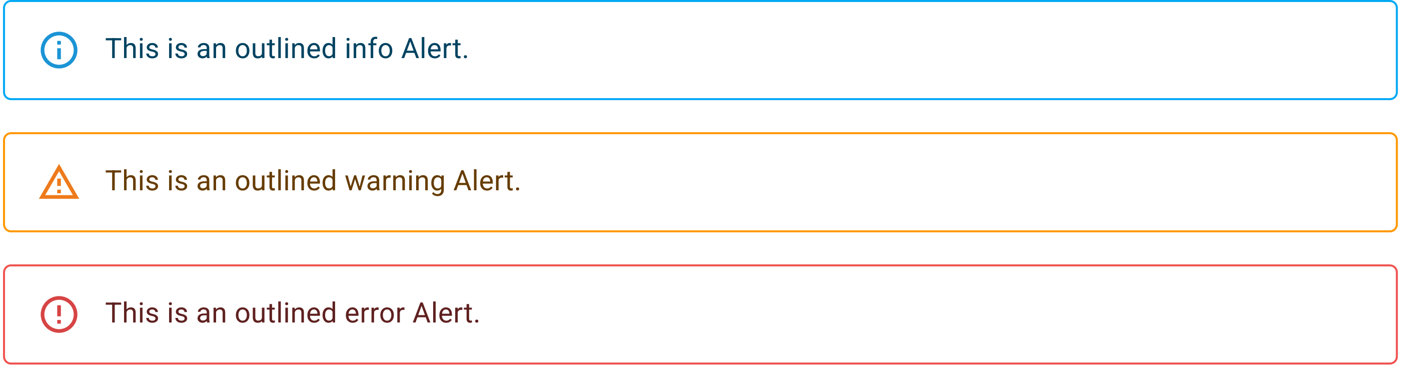
Edit in Chat JS TS Collapse code

```
import Alert from '@mui/material/Alert';
import Stack from '@mui/material/Stack';

export default function FilledAlerts() {
  return (
    <Stack sx={{ width: '100%' }} spacing={2}>
      <Alert variant="filled" severity="success">
        This is a filled success Alert.
      </Alert>
      <Alert variant="filled" severity="info">
        This is a filled info Alert.
      </Alert>
      <Alert variant="filled" severity="warning">
        This is a filled warning Alert.
      </Alert>
      <Alert variant="filled" severity="error">
        This is a filled error Alert.
      </Alert>
    </Stack>
  );
}
```

Outlined

This is an outlined success Alert.



This is an outlined info Alert.

This is an outlined warning Alert.

This is an outlined error Alert.

Edit in Chat JS TS Collapse code

```
import Alert from '@mui/material/Alert';
import Stack from '@mui/material/Stack';

export default function OutlinedAlerts() {
  return (
    <Stack sx={{ width: '100%' }} spacing={2}>
      <Alert variant="outlined" severity="success">
        This is an outlined success Alert.
      </Alert>
      <Alert variant="outlined" severity="info">
        This is an outlined info Alert.
      </Alert>
      <Alert variant="outlined" severity="warning">
        This is an outlined warning Alert.
      </Alert>
      <Alert variant="outlined" severity="error">
        This is an outlined error Alert.
      </Alert>
    </Stack>
  );
}
```

⚠ When using an outlined Alert with the Snackbar component, background content will be visible and bleed through the Alert by default. You can prevent this by adding `bgcolor: 'background.paper'` to the `sx` prop on the Alert component:

```
<Alert sx={{ bgcolor: 'background.paper' }} />
```

Check out the Snackbar—customization doc for an example of how to use these two components together.

Color

Use the `color` prop to override the default color for the specified `severity`—for instance, to apply `warning` colors to a `success` Alert:

This is a success Alert with warning colors.

Edit in Chat JS TS Collapse code

```
import Alert from '@mui/material/Alert';

export default function ColorAlerts() {
  return (
    <Alert severity="success" color="warning">
      This is a success Alert with warning colors.
    </Alert>
  );
}
```

Actions

Add an action to your Alert with the `action` prop. This lets you insert any element—an HTML tag, an SVG icon, or a React component such as a Material UI Button—after the Alert's message, justified to the right.

If you provide an `onClose` callback to the Alert without setting the `action` prop, the component will display a close icon (✕) by default.

