


展示行列数据。

## 设计师专属

安装 [Kitchen Sketch 插件](#) , 两步就可以自动生成 Ant Design 表格组件。

## 何时使用

- 当有大量结构化的数据需要展现时；
- 当需要对数据进行排序、搜索、分页、自定义操作等复杂行为时。

## 如何使用

指定表格的数据源 `dataSource` 为一个数组。

```
const dataSource = [
  {
    key: '1',
    name: '胡彦斌',
    age: 32,
    address: '西湖区湖底公园1号',
  },
  {
    key: '2',
    name: '胡彦祖',
    age: 42,
    address: '西湖区湖底公园1号',
  },
];

const columns = [
  {
    title: '姓名',
    dataIndex: 'name',
    key: 'name',
  },
  {
    title: '年龄',
    dataIndex: 'age',
    key: 'age',
  },
  {
    title: '住址',
    dataIndex: 'address',
    key: 'address',
  },
];

<Table dataSource={dataSource} columns={columns} />;
```

## API

另外我们封装了 [ProTable](#)，在 `antd` Table 之上扩展了更多便捷易用的功能，内置搜索、筛选、刷新等常用表格行为，并为多种类型数据展示提供了内置格式化，欢迎尝试使用。

### Table

参数	说明	类型	默认值	版本
bordered	是否展示外边框和列边框	boolean	false	
columns	表格列的配置描述，具体项见下表	<a href="#">ColumnsType[]</a>	-	
components	覆盖默认的 table 元素	<a href="#">TableComponents</a>	-	
dataSource	数据数组	object[]	-	
expandable	配置展开属性	<a href="#">expandable</a>	-	
footer	表格尾部	function(currentPageData)	-	
getPopupContainer	设置表格内各类浮层的渲染节点，如筛选菜单	(triggerNode) => HTMLInputElement	() => TableHtmlElement	
loading	页面是否加载中	boolean   <a href="#">Spin Props</a>	false	
locale	默认文案设置，目前包括排序、过滤、空数据文案	object	<a href="#">默认值</a>	
pagination	分页器，参考 <a href="#">配置项</a> 或 <a href="#">pagination</a> 文档，设为 false 时不展示和进行分页	object	-	
rowClassName	表格行的类名	function(record, index): string	-	
rowKey	表格行 key 的取值，可以是字符串或一个函数	string   function(record): string	key	
rowSelection	表格行是否可选择， <a href="#">配置项</a>	object	-	

scroll	表格是否可滚动，也可以指定滚动区域的宽、高， <a href="#">配置项</a>	object	-	
showHeader	是否显示表头	boolean	true	
showSorterTooltip	表头是否显示下一次排序的 tooltip 提示。 当参数类型为对象时，将被设置为 Tooltip 的属性	boolean   <a href="#">Tooltip.props</a>	true	
size	表格大小	default   middle   small	default	
sortDirections	支持的排序方式，取值为 ascend descend	Array	[ascend, descend]	
sticky	设置粘性头部和滚动条	boolean   {offsetHeader?: number, offsetScroll?: number, getContainer?: () => HTMLElement}	-	4.6.0 (getContainer: 4.7.0)
summary	总结栏	(currentData) => ReactNode	-	
tableLayout	表格元素的 <a href="#">table-layout</a> 属性，设为 fixed 表示内容不会影响列的布局	-   auto   fixed	无 <hr/> 固定表头/列或使用了 column.ellipsis 时，默认值为 fixed	
title	表格标题	function(currentPageData)	-	
onChange	分页、排序、筛选变化时触发	function(pagination, filters, sorter, extra: { currentDataSource: [], action: paginate   sort   filter })	-	
onHeaderRow	设置头部行属性	function(columns, index)	-	
onRow	设置行属性	function(record, index)	-	

