To DO! PWA 개발

계획서 및 분석보고서

2018.09.21 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**

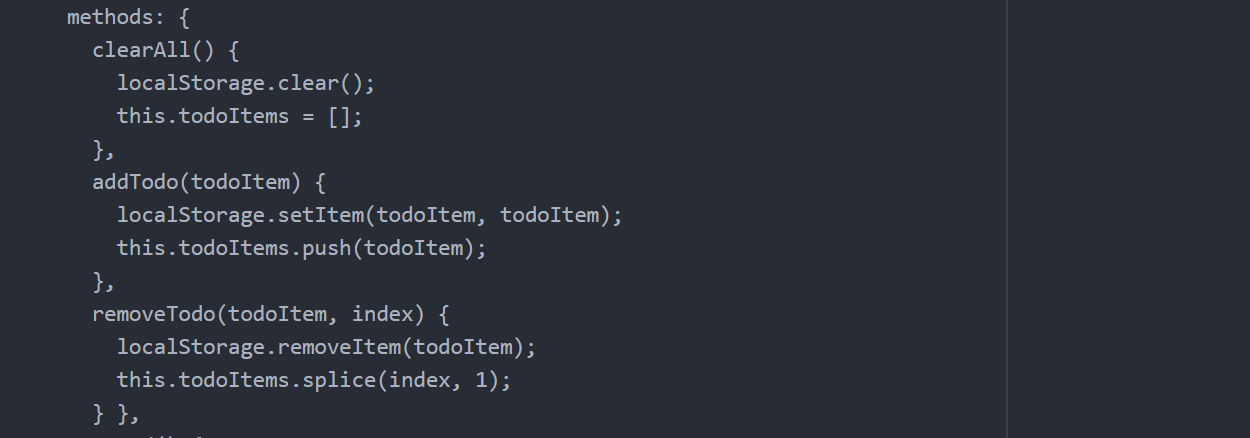
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| (수행 기간 : 2018년 9 월 17 일 ~ 2018년 10 월 17 일) | | | | | |
| 구분  과제내용 | 1주 | 2주 | 3주 | 4주 | 5주 |
| 계획서 및 보고서 작성 |  |  |  |  |  |
| 기본 기능 구현 |  |  |  |  |  |
| 중요도 설정 기능 구현 |  |  |  |  |  |
| 마감 알림 기능 구현 |  |  |  |  |  |
| Top/Bottom 기능 구현 |  | 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

# **ToDo App의 구조 분석**

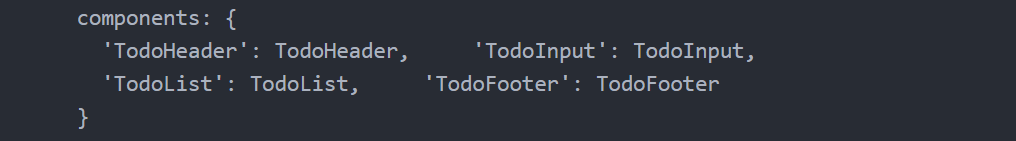
기본 제공된 ToDo App은 총 6개의 \*.vue 파일을 사용하고 있으며 각각의 vue 파일은 담당하는 영역에 따라 분할되어 있다. TodoHeader는 ToDo App의 로고 정보를 담고 있으며 TodoInput은 할 일을 List에 추가하는 부분을 담당한다. 추가된 할 일은 TodoList를 통해 화면에 출력되어 사용자에게 가시적으로 노출된다. TodoFooter는 ToDo App의 하단을 담당하고 있으며 사용자가 쉽게 List를 초기화 할 수 있는 버튼을 제공한다.



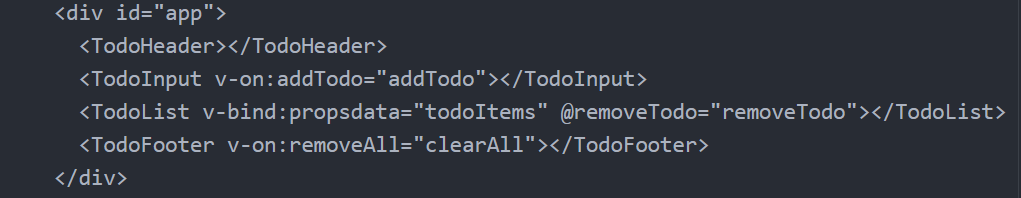
Vue인스턴스는 data객체에 있는 모든 속성을 프록시 처리하여 값이 변경되면 화면이 다시 렌더링 된다. data는 컴포넌트 인스턴스 함수의 형식을 따른다. 데이터 값을 반환 할 경우 새로운 데이터 객체를 반환시켜 각각의 컴포넌트에 값이 반환되도록 한다.



