

11. 리액트 라우터 (2)

Prof. Seunghyun Park (sp@hansung.ac.kr)

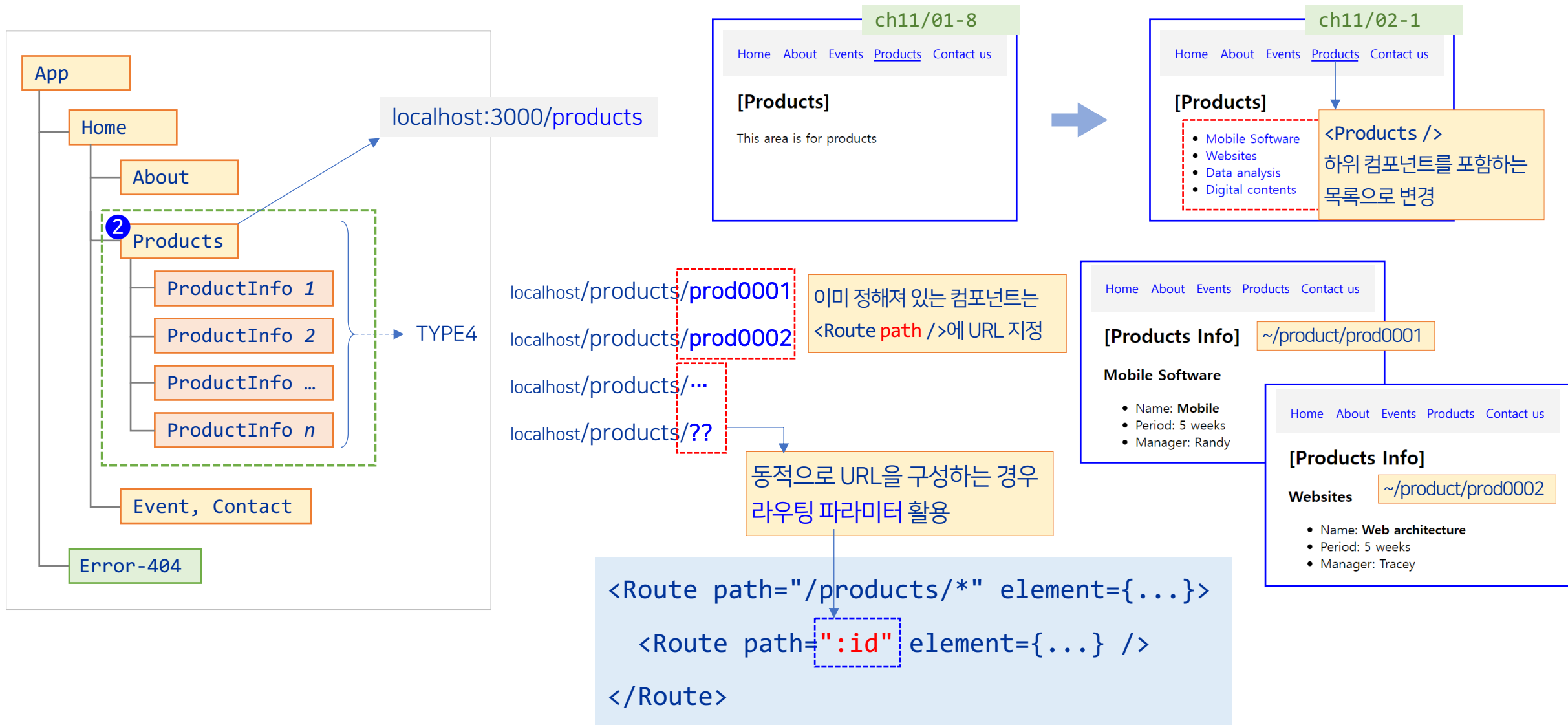
Division of Computer Engineering

학습 목표: 11장. 리액트 라우터 (2)

