

```

import React, { memo, useState, useEffect } from "react";
import { Upload, message, Image } from "antd";
import styled from "styled-components";
import { isEmpty } from "lodash";
import { LoadingOutlined, PlusOutlined } from "@ant-design/icons";
import { withFormContext } from "components/DataEntry/Form/withFormContext";
import { useDispatch } from "react-redux";
import { uploadFileURL } from "store/slices/uploadFileSlice";

```

```

export default withFormContext(
  memo(function Avatar({
    data,
    onChange,
    disabled,
    name = "avatarUrl",
    folder = null,
  }) {
    const dispatch = useDispatch();
    const [state, setState] = useState({
      loading: false,
      imageUrl: data?.location?.replace("http:", "https:") || "",
    });

    useEffect(() => {
      if (!isEmpty(data))
        setState({ imageUrl: data?.location?.replace("http:", "https:") });
    }, [data]);

    function getBase64(img, callback) {
      const reader = new FileReader();
      reader.addEventListener("load", () => callback(reader.result));
      reader.readAsDataURL(img);
    }

    const handleChange = (info) => {
      // let fileList = [...info.fileList];
      if (info.file.status === "uploading") {
        setState({ loading: true });
        return;
      }
      if (info.file.status === "error") {
        // Get this url from response in real world.
        getBase64(info.file.originFileObj, (imageUrl) =>
          setState({
            imageUrl,
            loading: false,
          })
        );
      }
      if (info.file.status === "done") {
        // Get this url from response in real world.
        setState({
          imageUrl: info.file.response,
          loading: false,
        });
        if (onChange) onChange(info.file.response);
      }
    };

    function beforeUpload(file) {
      const isJpgOrPng =
        (file.type === "image/jpeg" || file.type === "image/png") &&
        /\.png|.jpeg|.jpg/.test(file.name);
      if (!isJpgOrPng) {
        message.error("You can only upload JPG/PNG file!");
      }
      const isLimit = file.size / 1024 / 1024 < 4;
      if (!isLimit) {
        message.error(
          "File tải lên vượt quá dung lượng 2MB, vui lòng kiểm tra lại!"
        );
      }
    }
  })
);

```


