



Search...

Ctrl+K



Menu

Menus display a list of choices on temporary surfaces.



Design and Development tips in your inbox. Every weekday.
ads via Carbon

A menu displays a list of choices on a temporary surface. It appears when the user interacts with a button, or other control.

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

Introduction



Menus are implemented using a collection of related components:

- Menu: The container/surface of the menu.
- Menu Item: An option for users to select from the menu.
- Menu List (optional): Alternative composable container for Menu Items—see [Composition with Menu List](#) for details.

Basic menu



A basic menu opens over the anchor element by default (this option can be [changed](#) via props). When close to a screen edge, a basic menu vertically realigns to make sure that all menu items are completely visible.

You should configure the component so that selecting an option immediately confirms it and closes the menu, as shown in the demo below.

DASHBOARD

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

```
import * as React from 'react';
import Button from '@mui/material/Button';
import Menu from '@mui/material/Menu';
import MenuItem from '@mui/material/MenuItem';

export default function BasicMenu() {
  const [anchorEl, setAnchorEl] = React.useState<null | HTMLElement>(null);
  const open = Boolean(anchorEl);
  const handleClick = (event: React.MouseEvent<HTMLButtonElement>) => {
    setAnchorEl(event.currentTarget);
  };
  const handleClose = () => {
    setAnchorEl(null);
  };

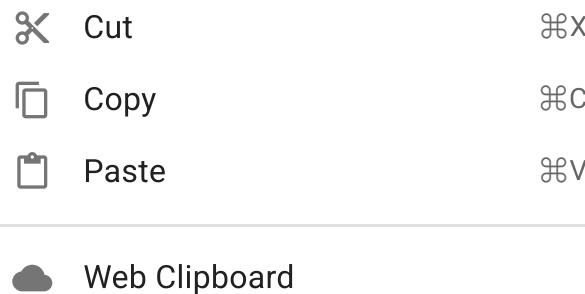
  return (
    <div>
      <Button
        id="basic-button"
        aria-controls={open ? 'basic-menu' : undefined}
        aria-haspopup="true"
        aria-expanded={open ? 'true' : undefined}
        onClick={handleClick}
      >
        Dashboard
      </Button>
      <Menu
        id="basic-menu"
        anchorEl={anchorEl}
        open={open}
        onClose={handleClose}
        slotProps={{
          list: {
            'aria-labelledby': 'basic-button',
          },
        }}
      >

```

Icon menu

In desktop viewport, padding is increased to give more space to the menu.



[Edit in Chat](#)

JS

TS

[Hide code](#)

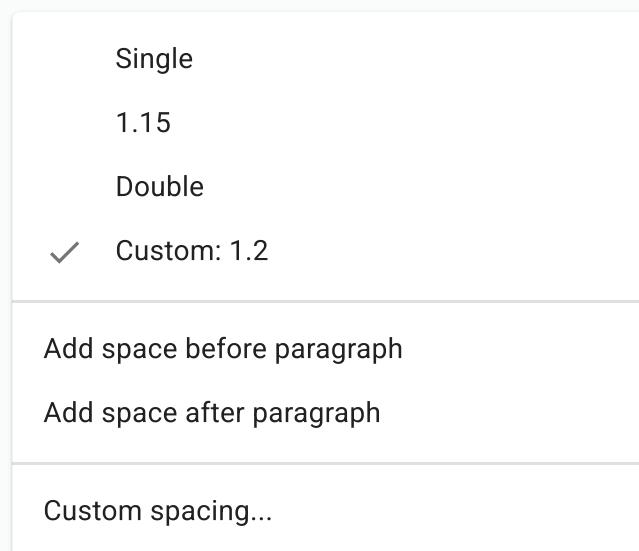
```
import Divider from '@mui/material/Divider';
import Paper from '@mui/material/Paper';
import MenuList from '@mui/material/MenuList';
import MenuItem from '@mui/material/MenuItem';
import ListItemText from '@mui/material/ListItemText';
import ListItemIcon from '@mui/material/ListItemIcon';
import Typography from '@mui/material/Typography';
import ContentCut from '@mui/icons-material/ContentCut';
import ContentCopy from '@mui/icons-material/ContentCopy';
import ContentPaste from '@mui/icons-material/ContentPaste';
import Cloud from '@mui/icons-material/Cloud';

export default function IconMenu() {
  return (
    <Paper sx={{ width: 320, maxWidth: '100%' }}>
      <MenuList>
        <MenuItem>
          <ListItemIcon>
            <ContentCut fontSize="small" />
          </ListItemIcon>
          <ListItemText>Cut</ListItemText>
          <Typography variant="body2" sx={{ color: 'text.secondary' }}>
            ⌘X
          </Typography>
        </MenuItem>
        <MenuItem>
          <ListItemIcon>
            <ContentCopy fontSize="small" />
          </ListItemIcon>
          <ListItemText>Copy</ListItemText>
          <Typography variant="body2" sx={{ color: 'text.secondary' }}>
            ⌘C
          </Typography>
        </MenuItem>
        <MenuItem>
          <ListItemIcon>
            <ContentPaste fontSize="small" />
          </ListItemIcon>
        </MenuItem>
      </MenuList>
    </Paper>
  )
}
```

