

# 11. 리액트 라우터 (실습, 1/2)

Prof. Seunghyun Park ([sp@hansung.ac.kr](mailto:sp@hansung.ac.kr))

Division of Computer Engineering

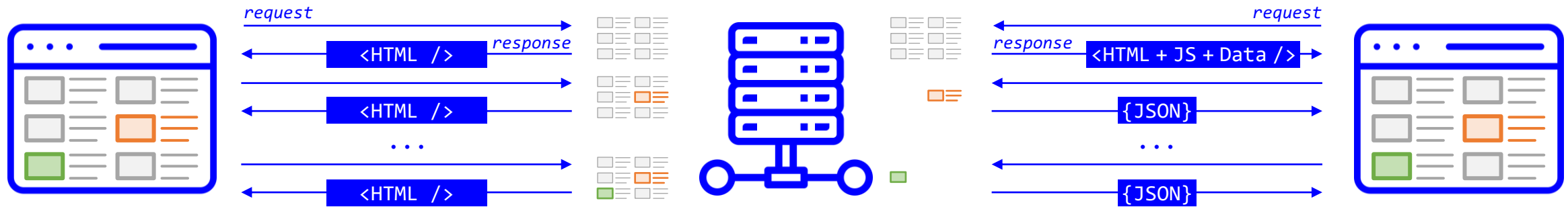
# 실습 목표: 11장. 리액트 라우터 (1)

---

## # 홈페이지 메인 화면 구성 (메인 메뉴, 서브 메뉴, 내용 영역)

- 실습43-1. 메인 메뉴 라우팅 처리 <Routes>, <Route>
- 실습43-2. 에러 페이지 준비
- 실습43-3. 메인 메뉴 링크 연결
  - 실습43-3a. <Link to />와 <a href /> 비교
- 실습43-4. 서브 메뉴의 페이지 생성 및 라우팅 <Outlet>
- 실습44-1. 리다이렉트, <Navigate>
- 실습44-2. 메뉴별 디폴트 컴포넌트 지정
- 실습44-3. 메뉴 구현 (메인메뉴, 서브메뉴)
- 실습44-4. 스타일 적용
- 실습45. 동적 라우팅: 라우팅 파라미터 구현

# Single Page Application



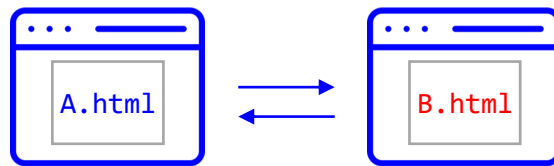
## # 페이지 요청 및 처리

- 매 요청시마다 서버는 렌더링 할 페이지를 반환
  - 변경되지 않는 영역까지 request & response 수행
  - 서버 리소스와 네트워크 트래픽 낭비

