Vue Essentials & Fritter Frontend

6.1040 Recitation 8
Fall 2022

Look at all these options I don't understand

Today’s goals

1. Review this week’s lecture content about Vue & Introduce new Vue concepts
2. Familiarize with the Fritter starter code

…in parallel

What we’re covering

Review of lecture & expanding on it:

- Components
- Lifecycle
- Data and methods
- Binding and event handling (v-bind, v-on, v-model)
- List rendering (v-for) and conditionals (v-if and v-else)
- Props (parent to child)
- this.$emit (child to parent)
- Store

New:

- Mixins

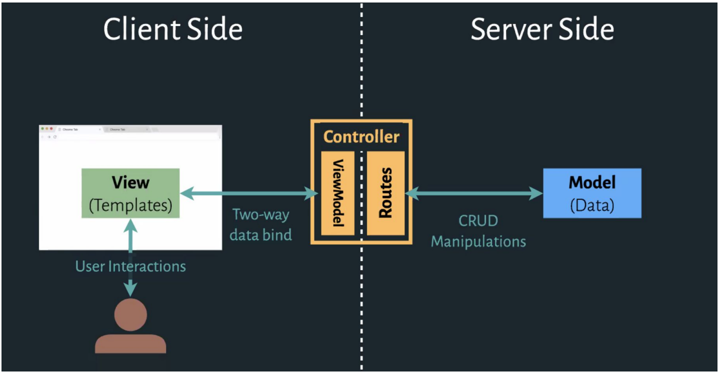
Quick demo of the Fritter frontend

http://localhost:8080/#/

Application Structure

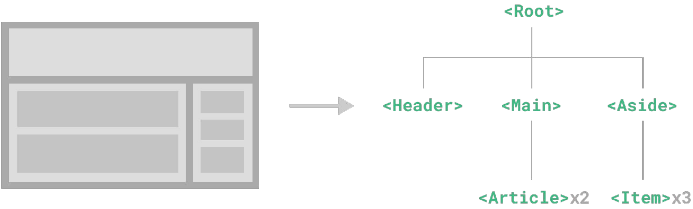
File structure

`/client` and `/server` directories correspond to Client and Server side



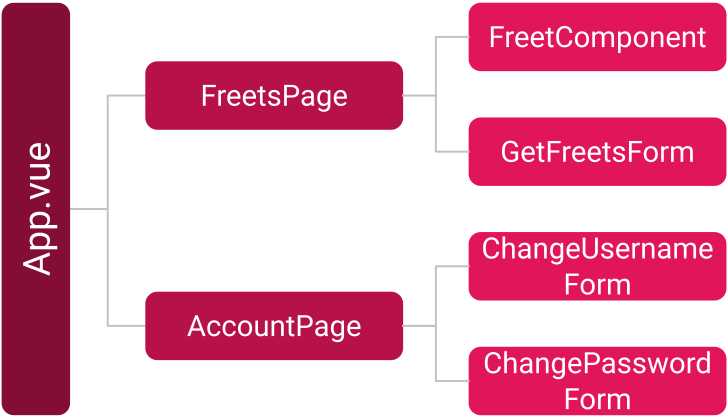
Components

Vue components group by functionality instead of separating your template, logic, and styling code (HTML/CSS/JS) into separate files



Components are organized as a tree

Diagram of Fritter (incomplete)



Putting the app on the page

Everything is inserted into the page via an HTML container

index.html:

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

main.ts:

```
app.mount('#app')
```

Component Essentials

Component Lifecycle

https://v2.vuejs.org/v2/guide/instance.html#Lifecycle-Diagram

For our purposes:

- `created()`
- `mounted()`
- `unmount()`

Examples:

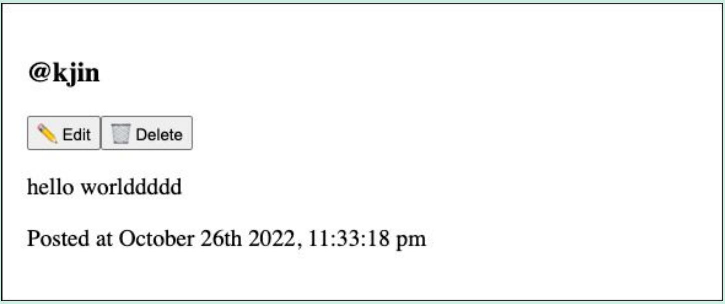
App.vue `beforeCreate()`

FreetsPage.vue `mounted()`

Data

- `data()` is a function
- `computed` properties *depend* on data
- When these values change, Vue **reacts** by updating the dependent components based on the new values

