



INTRODUCTION TO VUE.JS

---

# ROOM WITH A VUE



July 27, 2018

## ZACHARY KLEIN

- ▶ Grails developer since 2010
- ▶ Frontend developer since 2015
- ▶ Joined OCI Grails team in 2015
- ▶ OSS contributor



## ABOUT US

- ▶ Object Computing, Inc
  - ▶ Based in St Louis, MO
  - ▶ Consulting and training provider (24+ years experience)
  - ▶ Corporate sponsor to the **Grails** & **Micronaut** frameworks and **Groovy** language
  - ▶ <https://objectcomputing.com>



# GRAILS

<http://start.grails.org>

- ▶ Built over Spring Boot & Gradle
- ▶ Plugin system
- ▶ REST controllers, JSON views
- ▶ Single Page App support:
  - ▶ Angular
  - ▶ React
  - ▶ **Vue.js**

```
> curl -O start.grails.org/myapp.zip -d profile=vue
```

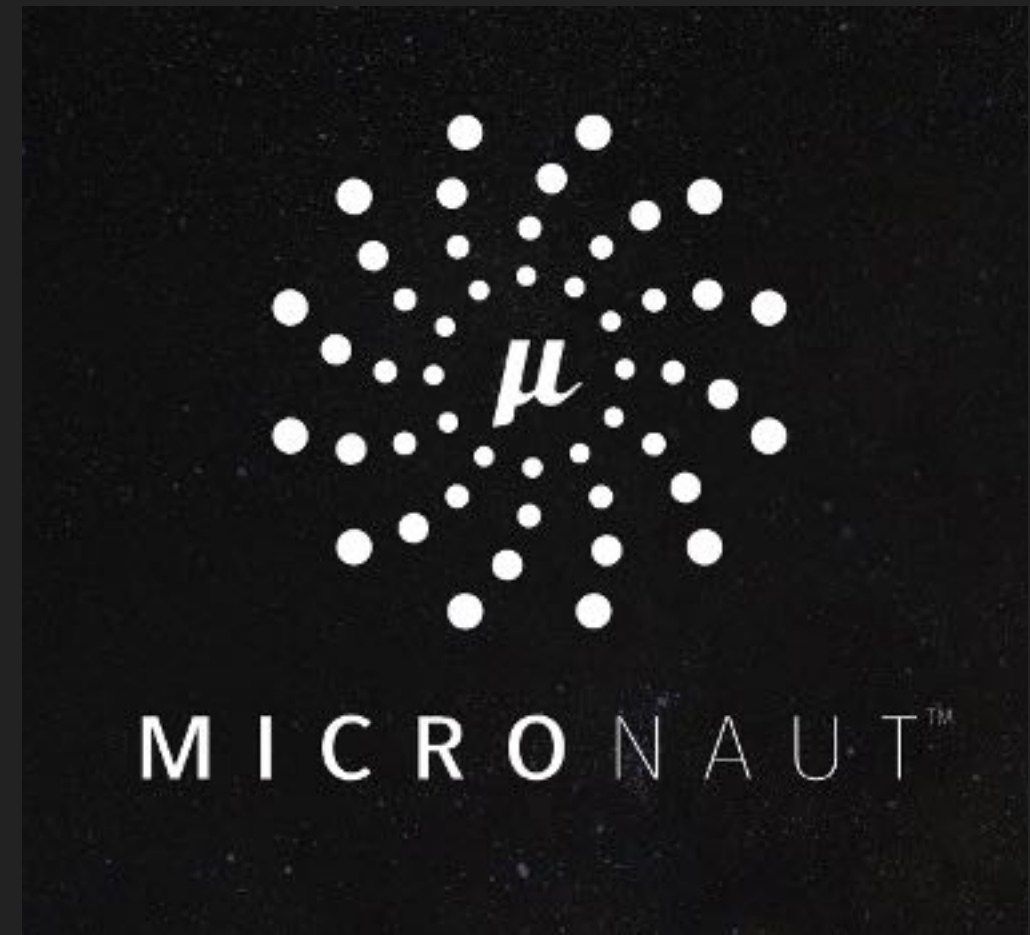
<http://guides.grails.org/tags/vue.html>