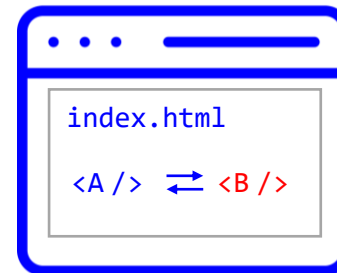
## • Single page application

- 뷰 렌더링을 브라우저가 담당 (client-side rendering)
- 하나의 페이지를 로딩한 후, 변경된 부분만 교체

## # 페이지 이동

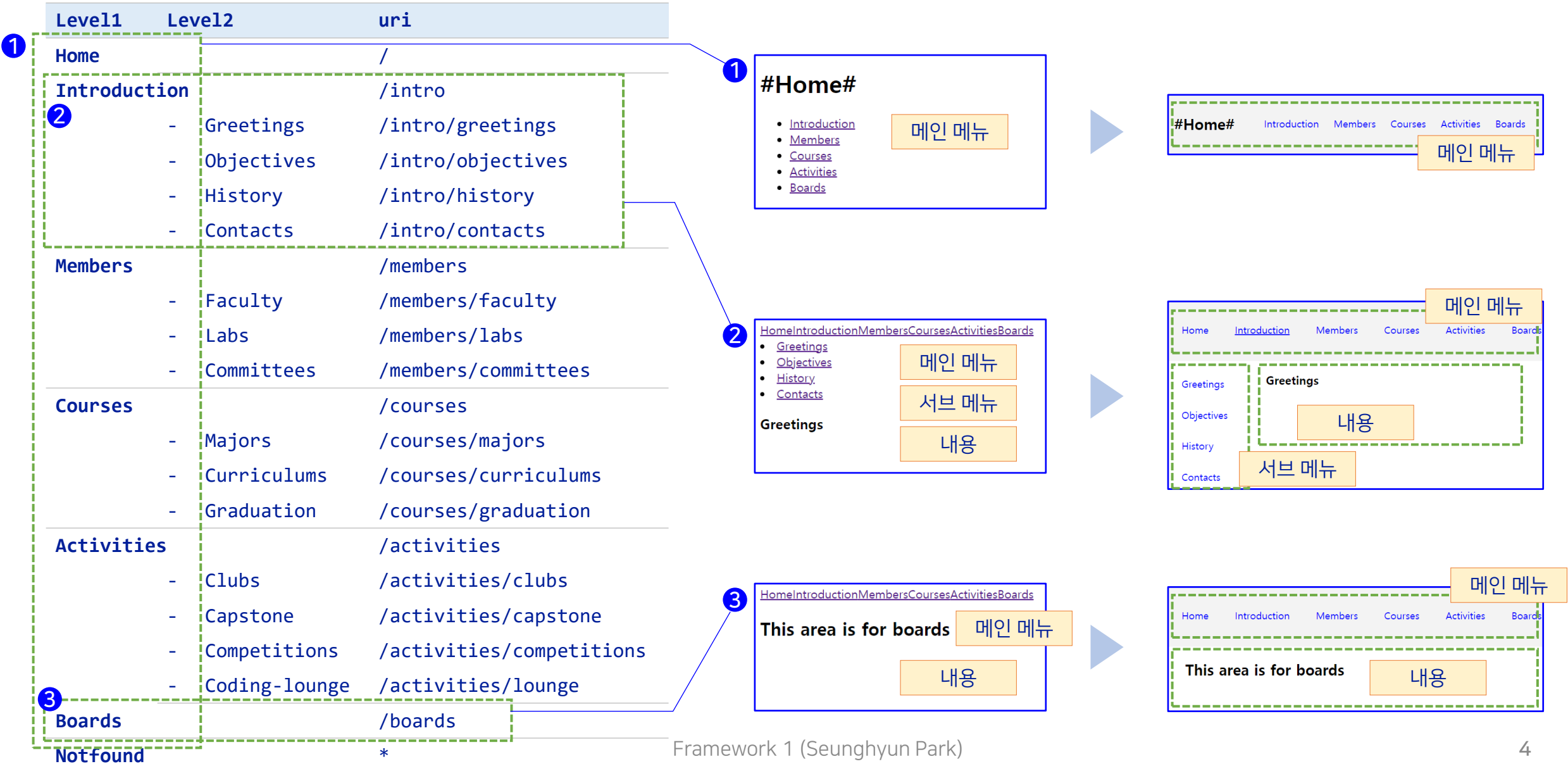


별도의 페이지로 전환



- 페이지 단위의 컴포넌트 반환을 위해 라우팅 필요
  - ※ 라우팅: 클라이언트 요청에 맞는 페이지 반환
- 리액트 라이브러리는 라우팅 기능 제공하지 않으므로 별도의 라우팅 라이브러리 필요

# 페이지 구성 (예: 학부 홈페이지)



# 실습43-1. 메인 메뉴 라우팅 처리 (<Routes>, <Route> 활용)

<https://reactrouter.com/en/main> [🔗](#)

❖ 메인 메뉴를 구성하는 6개의 컴포넌트를 생성하고, <Routes />와 <Route />를 활용하여 각 컴포넌트를 구조에 맞게 라우팅하세요.

```
/* ex43-1/src/index.js */
import { createRoot } from 'react-dom/client';
import App from './App';

const element = document.getElementById('root')
const root = createRoot(element);
root.render(
  <BrowserRouter />
);
```

