zh-CN

也可以使用 react-sortable-hoc 来实现一个拖拽操作列。

en-US

Alternatively you can implement drag sorting with handler using react-sortable-hoc.

```
import { Table } from 'antd';
import { SortableContainer, SortableElement, SortableHandle } from 'react-sortable-
hoc';
import { MenuOutlined } from '@ant-design/icons';
import { arrayMoveImmutable } from 'array-move';

const DragHandle = SortableHandle(() => <MenuOutlined style={{ cursor: 'grab',
color: '#999' }} />);

const columns = [
  {
    title: 'Sort',
    dataIndex: 'sort',
    width: 30,
    className: 'drag-visible',
    render: () => <DragHandle />,
  },
  {
    title: 'Name',
    dataIndex: 'name',
    className: 'drag-visible',
  },
  {
    title: 'Age',
    dataIndex: 'age',
  },
  {
    title: 'Address',
    dataIndex: 'address',
  },
];

const data = [
  {
    key: '1',
    name: 'John Brown',
    age: 32,
    address: 'New York No. 1 Lake Park',
    index: 0,
  },
  {
    key: '2',
    name: 'Jim Green',
```

```
    age: 42,
    address: 'London No. 1 Lake Park',
    index: 1,
  },
  {
    key: '3',
    name: 'Joe Black',
    age: 32,
    address: 'Sidney No. 1 Lake Park',
    index: 2,
  },
];

const SortableItem = SortableElement(props => <tr {...props} />);
const SortableBody = SortableContainer(props => <tbody {...props} />);

class SortableTable extends React.Component {
  state = {
    dataSource: data,
  };

  onSortEnd = ({ oldIndex, newIndex }) => {
    const { dataSource } = this.state;
    if (oldIndex !== newIndex) {
      const newData = arrayMoveImmutable([].concat(dataSource), oldIndex,
newIndex).filter(
        el => !!el,
      );
      console.log('Sorted items: ', newData);
      this.setState({ dataSource: newData });
    }
  };

  DraggableContainer = props => (
    <SortableBody
      useDragHandle
      disableAutoscroll
      helperClass="row-dragging"
      onSortEnd={this.onSortEnd}
      {...props}
    />
  );

  DraggableBodyRow = ({ className, style, ...restProps }) => {
    const { dataSource } = this.state;
    // function findIndex base on Table rowKey props and should always be a right
array index
    const index = dataSource.findIndex(x => x.index === restProps['data-row-key']);
    return <SortableItem index={index} {...restProps} />;
  };

  render() {
```

```
    const { dataSource } = this.state;

    return (
      <Table
        pagination={false}
        dataSource={dataSource}
        columns={columns}
        rowKey="index"
        components={{
          body: {
            wrapper: this.DraggableContainer,
            row: this.DraggableBodyRow,
          },
        }}
      />
    );
  }
}

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

```
.row-dragging {
  background: #fafafa;
  border: 1px solid #ccc;
}

.row-dragging td {
  padding: 16px;
}

.row-dragging .drag-visible {
  visibility: visible;
}
```