

Bibliotecas de roteamento

Por padrão, a navegação é realizada com um elemento `<a>` nativo. Você pode personalizá-lo para utilizar seu próprio roteador. Por exemplo, usando o Link do Next.js ou react-router.

Componentes de navegação

Existem dois componentes principais disponíveis para realizar navegações. It renders a native `<a>` element and applies the `href` as an attribute. The most common one is the `Link` as its name might suggest.

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

Você também pode fazer com que um botão execute ações de navegação. Se seu componente está estendendo `ButtonBase`, fornecer uma propriedade `href` habilita o modo de link. For instance, with a `Button` component:

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

Global theme Link

Em aplicações da vida real, usar um elemento `<a>` nativo é raramente o suficiente. Você pode melhorar a experiência do usuário usando sistematicamente um componente Link aprimorado. For instance, with react-router: The theme of Material-UI allows configuring this component once. For instance, with react-router:

```
const LinkBehavior = React.forwardRef<
  any,
  Omit<RouterLinkProps, 'to'> & { href: RouterLinkProps['to'] }
>((props, ref) => {
  const { href, ...other } = props;
  // Map href (Material-UI) -> to (react-router)
  return <RouterLink ref={ref} to={href} {...other} />;
});

const theme = createTheme({
  components: {
    MuiLink: {
      defaultProps: {
        component: LinkBehavior,
      },
    },
    MuiButtonBase: {
      defaultProps: {
        LinkComponent: LinkBehavior,
      },
    },
  },
});
```

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

⚠️ Esta abordagem tem limitações com TypeScript. A propriedade `href` só aceita uma string. No caso de você precisar fornecer uma estrutura mais rica, consulte a próxima seção.

Propriedade `component`

Você pode conseguir a integração com bibliotecas de roteamento de terceiros com a propriedade `component`. You can learn more about this prop in the [composition guide](#).

Link

Here are a few demos with [react-router](#). Você pode aplicar a mesma estratégia com todos os componentes: `BottomNavigation`, `Card`, etc.

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

Button

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

Nota: O componente base do botão adiciona o atributo `role="button"` quando identifica a intenção de renderizar um botão sem um `<button>` elemento nativo. Isso pode criar problemas ao renderizar um link. Se você não estiver usando um das propriedades `href`, `to`, ou `component="a"` você precisa substituir o atributo `role`. A demonstração acima consegue isso definindo `role={undefined}` **after** os "spread" "props".

```
const LinkBehavior = React.forwardRef((props, ref) => (  
  <RouterLink ref={ref} to="/" {...props} role={undefined} />  
));
```

Tabs

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

Lista

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

Mais exemplos

Next.js

O Next.js tem [um componente Link personalizado](#). The [example folder](#) provides adapters for usage with Material-UI.

- The first version of the adapter is the [NextLinkComposed](#) component. Este componente não tem estilo e é o único responsável pelo manuseio da navegação. The prop `href` was renamed `to` to avoid a naming conflict. This is similar to react-router's Link component.

```
import Button from '@material-ui/core/Button';  
import { NextLinkComposed } from '../src/Link';  
  
export default function Index() {  
  return (  
    <Button
```

```

    component={NextLinkComposed}
    to={{
      pathname: '/about',
      query: { name: 'test' },
    }}
  >
    Button link
  </Button>
);
}

```

- A segunda versão do adaptador é o componente `Link`. Este componente é estilizado. It leverages the [link component of Material-UI](#) with `NextLinkComposed`.

```

import Link from '../src/Link';

export default function Index() {
  return (
    <Link
      href={{
        pathname: '/about',
        query: { name: 'test' },
      }}
    >
      Link
    </Link>
  );
}

```

Gatsby

O componente [Link](#) do Gatsby é construído em `@reach/router`. Você pode usar a mesma documentação anterior para react-router. Ao contrário do Next.js, ele não exige que você faça uma "gambiarra".