I'm not angry, I'm happiness challenged.

I DON'T UNDERSTAND REACT, VUE.JS, ANGULAR

AND AT THIS POINT I'M TOO AFRAID TO ASK

makeameme.org

Module 3-11

VUE Component Communication

Objectives

- Identify candidate components in a web page/app
- Explain SRP (Single Responsibility Principle)
- Implement a simple SPA (Single Page Application)
- Implement a child component that is passed data from a parent using props
- Create child components from an array in a parent component
- Access data stored in Vuex from multiple components
- Modify data stored in Vuex using mutations

Components in Vue

- A component is a custom element in Vue. They help you extend basic HTML elements to encapsulate reusable code.
- Allows you to break your app into small, reusable pieces which you can then use to build more complicated apps with.
- Just like encapsulation!

- SRP – Single Responsibility Principle – each component of an app should handle only one job.
  - Limit functionality as much as possible

Working With Multiple Components

- So far we have dealt with a relatively simple project setup, with a single component doing all the hard work, then importing that component into the App.vue parent component.
- The whole idea of component based JS development is that we can take several components and integrate them together under a parent component.
  - This requires us to learn some additional techniques as individual components are not aware of each other.
- Namely, we will be covering how to transmit data from a parent component to a child component and vice versa.

Importing Components

- Three step process
  1. Import the component in the script area
  2. Declare the component in the components object
  3. Add the component to your template object in the mark-up

```
<template>
    <add-review></add-review>
</template>




<script>
import AddReview from './components/addReview';
export default {
    components: {
      AddReview
    }
}
</script>
```
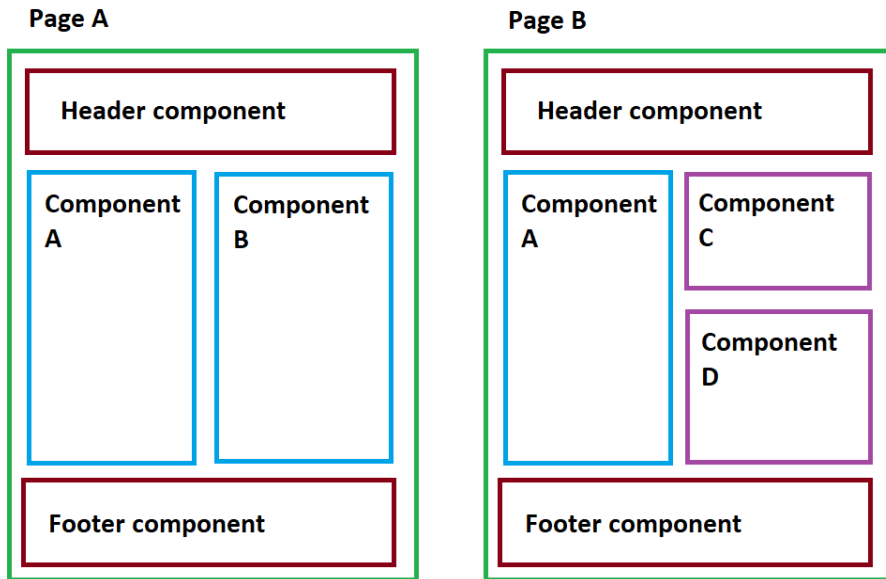
Working With Multiple Components

- Does not change functionality
- Makes app easier to test and more maintainable
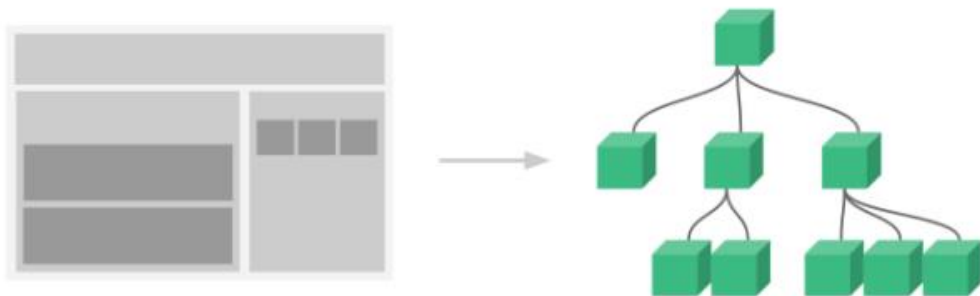
Working With Multiple Components