# MICRONAUT

<http://micronaut.io>

- ▶ Full stack OSS JVM framework for microservices
- ▶ Subsecond startup time
- ▶ Uses as little as 7MB memory
- ▶ Reactive HTTP server & client
- ▶ “Natively” Cloud-Native
- ▶ Supports Java, Groovy, and Kotlin



Introduction/Sample Project: <http://bit.ly/micronaut>



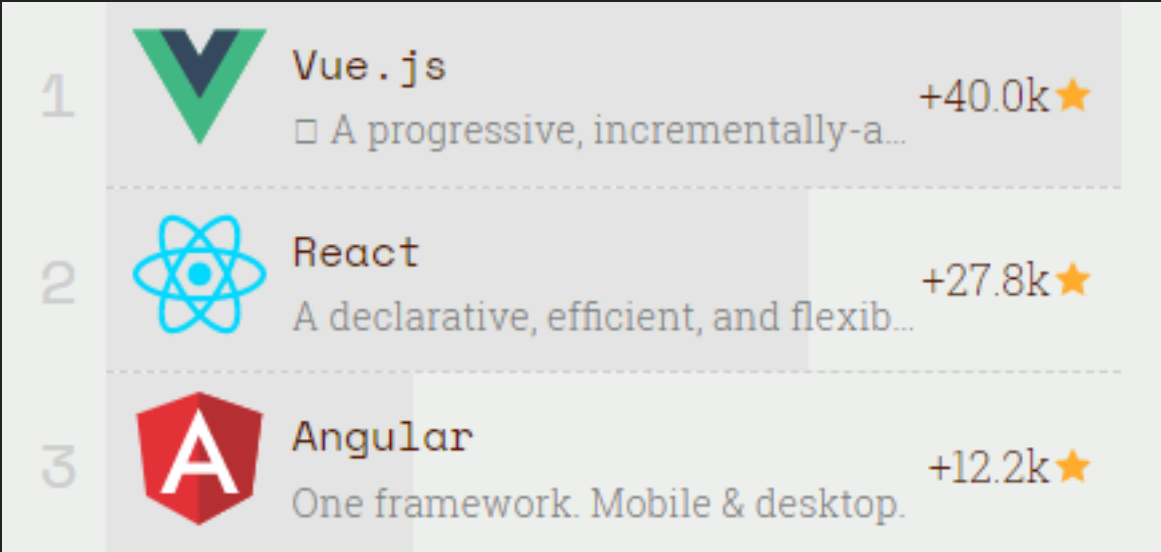
## AGENDA

- ▶ Introduction to Vue & Ecosystem
- ▶ Demo: Build a Vue app
- ▶ Review & Comparison

Seriously, *another* JavaScript framework?



# ROOM WITH A VUE



<https://risingstars.js.org/2017/en/#section-framework>



<https://stateofjs.com/2017/front-end/results/>



<http://www.timqian.com/star-history/#facebook/react&angular/angular&vuejs/vue>

## WHAT IS VUE?

- ▶ Released in 2014
- ▶ Current major release (2.x) in 2016
- ▶ Evan You (inventor & core developer) comes from a Google/Angular background
- ▶ Balances between the power/complexity of Angular and the simplicity/focus of React



## WHAT IS VUE?

- ▶ Small, focused view library
- ▶ Performant\*
- ▶ Full-featured API
- ▶ Mature\* ecosystem
- ▶ Powerful devtools
- ▶ Active community



## GETTING STARTED

- ▶ Install from NPM
- ▶ Install vue-cli:
  - ▶ `npm install -g @vue/cli`
  - ▶ `vue create my-project`



