

개요

1. npm 트렌트 사이트 : <https://npm trends.com/react-beautiful-dnd-vs-react-dnd-vs-react-draggable>

react-beautiful-dnd vs react-dnd vs react-draggable

Google Translate

Enter an npm package...

react-beautiful-dnd x

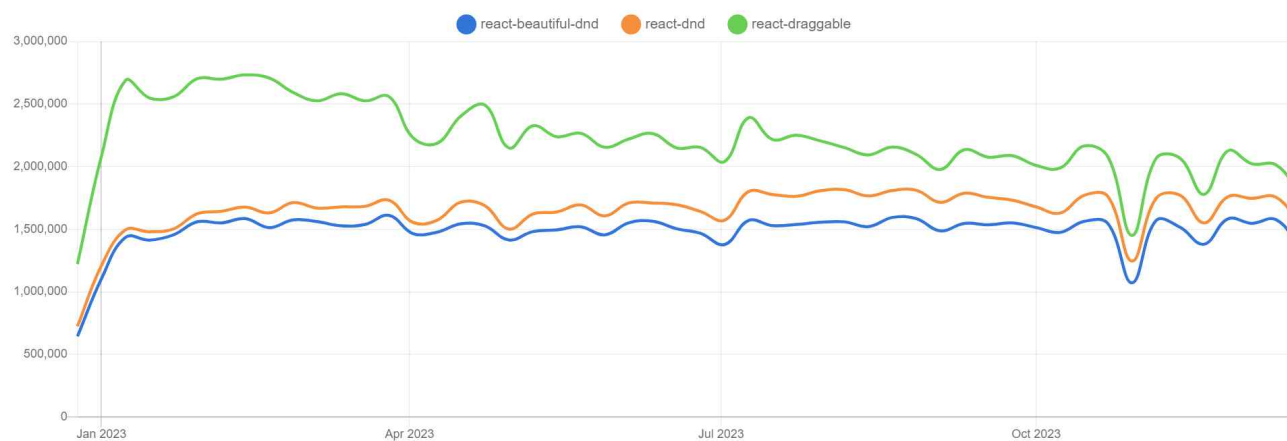
react-dnd x

react-draggable x

+ react-drag-and-drop

+ react-file-drop

Downloads in past 1 Year ▾



참고 :

<https://velog.io/@dowon938/react-dnd-%EC%9D%B4%EC%9A%A9%ED%95%98%EC%97%AC-drag-drop-%EA%B5%AC%ED%98%84%ED%95%98%EA%B8%B0>

1. react-draggable

react-draggable 라이브러리

<https://www.npmjs.com/package/react-draggable>

<https://codesandbox.io/p/sandbox/drag-and-drop-25j99?file=%2Fsrc%2FApp.js%3A60%2C26>

1. <https://7357.tistory.com/45>

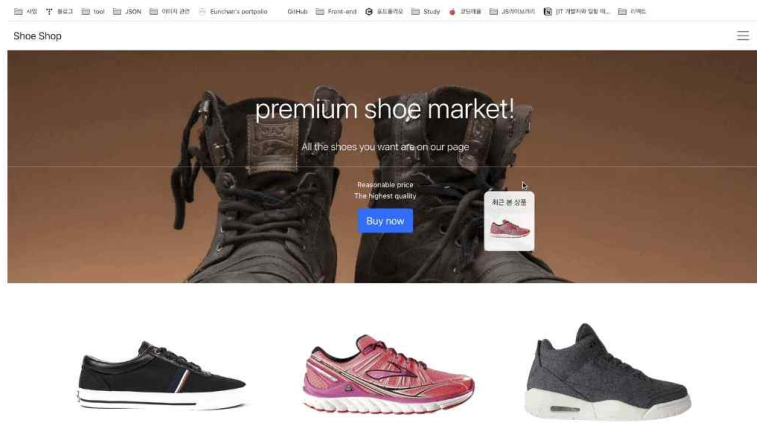
드래그 좌표 값을 trackPos가 가져간다.

```
const trackPos = (data) => {  
  setSidePosition({ x: data.x, y: data.y });  
};
```

해당 좌표값으로 Draggable을 이동시키는 원리.

