zh-CN

使用 `itemRender` ，我们可以集成 react-dnd 来实现对上传列表拖拽排序。

en-US

By using `itemRender` , we can integrate upload with react-dnd to implement drag sorting of uploadList.

```
import React, { useState, useCallback, useRef } from 'react';
import { Upload, Button, Tooltip } from 'antd';
import { DndProvider, useDrag, useDrop } from 'react-dnd';
import { HTML5Backend } from 'react-dnd-html5-backend';
import update from 'immutability-helper';
import { UploadOutlined } from '@ant-design/icons';

const type = 'DragableUploadList';

const DragableUploadListItem = ({ originNode, moveRow, file, fileList }) => {
  const ref = React.useRef();
  const index = fileList.indexOf(file);
  const [{ isOver, dropClassName }, drop] = useDrop({
    accept: type,
    collect: monitor => {
      const { index: dragIndex } = monitor.getItem() || {};
      if (dragIndex === index) {
        return {};
      }
      return {
        isOver: monitor.isOver(),
        dropClassName: dragIndex < index ? ' drop-over-downward' : ' drop-over-upward',
      };
    },
    drop: item => {
      moveRow(item.index, index);
    },
  });
  const [, drag] = useDrag({
    type,
    item: { index },
    collect: monitor => ({
      isDragging: monitor.isDragging(),
    }),
  });
  drop(drag(ref));
  const errorNode = <Tooltip title="Upload Error">{originNode.props.children}</Tooltip>;
  return (
    <div
      ref={ref}
      className={`ant-upload-draggable-list-item ${isOver ? dropClassName : ''}`}
      style={{ cursor: 'move' }}
    >
      {file.status === 'error' ? errorNode : originNode}
    </div>
  );
};

const DragSortingUpload = () => {
  const [fileList, setFileList] = useState([
    {
      uid: '-1',
      name: 'image1.png',
      status: 'done',
      url:
'https://zos.alipayobjects.com/rmsportal/jkjgkEfvpUPVyRjUImniVslZfWPnJuuZ.png',
    },
    {
      uid: '-2',
      name: 'image2.png',
      status: 'done',
      url:
'https://zos.alipayobjects.com/rmsportal/jkjgkEfvpUPVyRjUImniVslZfWPnJuuZ.png',
    },
    {
      uid: '-3',
      name: 'image3.png',
      status: 'done',
      url:
'https://zos.alipayobjects.com/rmsportal/jkjgkEfvpUPVyRjUImniVslZfWPnJuuZ.png',
    },
    {
      uid: '-4',
      name: 'image4.png',
      status: 'done',
      url:
'https://zos.alipayobjects.com/rmsportal/jkjgkEfvpUPVyRjUImniVslZfWPnJuuZ.png',
    },
    {
      uid: '-5',
      name: 'image.png',
      status: 'error',
    },
  ]);

  const moveRow = useCallback(
    (dragIndex, hoverIndex) => {
      const dragRow = fileList[dragIndex];
      setFileList(
        update(fileList, {
          $splice: [
            [dragIndex, 1],
            [hoverIndex, 0, dragRow],
          ],
        }),
      );
    },
    [fileList],
  );

  const onChange = ({ fileList: newFileList }) => {
    setFileList(newFileList);
  };

  return (
    <DndProvider backend={HTML5Backend}>
      <Upload
        action="https://www.mocky.io/v2/5cc8019d300000980a055e76"
        fileList={fileList}
        onChange={onChange}
        itemRender={(originNode, file, currFileList) => (
          <DragableUploadListItem
            originNode={originNode}
            file={file}
            fileList={currFileList}
            moveRow={moveRow}
          />
        )}
      >
        <Button icon={<UploadOutlined />}>Click to Upload</Button>
      </Upload>
    </DndProvider>
  );
};

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

```
#components-upload-demo-drag-sorting .ant-upload-draggable-list-item {
  border-top: 2px dashed rgba(0, 0, 0, 0);
  border-bottom: 2px dashed rgba(0, 0, 0, 0);
}
#components-upload-demo-drag-sorting .ant-upload-draggable-list-item.drop-over-downward {
  border-bottom-color: #1890ff;
}
#components-upload-demo-drag-sorting .ant-upload-draggable-list-item.drop-over-upward {
  border-top-color: #1890ff;
}
```