**onRow 用法**

适用于 `onRow` `onHeaderRow` `onCell` `onHeaderCell` 。

```
<Table
  onRow={record => {
    return {
      onClick: event => {}, // 点击行
      onDoubleClick: event => {},
      onContextMenu: event => {},
      onMouseEnter: event => {}, // 鼠标移入行
      onMouseLeave: event => {},
    };
  }}
  onHeaderRow={(columns, index) => {
    return {
      onClick: () => {}, // 点击表头行
    };
  }}
/>
```

Column

列描述数据对象，是 columns 中的一项，Column 使用相同的 API。

参数	说明	类型	默认值	版本
align	设置列的对齐方式	left   right   center	left	
className	列样式类名	string	-	
colSpan	表头列合并, 设置为 0 时，不渲染	number	-	
dataIndex	列数据在数据项中对应的路径，支持通过数组查询嵌套路径	string   string[]	-	
defaultFilteredValue	默认筛选值	string[]	-	
defaultSortOrder	默认排序顺序	ascend   descend	-	
editable	是否可编辑	boolean	false	
ellipsis	超过宽度将自动省略，暂不支持和排序筛选一起使用。 设置为 true 或 { showTitle?: boolean } 时，表格布局将变成 tableLayout="fixed"。	boolean   { showTitle?: boolean }	false	showTitle 4.3.0
filterDropdown	可以自定义筛选菜单，此函数只负责渲染图层，需要自行编写各种交互	ReactNode   (props: <a href="#">FilterDropdownProps</a> ) => ReactNode	-	
filterDropdownVisible	用于控制自定义筛选菜	boolean	-	

	单是否可见			
filtered	标识数据是否经过过滤，筛选图标会高亮	boolean	false	
filteredValue	筛选的受控属性，外界可用此控制列的筛选状态，值为已筛选的 value 数组	string[]	-	
filterIcon	自定义 filter 图标。	ReactNode   (filtered: boolean) => ReactNode	false	
filterMultiple	是否多选	boolean	true	
filterMode	指定筛选菜单的用户界面	'menu'   'tree'	'menu'	4.17.0
filterSearch	筛选菜单项是否可搜索	boolean   function(input, record):boolean	false	boolean:4 function:4
filters	表头的筛选菜单项	object[]	-	
fixed	(IE 下无效) 列是否固定，可选 true (等效于 left) left right	boolean   string	false	
key	React 需要的 key，如果已经设置了唯一的 dataIndex，可以忽略这个属性	string	-	
render	生成复杂数据的渲染函数，参数分别为当前行的值，当前行数据，行索引	function(text, record, index) {}	-	
responsive	响应式 breakpoint 配置列表。未设置则始终可见。	<a href="#">Breakpoint[]</a>	-	4.2.0
shouldCellUpdate	自定义单元格渲染时机	(record, prevRecord) => boolean	-	4.3.0
showSorterTooltip	表头显示下一次排序的 tooltip 提示, 覆盖 table 中 showSorterTooltip	boolean   <a href="#">Tooltip props</a>	true	
sortDirections	支持的排序方式，覆盖 Table 中 sortDirections，取值为 ascend descend	Array	[ascend, descend]	
sorter	排序函数，本地排序使	function   boolean	-	

	用一个函数(参考 <a href="#">Array.sort</a> 的 compareFunction)，需要服务端排序可设为 true			
sortOrder	排序的受控属性，外界可用此控制列的排序，可设置为 <code>ascend</code> <code>descend</code> <code>false</code>	boolean   string	-	
title	列头显示文字（函数用法 3.10.0 后支持）	ReactNode   ({ sortOrder, sortColumn, filters }) => ReactNode	-	
width	列宽度（ <a href="#">指定了也不生效?</a> ）	string   number	-	
onCell	设置单元格属性	function(record, rowIndex)	-	
onFilter	本地模式下，确定筛选的运行函数	function	-	
onFilterDropdownVisibleChange	自定义筛选菜单可见变化时调用	function(visible) {}	-	
onHeaderCell	设置头部单元格属性	function(column)	-	