- Single Page Application
 - Client-side rendering
- React Router
 - `<Routes />`, `<Route />`로 컴포넌트 전환 (주소 직접 입력)
 - `useLocation()` hooks로 주소정보 반환
 - `<Link />`로 라우터 링크 제공
 - 중첩 Route와 `<Outlet />`, `<Navigate />`
 - `useRoutes()` hooks
- 동적 라우팅
 - 라우팅 파라미터
 - 쿼리
- React Context (6장)

동적 라우팅: 라우팅 파라미터

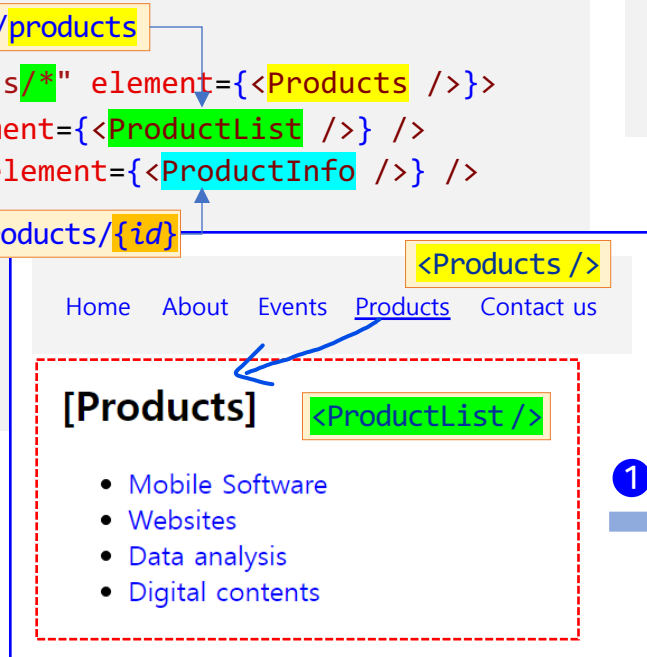
http://localhost:3000/products/prod0001



라우팅 정의: 목록 <ProductList />과 내용 <ProductInfo /> 컴포넌트

```
/* ch11/proj/02-1/src/App.js */
import { ..., Products, ProductList, ProductInfo, ... }
  from "../components/Pages";
```

```
const App = () => (
  <Routes>
    <Route path="/" element={<Home />} />
    <Route path="/about/" element={<About />} />
    <Route path="" element={<Company />} />
    <Route path="services" element={<Services />} />
    ...
  </Route>
  <Route path="/products/*" element={<Products />} />
    <Route path="" element={<ProductList />} />
    <Route path=":id" element={<ProductInfo />} />
  </Route>
  ...
</Routes>
);
export default App;
```

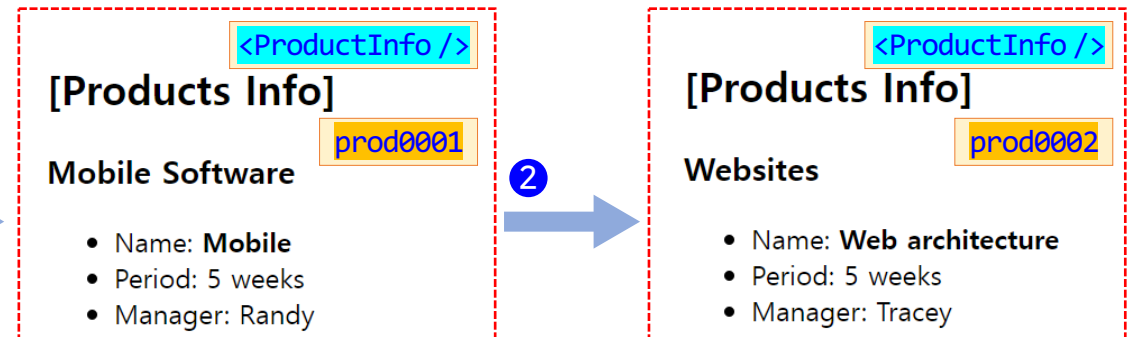


```
ch11/proj/02-1/src/data/product-data.json

[
  {
    "Category": "Mobile Software",
    "Id": "prod0001",
    "Name": "Mobile",
    "Period": "5 weeks",
    "Manager": "Randy"
  },
  {
    "Category": "Data analysis",
    "Id": "prod0003",
    ...
  },
  {
    "Category": "Digital contents",
    "Id": "prod0004",
    ...
  },
  {
    "Category": "Websites",
    "Id": "prod0002",
    ...
  },
  ...
],
```

