select-comp

select-comp is a Vue.js (>= 2.5) component similar to an HTML select tag where the parent can set the select value along with a collapsing panel of options from which to select.

select-comp can be installed via with the included `package.json` file for a local installation via the npm install command. **select-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 **select-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 **select-comp** instance(s) to an **select-comp** as a child component. **select-comp** has the following props for a parent to bind and send information to:

`items` -- the string array of items to select from (string array, default: null)

`select_value` -- when assigned by the parent, sets the current selected value if the value is a member of `items` Can be assigned to a static string or bound to a parent's data member. The demo illustrates both cases. If set to `null` and the `placeholder` property is not `null` then the `placeholder` value appears in the selection box. (string, default: null)

`heading` -- a heading to be displayed above the selection box (string,default: null)

`placeholder` -- sets an initial placeholder value (string, default: null)

`drop_panel_height` -- the height of drop down list panel (string, default: 6.5rem)

`blur_panel` -- if false, will not roll up item panel when component is out of focus (boolean, default: true)

`single_border` -- if a single bottom border is displayed or rectangular (boolean, default: false)

`css_variables` -- defines the css variables for **select-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 for a quick styling of **select-comp**:

```
{
    select_comp_font_family: 'Verdana,serif',

    select_comp_width: '10rem',
    select_comp_font_size: '1rem',
    select_comp_color: 'darkgray',
    select_comp_background: 'transparent',
    select_comp_border_color: 'black',

    select_comp_heading_font_size: '1rem',
    select_comp_heading_color: 'black',
```

```
        select_comp_heading_font_weight: 'bold',

        select_comp_hover_background_color: 'lightblue',

        select_comp_items_panel_color: 'black',
        select_comp_items_panel_background: 'white',
        select_comp_items_panel_border: '1px solid black',
        select_comp_items_panel_position: 'static',
        select_comp_items_panel_z_index: 'auto',

        select_comp_item_font_size: '.75rem',
        select_comp_item_hover_box_shadow: '0 0 10px gray',
        select_comp_item_hover_color: 'violet',
    }
```

As an example you could bind from the parent the following object to the `css_variables` prop for setting the width of **select-comp** and its border color to 'green':

```
{
  select_comp_width: '14rem',
  select_comp_border_color: 'green'
}
```

Note that multiple **select-comp** children of the parent can each be bound to a unique set of `css_variable` prop objects. To enforce the same styling across all **select-comp** children, simply bind just one `css_variable` prop object to all the **select-comp** children.

Events

select-comp has one event that notifies the parent component of the current selected item.

select-comp emits the following single named event:

```
 'select_comp_value_changed' -- returns the current selected item string
```

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.

The demonstration shows how the event can be incorporated.

Demonstration

One demonstration of **select-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 two independent **select-comp** components and shows how the item list is set. In addition buttons are provided which when clicked will select, add, or delete an item from one of the components. Note that callbacks are assigned to both **select-comp**s'.

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

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

```html
    <div class="demo_container">
      <div class="buttons_box">
        <button v-on:click="add_friend">Add Friend 'Johnny'</button>
        <button v-on:click="del_friend">Delete Friend 'Pete'</button>
        <button v-on:click="set_travel_value">Set Travel to 'Cleveland'</button>
        <button v-on:click="set_travel_null">Set Travel to null</button>
        <button v-on:click="set_new_travels">Set New Travel List</button>
        <button v-on:click="show_current_values">Show Current Values (at
console)</button>
      </div>

      <div class="selects_box">
        <div class="select_box">
          <select-comp
            heading="My Best Friend"
            placeholder="Select a friend"
            select_value="Jim"
            :items="items_friends"
            :blur_panel="blur_panel"
            :single_border="single_border_friends"
            :css_variables="css_variables_friends"
            v-on:select_comp_value_changed="value => {this.current_best_friend =
value}">
          </select-comp>
        </div>
        <div class="select_box">
          <select-comp
            heading="This Year's Travel Spot"
            placeholder="Select travel spot"
            :items="items_travels"
            :select_value="current_travel_spot"
            :css_variables="css_variables"
            v-on:select_comp_value_changed="value => {this.current_travel_spot =
value}">
          </select-comp>
        </div>
      </div>
      <section class="pbox">
        <p>
          rownwnn andlle  aljnd;3e; .akdnend  qnfdslns alne. andl;; irjlsek
sljdnel
          ewlnsl lsnkei s nelnsndl  slhelnskdn lslsn; a;ke;;;0ekd  lsnlsnel
sllls
          2nwlenkf lsmksnel; slasll slenen ls nellllsn 5nrlnelwn lsnl  slnwnl
rien
          wqnelnwlen tltnylnr lsnenl wqnsna slns yktnu iims wemr eeenelsn
ssysnwl snn
          rlrnelwn lsnen  rlnelwn alqnfkt ndlnwln eerlnw qnerndlsnl yytl alsn
lkd5lr
          endsbx alnsnr uyntlrnl lln wndjf prpt wenrnf qwns duu tnrhdhh slnfldn
wekr
        </p>
      </section>
    </div>
```

And the supporting data references:

```
data() {
  return {
    items_friends: [
      'Rick',
      'Nancy',
      'Jim',
      'Pete',
      'Harry',
      'Sally',
      'Jill',
      'Bob'
    ],

    items_travels: [
      'New York',
      'Boston',
      'Cleveland',
      'San Francisco',
      'Chicago',
      'Detroit',
      'St. Louis',
      'Toledo'
    ],
    current_best_friend: null,
    current_travel_spot: null,
    blur_panel: false,
    single_border_friends: true,
    css_variables_friends: {
      select_comp_width: '12.5rem',
      select_comp_border_color: 'green',
      select_comp_items_panel_background: 'lightgreen',
      select_comp_items_panel_position: 'absolute',
      select_comp_items_panel_z_index: '100'
    },
    css_variables: {
      select_comp_width: '14rem',
      select_comp_border_color: 'green'
    }
  }
},
components: {
  SelectComp
},
methods: {
  //parent to child -- add to friends list
  add_friend: function(){
    this.items_friends.push('Johnny');
  },
  //parent to child -- delete from friends list
  del_friend: function(){
    const idx = this.items_friends.indexOf('Pete');
    this.items_friends.splice(idx, 1);
  },
  //parent to child -- set value
  set_travel_value: function(){
    this.current_travel_spot = 'Cleveland';
  },
  //parent to child -- set value
```

```javascript
    set_travel_null: function(){
      this.current_travel_spot = null;
    },
    //parent to child -- reset the travels list
    set_new_travels: function(){
      this.items_travels =
['Spain','France','Germany','England','Canada','Mexico'];
    },
    show_current_values: function(){
      console.log(
        `\nBest Friend: ${this.current_best_friend}\n
        Travel Spot: ${this.current_travel_spot}`);
    }
  }
```