import React, { Component } from "react";

import Modal from "../../components/UI/Modal/Modal";

import Aux from "../../hoc/Auxiliary/Auxiliary";

const withErrorHandler = (WrappedComponent, axios) => {

  return class extends Component {

    state = {

      error: null

    };

    componentDidMount() {

      axios.interceptors.request.use(req => {

        this.setState({ error: null });

        return req;

      });

      axios.interceptors.response.use(

        response => response,

        error => {

          this.setState({ error: error });

        }

      );

    }

    errorConfirmHandler=()=>{

        this.setState({ error: null });

    }

    render() {

      return (

        <Aux>

          <Modal clicked={this.errorConfirmHandler} showBackdrop={this.state.error}>{this.state.error?this.state.error.message:null}</Modal>

          <WrappedComponent {...this.props} />

        </Aux>

      );

    }

  };

};

export default withErrorHandler;

hoc chưa wrappedComponent và axios, khi gọi thì dùng hoc này chứa component

modal chứa backdrop làm nền tối lại, đưa ra bảng thông báo chứa children

thực hiện trong didmount

 axios.interceptors.response.use

lấy error và set eror vào state, nghĩa là khi nhận reponse có eror thì set eror

axios.interceptors.request.use

đưa error về mặc định, nghĩa là khi request xong thì reset erorr

Khi sử dụng eror handler theo cách này, nhiều đối tượng được khởi tạo interceptor, dẫn tới bộ nhớ lớn lên ko cần thiết.

Vì vậy khi unmount thì ejct nó đi

Khi update đối tượng, nếu xảy ra error, đối tượng đó ko được mount vào, nên dùng did mount ko tối ưu - > dùng will mount

