```
import React, { useEffect, useRef, useState } from "react";
import { Modal, Input, message, Upload } from "antd";
import { useDispatch, useSelector } from "react-redux";
import { InboxOutlined } from "@ant-design/icons";
import { upload, createAttachment } from "store/slices/common/attachmentSlide";
const { Dragger } = Upload;

const ModalUpload = ({
  objectId,
  objectType,
  isVisibleModal,
  onCancelModal,
  onSubmitSuccess,
  title = "Thêm file đính kèm",
}) => {
  const dispatch = useDispatch();
  const { loading } = useSelector((state) => state.attachment);
  const [contentStatus, setContentStatus] = useState("");
  const [content, setContent] = useState("");
  const [fileList, setFileList] = useState([]);
  const fileListRef = useRef([]);
  useEffect(() => {
    setContent("");
    setFileList([]);
    fileListRef.current = [];
  }, [isVisibleModal]);
  const dummyRequest = ({ file, onSuccess, onError }) => {
    const formData = new FormData();
    formData.append("fileUpload", file);
    const subDir = `${objectType}_${objectId}`;
    if (subDir) formData.append("nameFolder", subDir); // sub dir
    const payload = {
      formData,
      onUploadSuccess: (result) => {
        if (onSuccess) onSuccess(result);
      },
      onUploadFailed: () => {
        if (onError) onError();
      },
    };
    dispatch(upload(payload));
  };
  const handleChange = (info) => {
    // console.log('handleChange info',info );
    const { status } = info.file;
    if (status !== "uploading") {
      // console.log(info.file, info.fileList);
    }
    if (status === "done") {
      message.success(`${info.file.name} file uploaded successfully.`);
    } else if (status === "error") {
      message.error(`${info.file.name} file upload failed.`);
    }
    var existed = fileListRef.current.find(
      (item) => item.uid === info.file.uid
    );
    if (existed) {
      const newFileList = fileListRef.current.map((item) => {
        if (item.uid === info.file.uid) return info.file;
        else return item;
      });
      fileListRef.current = newFileList;
      // setFileList([...newFileList]);
    } else {
      // setFileList(prev => ([...prev, info.file]))
      fileListRef.current.push(info.file);
    }
    //  clear removed item
    const dt = fileListRef.current.filter((item) => item.status !== "removed");
    fileListRef.current = dt;
    setFileList([...fileListRef.current]);
```

```
};
const props = {
  name: "file",
  multiple: true,
  maxCount: 5,
  customRequest: dummyRequest,
  // onRemove: onRemoveFile,
  fileList: fileList,
  // action: 'https://660d2bd96ddfa2943b33731c.mockapi.io/api/upload',
  onChange: handleChange,
  onDrop(e) {
    // console.log('Dropped files', e.dataTransfer.files);
  },
};
const onUploadSuccess = () => {
  if (!content) {
    setContentStatus("error");
    message.error(`Xin vui lòng nhập nội dung đính kèm`);
    return;
  }
  const doneFileList = fileList.filter((item) => item.status === "done");
  if (doneFileList.length < 1) {
    message.error(`Chưa có file đính kèm nào được tải lên`);
    return;
  }
  const payload = doneFileList.map((item, index) => ({
    ...item.response,
    description: `${content} ${index === 0 ? "" : ` (${index})`}`,
    objectId: objectId,
    objectType: objectType,
  }));
  const data = {
    payload: payload,
    onCreateSuccess: (result) => {
      if (onSubmitSuccess) onSubmitSuccess(result);
    },
  };
  dispatch(createAttachment(data));
};

const onContentChange = (e) => {
  const status = e?.target?.value ? "" : "error";
  setContentStatus(status);
  setContent(e?.target?.value);
};
return (
  <Modal
    title={title}
    open={isVisibleModal}
    onCancel={onCancelModal}
    width={"600px"}
    // height={"80vh"}
    // className="sign-modal"
    // footer={onRenderFooter()}
    onOk={onUploadSuccess}
    loading={loading}
    okText="Lưu"
    cancelText="Hủy"
    maskClosable={false}
  >
    <Input
      value={content}
      status={contentStatus}
      allowClear={true}
      placeholder="Nhập nội dung đính kèm"
      style={{ marginBottom: "24px" }}
      onChange={onContentChange}
    />
    <Dragger {...props}>
      <p className="ant-upload-drag-icon">
        <InboxOutlined />
      </p>
      <p className="ant-upload-text">
        Bấm hoặc kéo danh sách file vào đây để tải lên (tối đa 5 file)
        </p>
        <p className="ant-upload-hint">
          Hỗ trợ tải lên một hoặc nhiều file dữ liệu. Xin vui lòng tuân thủ quy
          định về dữ liệu của công ty trước khi tải lên.
        </p>
      </Dragger>
    </Modal>
  );
};
export default ModalUpload;
```