How to break apart ProductReviews

Product Reviews for Cigar Parties for Dummies

Host and plan the perfect cigar party for all of your squirrelly friends.

2.75	1	0	2	1	0
Average Rating	1 Star Review	2 Star Reviews	3 Star Reviews	4 Star Review	5 Star Reviews

Name:

Title:

Rating:

Review:

Save   Cancel

Malcolm Gladwell
⭐⭐⭐ What a book!

It certainly is a book. I mean, I can see that. Pages kept together with glue and there's writing on it, in some language.

Favorite? ☐

Tim Ferriss
⭐⭐⭐⭐ Had a cigar party started in less than 4 hours.

It should have been called the four hour cigar party. That's amazing. I have a new idea for muse because of this.

Favorite? ☐

Ramit Sethi
⭐ What every new entrepreneurs needs. A door stop.

When I sell my courses, I'm always telling people that if a book costs less than $20, they should just buy it. If they only learn one thing from it, it was worth it. Wish I learned something from this book.

Favorite? ☐

Gary Vaynerchuk
⭐⭐⭐ And I thought I could write

There are a lot of good, solid tips in this book. I don't want to ruin it, but prelighting all the cigars is worth the price of admission alone.

Favorite? ☐

Single Responsibility Principle

Displaying Counts & Averages

Managing the Add Review Form

Filtering Reviews

Displaying Review Information

Display product information

Letting the user filter on star count

Adding Reviews

Marking Reviews as Favorited

Product Reviews for Cigar Parties for Squirrels

Because squirrels LOVE their cigars

NaN	0	0	0	0	0
Average Rating	1 Star Reviews	2 Star Reviews	3 Star Reviews	4 Star Reviews	5 Star Reviews

Name:

Title:

Rating:

Review:

Save   Cancel

Malcolm Gladwell

★★★ What a book!

It certainly is a book. I mean, I can see that. Pages kept together with glue and there's writing on it, in some language.

Favorite? ☐

Tim Ferriss

★★★★ Had a cigar party started in less than 4 hours.

It should have been called the four hour cigar party. That's amazing. I have a new idea for muse because of this.

Favorite? ☐

Ramit Sethi

★ What every new entrepreneurs needs. A door stop.

When I sell my courses, I'm always telling people that if a book costs less than $20, they should just buy it. If they only learn one thing from it, it was worth it. Wish I learned something from this book.

Favorite? ☐

Gary Vaynerchuk

★★★ And I thought I could write

There are a lot of good, solid tips in this book. I don't want to ruin it, but prelighting all the cigars is worth the price of admission alone.

Favorite? ☐

Identifying Components

Props

- Vue has many properties, data, computed, methods and now props. Props is how a component can be given access to data from outside of that component.

- We will be using props to transmit data **from a parent component to a child component**.

Props Defining

- Within a child component, props are defined in JSON format within the script section.

- It is a peer of the other sections we've seen so far: data(), computed, methods.

```
<script>
export default {
    name: "product-review",
    props() {
      // your props go here
    },
    data() {
            ...
    },
    computed: {
            ...
    },
    methods: {
            ...
    }
}
</script>
```

Properties Example

Consider the following example, where we utilize props to send data from a parent component to a child component:

On App.VUE (or parent component)

```
<template>
  <div id="app">
...
    <product-review
      name="Toothbrush"
      description="To brush teeth"
       />
...
  </div>
</template>
```

Child Component (script)

```
props: {
  name: String,
  description: String,
},
```

The child component can now use these props, for instance having **{{ name }}** on the template would result in *Toothbrush* being rendered.