Dense menu

+

For the menu that has long list and long text, you can use the `dense` prop to reduce the padding and text size.

[Edit in Chat](#)

JS

TS

[Hide code](#)

```
import Paper from '@mui/material/Paper';
import Divider from '@mui/material/Divider';
import MenuList from '@mui/material/MenuList';
import MenuItem from '@mui/material/MenuItem';
import ListItemIcon from '@mui/material/ListItemIcon';
importListItemText from '@mui/material/ListItemText';
import Check from '@mui/icons-material/Check';

export default function DenseMenu() {
  return (
    <Paper sx={{ width: 320 }}>
      <MenuList dense>
        <MenuItem>
          <ListItemText inset>Single</ListItemText>
        </MenuItem>
        <MenuItem>
          <ListItemText inset>1.15</ListItemText>
        </MenuItem>
        <MenuItem>
          <ListItemText inset>Double</ListItemText>
        </MenuItem>
        <MenuItem>
          <ListItemIcon>
            <Check />
          </ListItemIcon>
          Custom: 1.2
        </MenuItem>
        <Divider />
        <MenuItem>
          <ListItemText>Add space before paragraph</ListItemText>
        </MenuItem>
        <MenuItem>
```

```
<ListItemText>Add space after paragraph</ListItemText>
</MenuItem>
<Divider />
<MenuItem>
  <ListItemText>Custom spacing...</ListItemText>
</MenuItem>
```

Selected menu

If used for item selection, when opened, simple menus places the initial focus on the selected menu item. The currently selected menu item is set using the `selected` prop (from [ListItemText](#)). To use a selected menu item without impacting the initial focus, set the `variant` prop to "menu".

When device is locked
Show all notification content

 Edit in Chat

JS

TS

[Hide code](#)



```
import * as React from 'react';
import List from '@mui/material/List';
import ListItemButton from '@mui/material/ListItemButton';
importListItemText from '@mui/materialListItemText';
import MenuItem from '@mui/material/MenuItem';
import Menu from '@mui/material/Menu';

const options = [
  'Show some love to MUI',
  'Show all notification content',
  'Hide sensitive notification content',
  'Hide all notification content',
];

export default function SimpleListMenu() {
  const [anchorEl, setAnchorEl] = React.useState<null | HTMLElement>(null);
  const [selectedIndex, setSelectedIndex] = React.useState(1);
  const open = Boolean(anchorEl);
  const handleClickListItem = (event: React.MouseEvent<HTMLElement>) => {
    setAnchorEl(event.currentTarget);
  };

  const handleMenuItemClick = (
    event: React.MouseEvent<HTMLElement>,
    index: number,
  ) => {
    setSelectedIndex(index);
    setAnchorEl(null);
  };
}
```

```
const handleClose = () => {
  setAnchorEl(null);
};

return (
  <div>
    <List
      component="nav"
```

Positioned menu

Because the `Menu` component uses the `Popover` component to position itself, you can use the same [positioning props](#) to position it. For instance, you can display the menu on top of the anchor:

DASHBOARD

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

[Hide code](#)



```
import * as React from 'react';
import Button from '@mui/material/Button';
import Menu from '@mui/material/Menu';
import MenuItem from '@mui/material/MenuItem';

export default function PositionedMenu() {
  const [anchorEl, setAnchorEl] = React.useState<null | HTMLElement>(null);
  const open = Boolean(anchorEl);
  const handleClick = (event: React.MouseEvent<HTMLElement>) => {
    setAnchorEl(event.currentTarget);
  };
  const handleClose = () => {
    setAnchorEl(null);
  };

  return (
    <div>
      <Button
        id="demo-positioned-button"
        aria-controls={open ? 'demo-positioned-menu' : undefined}
        aria-haspopup="true"
        aria-expanded={open ? 'true' : undefined}
        onClick={handleClick}
      >
        Dashboard
      </Button>
      <Menu
        id="demo-positioned-menu"
        aria-labelledby="demo-positioned-button"
        anchorEl={anchorEl}
```