This Alert displays the default close icon. ✕

This Alert uses a Button component for its action. UNDO

Edit in Chat JS TS Collapse code

```
import Alert from '@mui/material/Alert';
import Button from '@mui/material/Button';
import Stack from '@mui/material/Stack';

export default function ActionAlerts() {
  return (
    <Stack sx={{ width: '100%' }} spacing={2}>
      <Alert severity="warning" onClose={() => {}}>
        This Alert displays the default close icon.
      </Alert>
      <Alert
        severity="success"
        action={
          <Button color="inherit" size="small">
            UNDO
          </Button>
        }
      >
        This Alert uses a Button component for its action.
      </Alert>
    </Stack>
  );
}
```

Icons

Use the `icon` prop to override an Alert's icon. As with the `action` prop, your `icon` can be an HTML element, an SVG icon, or a React component. Set this prop to `false` to remove the icon altogether.

If you need to override all instances of an icon for a given `severity`, you can use the `iconMapping` prop instead. You can define this prop globally by customizing your app's theme. See Theme components—Default props for details.

✓ This success Alert has a custom icon.

This success Alert has no icon.

✓ This success Alert uses `iconMapping` to override the default icon.

Edit in Chat | JS | TS | Collapse code

```
import Alert from '@mui/material/Alert';
import CheckIcon from '@mui/icons-material/Check';
import CheckCircleOutlineIcon from '@mui/icons-material/CheckCircleOutline';
import Stack from '@mui/material/Stack';

export default function IconAlerts() {
  return (
    <Stack sx={{ width: '100%' }} spacing={2}>
      <Alert icon={<CheckIcon fontSize="inherit" />} severity="success">
        This success Alert has a custom icon.
      </Alert>
      <Alert icon={false} severity="success">
        This success Alert has no icon.
      </Alert>
      <Alert
        iconMapping={{
          success: <CheckCircleOutlineIcon fontSize="inherit" />,
        }}
      >
        This success Alert uses `iconMapping` to override the default icon.
      </Alert>
    </Stack>
  );
}
```

