zh-CN

使用 Table 组件作为自定义渲染列表。

en-US

Customize render list with Table component.

```
import { Transfer, Switch, Table, Tag } from 'antd';
import difference from 'lodash/difference';

// Customize Table Transfer
const TableTransfer = ({ leftColumns, rightColumns, ...restProps }) => (
  <Transfer {...restProps}>
    {({
      direction,
      filteredItems,
      onItemSelectAll,
      onItemSelect,
      selectedKeys: listSelectedKeys,
      disabled: listDisabled,
    }) => {
      const columns = direction === 'left' ? leftColumns : rightColumns;

      const rowSelection = {
        getCheckboxProps: item => ({ disabled: listDisabled || item.disabled }),
        onSelectAll(selected, selectedRows) {
          const treeSelectedKeys = selectedRows
            .filter(item => !item.disabled)
            .map(({ key }) => key);
          const diffKeys = selected
            ? difference(treeSelectedKeys, listSelectedKeys)
            : difference(listSelectedKeys, treeSelectedKeys);
          onItemSelectAll(diffKeys, selected);
        },
        onSelect({ key }, selected) {
          onItemSelect(key, selected);
        },
        selectedRowKeys: listSelectedKeys,
      };

      return (
        <Table
          rowSelection={rowSelection}
          columns={columns}
          dataSource={filteredItems}
          size="small"
          style={{ pointerEvents: listDisabled ? 'none' : null }}
          onRow={({ key, disabled: itemDisabled }) => ({
            onClick: () => {
              if (itemDisabled || listDisabled) return;
              onItemSelect(key, !listSelectedKeys.includes(key));
            },
          })}
        />
      );
    }}
  </Transfer>
);

const mockTags = ['cat', 'dog', 'bird'];

const mockData = [];
for (let i = 0; i < 20; i++) {
  mockData.push({
    key: i.toString(),
    title: `content${i + 1}`,
    description: `description of content${i + 1}`,
    disabled: i % 4 === 0,
    tag: mockTags[i % 3],
  });
}

const originTargetKeys = mockData.filter(item => +item.key % 3 > 1).map(item =>
item.key);

const leftTableColumns = [
  {
    dataIndex: 'title',
    title: 'Name',
  },
  {
    dataIndex: 'tag',
    title: 'Tag',
    render: tag => <Tag>{tag}</Tag>,
  },
  {
    dataIndex: 'description',
    title: 'Description',
  },
];
const rightTableColumns = [
  {
    dataIndex: 'title',
    title: 'Name',
  },
];

class App extends React.Component {
  state = {
    targetKeys: originTargetKeys,
    disabled: false,
    showSearch: false,
  };

  onChange = nextTargetKeys => {
    this.setState({ targetKeys: nextTargetKeys });
  };

  triggerDisable = disabled => {
    this.setState({ disabled });
  };

  triggerShowSearch = showSearch => {
    this.setState({ showSearch });
  };

  render() {
    const { targetKeys, disabled, showSearch } = this.state;
    return (
      <>
        <TableTransfer
          dataSource={mockData}
          targetKeys={targetKeys}
          disabled={disabled}
          showSearch={showSearch}
          onChange={this.onChange}
          filterOption={(inputValue, item) =>
            item.title.indexOf(inputValue) !== -1 || item.tag.indexOf(inputValue) !== -1
          }
          leftColumns={leftTableColumns}
          rightColumns={rightTableColumns}
        />
        <Switch
          unCheckedChildren="disabled"
          checkedChildren="disabled"
          checked={disabled}
          onChange={this.triggerDisable}
          style={{ marginTop: 16 }}
        />
        <Switch
          unCheckedChildren="showSearch"
          checkedChildren="showSearch"
          checked={showSearch}
          onChange={this.triggerShowSearch}
          style={{ marginTop: 16 }}
        />
      </>
    );
  }
}

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

```
#components-transfer-demo-table-transfer .ant-table td {
  background: transparent;
}
```