

Look at You, Vue!

Ken Rimple, Philly Tech Week 2018

@RimpleOnTech

Agenda

- What is Vue?
- The Vue CLI
- Vue Components Dive
- Components Together
- Vuex
- Routing (a quick demo)
- Example with GraphQL

What is Vue?

- A single-page Javascript framework
- Created by Evan You
- Former Googler who worked on AngularJS projects

Design goal

"I figured, what if I could just extract the part that I really liked about Angular and build something really lightweight without all the extra concepts involved?"

— Source: Wikipedia

Similar Concepts

- Components
- Directives
- Filters
- Attributes and Events with shortcuts

The Vue CLI

Install with

```
npm install -g @vue/cli
```

Create an app with

```
vue create foobar
```

Most common Vue commands

```
yarn serve  
yarn build
```

Vue Single-Page Components

A single page component

save with filename.vue

```
<template>
  <p>One parent node</p>
</template>

<script>
export default {
  ...
}
</script>

<style scoped>
...
</style>
```

A simple Vue Component - Template

TodoItem.vue - Template

```
<template>
  <li>
    <input type="checkbox" v-model="todoItem.complete">
    <span>{{ todoItem.todo }}</span>
  </li>
</template>
```

A simple Vue Component - Script

TodoItem.vue - Script

```
<script>
export default {
  name: 'TodoItem',
  props: {
    todoItem: { type: Object, required: true }
  }
}
</script>
```

Bindings

Custom properties (by component type)

```
<sui-sidebar
  :visible="sidebarVisible"
  :width="'wide'">
```

Style and property application...

```

```

Events

Builtins

```
<button @click="add()">+</button>
```

Event

```
export default {
  ...
  methods: {
    add: function() {
      ... do something
    }
  }
}
```

Events and Arguments

Builtins

```
<button @click="input(7)">7</button>
```

Event

```
export default {  
  ...  
  methods: {  
    input: function(value) {  
      this.currentInput += value;  
    }  
  }  
}
```

Props - defining

*Component will receive a required **todoItem** prop*

```
export default {  
  name: 'TodoItem',  
  props: {  
    todoItem: {  
      type: Object,  
      required: true  
    }  
  }  
}
```

Props - passing

Passing a prop

```
<ul>  
  <TodoItem v-for="todo of todos"  
    v-bind:key="todo.id"  
    :todo-item="todo">  
  </TodoItem>  
</ul>
```

- Use with **this.propName** in component

Events - Creating with **\$emit**

In the template

```
<button @click="addTodo">Add Task</button>
```

The component binding

```
methods: {  
  addTodo: function() {  
    this.$emit('add-todo', this.task);  
  }  
}
```


Vuex

- A state management engine
- Similar, but not the same as, Redux

Vuex Installation

- Use **Vue.use** to import the Vuex plugin

Base usage

```
import Vue from 'vue'
import Vuex from 'vuex'

...
Vue.use(Vuex)
```

- This enables Vue to provide a Vuex store

Enabling Vuex

- Create your Vuex store
- Add it to the properties of the **new Vue** creator

Define the Vuex store (my-store.js)

```
import Vuex from 'vuex'

export default new Vuex.store({
  state: ...,
  mutations: ...,
  actions: ...,
  plugins: ...,
  etc...
});
```

Enabling Vuex (continued)

Use it

```
import store from './my-store';

new Vue({
  ...
  store
  ...
})
```

Key Features

- Mutations
 - Change the store
 - Don't have to be 'pure functions' like Redux
- Actions
 - Dispatch mutations
 - Called from components
- Integration middleware
 - The store is available as **store**
 - **getState** can be used for computing properties
 - **getActions** can add actions to your components

Vuex Demo

Open Discussion / hacking