Example: FreetComponent.Vue



@kjin

✏️ Edit 🗑️ Delete

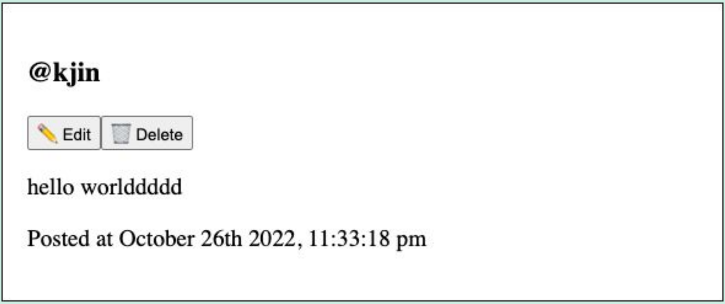
hello worlddddd

Posted at October 26th 2022, 11:33:18 pm

Methods

- `methods` is an object containing desired methods for the component
- Vue automatically binds the `this` value for methods

Example: FreetComponent.Vue



@kjin

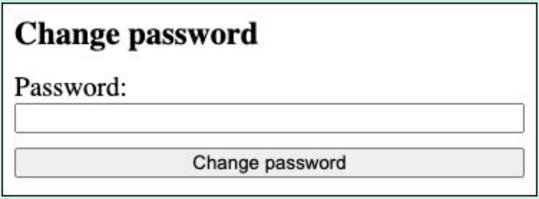
hello worlddddd

Posted at October 26th 2022, 11:33:18 pm

Attribute binding

- Directive: **`v-bind`**
- Shorthand: **`:`**
- Used to define an attribute on an element
  - `v-bind:class`
  - `v-bind:style`
  - `v-bind:src`
  - etc.

Example: BlockForm.vue



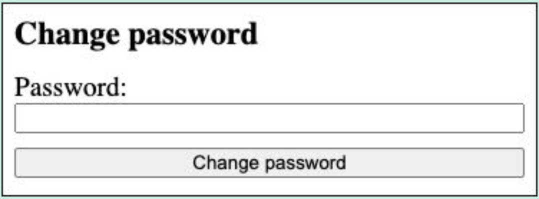
Change password

Password:

Event Handling

- Directive: **v-on**
- Shorthand: @
- Used to attach an event listener to the element

Example: BlockForm.vue



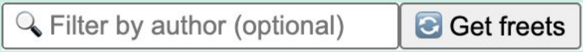
Change password

Password:

Form Input Bindings

- Directive: **`v-model`**
- Shorthand: **none**
- can be used on inputs of different types
- Two-way synchronizes the state of form input elements with the corresponding state

Example: InlineForm.vue



Summary of directives

- **`v-on aka @`** - attach an event listener to the element.
- **`v-bind aka :`**
  - *one-way* binding
  - Two uses: assign values for built-in HTML attributes OR a component prop to an expression
- **`v-model`**
  - *two-way* binding on a form input element or a component
  - Equivalent to combining **v-bind** and **v-on:input**

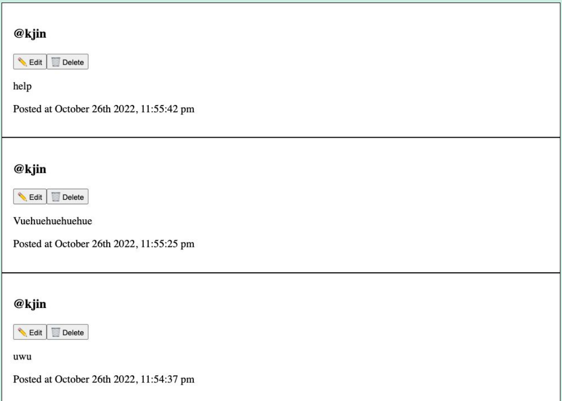
These two lines are equivalent:

```
<input

  v-bind:value="text"

  v-on:input="event => text =
event.target.value">


<input v-model="text">
```