이해를 위해 밑에 작성했을 뿐 useState랑 나란히 두면 됩니다.

5. 결과물을 감상한다.



사람들이 왜 라이브러리의 노예가 되는지 알 것 같다.
너무 편하다..

가. 활용

1) <https://wazacs.tistory.com/37>

2. 불러오기 및 적용

```
import Draggable from "react-draggable";

export default function App() {
  const [position, setPosition] = useState({ x: 0, y: 0 }); // box의 포지션 값
  // 업데이트 되는 값을 set 해줌
  const trackPos = (data) => {
    setPosition({ x: data.x, y: data.y });
  };
  return (
    <div className="App">
      <Draggable onDrag={(e, data) => trackPos(data)} >
        <div className="box" >
          <div>BOX</div>
          <div>x: {position.x.toFixed(0)}, y: {position.y.toFixed(0)}</div>
        </div>
      </Draggable>
    </div>
  );
}
```

```
// css
.box {
  position: absolute;
  cursor: move;
  color: black;
  width: 100px;
  border-radius: 5px;
  padding: 1em;
  margin: auto;
  user-select: none;
  background: lightgrey;
}
```

Draggable로 감싼 뒤 필요한 옵션과 함수를 props에 전달합니다.
x, y 좌표를 toFixed(0) 으로 소수점을 제외한 값을 표시합니다.
참고로 반드시 css에서 absolute를 줘야 작동이 되는것은 아닙니다.
onDrag 이외에 다양한 옵션값이 있습니다. (상단 react-draggable 사이트에서 확인 가능)

2) <https://stackoverflow.com/questions/63516368/is-the-react-draggable-not-support-on-mobile-platform>

모바일 적용

Is the react-draggable not support on mobile platform?

Asked 3 years, 4 months ago Modified 7 months ago Viewed 3k times

- ▲ My code is [here](#).
- 3 The p-in-p button of the MediaPlayer triggers the MiniPlayer component.
- ▼ And the react-draggable package is used in MiniPlayer component.
- 🔖 The top-left button works on the computer; however, it does not work in the mobile device.
- 🕒 When I remove the <Draggable> tag from the MiniPlayer.js, the top-left button works on mobile again.
- How can I fix the problem?

reactjs

Share Improve this question Follow

asked Aug 21, 2020 at 4:04
The KNVB
3,607 3 30 56

Add a comment

Answers

Sorted by: Highest score (default)

react-draggable works on mobiles. It seems onClick or onChange inside <Draggable> are disabled on mobiles (or on touch devices in general).

To make elements inside the drag component clickable, set as `cancel` the selector of those elements.

2. <https://velog.io/@gnoeyah/react-draggable-DragDrop-%EA%B5%AC%ED%98%84>

2. 모바일에서 클릭 이벤트가 먹히지 않는 이슈

같은 코드상에서 모바일로 실행을 했을 때는 드래그는 되지만 클릭이 되질 않았다.

즉 내가 고려해야하는 문제는 적절한 이벤트 충돌 방지 처리와 모바일에서 onTouch이벤트를 적용하는 것이었다.

🔗 해결

1. <Draggable> 에서 제공하는 API `onStop` 과 `useState`로 `flag`변수를 선언하여 드래그 이벤트를 제어해준다.

- `onStop: DraggableEventHandler` 드래그가 중지되면 트리거하는 이벤트 리스너이다. 드래그가 끝날 때의 이벤트 핸들러를 전달해준다.
- `useState` 로 드래그를 하는 동안에 `true`, 드래그가 끝날 때 `false`를 나타내는 `boolean` type의 `flag`변수를 지정해준다.
- 💡(핵심 포인트) `onStop`이벤트 핸들러에서 `flag`변수를 0.1초 뒤에 업데이트 시켜주는 이유
 - `setTimeout`함수를 사용하여 비동기적으로 `isDragging`변수를 업데이트 하면, 드래그가 완전히 종료된 후에 업데이트가 이루어진다. 따라서 정확한 드래그 상태를 파악할 수 있어 드래그 동작과 클릭 이벤트의 충돌을 방지하고 순차적으로 드래그 이벤트와 클릭 이벤트가 동작할 수 있는 것이다.

