

2020-2  
웹 시스템 설계  
학습과제

*Week 14*

*React Frontend with Express Backend*

Composed by:  
WISE Research Lab Ajou University



AJOU UNIVERSITY

WISE

## 학습목표

- Axios를 통해 구성된 백엔드-프론트엔드 간 비동기 통신을 활용하여 React 프론트 엔드 및 Express 백엔드를 연동할 수 있다.

# React Router

---

## 1. React Router 란?

- React Router는 React 기반의 웹 어플리케이션에서 선언적으로 구성되는(declaratively composed) navigational components의 컬렉션이다.
- React Router의 컴포넌트는 기존 React로 구성되는 SPA(Single Page Application) 과 달리 location이나 history 같은 브라우저의 내장 API와 연동되어 다음과 같은 이점을 얻을 수 있다.
  - ✓ 웹 어플리케이션 내 특정 페이지에 대한 주소를 정의할 수 있어, 특정 페이지에 대한 북마크 및 즐겨 찾기가 가능함.
  - ✓ 뒤로 가기 버튼을 누르는 경우, 그 전에 웹 사이트가 아닌 구성된 웹 앱의 이전 페이지로 이동 가능
  - ✓ 새로 고침 버튼을 누르면, 웹 앱의 홈 페이지로 이동하지 않고 웹 앱의 현재 페이지가 새로 고침 됨.

## 2. React Router 설치

- React Router는 웹 version과 네이티브 version이 있으며, 웹 version의 React Router를 활용하는 경우, 아래의 명령어로 패키지를 설치한다.

```
npm -i react-router-dom
```

## 3. React Router 핵심 컴포넌트

### 3.1 Router(BrowserRouter)

- Router는 3.2, 3.3에서 설명하는 모든 컴포넌트의 최상위 컴포넌트이다.

```
<Router>
  <Switch>
    <Route exact path="/" component={c} />
    <Route exact path="/a" component={a} />
    <Route exact path="/a/:id" component={b} />
  </Switch>
</Router>
```

## 3.2 Route Matcher

- Route Matcher는 Switch 컴포넌트와 Switch 컴포넌트의 하위 컴포넌트인 Route를 통해 수행된다.

### 3.2.1 Switch

- <Switch> 태그는 렌더링 될 때, 현재의 URL과 <Route> 태그 내 정의된 path 가 match 하는 경우를 탐색한다. <sup>1</sup>

```
<Switch>
  <Route exact path="/" component={c} />
  <Route exact path="/a" component={a} />
  <Route exact path="/a/:id" component={b} />
</Switch>
```

### 3.2.2 Route

- <Route> 컴포넌트는 현재 URL 경로와 path가 매치될 경우 렌더링 될 컴포넌트를 정의한다.

```
<Route path="/about" component={About} />
```

### 3.2.3 Navigation(Link)

- <Link> 컴포넌트는 웹 어플리케이션 내에서 링크를 생성하기 위해 활용된다. <Link> 컴포넌트는 기존 HTML의 <a> 태그와 동일한 역할을 한다.

```
<Link to="/">Home</Link>
```

### 3.2.4 withRouter

- 컴포넌트를 withRouter를 통해 wrapping한 경우, withRouter는 wrapping된 컴포넌트에 대해 1) 객체의 history 프로퍼티, 2) 가장 가까운 router의 match 정보, 3) location 정보에 접근할 수 있다. <sup>2</sup>

```
import React from "react";
```

---

<sup>1</sup> 탐색 시, path와 현재의 URL이 일치하는 경우를 차례대로 탐색하고, 탐색이 성공하면 나머지 Route는 무시된다. 즉, Route 태그를 기입하는 순서의 경우, 가장 자세한 경로를 먼저 기입하고, 나중에 단순한 경로를 가진 Route를 정의하여, 잘못된 Route로 이동하는 경우가 없도록 해야 한다.

<sup>2</sup> withRouter를 쓰지않는 경우 Route 태그 내 렌더링할 컴포넌트로 지정되지 않은 컴포넌트는 location, match, history 정보에 접근할 수 없음.

```

import PropTypes from "prop-types";
import { withRouter } from "react-router";

// A simple component that shows the pathname of the current location
class ShowTheLocation extends React.Component {
  static propTypes = {
    match: PropTypes.object.isRequired,
    location: PropTypes.object.isRequired,
    history: PropTypes.object.isRequired
  };

  render() {
    const { match, location, history } = this.props;

    return <div>You are now at {location.pathname}</div>;
  }
}

// Create a new component that is "connected" (to borrow redux
// terminology) to the router.
const ShowTheLocationWithRouter = withRouter(ShowTheLocation);

