

Tabela

Tabelas exibem conjuntos de dados. Elas podem ser totalmente customizadas.

As tabelas apresentam informação de uma forma que é fácil de digitalizar, para que os utilizadores possam procurar padrões e informações. Elas podem ser incorporadas no conteúdo principal, assim como cartões. Elas podem incluir:

- Uma visualização correspondente
- Navegação
- Ferramentas para consultar e manipular dados

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

Estrutura

Um exemplo simples sem frescuras.

Uma tabela de dados contém uma linha de cabeçalho no topo que lista os nomes das colunas, seguidas pelas linhas dos dados.

Tabela de dados

O componente `Table` tem um mapeamento próximo dos elementos nativos de `<table>`. Este requisito torna a construção de tabelas de dados ricas e desafiadora.

O [componente `DataGrid`](#) é projetado para casos de uso que se concentram em lidar com grandes quantidades de dados tabulares. Enquanto vem com uma estrutura mais rígida, em troca, você ganha recursos poderosos.

```
{{"demo": "DataTable.js", "bg": "inline"}}
```

Tabela Simples

Um exemplo simples de uma tabela densa sem muito enfeite.

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

Tabela Densa

Este exemplo demonstra o uso de linhas clicáveis com `Checkbox` para a seleção, e com um componente `Toolbar` customizado. Ele usa o componente `TableSortLabel` para ajudar no estilo dos cabeçalhos das colunas.

A tabela recebeu uma largura fixa para demonstrar a rolagem horizontal. Para impedir que os controles de paginação rolem, o componente `TablePagination` é usado fora da tabela. (O [Exemplo da 'ação de paginação customizada'](#) abaixo mostra a paginação dentro de um `TableFooter`.)

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

Tabelas Customizadas

Aqui está um exemplo de customização do componente. Você pode aprender mais sobre isso na [página de documentação de sobrescritas](#).

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

Opções de paginação customizada

É possível customizar as opções mostradas na seleção "Rows per page" usando a propriedade `rowsPerPageOptions`. Você deve fornecer um array de:

- **numbers**, cada número será usado para o rótulo e valor da opção.

```
<TablePagination rowsPerPageOptions={[10, 50]} />
```

- **objects**, as chaves `value` e `label` serão utilizadas, respectivamente para exibição do rótulo e valor da opção (útil para strings de idioma como 'Todos').

```
<TablePagination rowsPerPageOptions={[10, 50, { value: -1, label: 'All' }]} />
```

Ações de paginação customizada

A propriedade `ActionsComponent` do componente `TablePagination` permite a implementação de ações customizadas.

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

Cabeçalho fixo

Aqui está um exemplo de uma tabela com linhas roláveis e cabeçalhos de coluna fixos. Ela aproveita a propriedade `stickyHeader`. (⚠ no IE 11 support)

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

Agrupando colunas

Você pode agrupar cabeçalhos de coluna renderizando várias linhas de tabela dentro de um cabeçalho de tabela:

```
<TableHead>
  <TableRow />
  <TableRow />
</TableHead>
```

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

Tabela minimizável

Um exemplo de uma tabela com linhas expansíveis, revelando mais informações. Ela utiliza o componente [Collapse](#).

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

Abrangendo Tabela

Um exemplo simples com abrangência de linhas & colunas.

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

Tabela Virtualizada

No exemplo a seguir, nós demonstramos como usar [react-virtualized](#) com o componente `Table`. Ela renderiza 200 linhas e pode facilmente lidar com mais. A virtualização ajuda a lidar com problemas de desempenho.

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

Unstyled

If you would like to use an unstyled Table, you can use the primitive elements and enhance the table with the unstyled pagination as shown in the demo below.

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

Customized look and feel

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

Accessibility

(WAI tutorial: <https://www.w3.org/WAI/tutorials/tables/>)

Caption

A caption functions like a heading for a table. Most screen readers announce the content of captions. Captions help users to find a table and understand what it's about and decide if they want to read it.

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