To DO! PWA 개발

계획서 및 분석보고서

2018.09.20 오픈소스프로젝트 | 20160528 박정현

20140915 이상협

# **목차**

1. 개요
2. Challenges

* 중요도 설정 기능
* 마감 임박 알림 기능
* Top/Bottom 기능
* 검색 기능

1. 개발 일정
2. ToDo App의 구조 분석

# **개요**

Vue를 사용하여Progressive web app 기반인 To Do App을 구현한다. 기본적으로 제공되는 Web App의 소스코드에는 추가/삭제 기능만이 들어있다. 우리는 누구나 쉽고 간편하게 사용할 수 있도록 다양한 추가 기능을 구현하여 사용자에게 제공한다. 기능을 추가하기 위해 소스 코드를 분석 및 수정하고, 오픈소스를 참고하여 개발의 완성도를 높인다. 기본 개발 기능으로는 완료와 수정 및 상세정보 입력이 있고 추가 개발 기능으로는, 중요도 설정 기능, 마감 임박 알림 기능, Top/Bottom 기능, 검색 기능이 있다.

# **Challenges**

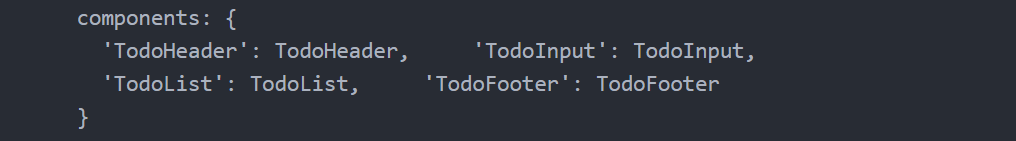
* 중요도 설정 기능
* 사용자가 To Do List를 작성하고 상세정보를 입력할 때 일의 우선순위를 설정한다. 우선순위에는 보통, 중요, 긴급이 있다.
* 상세 정보 입력 부분에 중요도를 설정하기 위해Radio 버튼을 구현한다. 이때 선택한 항목에 따라 빨간색 – 긴급, 노란색 – 중요, 색없음 – 보통 으로 Background를 지정한다.
* 마감 임박 알림 기능
* 사용자가 To Do List를 작성하고 상세정보를 입력할 때 마감 일정을 설정한다. 해당 날짜가 임박하면 사용자에게 알림을 보내 일정을 잊지 않게 도와준다.
* 사용자에게 알림을 보여 줄 때는 SnackBar/Toast 기능을 사용한다. 웹앱을 실행했을 때, 화면에 알림이 뜨도록 설정하며, 화면에 보이는 시간은 너무 짧지 않도록 설정한다. 알림은 일정 마감이 D-2가 되었을 때부터 노출되도록 한다. 날짜는 Javascript를 사용하여 계산하도록 한다.
* Top/Bottom 기능
* List가 액티비티에서 벗어날 경우 빠르게 상단, 하단으로 이동할 수 있도록 한다.
* 사용자가 처음 액티비티에 있을 경우 Bottom 버튼으로 최하단으로 이동할 수 있고, 최하단의 액티비티에 있을 경우 Top 버튼으로 최상단으로 이동할 수 있다. 만약 액티비티가 최상단, 최하단이 아닐 경우 각각의 버튼이 활성화되어 이동할 수 있도록 한다.
* 검색 기능
* 사용자가 원하는 일정을 찾을 수 있도록 한다.
* 사용자가 검색을 위해 단어를 입력하면 vue의 v-if와 v-for, javascript의 indexOf 기능을 적절히 이용하여 검색한 단어가 포함된 일정만 display 하도록 한다.

# **개발일정**

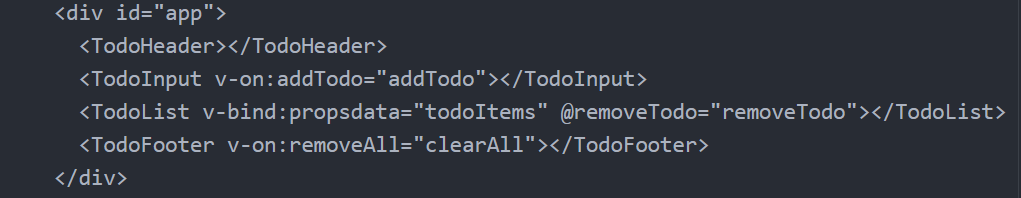
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| (수행 기간 : 2018년 9 월 17 일 ~ 2018년 10 월 17 일) | | | | | |
| 구분  과제내용 | 1주 | 2주 | 3주 | 4주 | 5주 |
| 계획서 및 보고서 작성 |  |  |  |  |  |
| 기본 기능 구현 |  |  |  |  |  |
| 중요도 설정 기능 구현 |  |  |  |  |  |
| 마감 알림 기능 구현 |  |  |  |  |  |
| Top/Bottom 기능 구현 |  |  |  |  |  |
| 검색 기능 구현 |  |  |  |  |  |
| 최종 발표 자료 작성 |  |  |  |  |  |

# **ToDo App의 구조 분석**

기본 제공된 ToDo App은 총 6개의 \*.vue 파일을 사용하고 있다. 각각의 vue 파일은 담당하는 영역에 따라 분할되어 있으며, 그 중 App.vue는 상위 컴포넌트의 역할을 하고 있다. App.vue에서는 TodoHeader, TodoInput, TodoList, TodoFooter로 4개의 하위 컴포넌트를 선언하여 사용한다.