<Products /> 하위의 컴포넌트:

- 1 페이지 전환시 목록은 <ProductList />로, 상세 내용은 <ProductInfo />로 컴포넌트 교체
- 2 <ProductInfo />으로 내용물을 보여주는 경우, 데이터 (JSON)에 따라 라우팅 전환



컴포넌트 교체를 위한 페이지 구성 <Products />

```
/* ch11/proj/02-1/src/components/Pages.js */
import { Link, Outlet, useParams } from "react-router-dom";
import productData from "../data/product-data.json";
```

```
...
const PageTemplate = ({ children }) => (
  <div className="page">
    <MainMenu />
    {children}
  </div>
);
```

```
export const Products = () => (
```

```
  <PageTemplate>
```

```
    <section className="products">
```

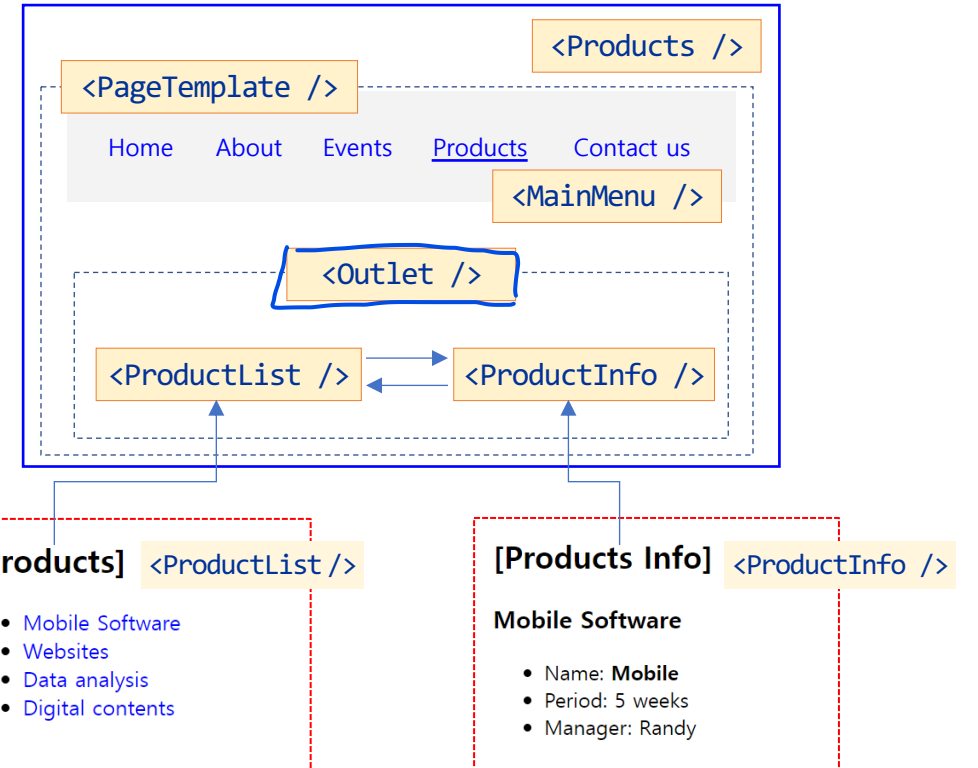
```
      <Outlet />
```

```
    </section>
```

```
  </PageTemplate>
```

```
);
```

```
...
```



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

```
export const ProductInfo = () => { ... };
```