```

#### 4. React Router 예제(첨부 파일 참조)

- 본 예제는 Link 컴포넌트를 활용하여 각 컴포넌트에 대한 링크를 구성하고, Switch 및 Route를 활용하여 특정 URL에 대한 접근 시, 렌더링 될 컴포넌트를 정의하는 예제이다.
- 예제의 D.js의 export default withRouter(D); 와 export default D; 를 비교하면서 withRouter가 어떠한 기능을 하는지 확인
- 예제 코드는 학습 보고서 업로드 시, 같이 업로드 되는 "react\_example\_react\_router.zip" 첨부 파일 참조
- 실행방법
  - ✓ Frontend 폴더에 접근 후, npm install -> npm start

# Axios

---

## 1. Axios란?

- Axios는 node.js와 브라우저를 위한 Promise 기반의 HTTP 클라이언트이다.
- Axios는 아래와 같은 기능들을 제공한다.
  - ✓ 브라우저로부터 XMLHttpRequest(즉, Ajax 기능수행가능) 생성
  - ✓ Node.js 로부터 Http 요청 생성
  - ✓ Promise API 지원
  - ✓ 요청 및 응답의 인터셉트
  - ✓ 요청 및 응답 데이터의 변환
  - ✓ 요청 취소
  - ✓ JSON 데이터의 자동 변환
  - ✓ 클라이언트 단에서의 XSRF에 대한 보호 지원

## 2. Axios 설치

```
npm install axios
```

## 3. Axios를 활용한 React-Express 간 연동 예시(첨부파일 참조)

- 본 예시는 Axios를 활용한 React-Express 간 연동을 통해 프론트엔드 측에서 전달한 정보를 백엔드 측에서 받은 후, 프론트 엔드 측으로 다시 동일하게 전달하는 과정(echo)을 수행한다.
- 백엔드 측에서 프론트엔드 측으로 다시 전달된 데이터는 console.log를 통해 출력된다.
- 예제 코드는 학습 보고서 업로드 시, 같이 업로드 되는 "react\_example\_axios.zip" 첨부파일 참조
- 실행방법
  - ✓ Frontend 폴더에 접근 후, npm install -> npm start
  - ✓ Backend 폴더에 접근 후, npm install -> npm start

# 학습과제

---

## 1. 과제 개요

- React로 구성된 프론트 엔드와 Express로 구성된 백 엔드를 연동하여 '영화 리뷰 관리' 웹 어플리케이션을 구현한다.
- 과제의 일부 코드에 대해 학습 보고서와 함께 올라오는 "react\_homework.zip" 첨부파일로 제공 예정
- CSS에 대한 요구사항은 없음.

## 2. 학습과제 파일 구성 및 과제 제출 양식

- 학습과제의 Frontend는 'npx create-react-app [프로젝트 이름]' 명령어로 통해 만들어진 스텁레톤 프로젝트 구성되며, 추가 및 변경되는 파일은 아래와 같다.
- Frontend 폴더

폴더 명	추가 및 변경 파일 이름
src/pages	<u>App.js</u>
	<u>CreateReview.js</u>
	<u>Reviews.js</u>
src/lib	<u>request.js</u>
	<u>star.js</u>
src/Components	<u>Form.js</u>
	<u>Header.js</u>
	<u>List.js</u>
	<u>ReviewCard.js</u>

- 학습과제의 Backend는 'express-generator [프로젝트 이름]' 명령어로 통해 만들어진 스텁레톤 프로젝트 구성되며, 추가 및 변경되는 파일은 아래와 같다.

- Backend 폴더

폴더 명	추가 및 변경 파일 이름
	<u>app.js</u>
<b>models</b>	<u>review.js</u>
<b>routes</b>	<u>review.js</u>

### 3. 제출방법 및 주의사항

- 압축 파일의 이름을 "Homework2\_자신의학번.zip"으로 지정하여 폴더(프로젝트)를 압축한다.
- 설치된 node package 들을 제외한 프로젝트 폴더를 압축해서 제출
  - ✓ 즉, node\_modules 폴더는 압축파일에 포함시키지 않는다.
- 압축된 프로젝트 폴더와 함께, 제출한 학습보고서 1~5 을 작성하여 함께 제출한다.
- 지각 제출 시, 0 점으로 처리한다.
- 채점 시 완성도의 평가는 Google 의 Chrome 브라우저에서 렌더링 된 화면을 기준으로 함

※ "4. 프로그램 설계", "5. 실행 예시"를 참고하여 프로그램이 올바르게 동작하도록 구현한다.