ColumnGroup

参数	说明	类型	默认值
title	列头显示文字	ReactNode	-

pagination

分页的配置项。

参数	说明	类型	默认值
position	指定分页显示的位置，取值为 <code>topLeft</code>   <code>topCenter</code>   <code>topRight</code>   <code>bottomLeft</code>   <code>bottomCenter</code>   <code>bottomRight</code>	Array	[bottomRight]

更多配置项，请查看 [Pagination](#)。

expandable

展开功能的配置。

参数	说明	类型	默认值	版本
childrenColumnName	指定树形结构的列名	string	children	

columnWidth	自定义展开列宽度	string   number	-	
defaultExpandAllRows	初始时，是否展开所有行	boolean	false	
defaultExpandedRowKeys	默认展开的行	string[]	-	
expandedRowClassName	展开行的 className	function(record, index, indent): string	-	
expandedRowKeys	展开的行，控制属性	string[]	-	
expandedRowRender	额外的展开行	function(record, index, indent, expanded): ReactNode	-	
expandIcon	自定义展开图标，参考 <a href="#">示例</a>	function(props): ReactNode	-	
expandRowByClick	通过点击行来展开子行	boolean	false	
fixed	控制展开图标是否固定，可选 true left right	boolean   string	false	4.16.0
indentSize	展示树形数据时，每层缩进的宽度，以 px 为单位	number	15	
rowExpandable	设置是否允许行展开	(record) => boolean	-	
showExpandColumn	设置是否展示行展开列	boolean	true	4.18.0
onExpand	点击展开图标时触发	function(expanded, record)	-	
onExpandedRowsChange	展开的行变化时触发	function(expandedRows)	-	

### rowSelection

选择功能的配置。

参数	说明	类型	默认值	版本
checkStrictly	checkable 状态下节点选择完全受控（父子数据选中状态不再关联）	boolean	true	4.4.0
columnTitle	自定义列表选择框标题	ReactNode	-	
columnWidth	自定义列表选择框宽度	string   number	32px	
fixed	把选择框列固定在左边	boolean	-	
getCheckboxProps	选择框的默认属性配置	function(record)	-	
hideSelectAll	隐藏全选勾选框与自定义选择项	boolean	false	4.3.0
preserveSelectedRowKeys	当数据被删除时仍然保留	boolean	-	4.4.0

	选项的 <code>key</code>			
<code>renderCell</code>	渲染勾选框，用法与 <code>Column</code> 的 <code>render</code> 相同	<code>function(checked, record, index, originNode) {}</code>	-	4.1.0
<code>selectedRowKeys</code>	指定选中项的 <code>key</code> 数组，需要和 <code>onChange</code> 进行配合	<code>string[]   number[]</code>	<code>[]</code>	
<code>defaultSelectedRowKeys</code>	默认选中项的 <code>key</code> 数组	<code>string[]   number[]</code>	<code>[]</code>	
<code>selections</code>	自定义选择项 <a href="#">配置项</a> ，设为 <code>true</code> 时使用默认选择项	<code>object[]   boolean</code>	<code>true</code>	
<code>type</code>	多选/单选	<code>checkbox</code>   <code>radio</code>	<code>checkbox</code>	
<code>onChange</code>	选中项发生变化时的回调	<code>function(selectedRowKeys, selectedRows)</code>	-	
<code>onSelect</code>	用户手动选择/取消选择某行的回调	<code>function(record, selected, selectedRows, nativeEvent)</code>	-	
<code>onSelectAll</code>	用户手动选择/取消选择所有行的回调	<code>function(selected, selectedRows, changeRows)</code>	-	
<code>onSelectInvert</code>	用户手动选择反选的回调	<code>function(selectedRowKeys)</code>	-	
<code>onSelectNone</code>	用户清空选择的回调	<code>function()</code>	-	