2. 모바일일 때에도 `Touch`가 가능하게 `onTouchEnd` 이벤트 리스너를 넣어주어 해결한다.

- `onTouchEnd` 화면에서 손을 뗄 때 발생하는 이벤트를 등록한다. 기존의 `onClick` 으로 등록한 이벤트 핸들러와 같은 이벤트를 등록해주었다.

```
import Draggable, { DraggableData, DraggableEvent } from 'react-draggable';
import { useEffect, useState } from 'react';
```

```
...
```

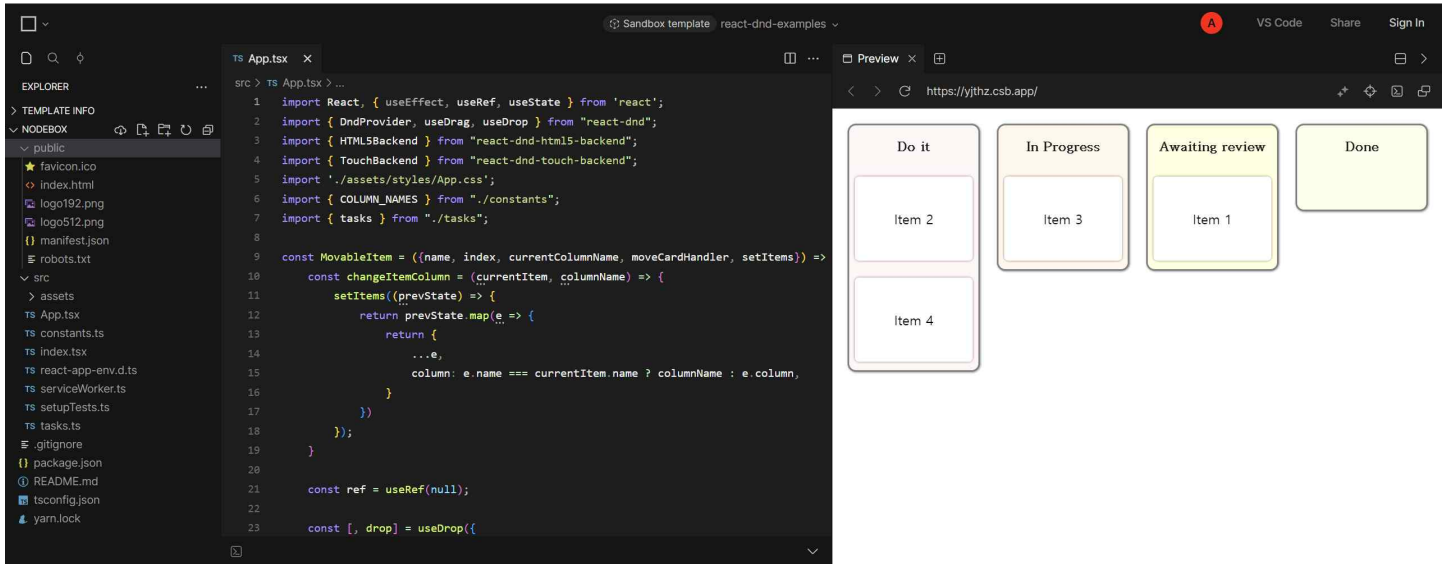
```
function FloatingButton() {
```

2. react DND

1.