<BrowserRouter> 활용 [🔗](#)

```
/* ex43-1/src/components/Pages.js */
export const Home = () => (
  <div>
    <h1>#Home#</h1>
  </div>
);
export const Introduction = () => (...);
export const Members = () => (...);
export const Courses = () => (...);
export const Activities = () => (...);
export const Boards = () => (...);
```

```
/* ex43-1/src/App.js */
```

```
const App = () => (
```

```
  <Home />
  <Introduction />
  <Members />
  <Courses />
  <Activities />
  <Boards />
);
```

```
export default App;
```

<Routes>, <Route> 활용 [🔗](#) [🔗](#)

```
> npm install react-router-dom
```

리액트 라우터 패키지 설치

# 실습43-1. 결과 확인

---

① localhost:3000에 <Home /> 컴포넌트 렌더링

**#Home#**

② localhost:3000/intro에 <Introduction /> 컴포넌트 렌더링

**#Introduction#**

③ <Members />, <Courses />, <Activities />, <Boards /> 컴포넌트도 확인

/members    /courses    /activities    /boards

④ 정의되지 않은 주소도 확인

/school

## 실습43-2. 에러 페이지 준비

❖ 정의되지 않은 주소를 처리하기 위한 에러 페이지 컴포넌트 `<Notfound />`를 생성하고 라우팅으로 연결하세요.

```
/* ex43-2/src/App.js */
import { Route, Routes } from 'react-router-dom';
import { Home, Introduction, Members, Courses, Activities,
  Boards, Notfound } from './components/Pages';

const App = () => (
  <Routes>
    <Route path="/" element={<Home />} />
    <Route path="/intro" element={<Introduction />} />
    <... />
    <Route path="*" element={<Notfound />} />
  </Routes>
);

export default App;
```

`<Notfound />` 컴포넌트 생성하여 참조

Routes 하위에 정의되지 않은 path는 `<Notfound />`로 이동

① 정의되지 않은 주소 확인

localhost:3000/school

Page not found at /school

```
/* ex43-2/src/components/Pages.js */
import { useLocation } from "react-router-dom";

export const Home = () => (...);
export const Introduction = () => (...);
export const Members = () => (...);
export const Courses = () => (...);
export const Activities = () => (...);
export const Boards = () => (...);

export const Notfound = () => {
  const locations = useLocation();
  return (
    <div>
      <h3>Page not found at {locations.pathname}</h3>
    </div>
  );
};
```

[[Prototype]]: Object  
hash: ""  
key: "default"  
pathname: "/no"  
search: ""  
state: null

**useLocation() 활용**

- 현재 URL 주소를 받아오는 hooks
- react-router-dom에 포함

## 실습43-3. 메인 메뉴 링크 연결 <Home />

❖ 실습43-1에서 생성한 메인 메뉴의 컴포넌트를 링크로 연결하세요.

```
/* ex43-3/src/components/Pages.js */
import { useLocation, Link } from "react-router-dom";

export const Home = () => (
  <div>
    <h1>#Home#</h1>
    <nav>
      <ul>
        <li><Link to="intro">Introduction</Link></li>
        <li><Link to="members">Members</Link></li>
        <li><Link to="courses">Courses</Link></li>
        <li><Link to="activities">Activities</Link></li>
        <li><Link to="boards">Boards</Link></li>
      </ul>
    </nav>
  </div>
);
...
```

localhost:3000

### #Home#

- Introduction
- Members
- Courses
- Activities
- Boards

localhost:3000/intro 로 이동

localhost:3000/intro

### #Introduction#

```
export const Introduction = () => (...);
export const Members = () => (...);
export const Courses = () => (...);
export const Activities = () => (...);
export const Boards = () => (...);
export const Notfound = () => {...};
```



# 실습43-3a. <Link to />와 <a href /> 비교

```
/* ex43-3a/src/components/Pages.js */
import { useLocation, Link } from "react-router-dom";
export const Home = () => (
```

```
...
<li><Link to="intro">Introduction Link_to</Link></li>
<li><a href="intro">Introduction a_href</a></li>
...
```

```
);
...
```

ex43-3a/public/index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>ex43</title>
</head>
<body>
  <div id="root"></div>
</body>
</html>
```

ex43-3a/public/intro/index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>ex43/intro</title>
</head>
<body>
  App: ex43-3a/intro/index.html
</body>
</html>
```