# GETTING STARTED

```
<script src="https://unpkg.com/vue"></script>
```

```
<div id="app">  
  {{ text }}  
</div>
```

```
<script>
```

```
var app = new Vue({  
  el: '#app',  
  data: {  
    text: 'Hello World!'  
  }  
})
```

```
</script>
```



# THE VUE INSTANCE

- ▶ Ground zero for a Vue.js app
- ▶ Accepts an object containing the Vue **Instance definition**
- ▶ Most definitions shared between instance and **components**

Vue Instance

```
new Vue({  
  el: '#app',  
  data: {...},  
  computed: {...},  
  methods: {...},  
  template: '<App/>'  
})
```

Vue Component

```
export default {  
  name: 'component',  
  data () { return {...} },  
  name: 'component',  
  methods: {...}  
}
```

## THE VUE INSTANCE

### ▶ **el**

- ▶ Specifies the DOM element where the app will be rendered

```
new Vue({  
  el: '#app'  
})
```

```
<div id="app"></div>
```

## THE VUE INSTANCE

### ▶ **data**

- ▶ An object describing the state of the app/component
- ▶ Properties are subject to reactive rendering within the app/component (one-way & two-way)
- ▶ New properties can be added, but will not be subject to reactive behavior
- ▶ Components must use a **data()** function that returns state



# THE VUE INSTANCE

## ► data

Vue Instance

```
new Vue({  
  data: {  
    myValue: 3,  
    myObject: {  
      prop: 'abc'  
    }  
  }  
})
```

Vue Component

```
export default {  
  data () {  
    return {  
      myValue: 3,  
      myObject: {  
        prop: 'abc'  
      }  
    }  
  }  
}
```

## THE VUE INSTANCE

### ▶ **Methods**

- ▶ Arbitrary functions
- ▶ Can access/manipulate data
- ▶ Can be called from templates, other methods

```
methods: {  
  toggleLinks: function () {  
    this.showLinks = !this.showLinks  
  }  
},
```

## THE VUE INSTANCE

- ▶ **computed (properties)**
  - ▶ Functions that return dynamic values
  - ▶ Accessed from templates & methods as state properties
  - ▶ Cached - only recomputed if their input data changes

```
data: {  
  firstNumber: 1  
},  
computed: {  
  number: function () {  
    return 2 + this.firstNumber  
  }  
},
```