scroll

参数	说明	类型	默认值
<code>scrollToFirstRowOnChange</code>	当分页、排序、筛选变化后是否滚动到表格顶部	<code>boolean</code>	-
<code>x</code>	设置横向滚动，也可用于指定滚动区域的宽，可以设置为像素值，百分比， <code>true</code> 和 <a href="#">'max-content'</a>	<code>string</code>   <code>number</code>   <code>true</code>	-
<code>y</code>	设置纵向滚动，也可用于指定滚动区域的高，可以设置为像素值	<code>string</code>   <code>number</code>	-

selection

参数	说明	类型	默认值
<code>key</code>	React 需要的 <code>key</code> ，建议设置	<code>string</code>	-
<code>text</code>	选择项显示的文字	<code>ReactNode</code>	-
<code>onSelect</code>	选择项点击回调	<code>function(changeableRowKeys)</code>	-



## 在 TypeScript 中使用

```
import { Table } from 'antd';
import { ColumnsType } from 'antd/es/table';

interface User {
  key: number;
  name: string;
}

const columns: ColumnsType<User> = [
  {
    key: 'name',
    title: 'Name',
    dataIndex: 'name',
  },
];

const data: User[] = [
  {
    key: 0,
    name: 'Jack',
  },
];

export default () => (
  <>
    <Table<User> columns={columns} dataSource={data} />
    /* 使用 JSX 风格的 API */
    <Table<User> dataSource={data}>
      <Table.Column<User> key="name" title="Name" dataIndex="name" />
    </Table>
  </>
);
```

TypeScript 里使用 Table 的 [CodeSandbox 实例](#)。

## 注意

按照 [React 的规范](#)，所有的数组组件必须绑定 `key`。在 Table 中，`dataSource` 和 `columns` 里的数据值都需要指定 `key` 值。对于 `dataSource` 默认将每列数据的 `key` 属性作为唯一的标识。

Warning: Each child in an array or iterator should have a unique "key" prop. Check the render method of `Table`. See <https://fb.me/react-warning-keys> for more information.

如果 `dataSource[i].key` 没有提供，你应该使用 `rowKey` 来指定 `dataSource` 的主键，如下所示。若没有指定，控制台会出现以上的提示，表格组件也会出现各类奇怪的错误。

```
// 比如你的数据主键是 uid
return <Table rowKey="uid" />;
// 或
return <Table rowKey={record => record.uid} />;
```

## 从 v3 升级到 v4

Table 移除了在 v3 中废弃的 `onRowClick`、`onRowDoubleClick`、`onRowMouseEnter`、`onRowMouseLeave` 等方法。如果你使用的 api 为文档中列举的 api，那你不用担心会丢失功能。

此外，比较重大的改动为 `dataIndex` 从支持路径嵌套如 `user.age` 改成了数组路径如 `['user', 'age']`。以解决过去属性名带 `.` 需要额外的数据转化问题。

## FAQ

### 如何在没有数据或只有一页数据时隐藏分页栏

你可以设置 `pagination` 的 `hideOnSinglePage` 属性为 `true`。

### 表格过滤时会回到第一页？

前端过滤时通常条目总数会减少，从而导致总页数小于筛选前的当前页数，为了防止当前页面没有数据，我们默认会返回第一页。

如果你在使用远程分页，很可能需要保持当前页面，你可以参照这个 [受控例子](#) 控制当前页面不变。

### 表格分页为何会出现 size 切换器？

自 4.1.0 起，`Pagination` 在 `total` 大于 50 条时会默认显示 size 切换器以提升用户交互体验。如果你不需要该功能，可以通过设置 `showSizeChanger` 为 `false` 来关闭。

### 为什么 更新 state 会导致全表渲染？

由于 `columns` 支持 `render` 方法，因而 Table 无法知道哪些单元会受到影响。你可以通过 `column.shouldCellUpdate` 来控制单元格的渲染。

### 固定列穿透到最上层该怎么办？

固定列通过 `z-index` 属性将其悬浮于非固定列之上，这使得有时候你会发现在 Table 上放置遮罩层时固定列会被透过的情况。为遮罩层设置更高的 `z-index` 覆盖住固定列即可。