

Ícones

Orientação e sugestões para usar ícones com o Material-UI.

Material-UI fornece suporte de ícones de três maneiras:

1. Padronizados como [ícones do Material Design](#) e exportados como componentes do React (ícones SVG).
2. Com o componente [SvgIcon](#), um wrapper React para ícones SVG customizados.
3. Com o componente [Icon](#), um wrapper React para ícones de fonte customizados.

Ícones Material

Google has created over 2,000 official Material icons, each in five different "themes" (see below). Para cada ícone SVG, exportamos o respectivo componente React do pacote @material-ui/icons. Você pode [pesquisar na lista completa destes ícones](#).

Instalação

Instale o pacote no diretório do seu projeto com:

```
// usando npm
npm install @mui/icons-material

// usando yarn
yarn add @mui/icons-material
```

Esses componentes usam o componente `SvgIcon` do Material-UI para renderizar o caminho SVG de cada ícone, e por isso tem uma dependência com `@mui/material`.

Se você ainda não estiver usando Material-UI no seu projeto, você pode adicioná-lo com:

```
// utilizando o npm
npm install @material-ui/core

// utilizando o yarn
yarn add @material-ui/core
```

Uso

Importe ícones usando uma destas duas opções:

- Opção 1:

```
import AccessAlarmIcon from '@mui/icons-material/AccessAlarm';
import ThreeDRotation from '@mui/icons-material/ThreeDRotation';
```

- Opção 2:

```
import { AccessAlarm, ThreeDRotation } from '@mui/icons-material';
```

O mais seguro para o tamanho do pacote é a opção 1, mas alguns desenvolvedores preferem a opção 2. Certifique-se de seguir o guia [minimizando o tamanho do pacote](#) antes de usar a segunda abordagem.

Cada ícone Material também tem um "tema": Filled (padrão), Outlined, Rounded, Two-tone, e Sharp. Para importar o componente de ícone com um tema diferente do padrão, acrescente o nome do tema ao nome do ícone. Por exemplo, para usar o ícone `@material-ui/icons/Delete`, temos:

- Tema Filled (preenchido que é o padrão) é exportado como `@material-ui/icons/Delete`,
- Tema Outlined (contornado) é exportado como `@material-ui/icons/DeleteOutlined`,
- Tema Rounded (arredondado) é exportado como `@material-ui/icons/DeleteRounded`,
- Tema Two tone (dois tons) é exportado como `@material-ui/icons/DeleteTwoTone`,
- Tema Sharp (pontagudo) é exportado como `@material-ui/icons/DeleteSharp`.

Note: The Material Design guidelines name the icons using "snake_case" naming (for example `delete_forever`, `add_a_photo`), while `@material-ui/icons` exports the respective icons using "PascalCase" naming (for example `DeleteForever`, `AddAPhoto`). Há três exceções a essa regra de nomenclatura: `3d_rotation` exportado como `ThreeDRotation`, `4k` exportado como `FourK` e `360` exportado como `ThreeSixty`.

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

Testando

Para fins de teste, cada ícone exposto do `@material-ui/icons` tem um atributo `data-testid` com o nome do ícone. Por exemplo:

```
import DeleteIcon from '@mui/icons-material/Delete';
```

tem o seguinte atributo assim que montado:

```
<svg data-testid="DeleteIcon"></svg>
```

SvgIcon

Se você precisa de um ícone SVG customizado (não disponível nos [ícones Material](#)) você pode usar encapsular com `SvgIcon`. Este componente estende o elemento nativo `<svg>`:

