

Modal 模态框组件

模态框组件可以用来快速创建对话框、弹出窗口，灯箱等任何你所需的组件。

组件会在背景组件上层渲染其 `children` 节点。 模态框 提供了一些重要的功能：

- 🔒 管理了当一次只显示一个不能满足时的模态框堆叠。
- 🔒 创建了一个背景暗化组件，这样能禁用模态框之外的交互。
- 🔒 在模态框打开时禁用页面内容的滚动。
- 🔗 它妥善管理焦点；移动到模态内容，并保持内容一直存在直到模态框关闭。
- 🔗 自动添加适当的 ARIA 角色。

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

术语注释。“模态框” (Modal) 这个词有时也被用来指代“对话框”，但是这种用法属于误用。模态框的窗口描述了 UI 的一部分。如果一个元素 阻挡了用户与应用的其它部分的互动，这个元素就是模态的。

当你创建一个模态对话框时，使用 [对话框 \(Dialog\)](#) 组件比直接使用模态框更佳。以下的组件将模态框作为一个低级别的组件运用：

- [Dialog](#)
- [Drawer \(抽屉\)](#)
- [Menu](#)
- [弹出框 \(Popover\)](#)

Basic modal

```
{{"demo": "BasicModal.js"}}
```

请注意，您可以通过 `outline: 0` 属性来禁用模态框的边缘（通常为蓝色或金色）。

Unstyled

- 📦 [4.7 kB gzipped](#)

The modal also comes with an unstyled version. The modal also comes with an unstyled version. It's ideal for doing heavy customizations and minimizing bundle size. It's ideal for doing heavy customizations and minimizing bundle size.

```
import ModalUnstyled from '@mui/base/ModalUnstyled';
```

```
{{"demo": "ModalUnstyled.js"}}
```

Nested modal

Modals can be nested, for example a select within a dialog, but stacking of more than two modals, or any two modals with a backdrop is discouraged.

```
{{"demo": "NestedModal.js"}}
```

过渡动画

通过使用一个过渡组件，您可以给模态框的打开/关闭状态加上动画效果。此组件应遵守以下条件：

- 作为模态框的直接子元素。
- 有一个 `in` 属性。这对应于打开/关闭的状态。
- 当进入过渡时调用 `onEnter` 回调属性。
- 当退出过渡完成后应该调用 `onExited` 回调属性。这两个回调属性保证了模态框在关闭并展示完过渡动画时，将会移除子内容。

模态框已经内嵌支持 [react-transition-group](#)。

```
{{"demo": "TransitionsModal.js"}}
```

Alternatively, you can use [react-spring](#).

```
{{"demo": "SpringModal.js"}}
```

性能

模态的内容在关闭时是不被加载的。如果你需要将内容提供给搜索引擎或在你的模态框中渲染昂贵的组件树，同时还要优化交互响应能力，那么你可以启用 `keepMounted` 属性来改变这一默认行为：

```
<Modal keepMounted />
```

```
{{"demo": "KeepMountedModal.js", "defaultCodeOpen": false}}
```

As with any performance optimization, this is not a silver bullet. As with any performance optimization, this is not a silver bullet. Be sure to identify bottlenecks first, and then try out these optimization strategies. Be sure to identify bottlenecks first, and then try out these optimization strategies.

服务端渲染的模态框

React [不支持](#)服务端渲染的 [createPortal\(\)](#) API。若您想显示模态框，则需要通过 `disablePortal` 这个属性来禁用 portal 功能：

```
{{"demo": "ServerModal.js"}}
```

设计局限

焦点陷阱

如果用户试图将焦点离开模态框，模态框会将丢失的焦点移回到组件的主体。

This is done for accessibility purposes. However, it might create issues. This is done for accessibility purposes. However, it might create issues. 如果用户需要与页面的其他部分进行交互，例如当您需要使用聊天窗口时，那么就可以禁用该行为：

```
<Modal disableEnforceFocus />
```

无障碍设计

(WAI-ARIA: https://www.w3.org/TR/wai-aria-practices/#dialog_modal)

- 记得用 `aria-labelledby="id..."` 属性来指向 `Modal` 的标题。此外，您可以使用 `aria-describedby="id..."` 属性来为 `Modal` 组件添加一段描述。

```
<Modal aria-labelledby="modal-title" aria-describedby="modal-description">
  <h2 id="modal-title">我的标题</h2>
  <p id="modal-description">我的描述</p>
</Modal>
```

- 这篇 [WAI-ARIA authoring practices](#) 里的方法帮助你通过模态窗口里的内容，为最相关的元素设置初始焦点。
- Keep in mind that a "modal window" overlays on either the primary window or another modal window. Windows under a modal are **inert**. 也就是说，用户不能与当前处于活跃状态下的模态框之外的内容进行交互。因为这可能会造成[冲突行为](#)。