

## zh-CN

最基本的用法，展示了 `dataSource`、`targetKeys`、每行的渲染函数 `render` 以及回调函数 `onChange` `onSelectChange` `onScroll` 的用法。

## en-US

The most basic usage of `Transfer` involves providing the source data and target keys arrays, plus the rendering and some callback functions.

```
import React, { useState } from 'react';
import { Transfer } from 'antd';

const mockData = [];
for (let i = 0; i < 20; i++) {
  mockData.push({
    key: i.toString(),
    title: `content${i + 1}`,
    description: `description of content${i + 1}`,
  });
}

const initialTargetKeys = mockData.filter(item => +item.key > 10).map(item =>
item.key);

const App = () => {
  const [targetKeys, setTargetKeys] = useState(initialTargetKeys);
  const [selectedKeys, setSelectedKeys] = useState([]);
  const onChange = (nextTargetKeys, direction, moveKeys) => {
    console.log('targetKeys:', nextTargetKeys);
    console.log('direction:', direction);
    console.log('moveKeys:', moveKeys);
    setTargetKeys(nextTargetKeys);
  };

  const onSelectChange = (sourceSelectedKeys, targetSelectedKeys) => {
    console.log('sourceSelectedKeys:', sourceSelectedKeys);
    console.log('targetSelectedKeys:', targetSelectedKeys);
    setSelectedKeys([...sourceSelectedKeys, ...targetSelectedKeys]);
  };

  const onScroll = (direction, e) => {
    console.log('direction:', direction);
    console.log('target:', e.target);
  };

  return (
    <Transfer
      dataSource={mockData}
      titles={['Source', 'Target']}
      targetKeys={targetKeys}
    />
  );
};
```

```
      selectedKeys={selectedKeys}
      onChange={onChange}
      onSelectChange={onSelectChange}
      onScroll={onScroll}
      render={item => item.title}
    />
  );
};

export default () => <App />;
```