

# [React] url에 있는 파라미터 읽어오기

## ▼ Path Parameter (useParams)

url : `/book/view/1`

기존에 사용했던 코드

### App.js

```
<Route path="/book/view/:bookNo" element={<BookView />} />
```

`/book/view/bookNo`

`/book/view/1`

`/book/view/2`

과 같은 동적인 경로를 처리할 수 있는 방법

Route path = `"/book/view/*"` 로도 해봤지만

: 으로 해야 `/book/view`로 시작하는 모든 경로에 적용이 된다.

### list.js

```
<Link to={` /book/view/${v.bookNo}`} > </Link>
```

### view.js

```
const {bookNo} = useParams();  
console.log(bookNo) // 1
```

예를들어 현재 경로가 `/book/view/1` 일 때

`useParams` 혹은 사용하면 `1`이 `bookNo`에 담긴다.

## ▼ Query Parameter (useSearchParams)

url: /book/view?bookNo=3

### App.js

```
<Route path="/book/view" element={<BookView />} />
```

### list.js

흔히 알고 있는 쿼리 파라미터 경로로 링크 걸어주기

```
<Link to={` /book/view?bookNo=${v.bookNo}`} > </Link>
```

### view.js

```
const [searchParams, setSearchParams] = useSearchParams()  
const bookNo = searchParams.get("bookNo");  
console.log(bookNo); // 3
```

- list.js에서 넘겨받은 bookNo라는 파라미터의 값을 useSearchParams() 혹은 사용해 key에 따른 value 값을 읽어올 수 있다.
- searchParams.get() / searchParams.getAll() / searchParams.toString() / searchParams.set()

## 1. 값 읽어오는 메서드

- `searchParams.get(key)` : 특정 key의 value 가져오기
- `searchParams.getAll(key)` : 특정 key의 모든 value 가져오기

예시) `?sort=popular&sort=latest`의 경우  
`searchParams.get("sort")`의 리턴값: "popular"  
`searchParams.getAll("sort")`의 리턴값: ["popular","latest"]

- `searchParams.toString()`: 쿼리스트링을 string형태로 리턴

예시) `?sort=popular&sort=latest`의 경우  
`searchParams.toString()`의 리턴값: "?sort=popular&sort=latest"

## 2. 값을 변경하는 메서드

- `searchParams.set(key, value)`: 인자로 전달한 key값을 value로 설정. 만약 동일 key에 여러 value가 이미 존재하면 set 메서드를 호출하면서 설정한 값 외에는 삭제됨.

`?sort=popular&sort=latest` 의 경우  
`searchParams.set("sort", "clear")` 호출  
`searchParams.toString()`의 리턴값: "?sort=clear"

- `searchParams.append(key,value)`: 메서드를 호출하면서 인자로 전달한 key 값을 value로 추가. 기존 value를 건들지 않음.

`?sort=popular&sort=latest` 의 경우  
`searchParams.append("sort", "clear")` 호출  
`searchParams.toString()`의 리턴 값  
: "?sort=popular&sort=latest&sort=clear"