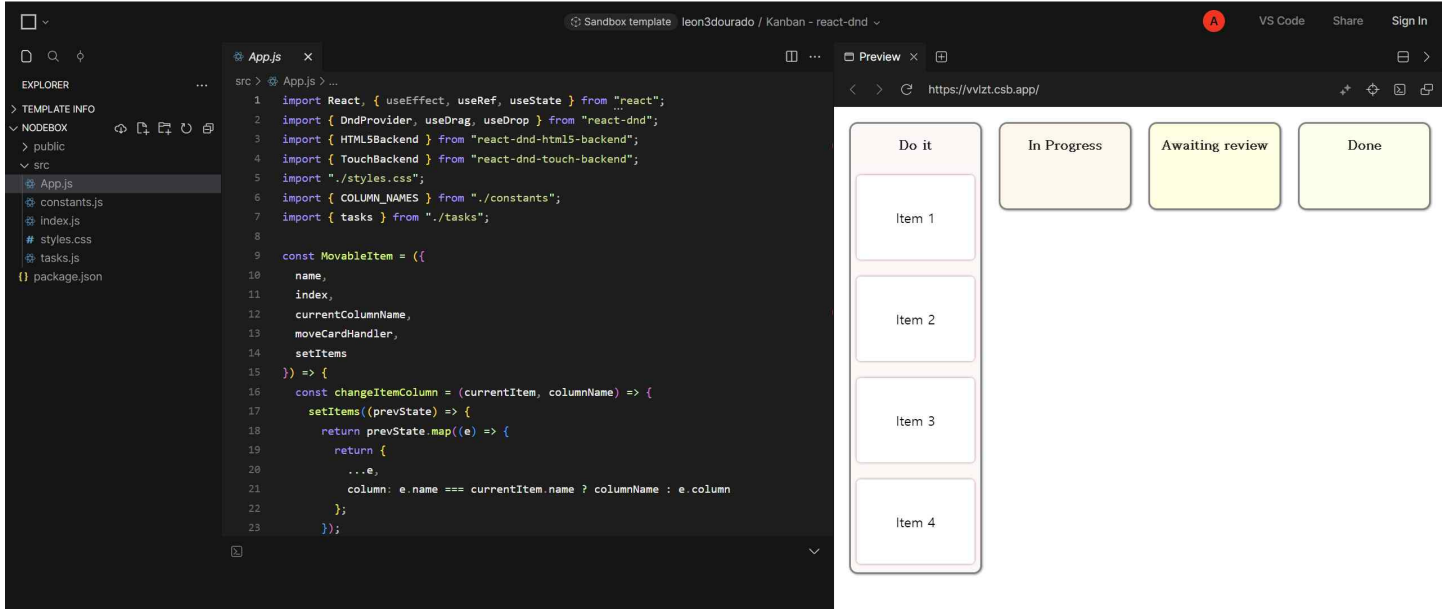
ts 버전

<https://codesandbox.io/p/sandbox/react-dnd-examples-yithz?file=%2Fsrc%2FApp.tsx>



js 버전

<https://codesandbox.io/p/sandbox/kanban-react-dnd-vvlzt?file=%2Fsrc%2FApp.js>



그 외 코드

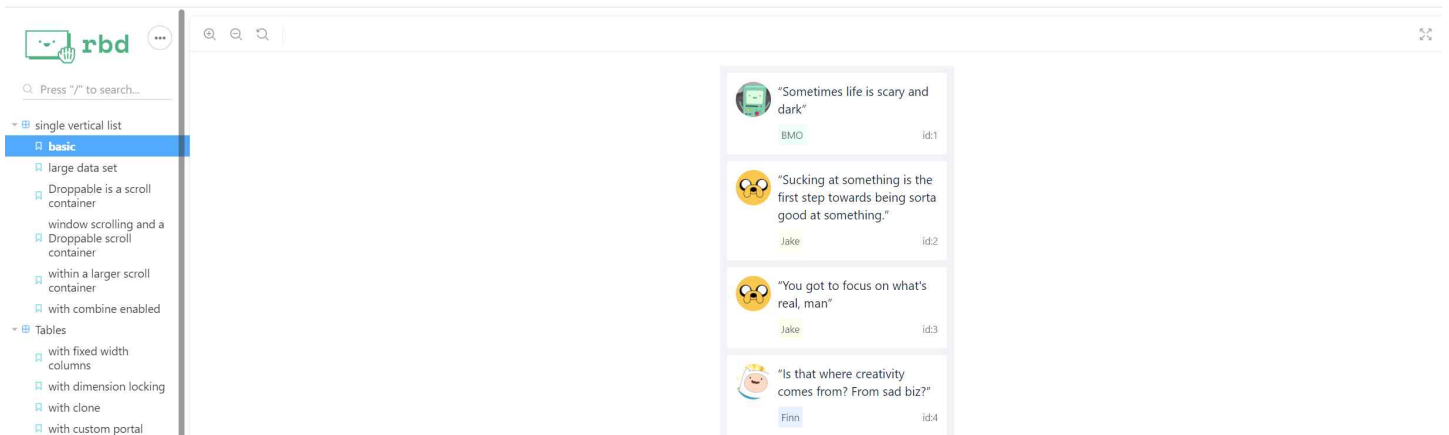
<https://codesandbox.io/examples/package/react-dnd-touch-backend>