Components for Product Reviews

Product Reviews for Cigar Parties for Dummies

Host and plan the perfect cigar party for all of your squirrelly friends.

AverageSummary

2.75	1	0	2	1	0
Average Rating	1 Star Review	2 Star Reviews	3 Star Reviews	4 Star Review	5 Star Reviews

StarSummary

AddReview

Name:

Title:

Rating:

Review:

Save   Cancel

ReviewList

ReviewDisplay

Malcolm Gladwell
⭐⭐⭐ What a book!

It certainly is a book. I mean, I can see that. Pages kept together with glue and there's writing on it, in some language.

Favorite? ☐

Tim Ferriss
⭐⭐⭐⭐ Had a cigar party started in less than 4 hours.

It should have been called the four hour cigar party. That's amazing. I have a new idea for muse because of this.

Favorite? ☐

Ramit Sethi
⭐ What every new entrepreneurs needs. A door stop.

When I sell my courses, I'm always telling people that if a book costs less than $20, they should just buy it. If they only learn one thing from it, it was worth it. Wish I learned something from this book.

Favorite? ☐

Gary Vaynerchuk
⭐⭐⭐ And I thought I could write

There are a lot of good, solid tips in this book. I don't want to ruin it, but prelighting all the cigars is worth the price of admission alone.

Favorite? ☐

Let's Look at the Lecture Code!

Sharing data across multiple components

- Best to have data live in central location
    - Separate from components that use it
    - Datastore
        - State
        - View
        - Actions
- Components read from store
    - Not keep own copy of data
- Only one store for each application
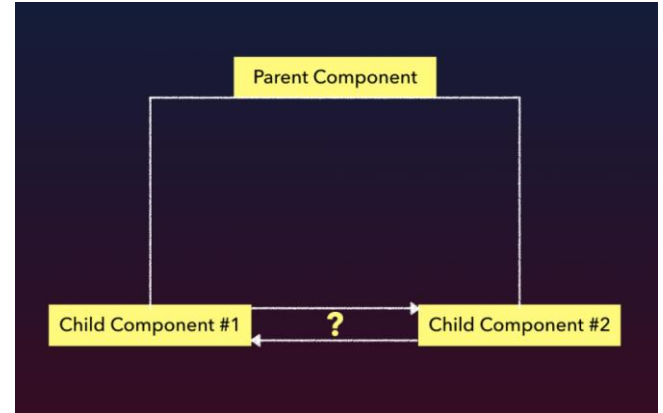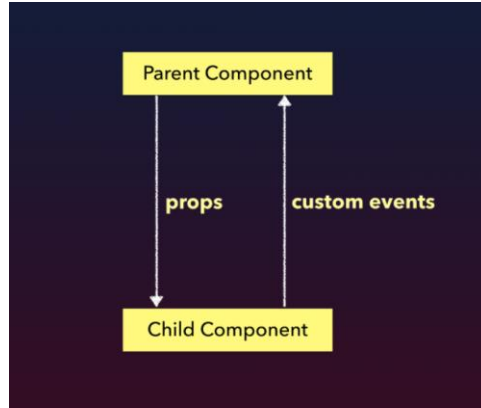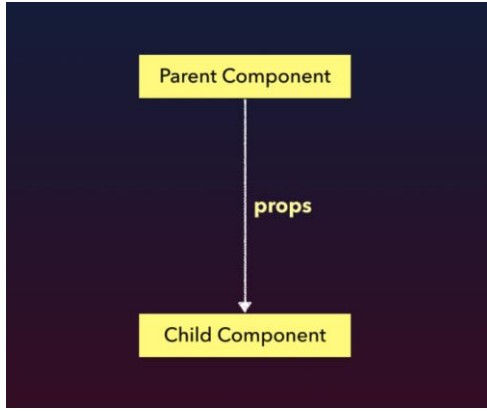   - Single source of truth
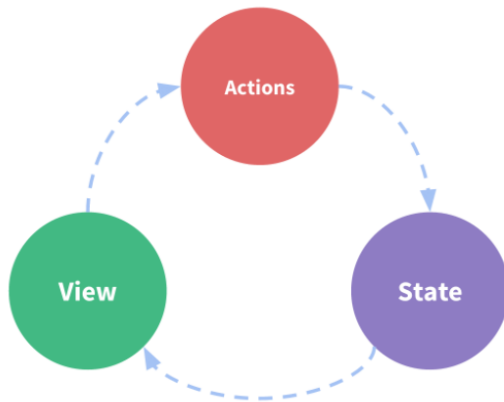
State and State management