## 4. 프로그램 설계

- 설명을 진행하기 전에, 컴포넌트의 참조 설명에 있어 아래와 같은 설명은 다음과 같은 코드에 대응된다.
  - ✓ A 가 B props의 매개변수로 전달되는 C 컴포넌트를 참조한다.

```
<C B={A} />
```

### 4.1 Frontend : Page 폴더 : Route 정의 및 컴포넌트로 이루어진 페이지 정의

#### 4.1.1 App.js

- App.js 코드는 학습보고서와 함께 올라오는 "react\_homework.zip" 첨부파일 참조
- render()
  - ✓ [Header 컴포넌트](#)를 참조한다.
  - ✓ Switch 및 Route 를 활용하여 페이지 폴더 내 정의된 컴포넌트에 대하여 특정 path 주소와 연동한다. (Route 태그 내 Path 정의 시, exact를 쓰거나, 긴 path부터 먼저 쓰도록 한다.)
    - ◆ Path "/" 와 Page 폴더 내 [Reviews 컴포넌트](#) 연동
    - ◆ Path "/review/new" 와 Page 폴더 내, [CreateReview 컴포넌트](#) 연동

### 4.1.1 Reviews 컴포넌트

- Component 폴더 내, List 컴포넌트를 참조하여 리뷰 리스트를 렌더링하는 컴포넌트이다.
- 서버의 리뷰 리스트를 받고, 해당 리스트를 활용하여 렌더링하기 위해 아래와 같이 constructor 내에 this.state를 정의한다.

```
this.state = { reviews: [] }
```

- async componentDidMount()
  - ✓ 리렌더링 수행을 위해 this.setState 내, await [request.getReviews\(\)](#) 의 매개변수로 정의하여 this.state 내 reviews 정보를 갱신한다.
- render()
  - ✓ Reviews 컴포넌트의 this.state의 reviews의 길이가 0인 경우, null을 렌더링하고 그렇지 않은 경우, this.state.reviews 가 data props로 전달되는 [List 컴포넌트](#)를 참조한다.

### 4.1.2 CreateReview 컴포넌트

- 새로운 리뷰 정보를 등록하는 컴포넌트이다
- render()
  - ✓ [request.createReview](#) 함수가 onSubmit props의 매개변수로 전달되는 [Form 컴포넌트](#)를 참조한다.

## 4.2 Frontend : components 폴더 : 재사용되는 컴포넌트를 정의

### 4.2.1 Header.js

- render()
  - ✓ 자신의 학번
  - ✓ Home 링크
    - ◆ React Router의 Link 태그로 정의되며, ["/" 경로](#)로 이동하기 위한 태그
  - ✓ New 버튼
    - ◆ React Router의 Link 태그로 정의되며, ["/review/new" 경로](#)로 이동하기 위한 태그

### 4.2.2 Form.js

- constructor
  - ✓ this.form 내 영화 리뷰에 대한 각 요소를 다음과 같이 정의하고, 각 요소에 대해 React.createRef()를 통해 Ref를 생성 후, 아래와 같이 부착한다.

```
this.form = {  
  movie_name: React.createRef(),  
  review_content: React.createRef(),  
  rate: React.createRef()  
}
```

- async submit()
  - ✓ this.form의 각 요소의 current.value를 활용하여 onSubmit props로 받은 함수인 request.createReview 를 위한 매개변수를 정의한다.
  - ✓ 정의된 매개변수와 함께 await this.props.onSubmit를 호출한다.
  - ✓ [this.props.history.push\("/"\)](#)를 통해 **홈화면** 으로 이동한다.
- cancel()
  - ✓ this.props.history.goBack()을 통해 이전페이지로 이동한다.
- render()
  - ✓ movie\_name, review\_content, rate 별로 input tag 를 구성한 후, input tag 에 this.form 내 의 대응되는 이름을 가진 엘리먼트에 대하여 ref 를 생성한다.
  - ✓ 제출 및 취소를 위한 버튼을 구성한다. 각 버튼의 onClick에는 앞서 정의 한 this.submit.bind(this) 와 this.cancel.bind(this)를 할당 한다.
- export : withRouter를 활용하여 export 한다.