3. react-beautiful-dnd

<https://github.com/atlassian/react-beautiful-dnd?tab=readme-ov-file>

<https://github.com/LeeHyungGeun/react-beautiful-dnd-kr> (한국어)

1. <https://react-beautiful-dnd.netlify.app/?path=/story/single-vertical-list--basic>



Core characteristics(핵심 특징):

- 아름답고 자연스러운 아이템 이동
- 깔끔하고 강력한 API 를 통한 간단한 사용
- 독착정인 스타일
- 추가 DOM 생성이 필요 없음 - 친숙한 flexbox 와 focus 관리
- anchor 태그 등 기존 노드와 잘 작동
- 상태(state) 주도의 드래깅 - 이 부분은 프로그래밍(programatic) 방식으로 많은 input 타입에서 드래깅(dragging)을 허용 한다. 현재는 마우스 와 키보드 드래깅(dragging)만 지원합니다.



Not for everyone(참고사항)

React를 사용한 많은 드래그 앤 드랍 라이브러리가 있습니다. 그중 가장 주목할 것은 [react-dnd](#) 입니다. 그것은 HTML5 드래그 앤 드랍 feature([wildly inconsistent](#))과 훌륭하게 매치되는 드래그 앤 드랍을 지원 합니다. [react-beautiful-dnd](#) 는 vertical 과 horizontal 리스트를 위해 특별히 high level로 추상화 되어있습니다. [react-beautiful-dnd](#) 는 기능적인 subset으로 파워풀하고 자연스럽고 아름다운 드래그 앤 드랍을 제안합니다. 그러나, 그것은 [react-dnd](#) 처럼 많은 기능들을 제공하지 않습니다. 그래서 이 라이브러리가 당신의 경우에 맞지 않을 수 있습니다.

Still young!(아직 젊습니다!)

이 라이브러리는 새것이기 때문에 상대적으로 작은 기능만 제공 합니다. 기다려 주세요! 빠르게 움직일 것입니다!

Currently supported feature set(현재 지원하는 특징)

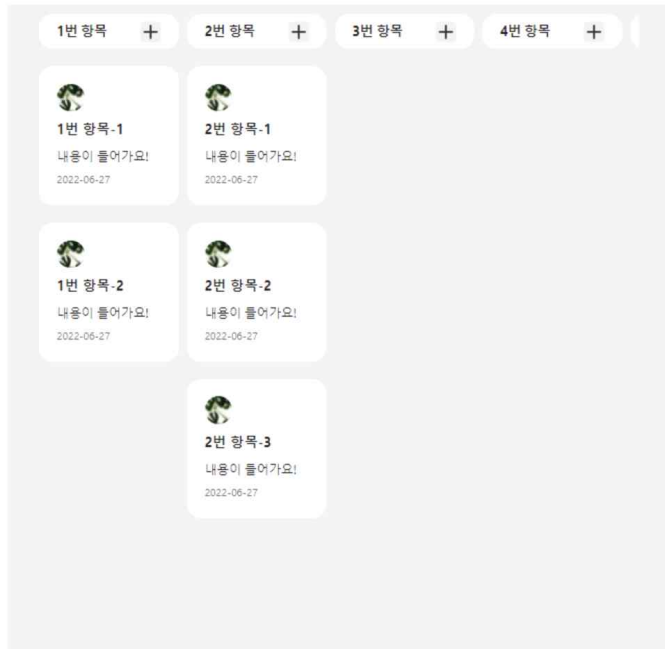
- vertical 리스트
- horizontal 리스트
- 한 페이지 안의 멀티 독립 리스트
- 마우스  와 키보드  드래깅
- 독립 중첩 리스트(리스트는 다른 리스트의 자식이 될 수 있습니다. 그러나 당신은 부모 리스트의 아이템을 자식 리스트로 드래그 할 수 없습니다.)
- 가변 높이 아이템 (아이템은 서로 다른 높이를 가질수 있다.)
- 커스텀 드래그 핸들 (일부 아이템만 드래그 할 수 있다.)
- vertical 리스트는 scoll container가 될 수 있다. (스크롤 부모 컨테이너 없이) 혹은 자식이 scoll container가 될 수 있다. (역시 스크롤 가능한 부모 없다)

