过渡动画

过渡动画有利于增强 UI 的表现力并且让人更易于使用。

Material-UI 提供了一系列的过渡效果，你可以将一些基本的 动作 添加到你的应用组件中。

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

Collapse 折叠

从子元素的起始边缘开始展开。 如果你需要水平折叠，请使用 `orientation` 属性。 `collapsedHeight` 属性可以用于设置未扩展时的最小高度值。

{{"demo": "SimpleCollapse.js", "bg": true}}

Fade 淡入淡出

从透明淡入至不透明。

{{"demo": "SimpleFade.js", "bg": true}}

Grow 扩展

Expands outwards from the center of the child element, while also fading in from transparent to opaque.

The second example demonstrates how to change the `transform-origin`, and conditionally applies the `timeout` prop to change the entry speed.

{{"demo": "SimpleGrow.js", "bg": true}}

Slide 滑动

从屏幕边缘滑入。 使用 `direction` 属性能够控制从屏幕的哪一个边缘开始过渡。

过渡组件的 `mountOnEnter` 属性，保证了只有 `in` 是 `true` 时，子组件才会被渲染。 这可以保证相对上定位好的组件不会从屏幕外面的位置滚动到视图中。 同样的， 在组件从屏幕中过渡完后， `unmountOnExit` 属性将次组件从 DOM 中移除。

{{"demo": "SimpleSlide.js", "bg": true}}

Slide relative to a container

The Slide component also accepts `container` prop, which is a reference to a DOM node. If this prop is set, the Slide component will slide from the edge of that DOM node.

从子元素的中心向外扩展。

Zoom 放大

从子元素的中心向外扩展。

此示例还演示了如何延迟过渡的开始。

{{"demo": "SimpleZoom.js", "bg": true}}

TransitionGroup 动画组

- 为了更好地支持服务器渲染，Material-UI 为一些动画组件的子组件提供了一个 `style` 属性，（Fade, Grow, Zoom, Slide）。 为了让动画如期实现，必须将 `style` 属性应用到 DOM 上。
- **Forward the ref**: The transition components require the first child element to forward its ref to the DOM node. For more details about ref, check out Caveat with refs For more details about ref, check out Caveat with refs
- **Single element**: The transition components require only one child element (`React.Fragment` is not allowed).

```
// 'props' 对象包含一个 'style' 属性。
// 你需要将这个属性提供给 `div` 元素，如下所示。
const MyComponent = React.forwardRef((props, ref) {
  return (
    <div ref={ref} {...props}>
      Fade
    </div>
  );
})

export default Main() {
  return (
    <Fade>
      {/* MyComponent must the only child */}
      <MyComponent />
    </Fade>
  );
}
```

TransitionComponent 属性

To animate a component when it is mounted or unmounted, you can use the `TransitionGroup` component from *react-transition-group*. As components are added or removed, the `in` prop is toggled automatically by `TransitionGroup`.

{{"demo": "TransitionGroupExample.js"}}

TransitionComponent 属性

有些 Material-UI 组件在内部也在使用这些过渡动画。 它们接受一个 `TransitionComponent` 属性来定制默认的动画。 您可以使用上述的任何组件或者是您自己的组件。 它应遵守以下条件：

- 接受一个 `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当在一个关闭的状态并展示完过渡动画时，才会移除子内容。

For more information on creating a custom transition, visit the *react-transition-group* `Transition` documentation. 你还可以访问一些组件的专用部分：

- Modal
- Dialog
- Popper
- Snackbar（消息条）
- Tooltip

Performance & SEO

The content of transition component is mounted by default even if `in={false}`. This default behavior has server-side rendering and SEO in mind. The content of transition component is mounted by default even if `in={false}`. This default behavior has server-side rendering and SEO in mind. If you render expensive component trees inside your transition it might be a good idea to change this default behavior by enabling the `unmountOnExit` prop:

```
<Fade in={false} unmountOnExit />
```

不过对所有情况下的性能优化，这并不是灵丹妙药。 Be sure to identify bottlenecks first and then try out these optimization strategies.