### 4.2.3 List.js

- constructor
  - ✓ 리뷰 삭제되는 경우에 대해 리렌더링을 수행하기 위해 this.state를 다음과 같이 정의한다.

```
this.state = { reviews: this.props.data}
```

- deleteReviewById
  - ✓ filter를 활용하여 함수의 매개변수로 받은 id에 대해 해당 id를 가진 리뷰를 현재 this.state에 저장된 review에서 삭제 한다.
  - ✓ 이후, 갱신된 리뷰리스트를 this.setState를 통해 반영한다.
- render()
  - ✓ 만약 data.length 가 0이 아닌 경우, map 을 활용하여 각 리뷰 정보가 1) review props의 매개변수로 전달되고 2) onDelete props의 매개변수로 {this.deleteReviewById.bind(this)} 가 전달되는 [ReviewCard](#)를 렌더링하여 리스트를 구성한다.

#### 4.2.4 ReviewCard.js

- async deleteReview()
  - ✓ 리뷰 삭제를 위해 review props 내의 \_id가 매개변수로 정의된 await request.deleteReview 를 호출한다.
  - ✓ 백엔드 측에서 삭제된 리뷰를 리뷰 리스트에 반영하기 위하여 review props내의 \_id가 매개변수로 정의된 onDelete 함수를 호출한다.
- render()
  - ✓ review props 를 통해 받아온 review를 다음과 같이 review 변수에 할당한다.

```
const { review } = this.props;
```

- ✓ [stars 함수](#)에 review.rate 를 매개변수로 부여 후 호출하여, 평점을 렌더링한다.
- ✓ review.movie\_name과 review.review\_content 를 렌더링한다.
- ✓ 삭제 버튼을 정의한다. 이때, 삭제 버튼의 onClick 이벤트에 {this.deleteReview.bind(this)} 를 할당한다.

### 4.3 Frontend : lib 폴더 : Axios 통신 함수 및 별찍기 함수 정의

#### 4.3.1 request.js

- Axios를 활용하여 다음과 같은 기능을 하는 Async 함수를 각각 정의한 후, export 한다.

함수명	매개변수	기능
getReviews		<ul style="list-style-type: none"> <li>• <a href="#">리뷰 리스트를 백엔드로 요청</a> 후, 응답으로 받은 리뷰 리스트를 반환한다.</li> </ul>
createReview	{movie_name, review_content, rate}	<ul style="list-style-type: none"> <li>• <a href="#">매개변수로 받은 신규 리뷰 정보에 대해 백엔드에 리뷰의 저장을 요청</a> 후, 저장의 결과값을 반환한다.</li> </ul>
deleteReview	{id}	<ul style="list-style-type: none"> <li>• <a href="#">매개변수로 받은 id를 가진 리뷰에 대하여 백엔드로 삭제 요청</a> 후, 결과 값을 반환한다.</li> </ul>

### 4.3.2 star.js

- 학습보고서와 함께 업로드되는 "react\_homework.zip" 첨부파일 참조

## 4.4 Backend : routes 폴더

- 백엔드에서 제공해야 하는 API는 다음과 같다.

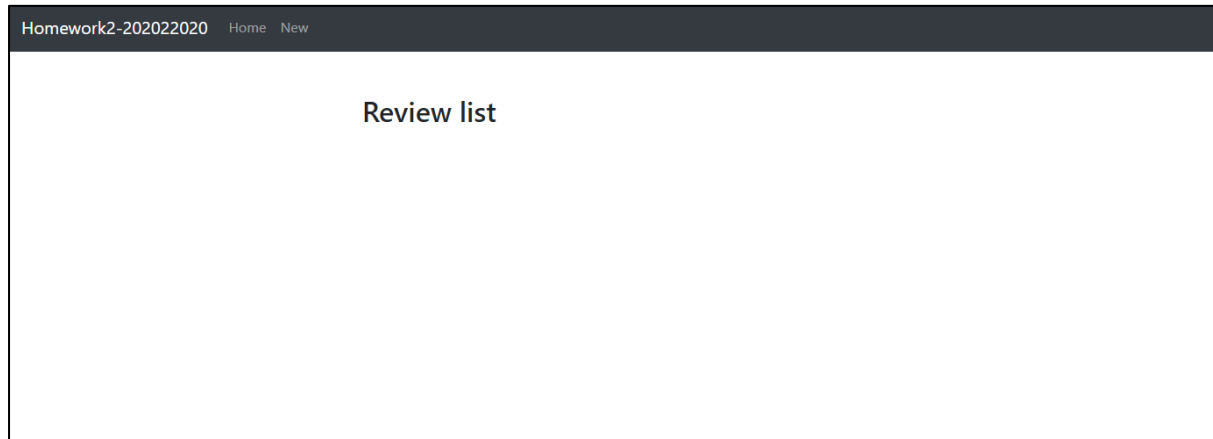
Method	URI	동작
GET	/	<ul style="list-style-type: none"> <li>• <u>현재 등록된 리뷰의 목록을 응답</u>한다.</li> <li>• 서버가 응답하는 리뷰의 목록은 [ <u>{movie_name, review_content, rate}</u> ] 형태의 객체 배열이다.</li> </ul>
POST	/	<ul style="list-style-type: none"> <li>• <u>리뷰 정보를 리뷰 목록에 추가</u>한다.</li> <li>• 먼저 추가되는 순서대로 목록에 저장되며, 클라이언트가 전송하는 영화 정보는 { <u>movie_name, review_content, rate</u>} 형태의 객체이다.</li> <li>• 서버에는 {<u>movie_name, review_content, rate, date</u>}의 형태로 리뷰 정보가 저장된다.</li> </ul>
DELETE	/:id	<ul style="list-style-type: none"> <li>• <u>id번째로 추가된 리뷰의 정보를 삭제</u>한다.</li> </ul>