2. <https://velog.io/@pbs1014/React-react-beautiful-dnd>

`source(...)` 는 시작지점을 의미하며

`droppableId`는 요소가 시작할 때의 영역에 대한 지정한 고유값이고,

`index`는 해당영역에서 Item이 영역의 Item들 중 몇 번째 에서 출발했는지를 나타내준다.



비교 및 결론

<https://seniyas.medium.com/react-drag-and-drop-libraries-what-should-we-choose-459ebd79ce01>

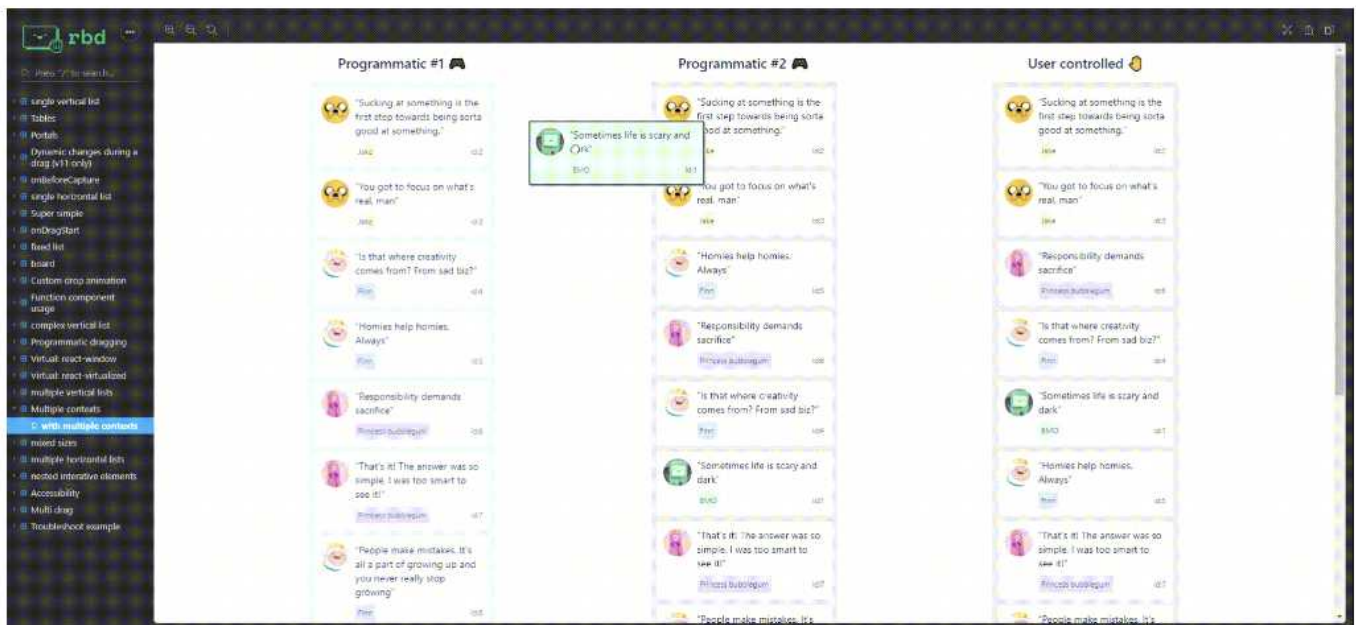
1. react-beautiful-dnd

(3-1. 링크 참조)

