

# Accordion

The Accordion component lets users show and hide sections of related content on a page.



Check out the latest remote job listings from Authentic Jobs.

ads via Carbon

[View as Markdown](#)[Feedback](#)[Bundle size](#)[Source](#)[WAI-ARIA](#)[Material Design](#)[Figma](#)[Sketch](#)

## Introduction



The Material UI Accordion component includes several complementary utility components to handle various use cases:

- **Accordion:** the wrapper for grouping related components.
- **Accordion Summary:** the wrapper for the Accordion header, which expands or collapses the content when clicked.
- **Accordion Details:** the wrapper for the Accordion content.
- **Accordion Actions:** an optional wrapper that groups a set of buttons.

Accordion 1



Accordion 2



Accordion Actions



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse malesuada lacus ex, sit amet blandit leo lobortis eget.


CANCEL

AGREE

[Edit in Chat](#)[JS](#)[TS](#)[Hide code](#)

```
import Accordion from '@mui/material/Accordion';
import AccordionActions from '@mui/material/AccordionActions';
import AccordionSummary from '@mui/material/AccordionSummary';
import AccordionDetails from '@mui/material/AccordionDetails';
import Typography from '@mui/material/Typography';
import ExpandMoreIcon from '@mui/icons-material/ExpandMore';
import Button from '@mui/material/Button';

export default function AccordionUsage() {
  return (
    <div>
      <Accordion>
        <AccordionSummary
          expandIcon={<ExpandMoreIcon />}
          aria-controls="panel1-content"
          id="panel1-header"
        >
          <Typography component="span">Accordion 1</Typography>
        </AccordionSummary>
        <AccordionDetails>
          Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse
          malesuada lacus ex, sit amet blandit leo lobortis eget.
        </AccordionDetails>
      </Accordion>
      <Accordion>
        <AccordionSummary
          expandIcon={<ExpandMoreIcon />}
          aria-controls="panel2-content"
          id="panel2-header"
        >
          <Typography component="span">Accordion 2</Typography>
        </AccordionSummary>
        <AccordionDetails>
          Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse
          malesuada lacus ex, sit amet blandit leo lobortis eget.
        </AccordionDetails>
      </Accordion>
      <Accordion defaultExpanded>
```

 This component is no longer documented in the [Material Design guidelines](#), but Material UI will continue to support it.

## Basics



```
import Accordion from '@mui/material/Accordion';
import AccordionDetails from '@mui/material/AccordionDetails';
import AccordionSummary from '@mui/material/AccordionSummary';
```

Copy

## Expand icon



Use the `expandIcon` prop on the `AccordionSummary` component to change the expand indicator icon. The component handles the turning upside-down transition automatically.

Accordion 1



Accordion 2



Edit in Chat

JS

TS

Hide code



```
import Accordion from '@mui/material/Accordion';
import AccordionSummary from '@mui/material/AccordionSummary';
import AccordionDetails from '@mui/material/AccordionDetails';
import Typography from '@mui/material/Typography';
import ArrowDownwardIcon from '@mui/icons-material/ArrowDownward';
import ArrowDropDownIcon from '@mui/icons-material/ArrowDropDown';

export default function AccordionExpandIcon() {
  return (
    <div>
      <Accordion>
        <AccordionSummary
          expandIcon={<ArrowDownwardIcon />}
          aria-controls="panel1-content"
          id="panel1-header"
        >
          <Typography component="span">Accordion 1</Typography>
        </AccordionSummary>
        <AccordionDetails>
          <Typography>
            Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse
            malesuada lacus ex, sit amet blandit leo lobortis eget.
          </Typography>
        </AccordionDetails>
      </Accordion>
      <Accordion>
        <AccordionSummary
          expandIcon={<ArrowDropDownIcon />}
          aria-controls="panel2-content"
          id="panel2-header"
        >
          <Typography component="span">Accordion 2</Typography>
        </AccordionSummary>
        <AccordionDetails>
          <Typography>
            Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse
            malesuada lacus ex, sit amet blandit leo lobortis eget.
          </Typography>
        </AccordionDetails>
      </Accordion>
    </div>
  );
}
```

## Expanded by default



Use the `defaultExpanded` prop on the Accordion component to have it opened by default.