## 4.5 Backend : models

- 학습보고서와 함께 업로드되는 "react\_homework.zip" 첨부파일 참조

## 5. 실행 예시

### 1. 시작화면



### 2. 리뷰 추가 화면 (Header 컴포넌트의 New 버튼 클릭 시)

A screenshot of a web application interface showing a 'Movie Review' form. At the top, there is a dark grey header bar with the text 'Lab0-201824417' on the left and 'Home New' on the right. Below the header, the form is titled 'Movie Review'. It contains three input fields: 'I'm gonna review' with a placeholder 'Movie Name', 'My review is' with a placeholder 'Review', and 'So, I'll give this move a rating of' with a placeholder 'Rating'. At the bottom of the form, there are two buttons: 'OK' (blue) and 'Cancel' (yellow).

### 3. 시작화면 (리뷰 추가 완료 후, history 를 활용하여 이동됨)

## Review list

- Rate (☆☆☆☆)

### Inception

Have you ever felt like waking up from a dream to see its continuation? This idea disturbs Christopher Nolan ever since he was 16. Already in the future, as he developed this idea in his head, he was visited by another captivating feeling. And what if it were possible to study and manage the place of sleep? Now the British cinematographer is 39 years old, and the world has finally come to the opportunity to get to know what Nolan was so stunning, fast and loud all these years. "The Beginning" is a sci-fi thriller about the architecture of the mind. If for some reason this has not yet become a reason for the strongest intrigue, it is never too late to recreate it.

[Delete](#)

## 4. 복수의 리뷰가 추가된 예제

## Review list

- Rate (☆☆☆☆)

### Inception

Have you ever felt like waking up from a dream to see its continuation? This idea disturbs Christopher Nolan ever since he was 16. Already in the future, as he developed this idea in his head, he was visited by another captivating feeling. And what if it were possible to study and manage the place of sleep? Now the British cinematographer is 39 years old, and the world has finally come to the opportunity to get to know what Nolan was so stunning, fast and loud all these years. "The Beginning" is a sci-fi thriller about the architecture of the mind. If for some reason this has not yet become a reason for the strongest intrigue, it is never too late to recreate it.

[Delete](#)

- Rate (☆☆)

### Inside Out

First time I write a review, but this "movie" has made me to. Only gave 2 for the animation, otherwise, it would have been 1 when considering the enjoy factor it provided. The story is way too boring, the "mind" characters are devastatingly annoying. Not funny enough for a children movie. The message it tries to communicate is overly artificial, contrived and shallow. I terribly regret wasting the time watching it, waiting for it to become better. It was a real pain watching it, in the end. Cannot understand how it has a rating over 8.

[Delete](#)

- Rate (☆☆☆☆☆)

### Blade Runner

An incredibly beautiful-looking film as one would expect with director Ridley Scott

[Delete](#)

## 5. 삭제 버튼을 눌러 리뷰 중 하나가 삭제된 예시

Lab0-201824417   Home   New

### Review list

- Rate (☆☆☆☆)

#### Inception

Have you ever felt like waking up from a dream to see its continuation? This idea disturbs Christopher Nolan ever since he was 16. Already in the future, as he developed this idea in his head, he was visited by another captivating feeling. And what if it were possible to study and manage the place of sleep? Now the British cinematographer is 39 years old, and the world has finally come to the opportunity to get to know what Nolan was so stunning, fast and loud all these years. "The Beginning" is a sci-fi thriller about the architecture of the mind. If for some reason this has not yet become a reason for the strongest intrigue, it is never too late to recreate it.

Delete

- Rate (☆☆☆☆)

#### Blade Runner

An incredibly beautiful-looking film as one would expect with director Ridley Scott

Delete