- State of data
    - Data shared across components
    - Components receive this data
    - When data changes, component will update itself
    - Single source of truth in app
- State management
    - Managing data across components of an app
    - Store pattern where all actions that change state are in single stored class
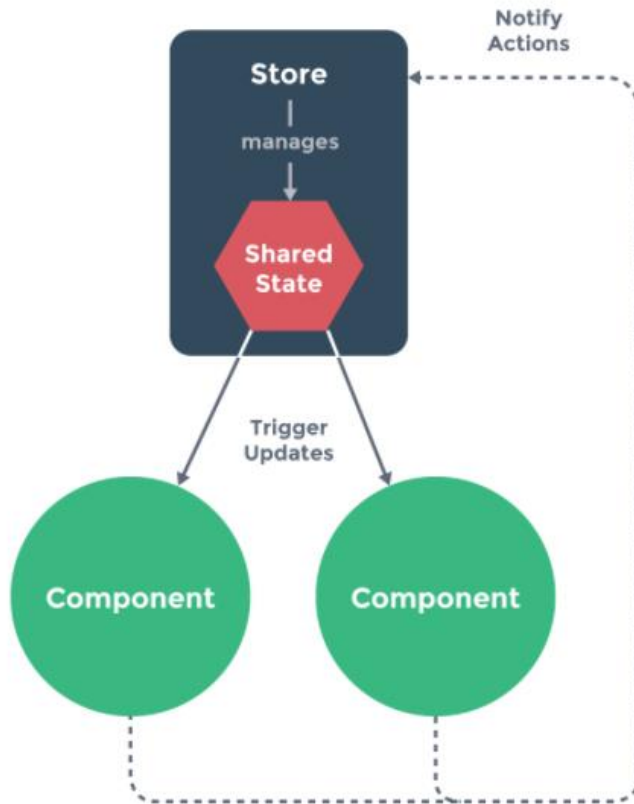   - Library such as Vuex makes this easy!

Sharing data across components

Vuex Store

Creating a Vuex Store

- During project creation, add Vuex.

- Add vuex to already created project

Vuex Store

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

Vue.use(Vuex)

export default new Vuex.Store({
  state: {
  },
  mutations: {
  },
  actions: {
  },
  modules: {
  }
})
```

Defining State

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

Vue.use(Vuex)

export default new Vuex.Store({
  state: {
    name: 'Cigar Parties for Dummies',
    description: 'Host and plan the perfect cigar party for all of your squirre
    filter: 0,
    reviews: [
      {
        reviewer: 'Malcolm Gladwell',
        title: 'What a book!',
        review:
          "It certainly is a book. I mean, I can see that. Pages kept together
            language.",
        rating: 3,
        favorited: false
      },
```

Accessing State

```html
<template>
  <div id="app" class="main">
    <h1>Product Reviews for {{ $store.state.name }}</h1>
    <p class="description">{{ $store.state.description }}</p>
    <div class="well-display">
      <average-summary />
      <star-summary rating="1" />
      <star-summary rating="2" />
      <star-summary rating="3" />
```