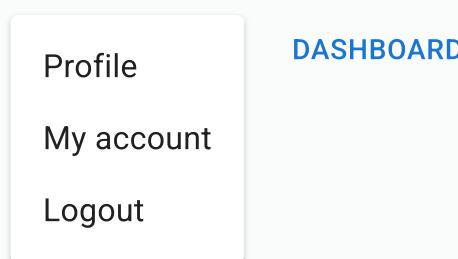
```
open={open}
onClose={handleClose}
anchorOrigin={{
  vertical: 'top',
  horizontal: 'left',
}}
transformOrigin={{
  vertical: 'top'
```

Composition with Menu List



The Menu component uses the Popover component internally. But you might want to use a different positioning strategy, or prefer not to block scrolling, for example.

The Menu List component lets you compose your own menu for these kinds of use cases—its primary purpose is to handle focus. See the demo below for an example of composition that uses Menu List and replaces the Menu's default Popover with a Popper component instead:



```
import * as React from 'react';
import Button from '@mui/material/Button';
import ClickAwayListener from '@mui/material/ClickAwayListener';
import Grow from '@mui/material/Grow';
import Paper from '@mui/material/Paper';
import Popper from '@mui/material/Popper';
import MenuItem from '@mui/material/MenuItem';
import MenuList from '@mui/material/MenuList';
import Stack from '@mui/material/Stack';

export default function MenuListComposition() {
  const [open, setOpen] = React.useState(false);
  const anchorRef = React.useRef<HTMLButtonElement>(null);

  const handleToggle = () => {
    setOpen((prevOpen) => !prevOpen);
  };

  const handleClose = (event: Event | React.SyntheticEvent) => {
    if (
      anchorRef.current &&
      anchorRef.current.contains(event.target as HTMLElement)
```

```
) {  
  return;  
}  
  
 setOpen(false);  
};  
  
function handleListKeyDown(event: React.KeyboardEvent) {  
  if (event.key === 'Tab') {  
    event.preventDefault();  
    setOpen(false);  
  } else if (event.key === 'Escape') {  
    setOpen(false);  
  }  
}
```

Account menu



Menu content can be mixed with other components like Avatar.

Contact Profile

M

Edit in Chat JS TS

Hide code

```
import * as React from 'react';  
import Box from '@mui/material/Box';  
import Avatar from '@mui/material/Avatar';  
import Menu from '@mui/material/Menu';  
import MenuItem from '@mui/material/MenuItem';  
import ListItemIcon from '@mui/material(ListItemIcon';  
import Divider from '@mui/material/Divider';  
import IconButton from '@mui/material/IconButton';  
import Typography from '@mui/material/Typography';  
import Tooltip from '@mui/material/Tooltip';  
import PersonAdd from '@mui/icons-material/PersonAdd';  
import Settings from '@mui/icons-material/Settings';  
import Logout from '@mui/icons-material/Logout';  
  
export default function AccountMenu() {  
  const [anchorEl, setAnchorEl] = React.useState<null | HTMLElement>(null);  
  const open = Boolean(anchorEl);  
  const handleClick = (event: React.MouseEvent<HTMLElement>) => {  
    setAnchorEl(event.currentTarget);  
};  
  const handleClose = () => {  
    setAnchorEl(null);  
};  
  return (  
    <React.Fragment>
```

```
<Box sx={{ display: 'flex', alignItems: 'center', textAlign: 'center' }}>
  <Typography sx={{ minWidth: 100 }}>Contact</Typography>
  <Typography sx={{ minWidth: 100 }}>Profile</Typography>
  <Tooltip title="Account settings">
    <IconButton
      onClick={handleClick}
      size="small"
      sx={{ ml: 2 }}
      aria-controls={open ? 'account-menu' : undefined}
      aria-haspopup="true"
      aria-expanded={open ? 'true' : undefined}
    >
      <Avatar sx={{ width: 32, height: 32 }}>M</Avatar>
    </IconButton>
  </Tooltip>
</Box>
```

Customization



Here is an example of customizing the component. You can learn more about this in the [overrides documentation page](#).

