Click-Away Listener

The Click-Away Listener component detects when a click event happens outside of its child element.

View as Markdown Feedback Bundle size

Introduction

Click-Away Listener is a utility component that listens for click events outside of its child. (Note that it only accepts *one* child element.) This is useful for components like the Popper which should close when the user clicks anywhere else in the document. Click-Away Listener also supports the Portal component.

The demo below shows how to hide a menu dropdown when users click anywhere else on the page:

Open menu dropdown

Edit in Chat JS TS Collapse code

```
import * as React from 'react';
import Box from '@mui/material/Box';
import ClickAwayListener from '@mui/material/ClickAwayListener';
import { SxProps } from '@mui/system';

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

  const handleClick = () => {
    setOpen((prev) => !prev);
  };

  const handleClickAway = () => {
    setOpen(false);
  };

  const styles: SxProps = {
    position: 'absolute',
    top: 28,
    right: 0,
    left: 0,
    zIndex: 1,
    border: '1px solid',
    p: 1,
    bgcolor: 'background.paper',
  };

  return (
    <ClickAwayListener onClickAway={handleClickAway}>
      <Box sx={{ position: 'relative' }}>
        <button type="button" onClick={handleClick}>
          Open menu dropdown
        </button>
        {open ? (
          <Box sx={styles}>
            Click me, I will stay visible until you click outside.
          </Box>
```

Basics

Import

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

Copy

Customization

Use with Portal

The following demo uses the Portal component to render the dropdown into a new subtree outside of the current DOM hierarchy:

Open menu dropdown

Edit in Chat JS TS Collapse code

```
import * as React from 'react';
import Box from '@mui/material/Box';
import ClickAwayListener from '@mui/material/ClickAwayListener';
import Portal from '@mui/material/Portal';
import { SxProps } from '@mui/system';

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

  const handleClick = () => {
    setOpen((prev) => !prev);
  };

  const handleClickAway = () => {
    setOpen(false);
  };

  const styles: SxProps = {
    position: 'fixed',
    width: 200,
    top: '50%',
    left: '50%',
    transform: 'translate(-50%, -50%)',
    border: '1px solid',
    p: 1,
    bgcolor: 'background.paper',
  };

  return (
    <ClickAwayListener onClickAway={handleClickAway}>
      <div>
        <button type="button" onClick={handleClick}>
          Open menu dropdown
        </button>
        {open ? (
          <Portal>
            <Box sx={styles}>
              Click me, I will stay visible until you click outside
```

Listening for leading events

By default, the Click-Away Listener component responds to **trailing events**—the *end* of a click or touch.

You can set the component to listen for **leading events** (the start of a click or touch) using the `mouseEvent` and `touchEvent` props, as shown in the following demo:

⚠ When the component is set to listen for leading events, interactions with the scrollbar are ignored.

Open menu dropdown

Edit in Chat JS TS Collapse code

```
import * as React from 'react';
import Box from '@mui/material/Box';
import ClickAwayListener from '@mui/material/ClickAwayListener';
import { SxProps } from '@mui/system';

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

  const handleClick = () => {
    setOpen((prev) => !prev);
  };

  const handleClickAway = () => {
    setOpen(false);
  };

  const styles: SxProps = {
    position: 'absolute',
    top: 28,
    right: 0,
    left: 0,
    zIndex: 1,
    border: '1px solid',
    p: 1,
    bgcolor: 'background.paper',
  };

  return (
    <ClickAwayListener
      mouseEvent="onMouseDown"
      touchEvent="onTouchStart"
      onClickAway={handleClickAway}
    >
      <Box sx={{ position: 'relative' }}>
        <button type="button" onClick={handleClick}>
          Open menu dropdown
        </button>
```

Accessibility

By default, Click-Away Listener adds an `onClick` handler to its child. This can result in screen readers announcing that the child is clickable, even though this `onClick` handler has no effect on the child itself.

To prevent this behavior, add `role="presentation"` to the child element:

```
<ClickAwayListener>
  <div role="presentation">
    <h1>non-interactive heading</h1>
  </div>
</ClickAwayListener>
```

This is also required to fix a known issue in NVDA when using Firefox that prevents the announcement of alert messages—see [this GitHub issue](#) for details.

API

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

- `<ClickAwayListener />`