Data-table은 3가지로 구성된다.  
1.page.tsx  
2.columns.tsx – export columns  
3.data-table.tsx – export <DataTable columns={columns} data={data}   
예시  
텍스트, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명  
  
1. page.tsx는 기본 페이지 여기에 DataTable컴포넌트를 가져와 사용한다.   
page.tsx에선 <DataTable>에 필요한 columns와 data를 가져와야 한다.  
columns는 columns.tsx 이고 data는 fetch를 통해 가져오거나 상수 데이터 등 보여줘야 할 data를 말한다.

컬럼

셀

2. columns.tsx 이건 보여줄 데이터를 정의하는 것으로

예를 들면 data의 형식이   
export type Exercise = {  
 \_id: string  
 title: string  
 description: string  
 url: string  
 tags: string[]  
}  
이런 형식 일 때 data-table에서 data를 보여줄 수 있도록 먼저 정의해주는 기능이다.  
예시로 보여주면   
  
텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

위 코드를 보면 accessorKey는 data-table에서 사용할 키값을 정의하는 것  
header는 함수 형식이나 또는 string형식 함수로 만들면 기능이나 css를 추가할 수 있고  
string으로 한 것은 Tags와 같은 식으로 보인다. 사용법은 header: ‘Tags’ 이렇게 적으면 된다.  
  
텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

filterFn은 필터링 하고 싶은 열이 있다면 사용하는 것으로 tags가 배열 형식이라 커스텀 함수를 만들어서 사용한 것이고 기본 사용 방식은 columns에선 따로 설정할 것 없이 바로 data-table.tsx에서 이런 식으로 사용한다.

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명   
텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

기본적으로 includes()로 작동하는 것 같다.  
만약 커스텀 함수가 필요하다면 아래 사진이 예시  
스크린샷, 텍스트, 폰트, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

cell이라는 속성은 이것은 데이터 테이블에서 셀을 어떻게 표시할지 수정할 수 있다. Tags를 보면 (getValue() as string[]).join(“, “)으로 tags는 배열 임으로 이걸 join으로 바꿔준 형태이다.   
url의 cell은 인자로 row가 있는데 row.original로 해당 행의 전체 객체를 가져와서 필요한 속성에 접근해 출력 형식을 바꿀 수 있다.  
cell에 들어가는 인자는 { row, getValue, column, table } 이런 것 들이 있다.  
  
**1. getValue**

* 셀의 데이터를 가져오는 함수입니다.
* getValue()를 호출하여 accessorKey에 해당하는 셀의 값을 가져옵니다.

**2. row**

* 셀이 속한 \*\*행(row)\*\*에 대한 정보를 담고 있는 객체입니다.
* row.original을 사용하여 해당 행의 원래 데이터(전체 객체)를 가져올 수 있습니다.
* 예를 들어, row.original.title로 특정 필드 값을 가져올 수 있습니다.

**3. column**

* \*\*열(column)\*\*에 대한 정보를 담고 있는 객체로, 열의 정의와 관련된 모든 메서드와 속성에 접근할 수 있습니다.
* 이를 통해 특정 열의 필터링이나 정렬 상태, 열 ID 등에 접근하거나 조작할 수 있습니다.

**4. table**

* 현재 **테이블 인스턴스**에 대한 정보입니다.
* table.getColumn()을 통해 특정 열을 가져와 필터링, 정렬, 페이지네이션 등의 다양한 메서드와 상호작용할 수 있습니다.

Id 라는 필드도 있는데 이건 accessorkey와 비슷한 기능이다 예를 들면 accessorkey가 tags로 같은 컬럼이 2개 있다면 이때 id로 구분할 수 있도록 하면 된다. 또는 id만 단독으로 사용하여 새로운 컬럼을 생성할 수 있다 맨위의 사진을 보면 … 이 있는데 이건   
id와 cell만 사용한 것이다 id: ‘actions’ cell: ( {row} ) ={ return() } 이걸로 클릭하면 DropdownMenu가 나오도록 만들었다.  
edit 버튼을 누르면 항목을 수정할 수 있는 창이 나오도록 하고 데이터베이스에 수정한 항목을 fetch하도록 코드를 작성 할 수 있다.  
텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명  
이런 식으로 할 수 있다. 코드가 긴 관계로 아래에 return문은 캡쳐 하지 않고 생략했다.  
텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

3. 제일 중요함 data-tablel.tsx 이건 공식 문서 보고하면 좋음   
기본적으로 초기 틀은 공식문서의 **Basic Table 설정하는 것 보고 따라하면됨**  
텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명  
  
텍스트, 스크린샷, 폰트, 디스플레이이(가) 표시된 사진

자동 생성된 설명  
아까 위에서 cell의 인자로 table 이 있었는데 이거 말하는 것임   
1. data는 page에서 받아온 data 값  
2. Columns도 page에서 받아온 columns.tsx의 columns 정의  
  
**getCoreRowModel: getCoreRowModel()**

* **기본 모델**: getCoreRowModel은 테이블의 기본 행(row) 모델을 정의하는 함수입니다.
* **기능**: 데이터에 대한 필터링, 정렬, 페이지네이션 등의 기능을 적용하지 않은 "기본" 상태의 데이터만 반환합니다.
* **용도**: 이 기본 모델을 바탕으로 다른 필터링 및 정렬된 데이터 모델이 생성됩니다.

**getSortedRowModel: getSortedRowModel()**

* **정렬 모델**: 테이블 데이터에 정렬을 적용할 수 있도록 설정합니다.
* **기능**: 이 모델은 setSorting 상태와 함께 작동하여 테이블 데이터를 특정 열에 따라 오름차순 또는 내림차순으로 정렬된 상태로 제공합니다.
* **용도**: 예를 들어, 열 제목을 클릭해 정렬을 토글하는 경우, 정렬된 데이터를 실시간으로 갱신해 화면에 표시합니다.

**getFilteredRowModel: getFilteredRowModel()**

* **필터링 모델**: 테이블 데이터에 필터링 기능을 추가합니다.
* **기능**: 각 열에 설정된 filterFn 함수를 이용해 특정 조건에 맞는 데이터만 필터링하여 표시할 수 있습니다. onColumnFiltersChange와 함께 사용하여 필터 상태를 업데이트할 수 있습니다.
* **용도**: 예를 들어, 검색창에 입력된 키워드에 맞춰 데이터가 필터링된 결과를 표시합니다.

onColumnFiltersChange: setColumnFilters는 table.getColumn("tags")?.setFilterValue(selectedTags)  
table.getColumn("title")?.setFilterValue(event.target.value)

이 걸 통해 값을 저장하고 화면에 반영해주는 역할 tags와 title 컬럼 모두 columnFilters 변수에 저장됨 콘솔로 보면

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

이런 형식으로 들어감 만약 tags에 값이 추가되면

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명  
배열에 값이 추가된다.