페이지 1: 콘텐츠 목록 <ProductList />

```
/* ch11/proj/02-1/src/components/Pages.js */
```

```
...
```

```
export const ProductList = () => (
```

```
<>
```

```
<h1>[Products]</h1>
```

```
<ul>
```

```
{productData.map( (prd, i) => (
```

```
<li key={i}>
```

```
<Link to={prd.Id}>{prd.Category}</Link>
```

```
</li>
```

```
)))
```

```
</ul>
```

```
</>
```

```
);
```

데이터 (배열 productData)에서
배열의 모든 요소를 탐색하면서 목록으로 <Link />를 생성

[Products]

- Mobile Software
- Websites
- Data analysis
- Digital contents

productData[#].Category

<ProductList />

```
productData = [
```

```
{
```

```
"Category": "Mobile Software",
```

```
"Id": "prod0001",
```

```
"Name": "Mobile",
```

```
"Period": "5 weeks",
```

```
"Manager": "Randy"
```

```
},
```

```
{ ... },
```

```
{ ... },
```

```
{ ... }
```

```
];
```

페이지2: 상세 콘텐츠 정보 <ProductInfo />

```
export const ProductInfo = () => {  
  const { id } = useParams();  
  console.log("id: ", id);
```

/products/:id

useParams() hooks:

- 현재 URL의 동적 매개변수 정보를 담고 있는 객체 반환

{ id: prod0001 }

```
  const product = productData.reduce(  
    (obj1, obj2) => obj1.Id == id ? obj1 : obj2 );
```

데이터 (배열 productData)에서

Id 값이 id와 동일한 값 (예: prod0001) 을 갖는 요소 (객체 obj1)
→ product에 반환

```
  if (!product) return <div>No product information</div>;  
  console.log(product);
```

```
  return (  
    <>
```

```
    <h1>[Products Info]</h1>
```

```
    <h3>{product.Category}</h3>
```

```
    <ul>
```

```
      <li>Name: <b>{product.Name}</b></li>
```

```
      <li>Period: {product.Period}</li>
```

```
      <li>Manager: {product.Manager}</li>
```

```
    </ul>
```

```
  </>
```

productData = [

```
{  
  "Category": "Mobile Software",  
  "Id": "prod0001",  
  "Name": "Mobile",  
  "Period": "5 weeks",  
  "Manager": "Randy"  
},  
...  
];
```

obj1
obj1.Id

window.location

- href: http://localhost:3000/products/prod0001
- host: localhost:3000
- hostname: localhost
- origin: http://localhost:3000
- pathname: /products/prod0001
- port: 3000
- protocol: http

[Products Info]