- Ele vem internamente com a acessibilidade.
- Os elementos SVG devem ser dimensionados para uma visualização de 24x24px, de modo que o ícone resultante possa ser usado como está, ou incluído como filho para outros componentes de Material-UI que usam ícones. This can be customized with the `viewBox` attribute. To inherit the `viewBox` value from the original image, the `inheritViewBox` prop can be used.
- Por padrão, o componente herda a cor atual. Opcionalmente, você pode aplicar uma das cores do tema usando a propriedade `color`.

```
function HomeIcon(props) {
  return (
    <SvgIcon {...props}>
      <path d="M10 20v-6h4v6h5v-8h3L12 3 2 12h3v8z" />
    </SvgIcon>
  );
}
```

```
    } ;  
  }
```

Cor

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

Tamanho

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

Propriedade Componente

Você pode usar o `SvgIcon` para encapsular seus ícones, mesmo que estes estejam salvos no formato `.svg`. [svgr](#) has loaders to import SVG files and use them as React components. Por exemplo, com webpack:

```
// webpack.config.js  
{  
  test: /\.svg$/,  
  use: ['@svgr/webpack'],  
}  
  
// ---  
import StarIcon from './star.svg';  
  
<SvgIcon component={StarIcon} viewBox="0 0 600 476.6" />
```

Também é possível usá-lo com "url-loader" ou "file-loader". Esta é a abordagem usada pelo Create React App.

```
// webpack.config.js  
{  
  test: /\.svg$/,  
  use: ['@svgr/webpack', 'url-loader'],  
}  
  
// ---  
import { ReactComponent as StarIcon } from './star.svg';  
  
<SvgIcon component={StarIcon} viewBox="0 0 600 476.6" />
```

createSvgIcon

O site materialdesignicons.com fornece mais de 2.000 ícones. Para o ícone desejado, copie o SVG `path` que eles fornecem, e use-o como elemento filho no componente `SvgIcon`, ou com `createSvgIcon()`.

```
const HomeIcon = createSvgIcon(  
  <path d="M10 20v-6h4v6h5v-8h3L12 3 2 12h3v8z" />,  
  'Home',  
);
```

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

Fonte Awesome

Se você descobrir que há problemas de layout ao usar FontAwesomelcon de `@fortawesome/react-fontawesome`, você pode tentar passar os dados SVG da Font Awesome diretamente para SvgIcon.

[Fonte Awesome](#) pode ser usada com o componente `Icon` da seguinte forma:

Nota: [mdi-material-ui](#) já agrupou cada um desses ícones SVG com o componente `SvgIcon`, para que você não precise fazer isso.

A propriedade `fullWidth` de FontAwesomelcon também pode ser usada para aproximar as dimensões corretas, mas não é garantida.

Fonte Material icons

MDI

[materialdesignicons.com](#) provides over 2,000 icons. Ele pode ser usado para encapsular um caminho SVG com um componente SvgIcon. Este componente estende o elemento nativo `<svg>`:

Nota: A biblioteca [mdi-material-ui](#) já agrupou cada um desses ícones SVG com o componente `SvgIcon`, para que você não precise fazer isso.

Ícone (ícones de fonte)

The `Icon` component will display an icon from any icon font that supports ligatures. As a prerequisite, you must include one, such as the [Material icon font](#) in your project. Para usar um ícone, simplesmente coloque o nome do ícone (font ligature) com o componente `Icon`, por exemplo:

```
import Icon from '@material-ui/core/Icon';

<Icon>star</Icon>;
```

Por padrão, um ícone herdar a cor do texto atual. Opcionalmente, você pode definir a cor do ícone usando uma das propriedades de cor do tema: `primary`, `secondary`, `action`, `erro` & `disabled`.

