

# dnd-comp

---

**dnd-comp** is a web component (Vue.js >= 2.5) that provides a titled, collapsible list of items that can be dragged to another target element.

**dnd-comp** can be installed via with the included `package.json` file for a local installation via the [npm install](#) command. **dnd-comp** depends on the [vue.js](#) framework. A demo folder is provided that used [Parcel](#) together with its associated `package.json` file to bundle **dnd-comp** 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 **dnd-comp** instance(s) to an **dnd-comp** as a child component.

**dnd-comp** has the following props for a parent to bind and send information to:

`items` -- an array of strings to set the items of the component (array, default: null)

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

`drop_panel_height` -- the height of drop down list panel (string, default: '100px')

`css_variables` -- defines the css variables for **dnd-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 **dnd-comp** :

```
{
  dnd_comp_font_family: 'verdana,serif',

  dnd_comp_heading_color: 'black',
  dnd_comp_heading_font_size: '1rem',
  dnd_comp_heading_font_weight: 'bold',

  dnd_comp_down_icon: '\21D3';
  dnd_comp_icon_color: 'black',

  dnd_comp_items_panel_position: 'static',
  dnd_comp_items_panel_z_index: 'auto',
  dnd_comp_items_panel_color: 'black',
  dnd_comp_items_panel_background: 'white',
  dnd_comp_items_panel_border: '1px solid black',

  dnd_comp_item_font_size: '.8rem',
  dnd_comp_item_hover_box_shadow: '0 0 10px gray',
  dnd_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 heading font size and the icon color of **dnd-comp** :

```
{dnd_comp_heading_font_size: '24px', dnd_comp_icon_color: 'brown'}
```

Note that multiple **dnd-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 **dnd-comp** children, simply bind just one `css_variable` prop object to all the **dnd-comp** children.

## Events

**dnd-comp** emits one single event that notifies the parent component when a drag-n-drop has been cancelled. The event responds with the source item of the drag-n-drop. The name of the event to be listened to is named: `dnd_comp_cancelled`. The parent component can listen to the this 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 **dnd-comp** is provided in the folder named `demo` and can be viewed by hosting the `index.html` file. The demo (templated in the `App.vue` file) shows **dnd-comp** with a list of cities that can be dragged to a `div` element. The `div` element is set up in simple javascript way to accept dragged objects. The parent is set up to listen to a cancelled drag-n-drop and `console.log` the item that was being dragged. Also a button is provided on the parent which when clicked will add a new item to the list.

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

The demo folder contains a `package.json` file that can be used to setup dependencies for this demo and as a template for other applications using **dnd-comp**.

Here is some example code for using **dnd-comp** taken from the demo:

```
<section class="button_section">
  <button v-on:click="reset_text">Reset Target Text</button>
  <button v-on:click="add_buffalo">Add Buffalo</button>
</section>
<dnd-comp>
  heading="Places to Travel"
  :items="items"
  :css_variables="css_variables"
  v-on:dnd_comp_cancelled="dnd_cancelled">
</dnd-comp>
```

And the supporting data references:

```
data() {
  return {
    items: [
      'New York',
      'Boston',
      'Cleveland',
```

```

        'San Francisco',
        'Chicago',
        'Detroit',
        'St. Louis',
        'Toledo'
    ],
    css_variables: {
        dnd_comp_heading_color: 'white',
        dnd_comp_icon_color: 'white',
        dnd_comp_items_panel_color: 'white',
        dnd_comp_items_panel_background: 'transparent',
        dnd_comp_items_panel_border: '1px solid white'
    }
},
methods: {
    reset_text: function(){
        this.$refs.target_box.innerText = 'DND Your Destination Here';
    },
    target_drop: function(e){
        const text = e.dataTransfer.getData("text");
        e.target.innerText = text;
        console.log(`Travel Destination: ${text}`);
    },
    target_dragover: function(e){
        e.preventDefault();
        e.dataTransfer.dropEffect = 'copy';
    },
    add_buffalo: function(){
        this.items.push('Buffalo');
    },
    //child to parent
    dnd_cancelled: function(source_text){
        console.log(`Drop cancelled from: ${source_text}`);
    }
}
}

```