Customization

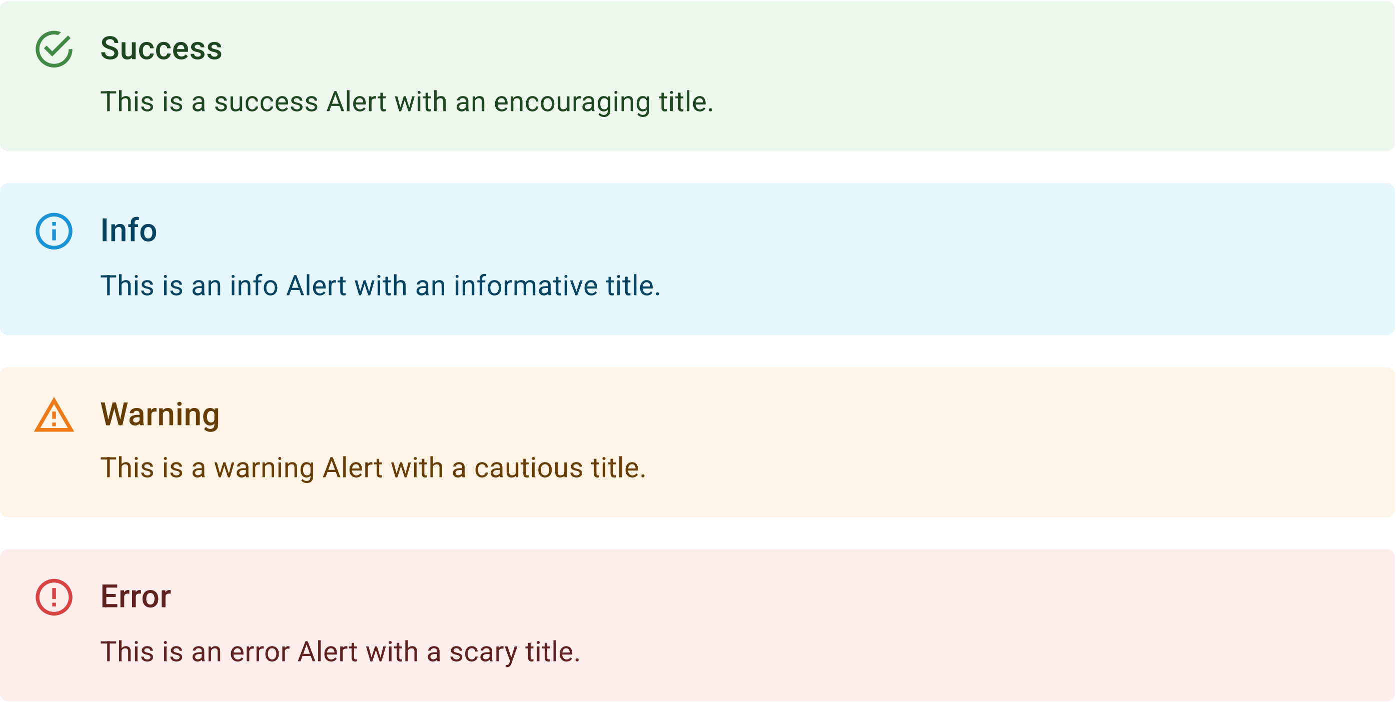
Titles

To add a title to an Alert, import the Alert Title component:

```
import AlertTitle from '@mui/material/AlertTitle';
```

Copy

You can nest this component above the message in your Alert for a neatly styled and properly aligned title, as shown below:



Success

This is a success Alert with an encouraging title.

Info

This is an info Alert with an informative title.

Warning

This is a warning Alert with a cautious title.

Error

This is an error Alert with a scary title.

Edit in Chat JS TS Collapse code

```
import Alert from '@mui/material/Alert';
import AlertTitle from '@mui/material/AlertTitle';
import Stack from '@mui/material/Stack';

export default function DescriptionAlerts() {
  return (
    <Stack sx={{ width: '100%' }} spacing={2}>
      <Alert severity="success">
        <AlertTitle>Success</AlertTitle>
        This is a success Alert with an encouraging title.
      </Alert>
      <Alert severity="info">
        <AlertTitle>Info</AlertTitle>
        This is an info Alert with an informative title.
      </Alert>
      <Alert severity="warning">
        <AlertTitle>Warning</AlertTitle>
        This is a warning Alert with a cautious title.
      </Alert>
      <Alert severity="error">
        <AlertTitle>Error</AlertTitle>
        This is an error Alert with a scary title.
      </Alert>
    </Stack>
  );
}
```

Transitions

You can use Transition components like Collapse to add motion to an Alert's entrance and exit.

✓ Click the close icon to see the Collapse transition in action! ✕

RE-OPEN

Edit in Chat JS TS Hide code

```
import * as React from 'react';
import Box from '@mui/material/Box';
import Alert from '@mui/material/Alert';
import IconButton from '@mui/material/IconButton';
import Collapse from '@mui/material/Collapse';
import Button from '@mui/material/Button';
import CloseIcon from '@mui/icons-material/Close';

export default function TransitionAlerts() {
  const [open, setOpen] = React.useState(true);

  return (
    <Box sx={{ width: '100%' }}>
      <Collapse in={open}>
        <Alert
          action={
            <IconButton
              aria-label="close"
              color="inherit"
              size="small"
              onClick={() => {
                setOpen(false);
              }}
            >
              <CloseIcon fontSize="inherit" />
            </IconButton>
          }
          sx={{ mb: 2 }}
        >
          Click the close icon to see the Collapse transition in action!
        </Alert>
      </Collapse>
      <Button
        disabled={open}
        variant="outlined"
        onClick={() => {
          setOpen(true);
        }}
```

Accessibility

Here are some factors to consider to ensure that your Alert is accessible:

- Because alerts are not intended to interfere with the use of the app, your Alert component should *never* affect the keyboard focus.
- If an alert contains an action, that action must have a `tabindex` of `0` so it can be reached by keyboard-only users.
- Essential alerts should not disappear automatically—timed interactions ↗ can make your app inaccessible to users who need extra time to understand or locate the alert.
- Alerts that occur too frequently can inhibit the usability ↗ of your app.
- Dynamically rendered alerts are announced by screen readers; alerts that are already present on the page when it loads are *not* announced.
- Color does not add meaning to the UI for users who require assistive technology. You must ensure that any information conveyed through color is also denoted in other ways, such as within the text of the alert itself, or with additional hidden text that's read by screen readers.

Anatomy

The Alert component is composed of a root Paper component (which renders as a `<div>`) that houses an icon, a message, and an optional action:

Copy

```
<div class="MuiPaper-root MuiAlert-root" role="alert">
  <div class="MuiAlert-icon">
    <!-- svg icon here -->
  </div>
  <div class="MuiAlert-message">This is how an Alert renders in the DOM.</div>
  <div class="MuiAlert-action">
```

```
    <!-- optional action element here -->
  </div>
</div>
```

API

See the documentation below for a complete reference to all of the props and classes available to the components mentioned here.

- `<Alert />`
- `<AlertTitle />`

Edit this page

Was this page helpful?

‹ Typography

Backdrop ›