Fonte Material icons

`Icon` irá por padrão definir o nome de classe base correto para a fonte Material Icons (variante filled). Tudo que você precisa fazer é carregar a fonte, por exemplo, através do Google Web Fonts:

```
<link
  rel="stylesheet"
  href="https://fonts.googleapis.com/icon?family=Material+Icons"
/>
```

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

Fonte customizada

Para outras fontes, você pode customizar o nome de classe base usando a propriedade `baseClassName`. Por exemplo, você pode exibir ícones de dois tons com Material Design:

```
import Icon from '@material-ui/core/Icon';

<link
  rel="stylesheet"
  href="https://fonts.googleapis.com/css?family=Material+Icons+Two+Tone"
  // Importe a variante MD de dois tons          ^^^^^^^^
/>;
```

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

Nome da classe base global

Modificar a propriedade `baseClassName` para cada uso feito do componente é repetitivo. Você pode alterar a propriedade padrão globalmente com o tema

```
const theme = createTheme({
  components: {
    MuiIcon: {
      defaultProps: {
        // Replace the `material-icons` default value.
        baseClassName: 'material-icons-two-tone',
      },
    },
  },
});
```

Então, você pode usar a fonte de dois tons diretamente:

```
<Icon>add_circle</Icon>
```

Fonte Awesome

A [fonte Awesome](#) pode ser usada com o componente `Icon` da seguinte forma:

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

Note que os ícones da fonte Awesome não foram projetados como os ícones do Material Design (compare as duas demonstrações anteriores). Os ícones são cortados para usar todo o espaço disponível. Você pode ajustar isso com uma sobrescrita global:

```
const theme = createTheme({
  components: {
    MuiIcon: {
      styleOverrides: {
        root: {
          // Match 24px = 3 * 2 + 1.125 * 16
          boxSizing: 'content-box',
          padding: 3,
          fontSize: '1.125rem',
        },
      },
    },
  },
});
```

```
    },  
    },  
  });  
};
```

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

Fonte vs SVG. Qual abordagem usar?

Ambas as abordagens funcionam bem, no entanto, existem algumas diferenças sutis, especialmente em termos de desempenho e qualidade de renderização. Sempre que possível, utilize o SVG, pois permite a divisão do código, suporta mais ícones e renderiza mais rápido e melhor.

Para maiores detalhes, dê uma olhada no [porque o GitHub migrou ícones de fonte para ícones SVG](#).

Acessibilidade

Ícones podem transmitir todos os tipos de informações significativas, então é importante garantir que eles estejam apropriadamente acessíveis. Há dois casos de uso que você deve considerar:

- **Ícones decorativos** que são usados apenas para reforço visual ou de marca. Se eles forem removidos da página, os usuários ainda entenderiam e poderiam usar sua página.
- **Semantic icons** are ones that you're using to convey meaning, rather than just pure decoration. Isso inclui ícones sem texto ao lado deles que são usados como controles interativos — botões, elementos de forma, toggles, etc.

Ícones decorativos

If your icons are purely decorative, you're already done! O atributo `aria-hidden=true` foi adicionado para que seus ícones estejam adequadamente acessíveis (invisíveis).

Ícones semânticos

Ícones SVG semânticos

Você deve incluir a propriedade `titleAccess` com um valor significativo. O atributo `role="img"` e o elemento `<title>` foram adicionados para que seus ícones sejam corretamente acessíveis.

No caso de elementos interativos focalizáveis, por exemplo, quando usado com um botão de ícone, você pode usar a propriedade `aria-label`:

```
import IconButton from '@material-ui/core/IconButton';  
import SvgIcon from '@material-ui/core/SvgIcon';  
  
// ...  
  
<IconButton aria-label="deletar">  
  <SvgIcon>  
    <path d="M20 121-1.41-1.41L13 16.17V4h-2v12.17l-5.58-5.59L4 1218 8 8-8z" />  
  </SvgIcon>  
</IconButton>;
```

Ícones de fonte semânticos

Você precisa fornecer um texto alternativo que só seja visível para tecnologia assistiva.

```
import Box from '@material-ui/core/Box';
import Icon from '@material-ui/core/Icon';
import { visuallyHidden } from '@material-ui/utils';

// ...

<Icon>add_circle</Icon>
<Box component="span" sx={visuallyHidden}>Create a user</Box>
```

Referência

- <https://www.tpgi.com/using-aria-enhance-svg-accessibility/>