## #Home#

- [Introduction](#)
- [Introduction using a href](#)
- [Members](#)
- [Courses](#)
- [Activities](#)
- [Boards](#)

```
▼<li>
  ::marker
  <a href="/intro">Introduction</a>
</li>
▼<li>
  ::marker
  <a href="/intro">Introduction using a_href</a>
</li>
```

<Link to>

localhost:3000/intro

기존 페이지에서

public/index.html

<App /> ← 컴포넌트 교체

Route <Introduction />

#Introduction#

<a href>

localhost:3000/intro

새로운 페이지 로드

public/intro/index.html

App: ex11-01/intro/index.html

# 실습43-4. 서브 메뉴의 페이지 생성 및 라우팅, <Outlet />

❖ 서브 메뉴가 렌더링 할 내용물 (컴포넌트)를 모두 생성하고, 서브 메뉴 페이지가 하위 컴포넌트를 포함하도록 <Outlet />을 활용하세요.

Level1	Level2	uri
Home	①	
Introduction		/intro
- Greetings		/intro/greetings
- Objectives		/intro/objectives
- History		/intro/history
- Contacts		/intro/contacts
Members		
...		

## ① 메인 메뉴 (Level1)

{ Home, Introduction, Members, Courses, Activities, Boards }

## ② Introduction의 서브 메뉴 (Level1→Level2)

{ Greetings, Objectives, History, Contacts }

localhost:3000

#Home#

- [Introduction](#)
- [Members](#)
- [Courses](#)
- [Activities](#)
- [Boards](#)

localhost:3000/intro

#Introduction#

localhost:3000/intro/greetings

#Introduction#

Greetings

localhost:3000/intro/objectives

#Introduction#

Education objectives

... {history, contacts}

# 실습43-4. 서브 메뉴의 하위 컴포넌트 구현 및 라우팅

```
/* ex43-4/src/components/Pages.js */
import { useLocation, Link, Outlet } from "react-router-dom";
export const Home = () => (...);
```

```
/* Introduction: {Greetings, Objectives, History, Contacts} */
```

```
export const Introduction = () => (
  <div>
    <h1>#Introduction#</h1>
    <div>
      <Outlet />
    </div>
  </div>
);
```

③

<Outlet /> 사용하여 레이아웃 구성:  
컴포넌트를 렌더링 할 때, 라우팅 구조에 따라  
해당 컴포넌트를 포함하는 상위 컴포넌트로 반환

```
export const Greetings = () => (
  <div>
    <h3>Greetings</h3>
  </div>
);
```

①

서브 메뉴의 하위 컴포넌트 생성

```
export const Objectives = () => (...);
...
```

```
/* ex43-4/src/App.js */
```

```
import { Route, Routes } from 'react-router-dom';
import { Home, NotFound,
  Introduction, Members, Courses, Activities, Boards,
  Greetings, Objectives, History, Contacts,
  Faculty, Labs, Committees,
  Majors, Curriculums, Graduation,
  Clubs, Capstone, Competitions, Lounge
} from "../components/Pages";
```

```
const App = () => (
```

```
  <Routes>
```

```
    <Route path="/" element={<Home />} />
```

```
    <Route path="/intro/*" element={<Introduction />} />
```

```
    <Route path="greetings" element={<Greetings />} />
```

```
    <Route path="objectives" element={<Objectives />} />
```

```
    <Route path="history" element={<History />} />
```

```
    <Route path="contacts" element={<Contacts />} />
```

```
  </Routes>
```

```
  <Route path="/members/*" element={<Members />} />
```

```
  ...
```

```
  <Route path="*" element={<NotFound />} />
```

```
</Routes>
```

```
);
```

```
export default App;
```

②

라우팅 구조에 따라  
상위/하위 컴포넌트를  
중첩하여 라우팅