Expanded by default



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse malesuada lacus ex, sit amet blandit leo lobortis eget.

Header



Edit in Chat

JS

TS

Hide code



```
import Accordion from '@mui/material/Accordion';
import AccordionSummary from '@mui/material/AccordionSummary';
import AccordionDetails from '@mui/material/AccordionDetails';
import Typography from '@mui/material/Typography';
import ExpandMoreIcon from '@mui/icons-material/ExpandMore';

export default function AccordionExpandDefault() {
  return (
    <div>
      <Accordion defaultExpanded>
        <AccordionSummary
          expandIcon={<ExpandMoreIcon />}
          aria-controls="panel1-content"
          id="panel1-header"
        >
          <Typography component="span">Expanded by default</Typography>
        </AccordionSummary>
        <AccordionDetails>
          <Typography>
            Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse
            malesuada lacus ex, sit amet blandit leo lobortis eget.
          </Typography>
        </AccordionDetails>
      </Accordion>
      <Accordion>
        <AccordionSummary
          expandIcon={<ExpandMoreIcon />}
          aria-controls="panel2-content"
          id="panel2-header"
        >
          <Typography component="span">Header</Typography>
        </AccordionSummary>
        <AccordionDetails>
          <Typography>
            Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse
            malesuada lacus ex, sit amet blandit leo lobortis eget.
          </Typography>
        </AccordionDetails>
      </Accordion>
    </div>
  );
}
```

## Transition



Use the `slots.transition` and `slotProps.transition` props to change the Accordion's default transition.

Custom transition using Fade



Default transition using Collapse



Edit in Chat

JS

TS

Hide code



```
import * as React from 'react';
import Accordion, {
  AccordionSlots,
  accordionClasses,
} from '@mui/material/Accordion';
import AccordionSummary from '@mui/material/AccordionSummary';
import AccordionDetails, {
  accordionDetailsClasses,
} from '@mui/material/AccordionDetails';
import Typography from '@mui/material/Typography';
import ExpandMoreIcon from '@mui/icons-material/ExpandMore';
import Fade from '@mui/material/Fade';

export default function AccordionTransition() {
  const [expanded, setExpanded] = React.useState(false);

  const handleExpansion = () => {
    setExpanded((prevExpanded) => !prevExpanded);
  };

  return (
    <div>
      <Accordion
        expanded={expanded}
        onChange={handleExpansion}
        slots={{ transition: Fade as AccordionSlots['transition'] }}
        slotProps={{ transition: { timeout: 400 } }}
        sx={[
          expanded
            ? {
                [`& .${accordionClasses.region}`]: {
                  height: 'auto',
                },
                [`& .${accordionDetailsClasses.root}`]: {
                  display: 'block',
                },
              }
            : {}
        ]}
      />
    </div>
  );
}
```

## Disabled item



Use the `disabled` prop on the Accordion component to disable interaction and focus.

Accordion 1



Accordion 2



Disabled Accordion



Edit in Chat

JS

TS

Hide code



```
import Accordion from '@mui/material/Accordion';
import AccordionSummary from '@mui/material/AccordionSummary';
import AccordionDetails from '@mui/material/AccordionDetails';
import Typography from '@mui/material/Typography';
import ExpandMoreIcon from '@mui/icons-material/ExpandMore';

export default function DisabledAccordion() {
  return (
    <div>
      <Accordion>
        <AccordionSummary
          expandIcon={<ExpandMoreIcon />}
          aria-controls="panel1-content"
          id="panel1-header"
        >
          <Typography component="span">Accordion 1</Typography>
        </AccordionSummary>
        <AccordionDetails>
          <Typography>
            Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse
            malesuada lacus ex, sit amet blandit leo lobortis eget.
          </Typography>
        </AccordionDetails>
      </Accordion>
      <Accordion>
        <AccordionSummary
          expandIcon={<ExpandMoreIcon />}
          aria-controls="panel2-content"
          id="panel2-header"
        >
          <Typography component="span">Accordion 2</Typography>
        </AccordionSummary>
        <AccordionDetails>
          <Typography>
            Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse
            malesuada lacus ex, sit amet blandit leo lobortis eget.
          </Typography>
        </AccordionDetails>
      </Accordion>
    </div>
  );
}
```

## Controlled Accordion



The Accordion component can be controlled or uncontrolled.

