**리액트로 에러 처리하기**

1. 다음처럼 props 가 넘어오지 않았을 경우 에러 처리

//User.js

funtion User({ user }) {

return (

<div>

ID : {user.id}

username: {user.user.name}

</div>

);

};

//App.js

function App() {

return <User />;

}

* 실제화면에선 아무것도 안뜨고, 개발환경에선 에러가 뜸.
* 이때는 User.js에서 적절히 처리해주면 됨.

//User.js

funtion User({ user }) {

if(!user) {

return null;

}

…

}

* user값이 없으면 null을 렌더링하게 함. -> 아무것도 안뜸
* 보통 데이터를 네트워크 요청을 통하여 나중에 받아오게 되는 상황에서, 당장 데이터가 없으면 null을 보여주거나 <div>로딩중</div> 같은 결과물을 렌더링하면 됨.

1. props가 제대로 안넘어왔을 때 defaultProps를 설정해주는 방법도 있다

function Users({ users, onToggle }) {

if(!users) return null;

return (

<ul>

{users.map(user => (

<li key={user.id} onClick={() => onToggle(user.id)}>

{user.username}

</li>

))}

</ul>

);

};

Users.defaultProps = {

onToggle: () => {

console.warn(‘onToggle is missing’);

}

};

1. PropTypes 를 사용하는 방법도 있다
   1. 필요한 데이터를 넣지 않았을 때 개발단계에서 경고를 볼 수 있기 대문에 실수로 props 설정을 깜박하는 일을 방지할 수 있음.
   2. 단 사용법이 귀찮아서, TypeScrips 또는 Flow를 사용해서 처리하는 게 권장사항

**componentDidCatch로 에러잡아내기**

1. componentDidCatch로 에러를 잡아 낼 수도 있다.

//ErrorBoundary.js

import React, { Component } from 'react';

class ErrorBoundary extends Component {

    state = {

        error: false

    };

    componentDidCatch(error, info) {

        console.log('에러가 발생했습니다.');

        console.log({

            error,

            info

        });

        this.setState({

            error: true

        });

    }

    render() {

        if(this.state.error) {

            return <h1>에러발생!</h1>;

        }

        return this.props.children;

    }

}

export default ErrorBoundary;

* componentDidCatch 메소드의 첫번째 파라미터는 에러의 내용, 두번째는 에러가 발생한 위치를 알려줌
* 에러가 발생하면 에러가 발생했다는 문구를 렌더링하고, 아니면 this.props.children을 렌더링하도록 처리해줌.

//App.js

 return (

    <div>

      <ErrorBoundary>

       <User />

      </ErrorBoundary>

    </div>

  );

}

* User에서 에러가 발생하면, ErrorBoundary로 넘어가서, ErrorBoundary의 componentDidCatch가 잡아냄.

**Sentry 연동**

1. componentDidCatch 를 사용하면 사용자에게 오류가 발생했다는 걸 알려줄 수 있지만, 원래 사용자에게 오류가 노출되면 안된다.
2. 따라서 오류가 발생하면 바로 고쳐야하는데, 사용자가 사용중에 발생한 오류를 알려주는 서비스가 Sentry이다
3. 나중에 사용할 때 살펴보자