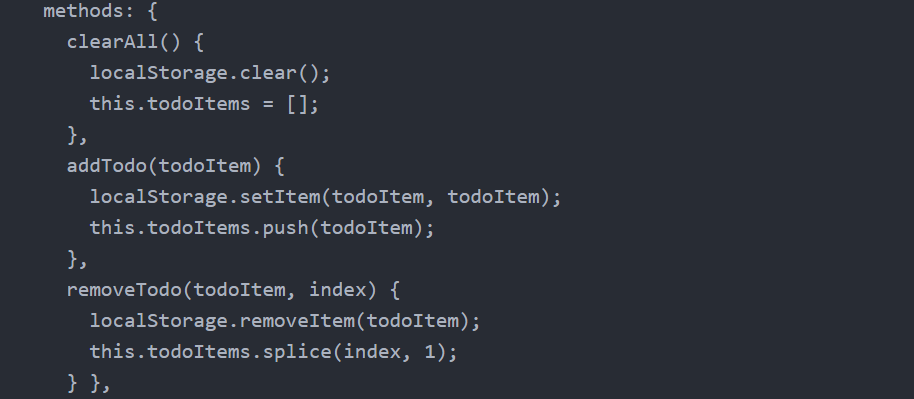


List는 상위 컴포넌트의 데이터에 props를 동적으로 바인딩 해주었다. ToDo App에서는 상위와 하위 컴포넌트 간의 데이터 전달이 원활해야 한다. TodoList에서는 todoItems의 데이터를 화면에 출력하여 보여주는 역할을 하므로 todoItems의 데이터가 갱신 될 때마다 새로운 데이터를 화면에 출력 해줘야 한다. 따라서 props를 동적으로 바인딩해주어 상위 컴포넌트에서 데이터가 업데이트될 때마다 연결된 하위 컴포넌트에 데이터가 전달되게 처리한다. Javascript에서는 이벤트 핸들러가 많아질수록 v-on을 속성값으로 보관하는 것이 간단하지 않기 때문에 각각의 v-on이 호출하고자 하는 메소드의 이름을 적어놓았다.



vue에서 data는 컴포넌트 인스턴스 함수여야 한다. 따라서 함수 형식으로 관리하는데, 데이터 객체를 반환시켜 각각의 컴포넌트에 값이 반환된다.





App.vue에서 사용되는 method는 총 3가지가 있다. Todo App에 가장 기본이 되는 addTodo, removeTodo, clearAll이고, 각각의 함수는 사용자의 데이터를 저장하는 localStorage와 연결되어있다.

TodoApp은 브라우저의 localStorage를 이용해 사용자의 데이터를 관리한다. 사용자가 list를 추가할 때

TodoInput.vue에서 App.vue에게 addTodo이벤트를 요청한다. addTodo함수에서는 받은 TodoItem을 localStorage에 setItem을 이용하여 저장한 뒤 push를 이용해 todoItems에 넣는다. TodoItems는 TodoList의 propsdata와 바인딩 되어있으므로 TodoItems가 업데이트 됨에 따라 List컴포넌트도 같이 업데이트 된다.

🡪 기본 제공된 ToDo App의 구조 분석 기술. 아래 내용을 포함할 것.

* 각 컴포넌트간의 용도 및 각 컴포넌트간의 관계
* 컴포넌트간의 메시지 교환 방법 및 목적
* 할 일 데이터의 저장 및 조회 방법

컴포넌트 : 화면에 비춰지는 vue의 단위를 쪼개어 재활용이 가능한 형태로 관리하는 것

부모->자식 : props down

자식->부모 : events up

상위 컴포넌트에 message를 바인딩해서 전달

하위에서 data의 message를 passedData에 전달 받음

컴포넌트간의 직접적인 통신은 불가능

Ch->pa->ch,ch로

List는 할 일을 보여주는 list에 관련된 이벤트를 담당함.

Header는 어플의 이름 header를 담당

Footer는 어플 중에 clearAll() 부분 담당

Input은 입력창 담당.

여기서 부모 컴포넌트는 app 자식 컴포넌트 (List,Header, Footer, Input)

props는 부모가 자식컴퍼넌트에게 data를 전달할 때 사용

먼저 자식컴퍼넌트는 이벤트리스너를 첨부해 인자로 전달할 값을 담아 보내고 부모컴퍼넌트는 이를 받아서 처리하는 구조이다.

할 일을 작성하여 추가 버튼을 누를 경우 App.vue / method의 addTodo가 작동된다. setItem을 통해 LocalStorage에 key와 value가 저장되며 todoItem에 push된다.

http://takeuu.tistory.com/35

https://blog.naver.com/wj8606/221238811743

<https://blog.naver.com/wj8606/221231688614>

<https://joshua1988.github.io/web-development/translation/essential-es6-features-for-vuejs/>

<https://kr.vuejs.org/v2/guide/syntax.html>

https://kr.vuejs.org/v2/guide/list.html