OPTIONS ▾

Edit in Chat

JS

TS

Hide code



```
import * as React from 'react';
import { styled, alpha } from '@mui/material/styles';
import Button from '@mui/material/Button';
import Menu, { MenuProps } from '@mui/material/Menu';
import MenuItem from '@mui/material/MenuItem';
import EditIcon from '@mui/icons-material/Edit';
import Divider from '@mui/material/Divider';
import ArchiveIcon from '@mui/icons-material/Archive';
import FileCopyIcon from '@mui/icons-material/FileCopy';
import MoreHorizIcon from '@mui/icons-material/MoreHoriz';
import KeyboardArrowDownIcon from '@mui/icons-material/KeyboardArrowDown';

const StyledMenu = styled((props: MenuProps) => (
  <Menu
    elevation={0}
    anchorOrigin={{
      vertical: 'bottom',
      horizontal: 'right',
    }}
    transformOrigin={{
      vertical: 'top',
      horizontal: 'right',
    }}
    {...props}
  />
))(({ theme }) => ({
```

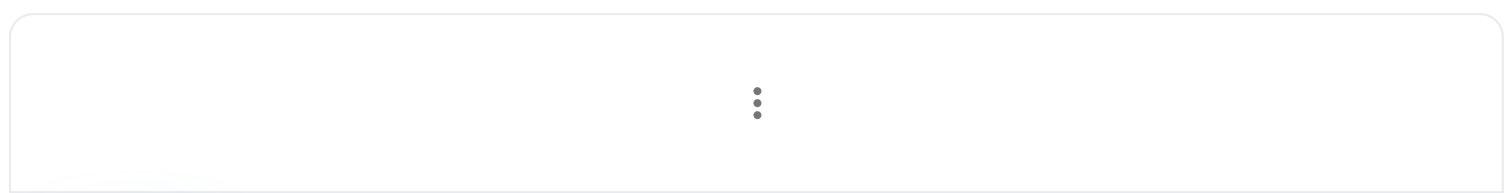
```
'& .MuiPaper-root': {
  borderRadius: 6,
  marginTop: theme.spacing(1),
  minWidth: 180,
  color: 'rgb(55, 65, 81)',
  boxShadow:
    'rgb(255, 255, 255) 0px 0px 0px 0px, rgba(0, 0, 0, 0.05) 0px 0px 0px 1px, rgba(0, 0, 0, 0.1) 0px
  '& .MuiMenu-list': {
    padding: '4px 0',
  },
  '& .MuiMenuItem-root': {
    '& .MuiSvgIcon-root': {
```

The `MenuItem` is a wrapper around `ListItem` with some additional styles. You can use the same list composition features with the `MenuItem` component:

 If you are looking for inspiration, you can check [MUI Treasury's customization examples](#).

Max height menu

If the height of a menu prevents all menu items from being displayed, the menu can scroll internally.



 [Edit in Chat](#) JS TS

[Hide code](#)      

```
import * as React from 'react';
import IconButton from '@mui/material/IconButton';
import Menu from '@mui/material/Menu';
import MenuItem from '@mui/material/MenuItem';
import MoreVertIcon from '@mui/icons-material/MoreVert';

const options = [
  'None',
  'Atria',
  'Callisto',
  'Dione',
  'Ganymede',
  'Hangouts Call',
  'Luna',
  'Oberon',
  'Phobos',
  'Pyxis',
  'Sedna',
  'Titania',
  'Triton',
  'Umbriel',
];
```

```
const ITEM_HEIGHT = 48;

export default function LongMenu() {
  const [anchorEl, setAnchorEl] = React.useState<null | HTMLElement>(null);
  const open = Boolean(anchorEl);
  const handleClick = (event: React.MouseEvent<HTMLElement>) => {
    setAnchorEl(event.currentTarget);
  };
  const handleClose = () => {
    setAnchorEl(null);
  };

  return (
    <div>
      <button>
```

Limitations

There is [a flexbox bug](#) that prevents `text-overflow: ellipsis` from working in a flexbox layout. You can use the `Typography` component with `nowrap` to workaround this issue:

- A short message
- ! A very long text that overflows
- ✉ A very long text that ...

