**React-tabulator 사용설명서**

**npm install**

npm install react-tabulator tabulator-tables

**import**

1. import "react-tabulator/lib/styles.css”;
2. import "react-tabulator/lib/css/tabulator.min.css";
3. import { ColumnDefinition, ReactTabulator, ReactTabulatorOptions } from "react-tabulator";

**ref**

* tableRef는 React의 useRef 훅을 사용하여 테이블 컴포넌트를 참조하는 변수
* tableRef.current를 사용하여 ReactTabulator 컴포넌트의 인스턴스에 접근하고, getSelectedData() 메서드를 호출하여 선택된 행의 데이터를 가져온다.

let tableRef = useRef<any>();

const btn\_onClick\_row\_remove = () => {

const selectedData = tableRef.current.getSelectedData();

if (selectedData.length === 0) {

alert("선택된 row가 없습니다.");

return;

}

};

**return**

<ReactTabulator

onRef={(ref) => (tableRef = ref)}

columns={defaultColumns}

data={data}

options={options}

events={{

rowClick: (e: any, row: any) => {},

....

}}

footerElement={<span>Footer || </span>}

/>

**1. columns**

* 테이블의 열을 정의하는 객체의 배열

1. title: 열의 제목을 나타내는 문자열
2. field: 열의 데이터를 나타내는 문자열
3. width: 열의 너비를 나타내는 숫자
4. hozAlign: 열의 텍스트 정렬을 나타내는 문자열
5. formatter: 열의 데이터를 포맷하는 문자열
6. editor: 열의 편집여부를 표현 “true”, “false”, "input", "textarea", "number", "autocomplete", "tickCross", "star" 등
7. editorParams: 에디터의 속성을 정의하는 객체
   1. elementAttributes: 이 속성은 에디터의 요소에 추가할 속성을 정의하는 객체.
   2. mask: 에디터의 입력을 마스킹하는 문자열 "AAA-999", "999-999-9999" 등
8. formatter: 열의 형식 지정
9. formatterParams: 포맷터의 속성을 정의하는 객체

**2. data**

* 테이블에 표시될 데이터를 제공하는 배열

**3. options**

* 테이블의 열을 정의하는 객체의 배열

1. movableRows: 테이블에서 행을 드래그하여 이동할 수 있는지 여부 설정
2. movableColumns: 테이블에서 열을 드래그하여 이동할 수 있는지 여부 설정
3. pagination: 페이지네이션 활성와 여부 설정
4. paginationSize: 페이지 당 표시할 행의 수
5. paginationSizeSelector: 사용자가 선택할 수 있는 페이지 크기를 나타내는 배열
6. height: 테이블 높이를 설정
7. tooltips: 테이블 셀에 마우스를 가져가면 표시되는 툴팁 사용 여부
8. rowClick: 사용자가 행을 클릭했을 때 실행될 콜백 함수
9. cellClick: 사용자가 셀을 클릭했을 때 실행될 콜백 함수

**4. events**

* ReactTabulator 컴포넌트에서 발생하는 다양한 이벤트에 대한 처리 함수들을 정의

1. rowClick: 테이블의 행을 클릭할 때 발생하는 이벤트
2. rowUpdated: 테이블의 행이 업데이트될 때 발생하는 이벤트
3. cellClick: 테이블의 셀을 클릭할 때 발생하는 이벤트
4. dataLoading: 데이터가 테이블에 로드되기 전에 발생하는 이벤트
5. dataLoaded: 데이터가 테이블에 로드된 후에 발생하는 이벤트
6. dataProcessing: 데이터가 처리되기 전에 발생하는 이벤트
7. dataProcessed: 데이터가 처리된 후에 발생하는 이벤트
8. dataChanged: 테이블의 데이터가 변경될 때 발생하는 이벤트
9. cellEditng: 셀 편집이 시작될 때 발생하는 이벤트
10. cellEditCancelled: 셀 편집이 취소될 때 발생하는 이벤트
11. cellEdited: 셀 편집이 완료된 후 발생하는 이벤트
12. cellEdit: 셀 편집이 발생할 때 발생하는 이벤트
13. rowSelected: 행이 선택될 때 발생하는 이벤트
14. rowDeselected: 행이 선택 해제될 때 발생하는 이벤트
15. rowAdded: 행이 추가될 때 발생하는 이벤트
16. rowDeleted: 행이 삭제될 때 발생하는 이벤트
17. rowMoved: 행이 이동될 때 발생하는 이벤트
18. rowSelectionchanged: 행 선택이 변경될 때 발생하는 이벤트

**5. footerElement**

* 테이블 하단에 표시될 커스텀 요소를 정의