import React, { Component } from 'react'

import Image from './Image';

import im1 from './logo192.png';

import im2 from './logo512.png';

import ErrorBoundary from './Error';

export default class  App2   extends Component {

  render() {

    return (

      <div>

        <ErrorBoundary>

        <Image im={im1}/> <br/>

        <Image im={im2}/> <br/>

        <Image im="No"/> <br/>

        </ErrorBoundary>

         </div>

    )

  }

}

import React, { Component } from 'react'

export default class Image extends Component {

  render() {

    if(this.props.im==="No")

    {

      console.log("kkjkjkjk");

  throw new Error("No image");

    }return (

      <div>Image

<img src={this.props.im} width="100px" height="100px"/>

<img src={this.props.im} width="100px" height="100px"/>

      </div>

    )

  }

}

import React,{Component} from "react";

class ErrorBoundary extends React.Component {

  constructor(props) {

    super(props);

    this.state = { error: null, errorInfo: null };

  }

  componentDidCatch(error, errorInfo) {

    // Catch errors in any components below and re-render with error message

    this.setState({

      error: error,

      errorInfo: errorInfo

    })

    // You can also log error messages to an error reporting service here

  }

  render() {

    if (this.state.errorInfo) {

      // Error path

      return (

        <div>

          <h2>Something went wrong.</h2>

          <details style={{ whiteSpace: 'pre-wrap' }}>

            {this.state.error && this.state.error.toString()}

            <br />

            {this.state.errorInfo.componentStack}

          </details>

        </div>

      );

    }

    // Normally, just render children

    return this.props.children;

  }

}

export default ErrorBoundary;