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

[Hide code](#)



```
import MenuList from '@mui/material/MenuList';
import MenuItem from '@mui/material/MenuItem';
import Paper from '@mui/material/Paper';
import ListItemIcon from '@mui/material/ListItemIcon';
import Typography from '@mui/material/Typography';
import DraftsIcon from '@mui/icons-material/Drafts';
import SendIcon from '@mui/icons-material/Send';
import PriorityHighIcon from '@mui/icons-material/PriorityHigh';

export default function TypographyMenu() {
  return (
    <Paper sx={{ width: 230 }}>
      <MenuList>
        <MenuItem>
          <ListItemIcon>
            <SendIcon fontSize="small" />
          </ListItemIcon>
          <Typography variant="inherit">A short message</Typography>
        </MenuItem>
      </MenuList>
    </Paper>
  );
}
```

```
<MenuItem>
  <ListItemIcon>
    <PriorityHighIcon fontSize="small" />
  </ListItemIcon>
  <Typography variant="inherit">A very long text that overflows</Typography>
</MenuItem>
<MenuItem>
  <ListItemIcon>
    <DraftsIcon fontSize="small" />
  </ListItemIcon>
  <Typography variant="inherit" nowrap>
    A very long text that overflows
  </Typography>
</MenuItem>
</MenuList>
</Paper>
);
}
```

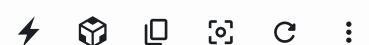
Change transition

Use a different transition.

DASHBOARD

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

[Hide code](#)



```
import * as React from 'react';
import Button from '@mui/material/Button';
import Menu from '@mui/material/Menu';
import MenuItem from '@mui/material/MenuItem';
import Fade from '@mui/material/Fade';

export default function FadeMenu() {
  const [anchorEl, setAnchorEl] = React.useState<null | HTMLElement>(null);
  const open = Boolean(anchorEl);
  const handleClick = (event: React.MouseEvent<HTMLElement>) => {
    setAnchorEl(event.currentTarget);
  };
  const handleClose = () => {
    setAnchorEl(null);
  };

  return (
    <div>
      <Button
        id="fade-button"
        aria-controls={open ? 'fade-menu' : undefined}
        aria-haspopup="true"
      >
```

```
    aria-expanded={open ? 'true' : undefined}
    onClick={handleClick}
  >
  Dashboard
</Button>
<Menu
  id="fade-menu"
  slotProps={{
    list: {
      'aria-labelledby': 'fade-button',
    },
  }}
  slots={{ transition: Fade }}
  anchorEl={anchorEl}
  open={open}
  onClose={handleClose}
```

Context menu



Here is an example of a context menu. (Right click to open.)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam ipsum purus, bibendum sit amet vulputate eget, porta semper ligula. Donec bibendum vulputate erat, ac fringilla mi finibus nec. Donec ac dolor sed dolor porttitor blandit vel vel purus. Fusce vel malesuada ligula. Nam quis vehicula ante, eu finibus est. Proin ullamcorper fermentum orci, quis finibus massa. Nunc lobortis, massa ut rutrum ultrices, metus metus finibus ex, sit amet facilisis neque enim sed neque. Quisque accumsan metus vel maximus consequat. Suspendisse lacinia tellus a libero volutpat maximus.

Edit in Chat

JS

TS

[Hide code](#)



```
import * as React from 'react';
import Menu from '@mui/material/Menu';
import MenuItem from '@mui/material/MenuItem';
import Typography from '@mui/material/Typography';

export default function ContextMenu() {
  const [contextMenu, setContextMenu] = React.useState<{
    mouseX: number;
    mouseY: number;
  } | null>(null);

  const handleContextMenu = (event: React.MouseEvent) => {
    event.preventDefault();

    setContextMenu(
      contextMenu === null
        ? {
            mouseX: event.clientX + 2,
            mouseY: event.clientY + 2,
          }
        : {
            ...contextMenu,
            mouseX: event.clientX + 2,
            mouseY: event.clientY + 2,
          }
    );
  };

  return (
    <div>
      <button>Open Context Menu</button>
      <Menu
        anchorEl={null}
        open={false}
        onClose={handleContextMenu}
      >
        <MenuItem>Action 1</MenuItem>
        <MenuItem>Action 2</MenuItem>
        <MenuItem>Action 3</MenuItem>
      </Menu>
    </div>
  );
}
```

```
    mouseY: event.clientY - 6,
  }
  : // repeated contextmenu when it is already open closes it with Chrome 84 on Ubuntu
  // Other native context menus might behave different.
  // With this behavior we prevent contextmenu from the backdrop to re-locale existing context
  null,
);

// Prevent text selection lost after opening the context menu on Safari and Firefox
const selection = document.getSelection();
if (selection && selection.rangeCount > 0) {
  const range = selection.getRangeAt(0);

  setTimeout(() => {
    selection.addRange(range);
  });
}
};

const handleClose = () => {
```

Grouped Menu

Display categories with the `ListSubheader` component.

