work-sheet-comp

Far from being an Excel worksheet **work-sheet-comp** is a Vue.js web component that can display a large table of data where row index headings freeze as you scroll horizontally or table headings freeze as you scroll vertically. Also when a table heading is clicked, the entire table is sorted based on the selected column. Repeated clicks of the column toggle the sort direction between ascending/descending.

work-sheet-comp can be installed via with the included `package.json` file for a local installation via the npm install command. **work-sheet-comp** depends on the vue.js framework. A demo folder is provided that used Parcel together with its associated `package.json` file to bundle together **work-sheet-comp** along with its vue.js dependency for a simple application. Further details are provided below for running the demo.

Props

A prop in Vue.js is a custom attribute for passing information from a parent component hosting **work-sheet-comp** instance(s) to a **work-sheet-comp** as a child component.

work-sheet-comp has the following props for a parent to bind to child:

`sheet_rows` -- an array of row arrays that can contain a mix of strings, Dates, and numbers (Array,of arrays default: null)

`headings` -- an array of strings for the sheet's headings (Array, default: null)

`column_widths` -- an array of numbers that specifies each of the column widths in pixels (Array of numbers, default: array of 120's)

`table_width` -- the width of the worksheet in pixels (number, default: 600)

`rows_per_view` -- the number of rows to view at one time (number, default: 10)

`sort_descending` -- a boolean which if `true` sets the column sorting to descending (boolean, default: `false`)

`sort_column_idx` -- a number that sets the sorting column starting with `0` for the index column (number, default: 0)

`css_variables` -- defines the css variables for **work-sheet-comp** (object, default: null)

Styling

The `css_variables` prop is a javascript object that contains any combination of css variable names as keys and associated values. The following list are the css variable names along with their default values:

```
  {
    worksheet_comp_font_family: 'Verdana,serif',
    worksheet_comp_border_color: 'black',
    worksheet_comp_background_color: 'white',

    worksheet_comp_table_color: 'black',

    worksheet_comp_row_hover_border: 'solid 1px black',
    worksheet_comp_cell_selected_color: '#0f9d58'
  }
```

Events

work-sheet-comp has an event that notifies the parent component when a table cell is clicked. **work-sheet-comp** emits the following named event:

```
'work_sheet_comp_cell' -- returns an object with the selected cell's row and
column index
```

The parent component can listen to the above event and provide a callback for further processing. Events emitted from a child component back to the parent is explained at Vue Custom Events.

Demonstration

One demonstration of **work-sheet-comp** is provided in the folder named `demo/dist` and can be viewed by hosting the `index.html` file. The demo (templated in the `App.vue` file) displays 15 random valued parameters with 100 rows. **work-sheet-comp** can accept numbers, strings, and Dates for data. In addition, column widths are set. The parent listens to the cell click event and displays the row/column indexes at console.log.

As a suggestion, install http-server locally/globally via npm then enter the command `http-server` in the **work-sheet-comp** `dist` directory. From a browser enter the url: `localhost:8080/` to view the demo.

Here is some example code for using **work-sheet-comp** taken from the `App.vue` file:

```
        <work-sheet-comp
          :sheet_rows="sheet_rows"
          :headings="headings"
          :column_widths="column_widths"
          :table_width="table_width"
          :rows_per_view="rows_per_view"
          :css_variables="css_variables"
          v-on:work_sheet_comp_cell="show_cell">
        </work-sheet-comp>
```

And the supporting data references:

```
     data: function() {
      return {
        sheet_rows: null,
        headings: [
          'Index',

  'Parameter_1','Big_Heading_Parameter_2','Parameter_3','Parameter_4','Parameter_5
  ',
          'Parameter_6','Parameter_7','Parameter_8','Parameter_9','Parameter_10',

  'Parameter_11','Parameter_12','Parameter_13','Parameter_14','Parameter_15'
        ],
        column_widths: [
          60,
          100,120,80,140,120,
          120,140,100,60,120,
          130,120,100,80,110
        ],

        //table_width: 1000,
        rows_per_view: 20,

        css_variables: {
          worksheet_comp_border_color: 'white',
          worksheet_comp_table_color: 'white',
          worksheet_comp_background_color: 'black',
          worksheet_comp_row_hover_border: 'solid 1px white'
        }
      }
    },
```

```
 methods: {
    show_cell: function(obj){
      console.log(`row_idx: ${obj.row} col_idx: ${obj.col}`);
    }
  },
  mounted: function() {
    this.sheet_rows = [];
    for(let i = 0; i < 100; i++) {
      const row = [];
      row.push(i+1);
      for(let ii = 0; ii < 15; ii++){
        const val = (Math.random() * 100.0) + 1.0;
        row.push(val.toFixed(2));
      }
      this.sheet_rows.push(row);
    }
    //assign strings and dates to rows
    //this.sheet_rows[0][3] = new Date('5/24/1949');
    this.sheet_rows[1][4] = 'A String';
  }
```