AUIGridT.vue 서브 컴포넌트 설명서

AUIGridT.vue 는 Typescript + Vue.js 프레임워크에서 AUIGrid 를 쉽게 사용할 수 있도록 작성한 컴포넌트입니다.

1. **정의 가능한 속성들**

속성명	유형(Type)	기본값	설명
autoResize	Booealn	true	window 리사이징 시 부모에 맞게 리사이징 할지 여부를 지정합니다.
columnLayout	IGrid.Column[]	[]	그리드 칼럼 레이아웃을 지정합니다.
footerLayout	IGrid.Footer[]	[]	그리드 푸터 레이아웃을 지정합니다.
gridProps	IGrid.Props	{}	그리드 속성을 지정합니다.
name	String	''	그리드 이름을 지정합니다.
resizeDelayTime	Number	300	autoResize 설정 시 해당 시간 이후 리사이징을 실행합니다.

* name 속성은 의무사항이 아닙니다. 그러나 name 을 지정하면 해당 name 값을 id 로 갖는 그리드가 생성됩니다. 예로 name='showcase01' 로 설정했다면 해당 그리드는 'aui-grid-wrap-showcase01' 을 id 로 갖는 그리드로 생성됩니다.

DOM 요소를 확인하면 다음과 같이 그리드 생성 div의 id 를 확인 할 수 있습니다.

```
▼<div id="aui-grid-wrap-showcase01" style="position: relative;"> == $0
  ▶<div class="aui-grid" style="position: relative; box-sizing: content-box
```

단, 서버사이드 렌더링(SSR)을 하는 프로젝트나 AUIGrid.vue 가 동적으로 생성/제거되는 경우에 name은 고유값 설정이 의무화 됩니다.

2. 그리드에 접근하여 메소드 사용하기

Vue.js 에서 제공하는 ref 를 이용하여 다음처럼 참조하도록 합니다.

```
// 그리드 객체
const myGrid = ref<AUIGrid | null>(null);
```

위와 같이 ref 로 myGrid 를 생성한 후 AUIGrid 생성 태그에 ref 로 설정합니다

```
<AUIGrid ref="myGrid"/>
```

그러면 언제든 Vue 자신의 스코프에서 다음처럼 그리드에 접근 할 수 있습니다.

```
const grid = myGrid.value as AUIGrid;

grid.addRow();
```

3. Vue.js에서 AUIGridT.vue 사용 예시

```
<script setup lang="ts">
    import { ref, onMounted } from 'vue';
    import * as IGrid from 'aui-grid';
    import AUIGrid from '@/static/AUIGrid-Vue/AUIGridT.vue';

    // 그리드 InstanceType
    type AUIGrid = InstanceType<typeof AUIGrid>;

    // 그리드 객체
    const myGrid = ref<AUIGrid | null>(null);

    // 그리드 속성 정의
    const gridProps: IGrid.Props = {
        width: '100%',
        height: 480,
        // 편집 가능 여부 (기본값 : false)
        editable: true,
        noDataMessage: '출력할 데이터가 없습니다.',
        groupingMessage: '여기에 칼럼을 드래그하면 그룹핑이 됩니다.'
    };

    // 그리드 칼럼 레이아웃 정의
    const columnLayout: IGrid.Column[] = [
        {
            dataField: 'id',
            headerText: 'ID',
            width: 120
        },
        {
            dataField: 'name',
            headerText: '이름',
            width: 140
        },
        {
            dataField: 'country',
            headerText: 'Country',
            width: 140
        },
        {
            dataField: 'product',
```

```
            headerText: 'Product',
            width: 140
        },
        {
            dataField: 'color',
            headerText: 'Color',
            width: 100
        },
        {
            dataField: 'price',
            headerText: 'Price',
            dataType: 'numeric',
            style: 'my-right-column',
            width: 120,
            editRenderer: {
                type: IGrid.EditRendererKind.InputEditRenderer,
                onlyNumeric: true, // 0~9 만 입력가능
                textAlign: 'right', // 오른쪽 정렬로 입력되도록 설정
                autoThousandSeparator: true // 천단위 구분자 삽입 여부
            }
        },
        {
            dataField: 'quantity',
            headerText: 'Quantity',
            dataType: 'numeric',
            style: 'my-right-column',
            width: 100,
            editRenderer: {
                type: IGrid.EditRendererKind.InputEditRenderer,
                onlyNumeric: true, // 0~9 만 입력가능
                textAlign: 'right', // 오른쪽 정렬로 입력되도록 설정
                autoThousandSeparator: true // 천단위 구분자 삽입 여부
            }
        },
        {
            dataField: 'date',
            headerText: 'Date',
            dataType: 'date',
            dateInputFormat: 'yyyy-mm-dd', // 데이터의 날짜 형식
            formatString: 'yyyy 년 mm 월 dd 일' // 그리드에 보여줄 날짜 형식
        }
    ];

    const cellClick = (event: IGrid.EventKind.CellClick) => {
        console.log(event);
    };
```

```
    const requestGridData = async () => {
        const grid = myGrid.value;
        grid?.showAjaxLoader();
        const response = await fetch(BASE_URL + '/data/normal_100.json');
        const jsonData = await response.json();
        console.log(jsonData);
        grid?.setGridData(jsonData);
        grid?.removeAjaxLoader();
    };

    onMounted(() => {
        requestGridData();
    });
</script>

<template>
    <div>
      <AUIGrid ref="myGrid" :gridProps="gridProps"
          :columnLayout="columnLayout" @cellClick="cellClick"/>
    </div>
</template>
```

- 감사합니다. -