

# Vue Todo App

# Overview

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

## To Do List

1 of 2 remaining Archive Completed

Add

☒ ~~learn Vue~~ Delete

☐ build a Vue app Delete

← a Todo component



# 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>
```

# Todo.vue

```
<script>
export default {
  name: 'Todo',
  props: {
    onDeleteTodo: {
      type: Function,
      required: true
    },
    onToggleDone: {
      type: Function,
      required: true
    },
    todo: {
      type: Object,
      required: true,
      validator(obj) {
        return obj.text && obj.done !== undefined;
      }
    },
    computed: {
      doneClass() {
        return 'done-' + this.todo.done;
      }
    }
  };
}</script>
```

```
<template>
  <li>
    <input
      type="checkbox"
      :checked="todo.done"
      @change="onToggleDone"
    >
    <span :class="doneClass">{{todo.text}}</span>
    <button class="delete-btn" @click="onDeleteTodo">
      Delete
    </button>
  </li>
</template>
```

```
<style scoped>
button {
  margin-left: 10px;
}

.done-true {
  color: gray;
  text-decoration: line-through;
}

li {
  align-items: center;
  display: flex;
  margin-top: 10px;
}

li > span {
  margin-left: 10px;
}
</style>
```

# 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>
  </div>
</template>
```



# ... TodoList.vue ...

```
<ul class="unstyled">
  <Todo v-for="todo in todos"
    :key="todo.id"
    :todo="todo"
    :onDeleteTodo="() => onDeleteTodo(todo.id) "
    :onToggleDone="() => onToggleDone(todo) "
  />
</ul>
</div>
</template>
```

# ... 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() {  
    this.todos = this.todos.filter(t => !t.done);  
  },  
  
  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>
```

just deletes todos that  
are done in this version



# ... 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>
```