List Rendering

- Directive: **`v-for`**
- Use with **key** attribute

Example: FreetsPage.vue



@kjin

help

Posted at October 26th 2022, 11:55:42 pm

@kjin

Vuehuehuehuehue

Posted at October 26th 2022, 11:55:25 pm

@kjin

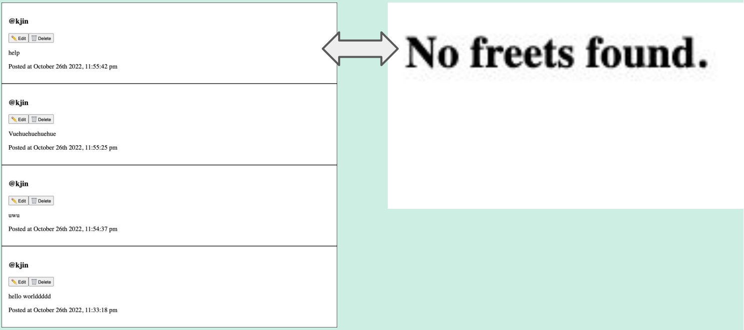
uwu

Posted at October 26th 2022, 11:54:37 pm

Conditional Rendering

- Directive: **`v-if`**, **`v-else-if`**, **`v-else`**
- Shorthand: **none**

Example: FreetsPage.vue



@kjin

help

Posted at October 26th 2022, 11:55:42 pm

@kjin

Vuehuehuehuehue

Posted at October 26th 2022, 11:55:25 pm

@kjin

uwu

Posted at October 26th 2022, 11:54:37 pm

@kjin

hello worldddddd

Posted at October 26th 2022, 11:33:18 pm

No freets found.

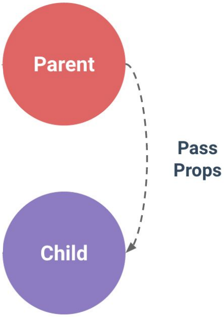
Small exercise:

Disable the Submit Edit button if there are no changes to the Freet content.

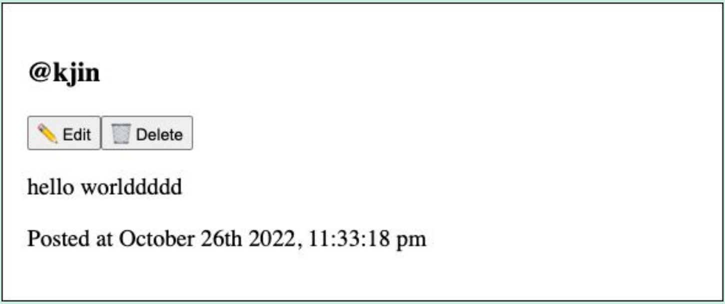
State management

Props: Parent to Child

Props are data that are passed from a parent component to child components



Example: FreetComponent.Vue

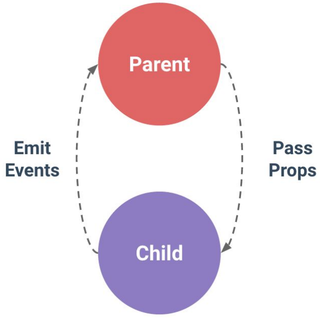

@kjin

hello worlddddd

Posted at October 26th 2022, 11:33:18 pm

$emit: Child to Parent

A component can “emit” custom events (`this.$emit`) with data, and (only) the parent can listen for the event



Example:

Child component:

```
<button @click="$emit('someEvent')">click me</button>
```

Parent component:

```
<MyComponent @some-event="callback" />
```

Store

- Use store when we need multiple components that share a common state
- We store this common state in a global “store”

Example: App.vue

When to use props and this.$emit vs. when to use store?

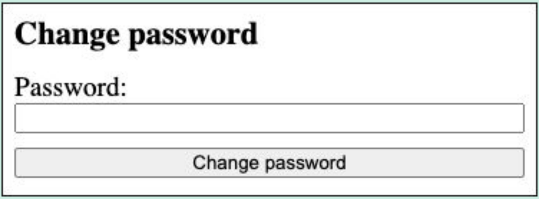
- Is there a direct hierarchical relationship between my parent and child components that store and use this prop?
  - If yes then use props/events
- How long must the data persist? If you’re only using this data for one form and then never needing it again, try to use props/events
- Do you need the same data across many “pages”?
  - Then use Vuex
- Are many components sharing and updating the same data object?
  - If yes, use Vuex

New Topics

Mixins

- Used for reusable logic
- Kind of like inheriting classes - e.g. “Dog” inherits “Animal”

Example: BlockForm.vue



Change password

Password:

Concepts in lecture

- Template syntax (e.g. `{{message}}`)
- Slots
- Scoped CSS
- Router

Reminder: we are using **Vue 2** not Vue 3

(You might see it as Options API vs Composition API)