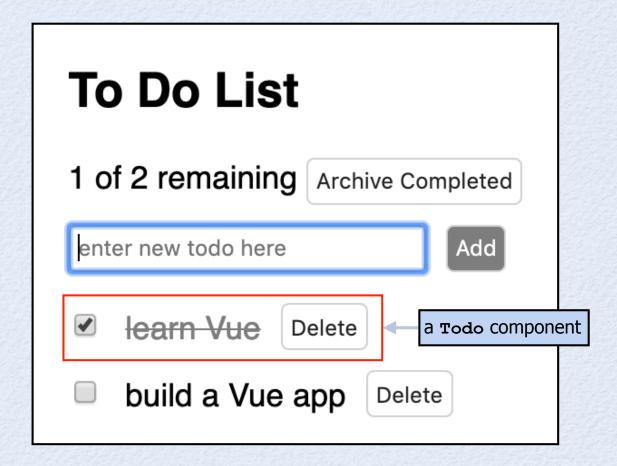
# Vue Todo App

#### Overview

 A basic todo app that pulls together much of what we've seen so far



# App.vue

```
<template>
  <div id="app">
    <TodoList />
 </div>
</template>
<script>
import TodoList from './components/TodoList.vue';
export default {
  name: 'app',
  components: {
    TodoList
};
</script>
<style>
#app {
  font-family: sans-serif;
  padding-left: 10px;
button, input {
 border: solid lightgray 1px;
 border-radius: 4px;
 padding: 4px;
</style>
```

3

#### Todo.vue

<template>

```
<1i>>
                             <input
<script>
                               type="checkbox"
export default {
                               :checked="todo.done"
  name: 'Todo',
                               @change="onToggleDone"
  props: {
                             >
    onDeleteTodo: {
                             <span :class="doneClass">{{todo.text}}
      type: Function,
                             <button class="delete-btn" @click="onDeleteTodo">
      required: true
                               Delete
    },
                             </button>
    onToggleDone: {
                           type: Function,
                         </template>
      required: true
    },
                                                        <style scoped>
                                                       button {
    todo: {
                                                         margin-left: 10px;
      type: Object,
      required: true,
      validator(obj) {
                                                        .done-true {
                                                         color: gray;
        return obj.text && obj.done !== undefined;
                                                         text-decoration: line-through;
  },
                                                       li {
                                                         align-items: center;
  computed: {
                                                         display: flex;
    doneClass() {
                                                         margin-top: 10px;
      return 'done-' + this.todo.done;
                                                       li > span {
                                                         margin-left: 10px;
</script>
                                                       </style>
```

4

#### TodoList.vue ...

```
<template>
 <div>
   <h2>To Do List</h2>
   <div>
      {{uncompletedCount}} of {{todos.length}} remaining
      <button
        class="archive-btn"
        @click="onArchiveCompleted"
     >Archive Completed</button>
   </div>
   <form @submit.prevent>
      <input
        class="todo-input"
        type="text"
        size="30"
        autofocus
       placeholder="enter new todo here"
        v-model="todoText"
        @change="onTextChange"
      <button class="add-btn" :disabled="!todoText" @click="onAddTodo">
        Add
     </button>
   </form>
```

### ... TodoList.vue ...

#### ... TodoList.vue ...

```
<script>
import Todo from './Todo.vue';
let lastId = 0;
const createTodo = (text, done = false) => ({id: ++lastId, text, done});
export default {
  name: 'TodoList',
  components: {Todo},
 computed: {
    uncompletedCount() {
      return this.todos.filter(t => !t.done).length;
  },
  data() {
    return {
      todos: [createTodo('learn Vue', true), createTodo('build a Vue app')],
      todoText: ''
    };
```

#### ... TodoList.vue ...

```
methods: {
    onAddTodo() {
      this.todos = this.todos.concat(createTodo(this.todoText));
     this.todoText = '';
    },
    onArchiveCompleted() {
                                                      just deletes todos that
      this.todos = this.todos.filter(t => !t.done);
                                                      are done in this version
    },
    onDeleteTodo(todoId) {
      this.todos = this.todos.filter(t => t.id !== todoId);
    },
    onTextChange (event) {
      this.todoText = event.target.value;
    },
    onToggleDone(todo) {
      const {id} = todo;
      this.todos = this.todos.map(t =>
        t.id === id ? {...t, done: !t.done} : t
      );
</script>
```

8

## ... TodoList.vue

```
<style scoped>
button:disabled {
  background-color: gray;
  color: white;
  margin-left: 10px;
}

input {
  margin-top: 10px;
}

ul.unstyled {
  list-style: none;
  margin-left: 0;
  padding-left: 0;
}
</style>
```