// reset App.js

import React from 'react';

// import PropTypes from "prop-types";

class App extends React.Component{

  state = {

    isLoding: true

  };

  //componentDidMount -> render 후에 작동

  // 6초 뒤 바뀐다!

  componentDidMount(){

    setTimeout(() => {

      this.setState({isLoding:false})

    }, 6000)

  }

  render() {

    const {isLoding} = this.state;

    return <div>{isLoding ? "Loading": "We are ready"}

    </div>;

  }

}

export default App;

**\*\* componentDidMount \*\***

setState가 true가 되는 “We are ready” 자리에 코드를 넣으면 된다!

**\*\* State 자리에 초기화할 필요 없이 setState에서 state 변수를 시작해도 된다.**

**취향 차이 \*\***

fetch() vs **axios**

axios = fetch 위에 있는 layer

**npm install axios**

**yts.lt -> 여기에서 movie api 사용 ( 토렌트 사이트 )**

**-> 하단의 api 누르고 List Movies의 json을 활용 할건데..**

**-> url이 종종 바뀌니까** [**https://yts-proxy.now.sh/list\_movies.json**](https://yts-proxy.now.sh/list_movies.json)

**-> 으로 해서 쓴다.**

  getMovies = async () => {

    const {data:{data:{movies}}} = await axios.get("https://yts-proxy.now.sh/list\_movies.json?sort\_by=rating")

  }

  componentDidMount(){

    this.getMovies();

  }

**비동기 방식 axios -> 끝날 때까지 기다린다.**

**<< json control >>**

**파일명.키.키.... => movies.data.data.movies**

    const {data:{data:{movies}}} = await axios.get("https://yts-proxy.now.sh/list\_movies.json")

    console.log(movies)

**위에서처럼 {키:{키:{키{...}}}}**

최종 키값이 변수로 사용된다.

**src/Movie.js**

import React from "react";

import PropTypes from "prop-types";

function Movie({id, year, title, summary, poster}){

    return <h5>{title}</h5>

}

Movie.propTypes = {

    id: PropTypes.number.isRequired,

    year: PropTypes.number.isRequired,

    title: PropTypes.string.isRequired,

    summary: PropTypes.string.isRequired,

    poster: PropTypes.string.isRequired

};

export default Movie;

- propTypes를 통해 변수의 타입 고정

- function에서 Movie의 동작을 선언

- 파라미터 여러 개 입력할 때, **{ } 없어서 30분 날려먹음**

얘로 불러오고 ( 경로 맞춰서 )

import Movie from "./Movie";

**App.js > render()**

  render() {

    const {isLoding, movies} = this.state;

    return (

      <div>

        {isLoding ? "Loading..."

        : movies.map(movie => {

          return (

            <Movie

              key={movie.id}

              id={movie.id}

              year={movie.year}

              title={movie.title}

              summary={movie.summary}

              poster={movie.medium\_cover\_image}

            />

          );

        })}

      </div>

    );

  }

}

- 삼항 연산자를 통해 **isLoading( F/T )**의 상태 분기 **– getMovies가 끝나면 setState**

  getMovies = async () => {

    const {data:{data:{movies}}} = await axios.get("https://yts-proxy.now.sh/list\_movies.json?sort\_by=rating")

    console.log(movies)

    // this.setState({movies:movies}) // 아래의 movies와 동일한 코드...

    this.setState({movies, isLoding:false})

  }

- state를 가져올 때는 **const { state명 , ... } = this.state;**

**-** map을 통해 movie마다 **<Movie />**를 선언하는데,

**파라미터를 다 넣어주어야 한다. -> isRequired이기 때문에**



**위 에러가 콘솔에 뜰 것이다! -> map은 unique key값이 필요하다.**

**movie에서는 id값이 키로 사용가능하다.**

              key={movie.id}

**App.js > render()**

import "./App.css";

해당 파일 이름으로 .css 만들어 스타일 관리

  render() {

    const {isLoding, movies} = this.state; console.log(movies)

    return (

      <section className="container">

        {isLoding ? (<div className="loader">

          <span className="loader\_\_text">Loading...</span>

        </div>

        ) : (

          <div className="movies">

            { movies.map(movie => (

              <Movie

                key={movie.id}

                year={movie.year}

                title={movie.title}

                summary={movie.summary}

                poster={movie.medium\_cover\_image}

                genres={movie.genres}

              />

            ))}

          </div>

        )}

      </section>

    );

  }

genres 추가 후 코드 변화

- section이나 span 등은 생략 ( 설명 X )

- div 내부에서 **“class”**라는 명을 써야하면, **(HTML로)** -> **className**으로 해줘라

- id는 더이상 필요없어서 삭제

**src/Movie.js**

import React from "react";

import PropTypes from "prop-types";

import "./Movie.css";

function Movie({year, title, summary, poster, genres}){

    return <div className="movie">

        <img src={poster} alt={title} title={title} />

        <div className="movie\_\_data">

            <h3 className="movie\_\_title">{title}</h3>

            <h5 className="movie\_\_year">{year}</h5>

            <ul className="genres">

                {genres.map ( (genre,index) =>

                    <li key={index} className="genres\_\_genre">

                        {genre}

                    </li>

                )}</ul>

            <p className="movie\_\_summary">{summary}</p>

        </div>

    </div>;

}

Movie.propTypes = {

    year: PropTypes.number.isRequired,

    title: PropTypes.string.isRequired,

    summary: PropTypes.string.isRequired,

    poster: PropTypes.string.isRequired,

    genres: PropTypes.arrayOf(PropTypes.string).isRequired

};

export default Movie;

- genres는 array형태, **PropTypes.arrayOf** 사용

- HTML 형태의 <ul></ul> 태그 활용

- map에서 키로 쓸 값이 없으면 index (아무 이름이나 가능하나 index 쓰자) 사용

**key={index}**로 잡아주면 unique key 경고는 사라진다.

끝