

zh-CN

集成 [react-resizable](#) 来实现可伸缩列。如果有排序需要，可以通过[额外标记](#)阻止触发排序。

en-US

Implement resizable column by integrate with [react-resizable](#). When sort needed, you can use [additional mark](#) to prevent resize trigger sort.

```
import { Table } from 'antd';
import { Resizable } from 'react-resizable';

const ResizableTitle = props => {
  const { onResize, width, ...restProps } = props;

  if (!width) {
    return <th {...restProps} />;
  }

  return (
    <Resizable
      width={width}
      height={0}
      handle={
        <span
          className="react-resizable-handle"
          onClick={e => {
            e.stopPropagation();
          }}
        />
      >
      <th {...restProps} />
    </Resizable>
  );
};

class Demo extends React.Component {
  state = {
    columns: [
      {
        title: 'Date',
        dataIndex: 'date',
        width: 200,
      },
      {
        title: 'Amount',
        dataIndex: 'amount',
        width: 100,
      },
    ],
  };
}
```

```

        sorter: (a, b) => a.amount - b.amount,
      },
      {
        title: 'Type',
        dataIndex: 'type',
        width: 100,
      },
      {
        title: 'Note',
        dataIndex: 'note',
        width: 100,
      },
      {
        title: 'Action',
        key: 'action',
        render: () => <a>Delete</a>,
      },
    ],
  };

  components = {
    header: {
      cell: ResizableTitle,
    },
  };

  data = [
    {
      key: 0,
      date: '2018-02-11',
      amount: 120,
      type: 'income',
      note: 'transfer',
    },
    {
      key: 1,
      date: '2018-03-11',
      amount: 243,
      type: 'income',
      note: 'transfer',
    },
    {
      key: 2,
      date: '2018-04-11',
      amount: 98,
      type: 'income',
      note: 'transfer',
    },
  ];

  handleResize =
    index =>

```

```

(e, { size }) => {
  this.setState(({ columns }) => {
    const nextColumns = [...columns];
    nextColumns[index] = {
      ...nextColumns[index],
      width: size.width,
    };
    return { columns: nextColumns };
  });
};

render() {
  const columns = this.state.columns.map((col, index) => ({
    ...col,
    onHeaderCell: column => ({
      width: column.width,
      onResize: this.handleResize(index),
    }),
  }));

  return <Table bordered components={this.components} columns={columns}
dataSource={this.data} />;
}
}

export default () => <Demo />;

```

```

#components-table-demo-resizable-column .react-resizable {
  position: relative;
  background-clip: padding-box;
}

#components-table-demo-resizable-column .react-resizable-handle {
  position: absolute;
  right: -5px;
  bottom: 0;
  z-index: 1;
  width: 10px;
  height: 100%;
  cursor: col-resize;
}

```