

zh-CN

使用 Tree 组件作为自定义渲染列表。

en-US

Customize render list with Tree component.

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

// Customize Table Transfer
const isChecked = (selectedKeys, eventKey) => selectedKeys.indexOf(eventKey) !== -1;

const generateTree = (treeNodes = [], checkedKeys = []) => {
  treeNodes.map(({ children, ...props }) => ({
    ...props,
    disabled: checkedKeys.includes(props.key),
    children: generateTree(children, checkedKeys),
  }));
};

const TreeTransfer = ({ dataSource, targetKeys, ...restProps }) => {
  const transferDataSource = [];
  function flatten(list = []) {
    list.forEach(item => {
      transferDataSource.push(item);
      flatten(item.children);
    });
  }
  flatten(dataSource);

  return (
    <Transfer
      {...restProps}
      targetKeys={targetKeys}
      dataSource={transferDataSource}
      className="tree-transfer"
      render={item => item.title}
      showSelectAll={false}
    >
    {({ direction, onItemSelect, selectedKeys }) => {
      if (direction === 'left') {
        const checkedKeys = [...selectedKeys, ...targetKeys];
        return (
          <Tree
            blockNode
            checkable
            checkStrictly
            defaultExpandAll
            checkedKeys={checkedKeys}
            treeData={generateTree(dataSource, targetKeys)}
          />
        );
      }
    }}
  );
};
```

```

        onCheck={(_ , { node: { key } }) => {
            onItemSelect(key, !isChecked(checkedExceptions, key));
        }}
        onSelect={(_ , { node: { key } }) => {
            onItemSelect(key, !isChecked(checkedExceptions, key));
        }}
    />
    );
}
}}
</Transfer>
);
};

const treeData = [
  { key: '0-0', title: '0-0' },
  {
    key: '0-1',
    title: '0-1',
    children: [
      { key: '0-1-0', title: '0-1-0' },
      { key: '0-1-1', title: '0-1-1' },
    ],
  },
  { key: '0-2', title: '0-3' },
];

const App = () => {
  const [targetKeys, setTargetKeys] = useState([]);
  const onChange = keys => {
    setTargetKeys(keys);
  };
  return <TreeTransfer dataSource={treeData} targetKeys={targetKeys} onChange={
onChange} />;
};

export default () => <App />;

```