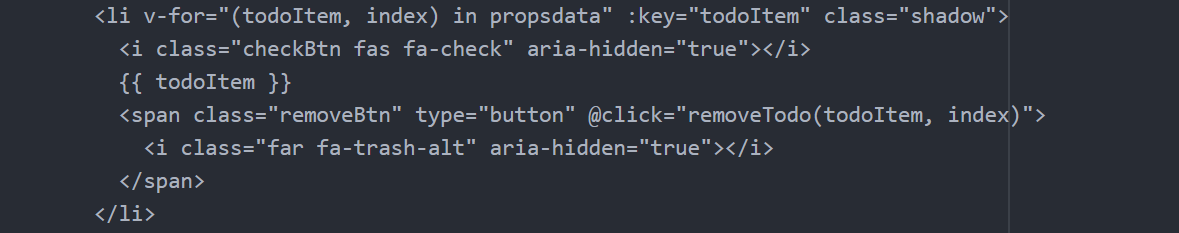
App.vue에서 사용되는 메소드는 Todo App에 가장 기본이 되는 addTodo와 removeTodo, clearAll 총 3가지가 있다. 각각의 함수는 사용자의 데이터 저장/삭제와 관련이 있으며, 브라우저의 localStorage를 이용해 사용자의 데이터를 관리한다. App을 실행하면 created가 실행된다. created는 localStorage.length를 사용하여 localStorage에 저장된 key값의 개수를 받는다. for문을 통해 찾은 키 값을 todoItems에 푸시하여 localStorage에 저장된 데이터를 App으로 가져온다. 사용자가 데이터를 추가하면 addTodo 함수를 통해 localStorage에 키가 저장되고, 데이터를 삭제하면 removeTodo 함수를 통해 키가 삭제된다.



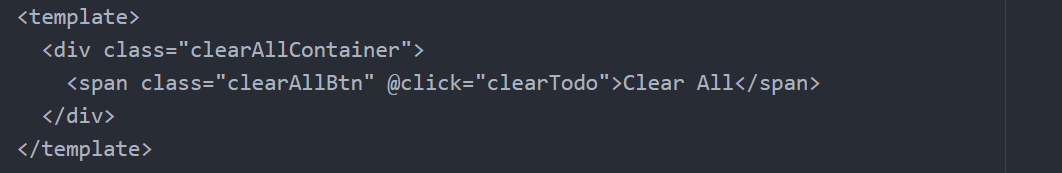
Vue에서 상위 컴포넌트는 하위컴포넌트에게 메시지를 전달하며, 하위 컴포넌트는 상위 컴포넌트에게 이벤트를 요청한다. TodoApp에서는 App.vue가 상위 컴포넌트에 해당되고, 그 외 4개의 컴포넌트인 TodoHeader, TodoInput, TodoList, TodoFooter는 하위 컴포넌트로 선언하여 사용한다.



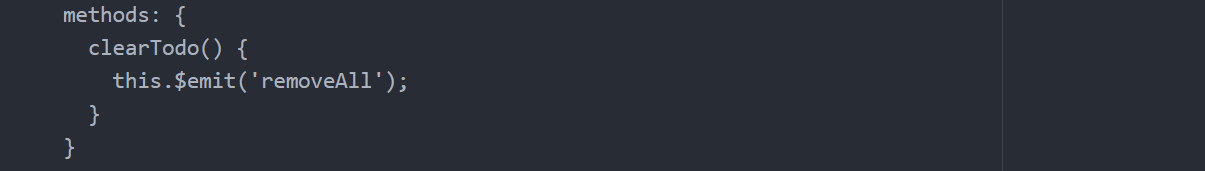
ToDoList는 상위 컴포넌트의 데이터에 props를 동적으로 바인드 되어있다. ToDo App에서는 상위와 하위 컴포넌트 간의 데이터 전달이 원활해야 한다. TodoList에서는 todoItems의 데이터를 화면에 출력하여 보여주는 역할을 하므로 todoItems의 데이터가 갱신 될 때마다 새로운 데이터를 화면에 출력 해줘야 한다. 따라서 props를 동적으로 바인드하여 상위 컴포넌트에서 데이터가 업데이트 될 때마다 연결된 하위 컴포넌트에 데이터가 전달되게 처리한다. Javascript에서는 이벤트 핸들러가 많아질수록 v-on을 속성값으로 보관하는 것이 간단하지 않기 때문에 각각의 v-on이 호출하고자 하는 메소드의 이름을 적어놓았다. todoItems는 TodoList의 propsdata와 바인딩 되어있으므로 todoItems가 업데이트됨에 따라 TodoList 컴포넌트도 같이 업데이트된다.



TodoList.vue에서는 App.vue에서 propsdata 를 수신한다. propsdata에는 todoItem와 index가 포함되어있고 v-for반복문을 통해 data를 화면에 노출시킨다. 이때 removeBtn을 클릭하면 App.vue에게 click 이벤트가 요청된다. removeTodo에서는 localStorage와 todoItems에 들어간 데이터를 제거한다. 이때 todoItems가 업데이트되어 다시 TodoList 컴포넌트가 출력된다.



TodoFooter.vue에서는 사용자가 clearAllBtn을 클릭 할 때 clearTodo메소드를 실행하며, 상위 컴포넌트에게 이벤트를 요청한다. 상위 컴포넌트인 App.vue에서는 removeAll리스너를 통해 clearAll을 실행한다.





clearAll 메소드는 localStorage에 저장된 data를 모두 지우고 todoItems를 초기화시키며, 데이터가 다시 업데이트되어 초기화된 List를 재출력한다.