Moving State to Vuex with Mutations

```javascript
1 import Vue from 'vue'
2 import Vuex from 'vuex'
3
4 Vue.use(Vuex)
5
6 export default new Vuex.Store({
7   // State contains the global application state. Think of it as app-wide data
8   state: {
9     isAddQuestionVisible: false,
10   },
11
12   // Mutations are used to make discrete changes to state from a central place
13   mutations: {
14     SHOW_ADD_QUESTION(state) {
15       state.isAddQuestionVisible = true;
16     },
17     HIDE_ADD_QUESTION(state) {
18       state.isAddQuestionVisible = false;
19     },
20   }
21 })
22
```

Calling Mutations from Component

```
1 export default {
2   // Methods contain functions which can be invoked from event handlers or other code
3   methods: {
4     /**
5      * Shows the add question form.
6      */
7     showAddQuestionForm() {
8       this.$store.commit('SHOW_ADD_QUESTION');
9     },
10   },
11   // Computed contains the Vue.js equivalent of getters that rely on other data to compute results.
12   // Think of computed as computed or auto-calculated properties that are bound to from the template.
13   computed: {
14     isAddQuestionVisible() {
15       return this.$store.state.isAddQuestionVisible;
16     }
17   }
18 };
```
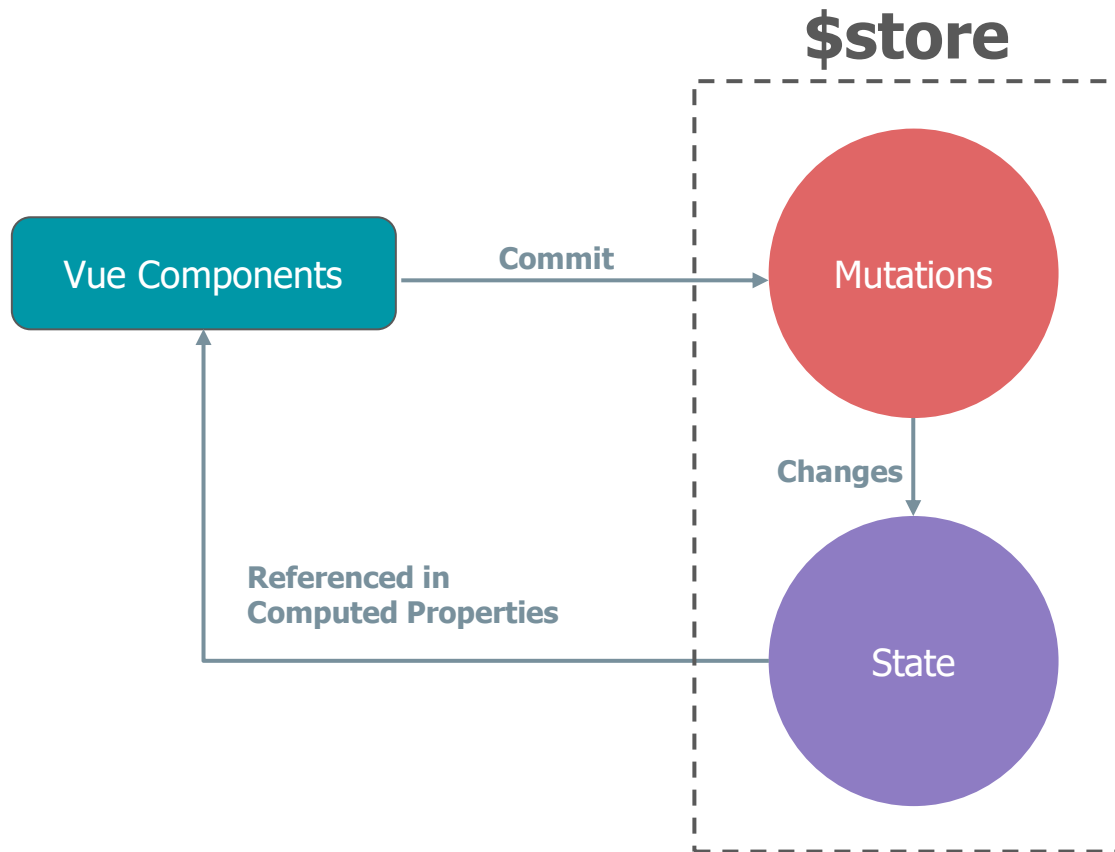
Sending parameters from Component to Store

```
1 export default {
2   // Methods contain functions which can be invoked from event handlers or other code
3   methods: {
4     someMethod() {
5       // Call the SOME_MUTATOR_NAME mutator with a payload of 42
6       this.$store.commit('SOME_MUTATOR_NAME', 42);
7     },
8   },
9   // Other members omitted...
10 };
```

