zh-CN

使用 `react-dnd@15+` 实现标签可拖拽。

en-US

Use `react-dnd@15+` to make tabs draggable.

```
import React, { useRef, cloneElement } from 'react';
import { Tabs } from 'antd';
import { DndProvider, useDrag, useDrop } from 'react-dnd';
import { HTML5Backend } from 'react-dnd-html5-backend';

const { TabPane } = Tabs;

const type = 'DraggableTabNode';

const DraggableTabNode = ({ index, children, moveNode }) => {
  const ref = useRef();
  const [{ isOver, dropClassName }, drop] = useDrop({
    accept: type,
    collect: monitor => {
      const { index: dragIndex } = monitor.getItem() || {};
      if (dragIndex === index) {
        return {};
      }
      return {
        isOver: monitor.isOver(),
        dropClassName: 'dropping',
      };
    },
    drop: item => {
      moveNode(item.index, index);
    },
  });
  const [, drag] = useDrag({
    type,
    item: { index },
    collect: monitor => ({
      isDragging: monitor.isDragging(),
    }),
  });
  drop(drag(ref));
  return (
    <div ref={ref} className={`dragnode ${isOver ? dropClassName : ''}`}>
      {children}
    </div>
  );
};

class DraggableTabs extends React.Component {
```

```
  state = {
    order: [],
  };

  moveTabNode = (dragKey, hoverKey) => {
    const newOrder = this.state.order.slice();
    const { children } = this.props;

    React.Children.forEach(children, c => {
      if (newOrder.indexOf(c.key) === -1) {
        newOrder.push(c.key);
      }
    });

    const dragIndex = newOrder.indexOf(dragKey);
    const hoverIndex = newOrder.indexOf(hoverKey);

    newOrder.splice(dragIndex, 1);
    newOrder.splice(hoverIndex, 0, dragKey);

    this.setState({
      order: newOrder,
    });
  };

  renderTabBar = (props, DefaultTabBar) => (
    <DefaultTabBar {...props}>
      {node => (
        <DraggableTabNode key={node.key} index={node.key} moveNode=
{this.moveTabNode}>
          {node}
        </DraggableTabNode>
      )}
    </DefaultTabBar>
  );

  render() {
    const { order } = this.state;
    const { children } = this.props;

    const tabs = [];
    React.Children.forEach(children, c => {
      tabs.push(c);
    });

    const orderTabs = tabs.slice().sort((a, b) => {
      const orderA = order.indexOf(a.key);
      const orderB = order.indexOf(b.key);

      if (orderA !== -1 && orderB !== -1) {
        return orderA - orderB;
      }
```

```
      if (orderA !== -1) {
        return -1;
      }
      if (orderB !== -1) {
        return 1;
      }

      const ia = tabs.indexOf(a);
      const ib = tabs.indexOf(b);

      return ia - ib;
    });

    return (
      <DndProvider backend={HTML5Backend}>
        <Tabs renderTabBar={this.renderTabBar} {...this.props}>
          {orderTabs}
        </Tabs>
      </DndProvider>
    );
  }
}

export default () => (
  <DraggableTabs>
    <TabPane tab="tab 1" key="1">
      Content of Tab Pane 1
    </TabPane>
    <TabPane tab="tab 2" key="2">
      Content of Tab Pane 2
    </TabPane>
    <TabPane tab="tab 3" key="3">
      Content of Tab Pane 3
    </TabPane>
  </DraggableTabs>
);
```

```
.dropping {
  background: #fefefe;
  transition: all 0.3s;
}

.dragnode {
  margin-inline-end: 24px;
}
```