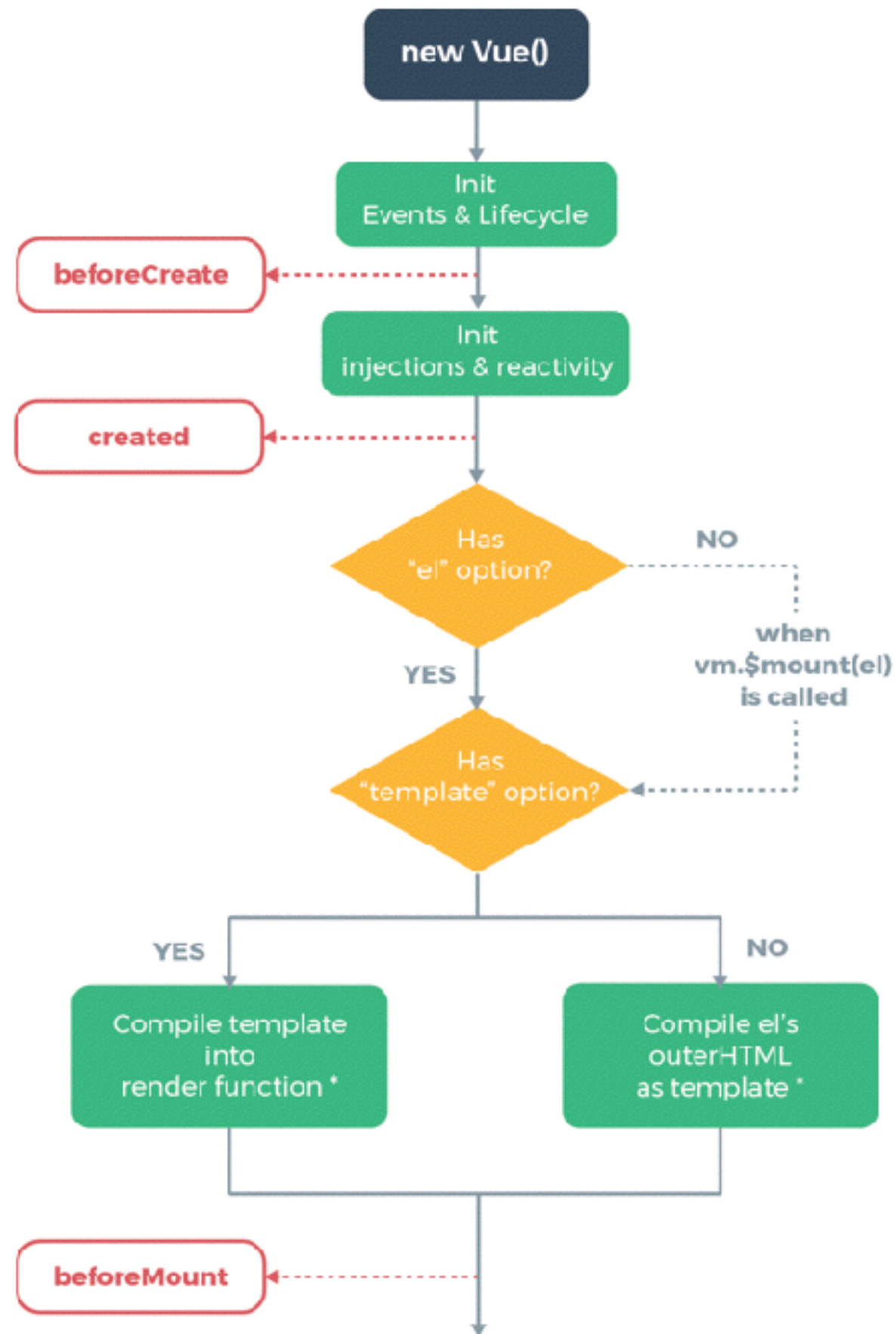
# THE VUE INSTANCE

## ▶ Lifecycle Hooks

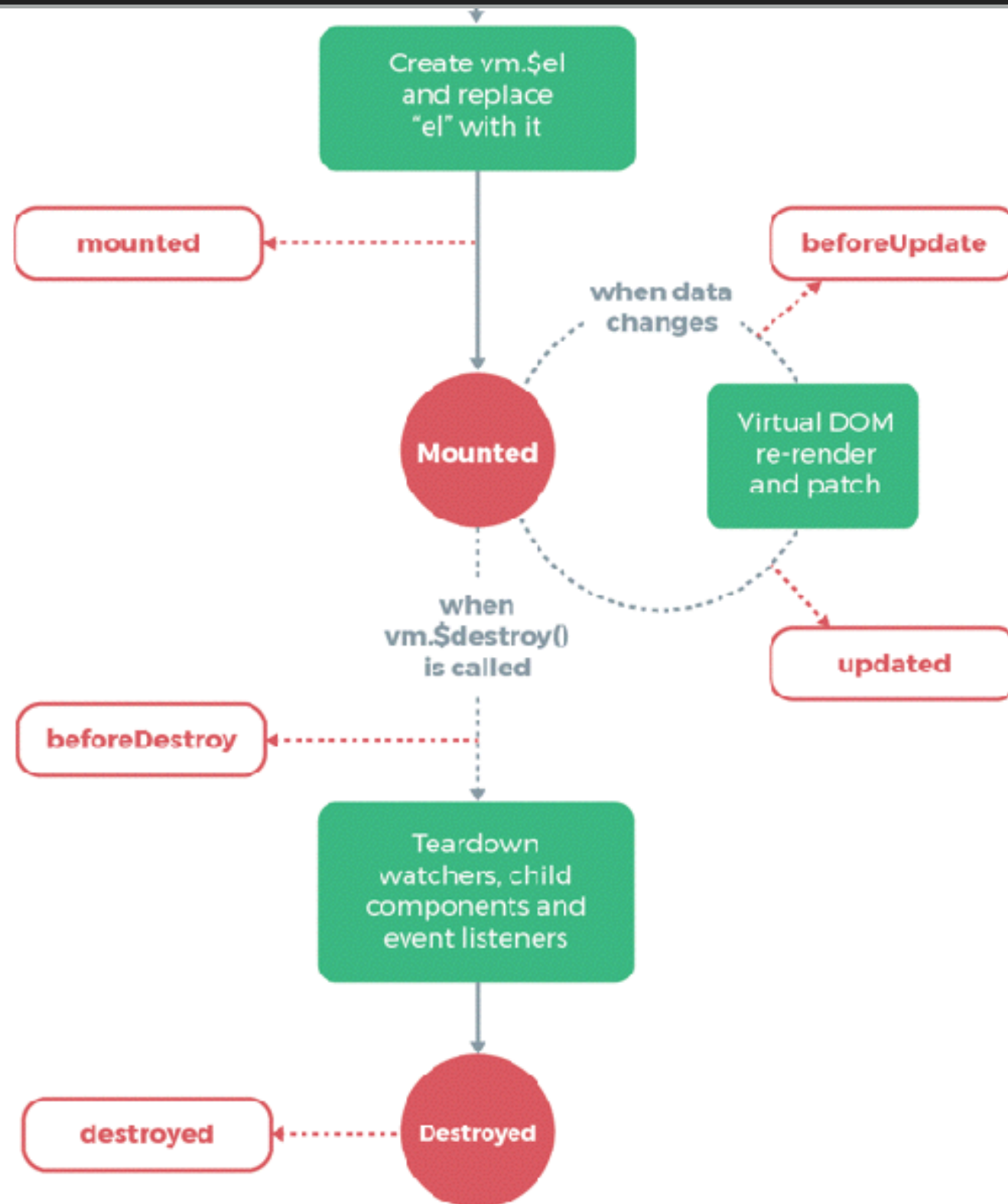
- ▶ Functions that are called at specific points in the app/  
component lifecycle

```
created: function () {  
  fetch(`${this.$data.serverURL}/application`)  
    .then(r => r.json())  
    .then(json => {  
      this.serverInfo = json  
    })  
},
```

# ROOM WITH A VUE



# ROOM WITH A VUE



\* template compilation is performed ahead-of-time if using a build step, e.g. single-file components

# TEMPLATES

- ▶ HTML based Templating syntax
- ▶ “Mustache” `{{}}` syntax for string interpolation and dynamic variables
- ▶ Directives
  - ▶ prefixed with “**v-**”
  - ▶ accept **arguments** separated by “**:**”
  - ▶ support **modifiers** prefixed with “**.**”
- ▶ Rendered via a virtual DOM, for efficient updates

# TEMPLATES

```
<template>
  <div class="form">
    <div class="title cell">
      <label>Title</label>
      <input ref="bookTitle" v-model="book.title" type="text"/>
    </div>
    <div class="pages cell">
      <label>Pages</label>
      <input v-model="book.pages" type="text"/>
    </div>
    <div class="author cell">
      <label>Author</label>

      <select v-model="book.author">
        <option disabled selected value="">Choose Author</option>
        <option v-if="author !== null"
          v-bind:value="{ id: author.id }"
          v-for="author in authors">{{author.name}}</option>
      </select>
    </div>
    <div class="save cell">
      <button @click="submitNewBook()" >Add Book</button>
    </div>
  </div>
</template>
```