General settings	I am an accordion	▼
Users	You are currently not an owner	▼
Advanced settings	Filtering has been entirely disabled for whole web server	▼
Personal data		▼

[Edit in Chat](#)

JS

TS

[Hide code](#)

```
import * as React from 'react';
import Accordion from '@mui/material/Accordion';
import AccordionDetails from '@mui/material/AccordionDetails';
import AccordionSummary from '@mui/material/AccordionSummary';
import Typography from '@mui/material/Typography';
import ExpandMoreIcon from '@mui/icons-material/ExpandMore';

export default function ControlledAccordions() {
  const [expanded, setExpanded] = React.useState<string | false>(false);

  const handleChange =
    (panel: string) => (event: React.SyntheticEvent, isExpanded: boolean) => {
      setExpanded(isExpanded ? panel : false);
    };

  return (
    <div>
      <Accordion expanded={expanded === 'panel1'} onChange={handleChange('panel1')}>
        <AccordionSummary
          expandIcon={<ExpandMoreIcon />}
          aria-controls="panel1bh-content"
          id="panel1bh-header"
        >
          <Typography component="span" sx={{ width: '33%', flexShrink: 0 }}>
            General settings
          </Typography>
          <Typography component="span" sx={{ color: 'text.secondary' }}>
            I am an accordion
          </Typography>
        </AccordionSummary>
        <AccordionDetails>
          <Typography>
```

```

        Nulla facilisi. Phasellus sollicitudin nulla et quam mattis feugiat.
        Aliquam eget maximus est, id dignissim quam.
    </Typography>
</AccordionDetails>
</Accordion>

```

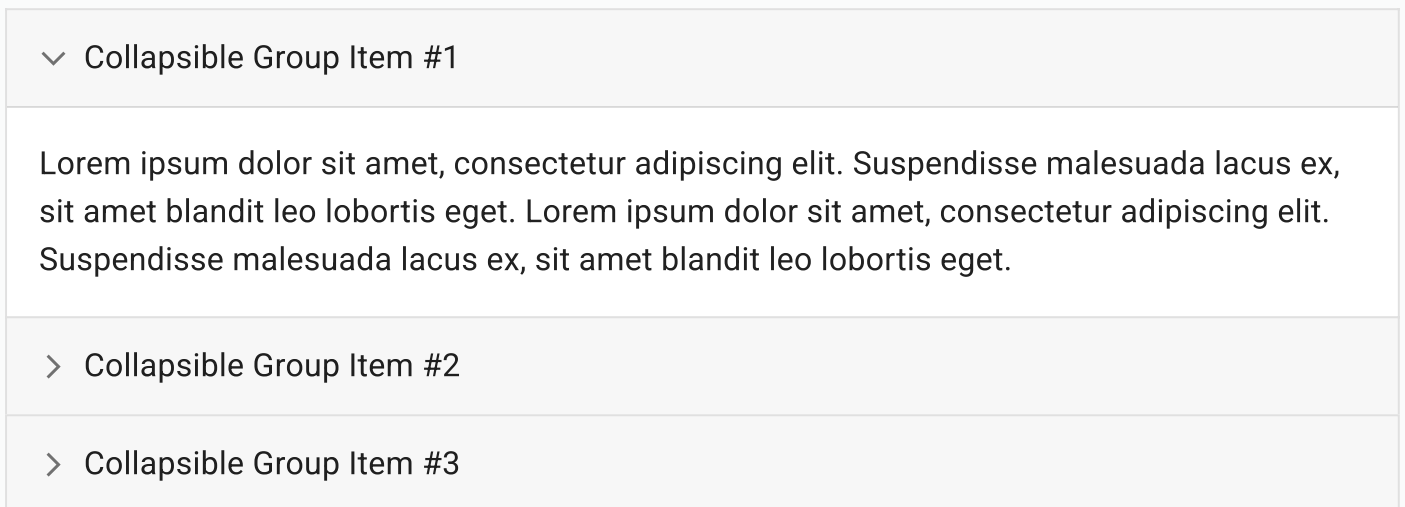
- A component is **controlled** when it's managed by its parent using props.
- A component is **uncontrolled** when it's managed by its own local state.

