zh-CN

根据类型默认显示对应 icon

en-US

Displays the corresponding by default by type icon

```
import { Upload, Modal } from 'antd';
import {
  LoadingOutlined,
  PaperClipOutlined,
  PictureTwoTone,
  FilePdfTwoTone,
  FileWordTwoTone,
  FileExcelTwoTone,
  PlusOutlined,
} from '@ant-design/icons';

function getBase64(file) {
  return new Promise((resolve, reject) => {
    const reader = new FileReader();
    reader.readAsDataURL(file);
    reader.onload = () => resolve(reader.result);
    reader.onerror = error => reject(error);
  });
}

class PicturesWall extends React.Component {
  state = {
    previewVisible: false,
    previewImage: '',
    fileList: [
      {
        uid: '-2',
        name: 'pdf.pdf',
        status: 'done',
        url: 'http://cdn07.foxitsoftware.cn/pub/foxit/cpdf/FoxitCompanyProfile.pdf',
      },
      {
        uid: '-3',
        name: 'doc.doc',
        status: 'done',
        url:
'https://zos.alipayobjects.com/rmsportal/jkjgkEfvpUPVyRjUImniVslZfWPnJuuZ.doc',
      },
      {
        uid: '-4',
        name: 'image.png',
        status: 'error',
      },
        {
          uid: '-5',
          name: 'pdf.pdf',
          status: 'error',
        },
        {
          uid: '-6',
          name: 'doc.doc',
          status: 'error',
        },
      ],
    };

    handleCancel = () => this.setState({ previewVisible: false });

    handlePreview = async file => {
      if (!file.url && !file.preview) {
        file.preview = await getBase64(file.originFileObj);
      }

      this.setState({
        previewImage: file.url || file.preview,
        previewVisible: true,
      });
    };

    handleChange = ({ fileList }) => this.setState({ fileList });

    handleIconRender = (file, listType) => {
      const fileSufIconList = [
        { type: <FilePdfTwoTone />, suf: ['.pdf'] },
        { type: <FileExcelTwoTone />, suf: ['.xlsx', '.xls', '.csv'] },
        { type: <FileWordTwoTone />, suf: ['.doc', '.docx'] },
        {
          type: <PictureTwoTone />,
          suf: ['.webp', '.svg', '.png', '.gif', '.jpg', '.jpeg', '.jfif', '.bmp',
'.dpg'],
        },
      ];
      // console.log(1, file, listType);
      let icon = file.status === 'uploading' ? <LoadingOutlined /> :
<PaperClipOutlined />;
      if (listType === 'picture' || listType === 'picture-card') {
        if (listType === 'picture-card' && file.status === 'uploading') {
          icon = <LoadingOutlined />; // or icon = 'uploading...';
        } else {
          fileSufIconList.forEach(item => {
            if (item.suf.includes(file.name.slice(file.name.lastIndexOf('.')))) {
              icon = item.type;
            }
          });
        }
    }
    return icon;
  };

  render() {
    const { previewVisible, previewImage, fileList } = this.state;
    const uploadButton = (
      <div>
        <PlusOutlined />
        <div style={{ marginTop: 8 }}>Upload</div>
      </div>
    );
    return (
      <>
        <Upload
          action="https://www.mocky.io/v2/5cc8019d300000980a055e76"
          listType="picture-card"
          fileList={fileList}
          onPreview={this.handlePreview}
          onChange={this.handleChange}
          iconRender={this.handleIconRender}
        >
          {fileList.length >= 8 ? null : uploadButton}
        </Upload>
        <Modal visible={previewVisible} footer={null} onCancel={this.handleCancel}>
          <img alt="example" style={{ width: '100%' }} src={previewImage} />
        </Modal>
      </>
    );
  }
}

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