

AUIGrid.vue 서버 컴포넌트 설명서

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

1. 정의 가능한 속성들

속성명	유형(Type)	기본값	설명
autoResize	Boolean	True	window 리사이징 시 부모에 맞게 리사이징 할지 여부를 지정합니다.
columnLayout	Array	[]	그리드 칼럼 레이아웃을 지정합니다.
footerLayout	Array	[]	그리드 푸터 레이아웃을 지정합니다.
gridProps	Object	{}	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"
    <div>
```

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 속성을 이용하여 다음처럼 참조하도록 합니다.

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

그러면 언제나 Vue.js 자신의 인스턴스 스코프에서 다음처럼 그리드에 접근 할 수 있습니다.

```
const grid = this.$refs.myGrid;
grid.addRow();
```

3. Vue.js에서 AUIGrid.vue 사용 예시

```
<template>
  <AUIGrid ref="myGrid"
    :columnLayout="columnLayout"
    :gridProps="gridProps" />
</template>
<script>
  // AUIGrid 컴포넌트
  import AUIGrid from '../static/AUIGrid-Vue.js/AUIGrid.vue';
  import axios from 'axios';
  import './samples.css';

  export default {
    components: {
      AUIGrid,
    },

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

      // 그리드 칼럼 레이아웃
      columnLayout: [
        {
          dataField: 'id',
          headerText: 'ID',
          width: 120,
        },
        {
          dataField: 'name',
          headerText: 'Name',
          width: 140,
        },
        {
          dataField: 'country',
          headerText: 'Country',
          width: 140,
        },
      ],
    })
  }
}
```

```

        {
            dataField: 'price',
            headerText: 'Price',
            dataType: 'numeric',
            style: 'my-column',
            width: 120,
        },
        {
            dataField: 'quantity',
            headerText: 'Quantity',
            dataType: 'numeric',
            style: 'my-column',
            width: 100,
        },
        {
            dataField: 'date',
            headerText: 'Date',
            dataType: 'date',
        },
    ],
}),
created() {
    console.log('SampleDefault 생성됨');
},

mounted() {
    console.log('SampleDefault 마운트됨');
    // 초기 데이터 얻기
    this.requestGridData();
},

unmounted() {
    console.log('SampleDefault 언마운트됨');
},
methods: {
    // 그리드 데이터 조회하여 삽입
    requestGridData() {
        const grid = this.$refs.myGrid;
        const REQ_URL = './data/normal_100.json';

        grid.showAjaxLoader();
        axios.get(REQ_URL).then((result) => {
            //console.log(result);
            // 그리드 데이터 삽입
            grid.setGridData(result.data);
            grid.removeAjaxLoader();
        });
    },

```

```
    },  
  },  
};  
</script>
```

- 감사합니다. -