스토리북 — <https://react-beautiful-dnd.netlify.app/?path=/story/single-vertical-list--basic>

지원되는 입력 방법 — 마우스, 키보드, 터치

적합 — 단방향 드래그 가능한 구성 요소



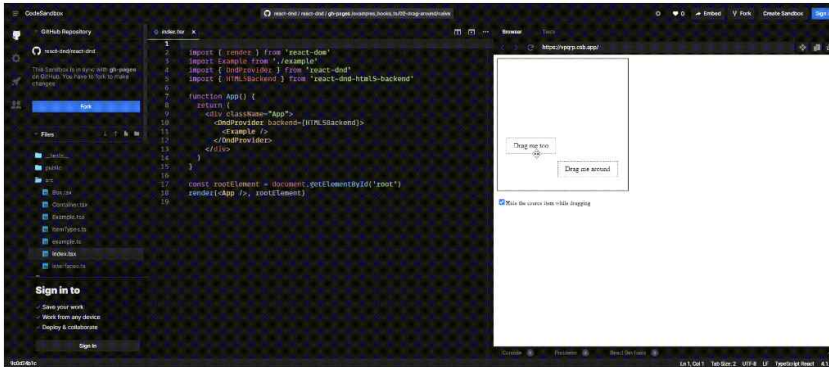
2. react dnd

(https://codesandbox.io/s/github/react-dnd/react-dnd/tree/gh-pages/examples_hooks_ts/02-drag-around/naive?from-embed)

예 — https://codesandbox.io/s/github/react-dnd/react-dnd/tree/gh-pages/examples_hooks_ts/02-drag-around/naive?from-embed

지원되는 입력 방법 — 마우스, 터치

장점 — 기본 HTML DnD API 지원



React dnd는 내부적으로 기본 HTML5 드래그 앤 드롭 API를 사용합니다.

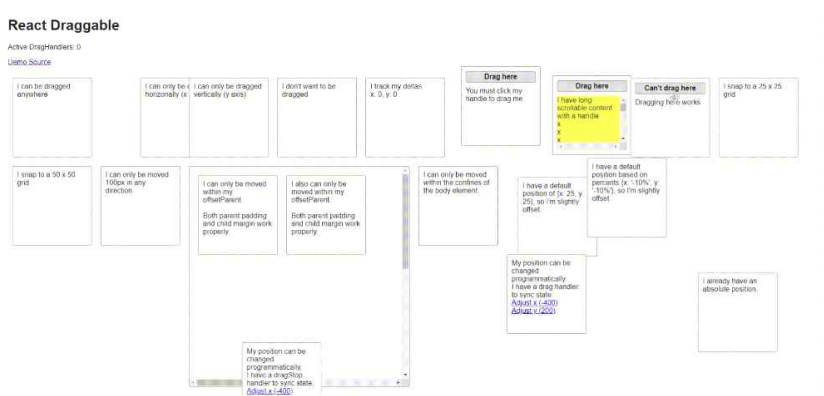
React-dnd 코어는 DOM이나 UI 부분을 조작하지 않습니다. 이는 x/y 좌표, 구성 요소 유형 등과 같은 드래그 가능한 구성 요소의 데이터를 추적합니다. 따라서 드래그 가능한 구성 요소를 위해 만들어진 상태 관리 라이브러리와 같습니다.

3. react- draggable (<http://react-grid-layout.github.io/react-draggable/example/>)

예 — <http://react-grid-layout.github.io/react-draggable/example/>

지원되는 입력 방법 — 마우스, 터치

단일 드래그 영역에 적합합니다.



이것은 **Draggable** 구성 요소에서 구성 요소를 래핑하여 드래그 가능한 구성 요소를 만드는 데 도움이 되는 매우 단순화된 라이브러리입니다.

여기에는 간단한 드래그 가능한 구성 요소를 구현하는 데 필요한 거의 모든 기능이 있습니다. 터치 입력을 지원하며, Draggable 컴포넌트에 Grid Prop을 전달하는 것만으로도 그리드 레이아웃을 구현할 수 있습니다.

부록

1. <https://codesandbox.io/search>