<ProductInfo />

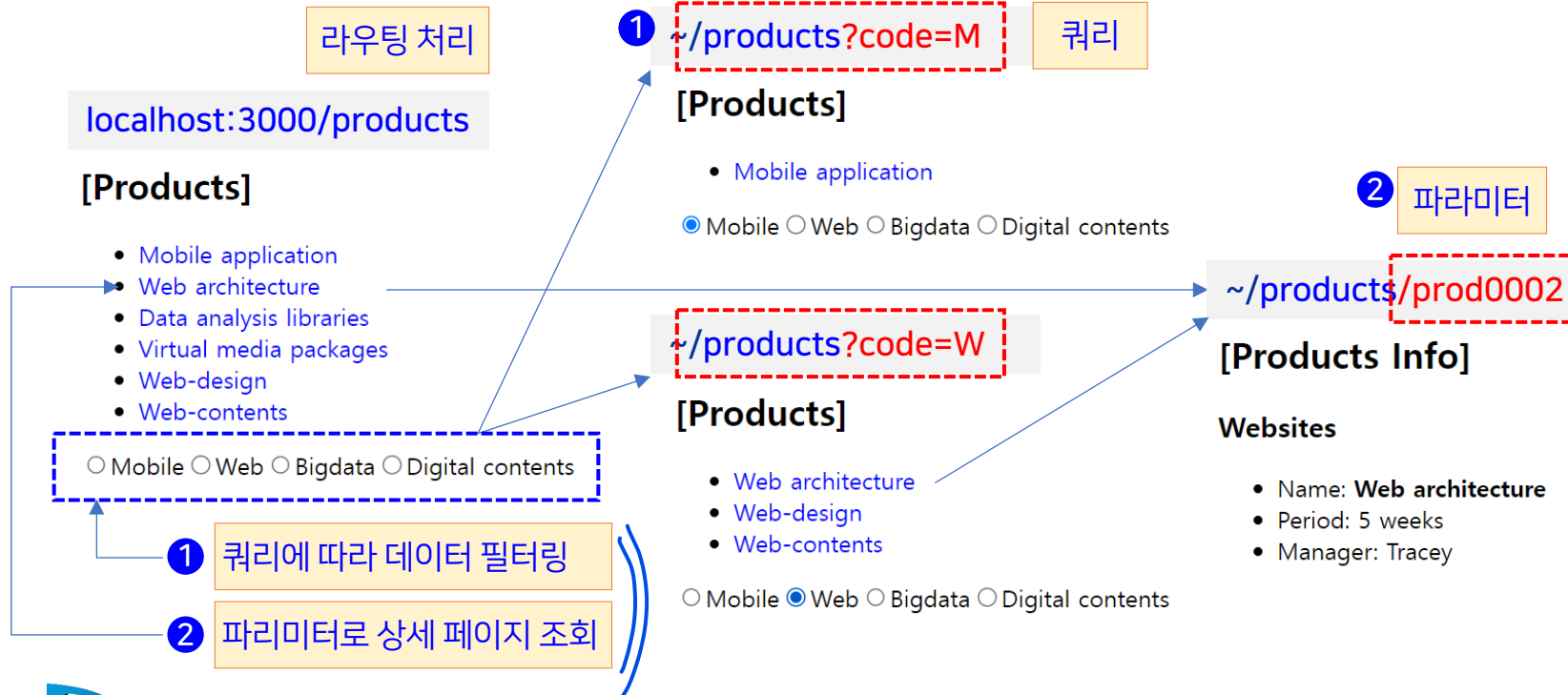
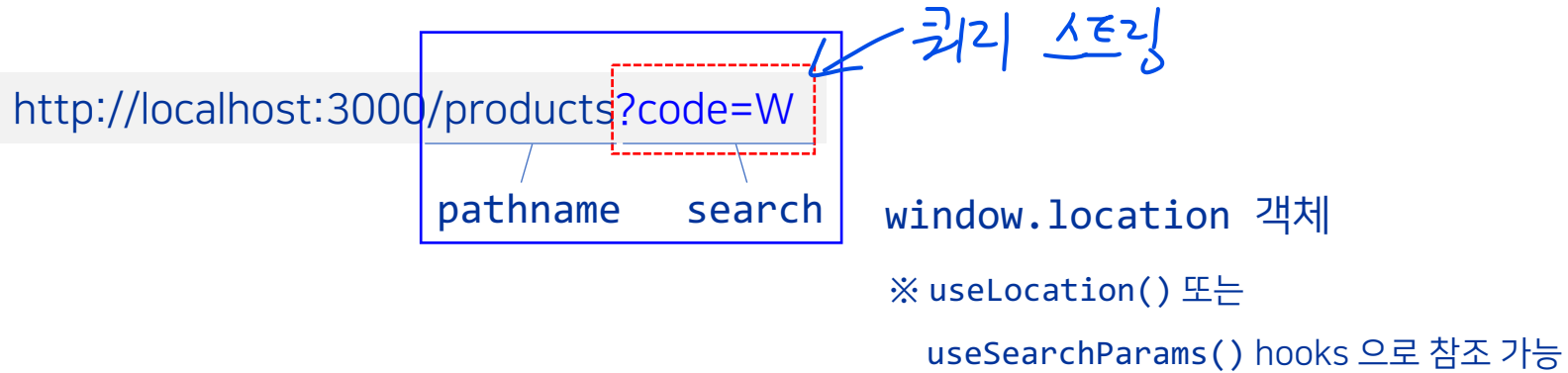
Mobile Software