DASHBOARD

 Edit in Chat JS TS

Hide code



```
import * as React from 'react';
import Button from '@mui/material/Button';
import ListSubheader from '@mui/material/ListSubheader';
import Menu from '@mui/material/Menu';
import MenuItem from '@mui/material/MenuItem';
import { styled } from '@mui/material/styles';

const StyledListHeader = styled(ListSubheader)({
  backgroundImage: 'var(--Paper-overlay)',
});

export default function GroupedMenu() {
  const [anchorEl, setAnchorEl] = React.useState<null | HTMLElement>(null);
  const open = Boolean(anchorEl);
  const handleClick = (event: React.MouseEvent<HTMLButtonElement>) => {
    setAnchorEl(event.currentTarget);
  };
  const handleClose = () => {
    setAnchorEl(null);
};
```

```

return (
  <div>
    <Button
      id="basic-button"
      aria-controls={open ? 'grouped-menu' : undefined}
      aria-haspopup="true"
      aria-expanded={open ? 'true' : undefined}
      onClick={handleClick}
    >
      Dashboard
    </Button>
    <Menu
      id="grouped-menu"
      anchorEl={anchorEl}
      open={open}
      onClose={handleClose}
      slotProps={...}
    >

```

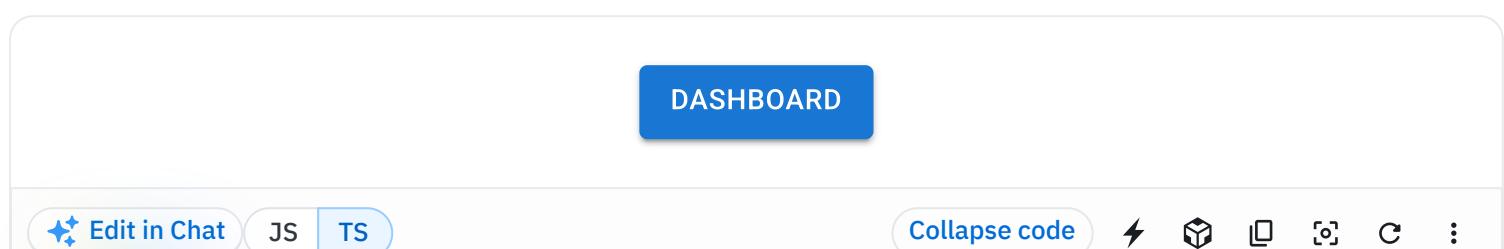
Supplementary projects

For more advanced use cases you might be able to take advantage of:

material-ui-popup-state

 Star 472 downloads 1.1M/month

The package [material-ui-popup-state](#) that takes care of menu state for you in most cases.



DASHBOARD

Edit in Chat JS TS Collapse code ⚡ 📁 🖨️ ⚙️ C ⋮

```

import * as React from 'react';
import Button from '@mui/material/Button';
import Menu from '@mui/material/Menu';
import MenuItem from '@mui/material/MenuItem';
import PopupState, { bindTrigger, bindMenu } from 'material-ui-popup-state';

export default function MenuPopupState() {
  return (
    <PopupState variant="popover" popupId="demo-popup-menu">
      {(popupState) => (
        <React.Fragment>
          <Button variant="contained" {...bindTrigger(popupState)}>
            Dashboard
          </Button>
          <Menu {...bindMenu(popupState)}>
            <MenuItem onClick={popupState.close}>Profile</MenuItem>

```

```
<MenuItem onClick={popupState.close}>My account</MenuItem>
  <MenuItem onClick={popupState.close}>Logout</MenuItem>
</Menu>
</React.Fragment>
)}
</PopupState>
);
}
```

API



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

- [`<ClickAwayListener />`](#)
- [`<Menu />`](#)
- [`<MenuItem />`](#)
- [`<MenuList />`](#)
- [`<Popover />`](#)
- [`<Popper />`](#)

Edit this page

Was this page helpful?

[Link](#)

[Pagination](#)