Receiving parameters from Component

```
1 export default new Vuex.Store({
2   // State contains the global application state. Think of it as app-wide data
3   state: {
4     isAddQuestionVisible: false,
5     someVariable = 0,
6   },
7
8   // Mutations are used to make discrete changes to state from a central place
9   mutations: {
10     SHOW_ADD_QUESTION(state) {
11       state.isAddQuestionVisible = true;
12     },
13     HIDE_ADD_QUESTION(state) {
14       state.isAddQuestionVisible = false;
15     },
16     SOME_MUTATOR_NAME(state, payload) {
17       state.someVariable = payload;
18     },
19   },
20 })
```

Big Picture

Component

```
1  export default {
2    // Methods contain functions which can be invoked f
3    methods: {
4      /**
5       * Shows the add question form.
6       */
7      showAddQuestionForm() {
8        this.$store.commit('SHOW_ADD_QUESTION');
9      },
10   },
11   // Computed contains the Vue.js equivalent of gette
12   // Think of computed as computed or auto-calculate
13   computed: {
14     isAddQuestionVisible() {
15       return this.$store.state.isAddQuestionVisible;
16     }
17   }
18 };
```

Vuex

```
1  export default new Vuex.Store({
2    // State contains the global application state. Think of it as app-wide data
3    state: {
4      isAddQuestionVisible: false,
5    },
6
7    // Mutations are used to make discrete changes to state from a central place
8    mutations: {
9      SHOW_ADD_QUESTION(state) {
10       state.isAddQuestionVisible = true;
11     },
12     HIDE_ADD_QUESTION(state) {
13       state.isAddQuestionVisible = false;
14     },
15   },
16 })
```

Big Picture

Component

```
 1 export default {
 2   // Methods contain functions which can be invoked f
 3   methods: {
 4     /**
 5      * Shows the add question form.
 6      */
 7     showAddQuestionForm() {
 8       this.$store.commit('SHOW_ADD_QUESTION');
 9     },
10   },
11   // Computed contains the Vue.js equivalent of gette
12   // Think of computed as computed or auto-calculated
13   computed: {
14     isAddQuestionVisible() {
15       return this.$store.state.isAddQuestionVisible;
16     }
17   }
18 };
```

Vuex

```
 1 export default new Vuex.Store({
 2   // State contains the global application state. Think of it as app-wide data
 3   state: {
 4     isAddQuestionVisible: false,
 5   },
 6
 7   // Mutations are used to make discrete changes to state from a central place
 8   mutations: {
 9     SHOW_ADD_QUESTION(state) {
10       state.isAddQuestionVisible = true;
11     },
12     HIDE_ADD_QUESTION(state) {
13       state.isAddQuestionVisible = false;
14     },
15   },
16 })
```

Immutability

```
1 export default new Vuex.Store({
2   // When strict is set to true, state can ONLY be mutated from the store. This should be true.
3   strict: true,
4
5   // State contains the global application state. Think of it as app-wide data
6   state: {
7     // Omitted...
8   },
9
10   // Mutations are used to make discrete changes to state from a central place
11   mutations: {
12     // Omitted...
13   },
14 })
```

Let's Code!