{id: prod0001}

- Name: Mobile
- Period: 5 weeks
- Manager: Randy

동적 라우팅: 쿼리

<https://www.npmjs.com/package/qs> 



```
ch11/02-2/src/data/product-data.json
[
  {
    "Category": "Mobile Software",
    "Code": "M",
    "Id": "prod0001",
    "Name": "Mobile",
    "Period": "5 weeks",
    "Manager": "Randy"
  },
  {
    "Category": "Websites",
    "Code": "W",
    "Id": "prod0002",
    ...
  },
  { ..., "Code": "B", ... },
  { ..., "Code": "D", ... },
  { ..., "Code": "W", ... },
  { ..., "Code": "W", ... }
],
```


쿼리에 따라 데이터 필터링 <ProductList />

<https://reactrouter.com/en/main/hooks/use-navigate>
<https://reactrouter.com/en/main/hooks/use-search-params>

```
/* ch11/proj/02-2/src/components/Pages.js */
export const ProductList = () => {
  const [ code, setCode ] = useState();
  const navigate = useNavigate();
  const [ searchParams ] = useSearchParams();
```

```
  useEffect( () => {
    if (!code) return;
    3 navigate(`/products?code=${code}`);
  }, [code]);
```

```
4 const filteredData = searchParams.toString() == "" ? productData :
  productData.filter(data => data.Code == searchParams.get('code'));
```

```
2 const handleRadio = e => setCode(e.target.value);
```

```
5 return (
  <>
```

```
  <h1>[Products]</h1>
  <ul>{filteredData.map( (prd, i) => (<li key={i}><Link to={prd.Id}>{prd.Name}</Link></li>))}</ul>
  <div>
    <label><input type="radio" name="code" value="M" onChange={handleRadio} />Mobile</label>
    <label><input type="radio" name="code" value="W" onChange={handleRadio} />Web</label>
    <label><input type="radio" name="code" value="B" onChange={handleRadio} />Bigdata</label>
    <label><input type="radio" name="code" value="D" onChange={handleRadio} />Digital contents</label>
  </div>
  </> );
};
```

localhost:3000/products

[Products]

- Mobile application
- Web architecture ✓
- Data analysis libraries
- Virtual media packages
- Web-design ✓
- Web-contents ✓

☐ Mobile ☒ Web ☐ Bigdata ☐ Digital contents

*/products?code=W

[Products]

- Web architecture
- Web-design
- Web-contents

☐ Mobile ☒ Web ☐ Bigdata ☐ Digital contents

쿼리스트링 중,
key가 'code'인 값을 반환받아 data와 비교

```
productData → data = [
  { ..., "Code": "M", ... },
  { ..., "Code": "W", ... },
  { ..., "Code": "B", ... },
  { ..., "Code": "D", ... },
  { ..., "Code": "W", ... },
  { ..., "Code": "W", ... },
],
```

라디오 컨트롤이 선택되면 이벤트 처리

1

```
onChange={handleRadio} />Mobile</label>
onChange={handleRadio} />Web</label>
onChange={handleRadio} />Bigdata</label>
onChange={handleRadio} />Digital contents</label>
```

학습 정리: 11장. 리액트 라우터

- Single Page Application
 - Client-side rendering
- React Router
 - `<Routes />`, `<Route />`로 컴포넌트 전환 (주소 직접 입력)
 - `useLocation()` hooks로 주소정보 반환
 - `<Link />`로 라우터 링크 제공
 - 중첩 Route와 `<Outlet />`, `<Navigate />`
 - `useRoutes()` hooks
- 동적 라우팅
 - 라우팅 파라미터
 - 쿼리
- React Context (6장)