Learn more about controlled and uncontrolled components in the [React documentation](#).

## Customization

## Only one expanded at a time

Use the `expanded` prop with React's `useState` hook to allow only one Accordion item to be expanded at a time. The demo below also shows a bit of visual customization.



JS

TS

Hide code



```
import * as React from 'react';
import { styled } from '@mui/material/styles';
import ArrowForwardIosSharpIcon from '@mui/icons-material/ArrowForwardIosSharp';
import MuiAccordion, { AccordionProps } from '@mui/material/Accordion';
import MuiAccordionSummary, {
  AccordionSummaryProps,
  accordionSummaryClasses,
} from '@mui/material/AccordionSummary';
import MuiAccordionDetails from '@mui/material/AccordionDetails';
import Typography from '@mui/material/Typography';

const Accordion = styled((props: AccordionProps) => {
```



```

<MuiAccordion disableGutters elevation={0} square {...props} />
))(({ theme }) => ({
  border: `1px solid ${theme.palette.divider}`,
  '&:not(:last-child)': {
    borderBottom: 0,
  },
  '&::before': {
    display: 'none',
  },
}));

const AccordionSummary = styled((props: AccordionSummaryProps) => (
  <MuiAccordionSummary
    expandIcon={<ArrowForwardIosSharpIcon sx={{ fontSize: '0.9rem' }} />}
    {...props}
  />
))(({ theme }) => ({
  backgroundColor: 'rgba(0, 0, 0, .03)',
  flexDirection: 'row-reverse',
  [`&.${accordionSummaryClasses.expandIconWrapper}.${accordionSummaryClasses.expanded}`]:
    {
      transform: 'rotate(90deg)',
    },
  [`&.${accordionSummaryClasses.content}`]: {
    marginLeft: theme.spacing(1),
  },
}));

```

## Changing heading level



By default, the Accordion uses an `h3` element for the heading. You can change the heading element using the `slotProps.heading.component` prop to ensure the correct heading hierarchy in your document.

```

<Accordion slotProps={{ heading: { component: 'h4' } }}>
  <AccordionSummary
    expandIcon={<ExpandMoreIcon />}
    aria-controls="panel1-content"
    id="panel1-header"
  >
    Accordion
  </AccordionSummary>
  <AccordionDetails>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse malesuada
    lacus ex, sit amet blandit leo lobortis eget.
  </AccordionDetails>
</Accordion>

```

Copy

## Performance



The Accordion content is mounted by default even if it's not expanded. This default behavior has server-side rendering and SEO in mind.

If you render the `AccordionDetails` with a big component tree nested inside, or if you have many `Accordions`, you may want to change this behavior by setting `unmountOnExit` to `true` inside the `slotProps.transition` prop to improve performance:

```
<Accordion slotProps={{ transition: { unmountOnExit: true } }} />
```

Copy

## Accessibility



The [WAI-ARIA guidelines for accordions](#) recommend setting an `id` and `aria-controls`, which in this case would apply to the `AccordionSummary` component. The `Accordion` component then derives the necessary `aria-labelledby` and `id` from its content.

```
<Accordion>
  <AccordionSummary id="panel-header" aria-controls="panel-content">
    Header
  </AccordionSummary>
  <AccordionDetails>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  </AccordionDetails>
</Accordion>
```

Copy

## Anatomy



The `Accordion` component consists of a root `<div>` that contains the `AccordionSummary`, `AccordionDetails`, and optional `AccordionActions` for action buttons.

```
<div class="MuiAccordion-root">
  <h3 class="MuiAccordion-heading">
    <button class="MuiButtonBase-root MuiAccordionSummary-root" aria-expanded="">
      <!-- Accordion summary goes here -->
    </button>
  </h3>
  <div class="MuiAccordion-region" role="region">
    <div class="MuiAccordionDetails-root">
      <!-- Accordion content goes here -->
    </div>
  </div>
</div>
```

Copy


## API





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

- [<Accordion />](#)

- [<AccordionActions />](#)
- [<AccordionDetails />](#)
- [<AccordionSummary />](#)

 [Edit this page](#)

Was this page helpful?  

[< Snackbar](#)

[AppBar >](#)