

# 체크리스트 관리용 웹 서비스 개발

개발자 안 승진 (2023-11-20)

# 목차

1. 주요 기능 및 기술(개발 환경)
2. 개발 일정
3. 주요 기능별 구현 사항
4. 주요 소스 트리

# 소스코드 : <https://github.com/sjahnsj/itemchecker>

# I. 주요 기능 및 기술(개발 환경)

## ❖ 주요 기능

- 체크리스트 작성 및 추가
- 체크리스트 검색
- 체크리스트 체크 및 표시
- 체크리스트 삭제

## ❖ 주요 기술 및 개발 환경

- React-JS
- Node-JS
- LocalStorage
- HTML/CSS

## 2. 개발 일정

- 개발 환경 구축 및 체크 리스트 컴포넌트 구현 : 2023-11-18
- 체크리스트 작성 및 검색 컴포넌트 구현 : 2023-11-19
- 웹 브라우저 상의 Local Storage CRUD 구현 : 2023-11-20

### 3. 주요 기능 별 구현 사항

✍ 체크 리스트 3개

리스트 작성

+

검색

☐ 개발 일정 점검

🗑

☐ 이슈 정리하기

🗑

☒ 스프링 시큐리티 보완

🗑

Copyright © 2023 by SJ

- 할일, 쇼핑 리스트 등 점검 및 관리가 필요한 항목들을 체크리스트로 관리하는 기능 제공
- 타이틀, 아이템 추가, 검색, 체크리스트 보기 및 삭제, 저작권 및 개발자 표시 기능을 각각 담당하는 5개의 React component로 구성하여 개발

## 3.1 REACT COMPONENT 구성

```
<div className="App">
  <Header title="체크 리스트" length={items.length} />
  <AddItem
    newItem={newItem}
    setNewItem={setNewItem}
    items={items}
    setItems={setItems}
  />
  <SearchItem search={search} setSearch={setSearch} />
  <Content
    items={items.filter((item) => item.item.includes(search))}
    setItems={setItems}
  />
  <Footer />
</div>
```

- <Header />  
타이틀 및 체크리스트 갯수 표시
- <AddItem />  
체크리스트 작성 및 추가
- <SearchItem />  
검색 입력
- <Content />  
체크리스트 내용 및 체크 표시 및 삭제
- <Footer />  
Copyright 및 개발자 표시

## 3.2 체크 리스트 저장소 및 USESTATE

```
import { useState } from "react";
```

```
const [items, setItems] = useState(  
  localStorage.getItem("checklist")  
  ? JSON.parse(localStorage.getItem("checklist"))  
  : []  
);
```

- 주요 데이터와 연동하여 화면 업데이트를 위한 useState를 준비
- 웹 브라우저 상의 local storage로 생성된 “checklist”로 부터 데이터를 가져와서 item list를 초기화

### 3.3 체크리스트 아이템 추가

```
const handleSubmit = (e) => {  
  e.preventDefault();  
  if (!newItem) return;  
  const id = items.length ? items[items.length - 1].id + 1 : 1;  
  const myNewItem = { id, checked: false, item: newItem };  
  const listItems = [...items, myNewItem];  
  setItems(listItems);  
  localStorage.setItem("checklist", JSON.stringify(listItems));  
  setNewItem("");  
};
```

- AddItem component에서 아이템 작성 후 호출하는 함수
- 사용자로부터 입력 받은 아이템 내용을 기존 체크리스트에 신규 아이템으로 추가 후 localStorage에 저장
- useState에 등록된 setItems를 통해 체크리스트 화면 업데이트



## 3.4 체크리스트 아이템 체크 및 삭제

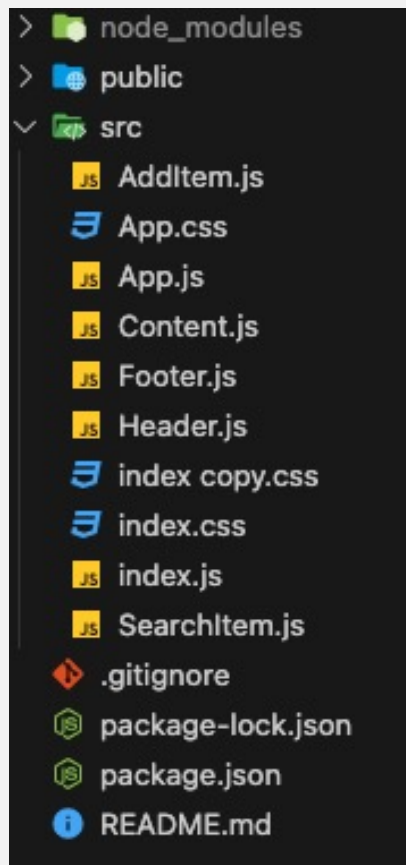
```
const handleCheck = (id) => {  
  const listItems = items.map((item) =>  
    item.id === id ? { ...item, checked: !item.checked } : item  
  );  
  setItems(listItems);  
  localStorage.setItem("checklist", JSON.stringify(listItems));  
};
```

- 체크리스트를 보여주는 Content component에서 각 아이템의 체크박스 설정 시 호출되는 함수
- Map 함수를 이용, 리스트의 아이템들 중 해당 아이템에 대해서 토글 설정 및 localStorage 저장

```
const handleDelete = (id) => {  
  const listItems = items.filter((item) => item.id !== id);  
  setItems(listItems);  
  localStorage.setItem("checklist", JSON.stringify(listItems));  
};
```

- 체크리스트를 보여주는 Content component에서 각 아이템의 삭제 버튼 클릭 시 호출되는 함수
- Filter 함수를 이용, 리스트의 아이템들 중 해당 아이템에 삭제 후 localStorage에 체크리스트 저장

## 4. 주요 소스 트리



- Header.js  
<Header />
- AddItem.js  
<AddItem />
- SearchItem.js  
<SearchItem />
- Content.js  
<Content />
- Footer.js  
<Footer />
- Package.json  
앱에서 다양한 아이콘 사용을 위해 dependencies에  
"react\_icons" 모듈을 인스톨하여 추가