

App Bar 应用栏

应用栏组件展示了与当前屏幕息息相关的信息和操作。

而顶部应用栏（App Bar）则提供与当前屏幕相关的内容和操作。该组件常用于展示品牌、展示标题、提供导航和一些可操作的内容。

它既可以用于转换为上下文相关的操作栏，又可以直接充当导航栏。

```
{{"component": "modules/components/ComponentLinkHeader.js"}}
```

简单的应用栏

```
{{"demo": "ButtonAppBar.js", "bg": true}}
```

带有菜单的应用栏

```
{{"demo": "MenuAppBar.js", "bg": true}}
```

带有响应式菜单的应用栏

```
{{"demo": "ResponsiveAppBar.js", "bg": true}}
```

带有搜索输入框的应用栏

侧边搜索栏。

```
{{"demo": "SearchAppBar.js", "bg": true}}
```

带有主要搜索输入框的应用栏

主要搜索栏。

```
{{"demo": "PrimarySearchAppBar.js", "bg": true}}
```

紧凑模式（仅限桌面模式）

```
{{"demo": "DenseAppBar.js", "bg": true}}
```

Prominent

A prominent app bar.

```
{{"demo": "ProminentAppBar.js", "bg": true}}
```

底部应用栏

```
{{"demo": "BottomAppBar.js", "iframe": true, "maxWidth": 400}}
```

固定位置

当渲染一个固定位置的应用栏时，元素的尺寸不会影响页面的其余内容。这可能导致部分内容会被挡在应用栏后面使其无法可见。下面有 3 种解决方案：

1. 使用 `position = "sticky"` 代替 `fixed`。⚠️ `sticky` 不支持 IE11。
2. 可以渲染第二个 `<Toolbar />` 组件：

```
function App() {
  return (
    <React.Fragment>
      <AppBar position="fixed">
        <Toolbar>{/* 内容 */</Toolbar>
      </AppBar>
      <Toolbar />
    </React.Fragment>
  );
}
```

3. 也可以用 `theme.mixins.toolbar` 的 CSS：

```
const Offset = styled('div')(({ theme }) => theme.mixins.toolbar);

function App() {
  return (
    <React.Fragment>
      <AppBar position="fixed">
        <Toolbar>{/* 内容 */</Toolbar>
      </AppBar>
      <Offset />
    </React.Fragment>
  );
}
```

滚动

您可以使用 `useScrollTrigger()` 这个 hook 来响应用户触发的滚动操作。

隐藏应用栏

向下滚动会隐藏应用栏，这样一来会留有更多的空间进行阅读。

```
{{"demo": "HideAppBar.js", "iframe": true}}
```

变高的应用栏

应用栏阴影会在滚动时加深，以表明用户还未到页面的顶部。

```
{{"demo": "ElevateAppBar.js", "iframe": true}}
```

回到顶部

在滚动的时候，会出现一个浮动操作按钮，这样以便于返回页面的顶部。

```
{{"demo": "BackToTop.js", "iframe": true}}
```

```
useScrollTrigger([options]) => trigger
```

参数

1. `options` (*object* [optional]):
 - `options.disableHysteresis` (*bool* [optional]): 默认值是 `false`。禁用迟滞的效果。在决定 `trigger` 的值时会忽略在滚动的方向。
 - `options.target` (*Node* [optional]): 默认值是 `window`。
 - `options.threshold` (*number* [optional]): 默认值是 `100`。当垂直滚动恰好超过（但不包括）此阈值时，会改变 `trigger` 的值。

返回结果

`trigger`：滚动位置是否与目标值匹配？

例子

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

function HideOnScroll(props) {
  const trigger = useScrollTrigger();
  return (
    <Slide in={!trigger}>
      <div>Hello</div>
    </Slide>
  );
}
```

在深色模式上启用颜色

根据 [Material Design 规范](#)，`color` 属性在深色模式下对应用栏的外观没有影响。您可以通过设置 `enableColorOnDark` 属性为 `true` 来覆盖此行为。

```
{{"demo": "EnableColorOnDarkAppBar.js", "bg": true}}
```