# DIRECTIVES

- ▶ Vue-specific element attributes
- ▶ Add control logic, behavior & data-binding to elements
- ▶ You can write your own directives

```
<option v-if="author !== null" >{{author.name}}</option>
```

```
<li v-for="book in books" >{{book.title}}</li>
```

```
<select v-model="book.author">
```

### Shorthand

```
<input v-bind:value="{ id: author.id }"/> OR <input :value="{ id: author.id }"/>
```

```
<button v-on:click="save()">Save</button> OR <button @click="save()">Save</button>
```

## COMPONENTS

- ▶ Nested component hierarchy
- ▶ Each component renders either a template or returns createElement() calls (JSX is supported)
- ▶ Components typically defined as **Single File Component** with **<template>**, **<script>**, and **<style>** sections

# SINGLE FILE COMPONENTS

(\* .vue files)

```
<template>
  <div class="hello">
    <h1>{{ msg }}</h1>
  </div>
</template>
```

Template Section

```
<script>
export default {
  name: 'HelloWorld',
  data () {
    return {
      msg: 'Welcome to Your Vue.js App'
    }
  }
}
</script>
```

Script Section

```
<style scoped>
h1 {
  font-color: blue;
}
</style>
```

Style Section

DEMO

## ECOSYSTEM



- ▶ **Vue CLI**
- ▶ **Vuex** - State Management
- ▶ **Vue Router**
- ▶ **Vue SSR** - Server-Side Rendering
- ▶ Devtools
- ▶ UI Libraries:
  - ▶ **Bootstrap-Vue** - Twitter Bootstrap (v4)
  - ▶ **Vuetify** - Material UI

## VUE VS ANGULAR



- ▶ A cleaner, simpler way to build apps
- ▶ Similar concepts/syntax: directives, templates, v-model
- ▶ More understandable, readable code
- ▶ A rich, featureful API that you can grow into
- ▶ Not a framework - you'll still need to add packages

## VUE VS REACT



- ▶ Adds a bit more magic
- ▶ Similar conceptually: components, virtual DOM, lifecycle methods
- ▶ Supports JSX, createElement API, render functions
- ▶ Less code... sometimes
- ▶ Favors pragmatism over “pure” simplicity

# SUMMARY

- ▶ Vue.js is not “just” another JavaScript framework
- ▶ Aims for balance between rich, developer-friendly features and clean, understandable code
- ▶ Community is vibrant and engaged, and shows every sign of being in this for the long haul
- ▶ Vue.js is definitely worth learning, either as a first-timer’s JavaScript framework or for developers looking for a fresh view of SPAs



# LINKS

- ▶ Official website: <http://vuejs.org>
- ▶ Awesome Vue: <https://github.com/vuejs/awesome-vue>
- ▶ Comparison: <https://medium.com/unicorn-supplies/angular-vs-react-vs-vue-a-2017-comparison-c5c52d620176>
- ▶ Interview with Evan You: <https://medium.freecodecamp.org/between-the-wires-an-interview-with-vue-js-creator-evan-you-e383cbf57cc4>

# LINKS

- ▶ Vue Profile docs: <https://grails-profiles.github.io/vue/latest/guide/>
- ▶ Building a Vue App: <http://guides.grails.org/building-a-vue-app/guide>
- ▶ Using the Vue Profile: <http://guides.grails.org/using-the-vue-profile/guide>
- ▶ Grails/React Quickcast: <https://www.youtube.com/watch?v=JVLn7kWkjp0>



# THANK YOU

Slides/Sample Code: <https://github.com/ZacharyKlein/room-with-a-vue-midwestjs>



Twitter: **@ZacharyAKlein**. Email: [kleinz@objectcomputing.com](mailto:kleinz@objectcomputing.com)

August 9, 2018