

Да что такого в этом Vue.js?

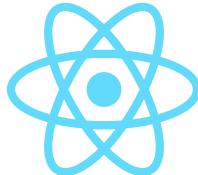


Github stars



★ Star

121,869



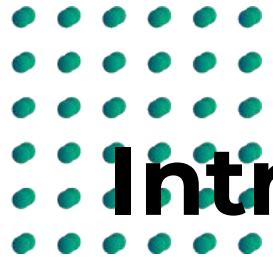
★ Star

117,437



★ Star

43,442

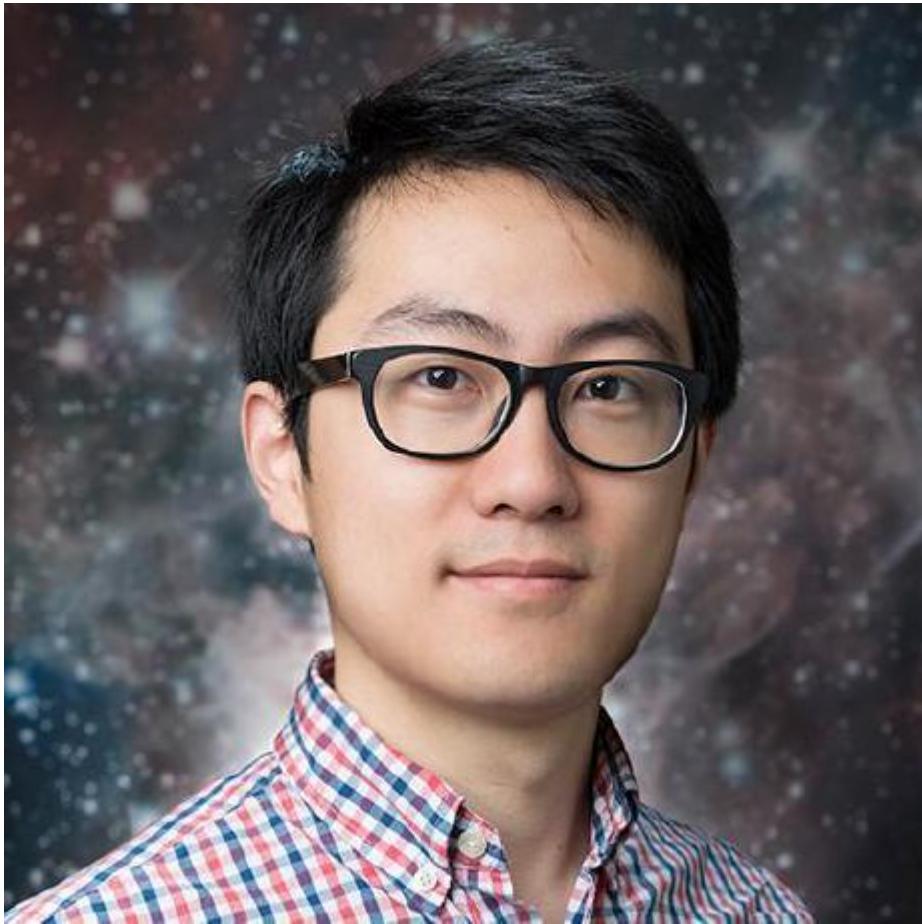


Introduction



Evan You

Vue was first released in February 2014 by Evan You after working for Google using AngularJS in a number of projects





NETFLIX



9GAG



Xiaomi





euronews.

LARAC ➔ STS



GitLab



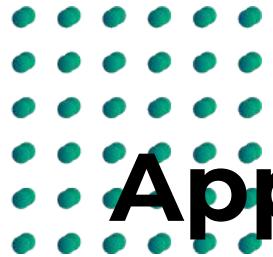
ІНТЕРНЕТ-ГІПЕРМАРКЕТ





What Vue is about:

- Reactive & composable components
- Virtual DOM
- Predictable & clear data flow
- All-inclusive framework
- Simplicity & Lowest learning curve as of 2018



App Mounting

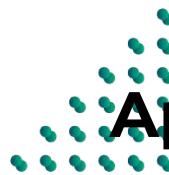




App Mounting



```
63 // main.js
64 import Vue from 'vue';
65 import App from '@/components';
66
67 const app = new Vue({
68   el: '#app',
69   components: {
70     App,
71   },
72 });
73
```



App Mounting



```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Document</title>
8  </head>
9  <body>
10     <div id="app">
11         <App/>
12     </div>
13     <script src="/js/main.js"></script>
14 </body>
15 </html>
```



App Mounting

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Document</title>
8  </head>
9  <body>
10     <div id="app">
11         <App>
12             <AppHeader/>
13             <AppContent>
14                 <Sidebar/>
15                 <ContentBlock>
16                     <h1>{{ content.welcome }}</h1>
17                     <p>{{ content.description }}</p>
18                     <button v-on:click="subscribeToUpdates">
19                         Subscribe to our updates
20                     </button>
21                 </ContentBlock>
22             </AppContent>
23             <AppFooter/>
24         </App>
25     </div>
26     <script src="/js/main.js"></script>
27 </body>
28 </html>
```

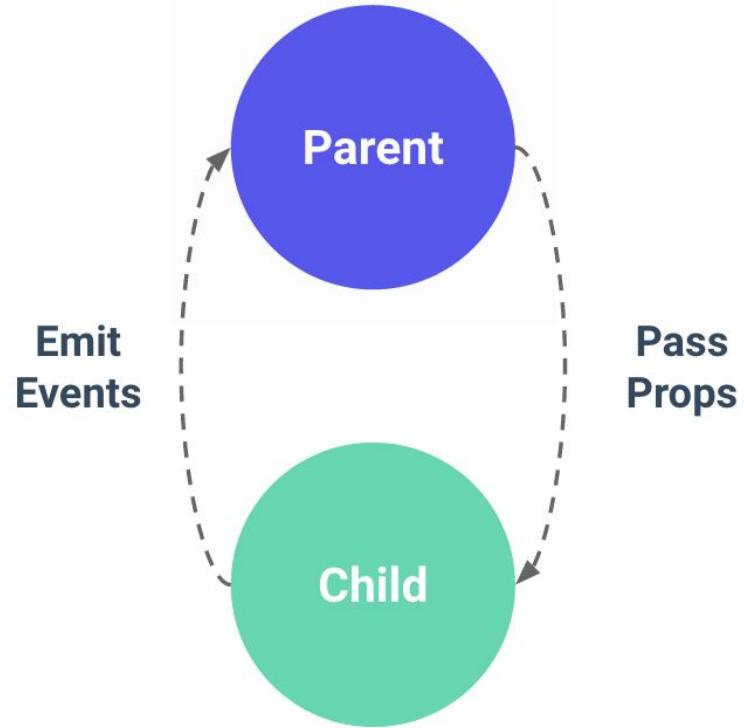


Parent Child Relationships





Parent Child Relationships





Parent-Child Relationships

```
<template>
  <div class="parent-component">
    <Child v-on:nameChange="handleNameChange" v-bind:name="parentData.name" />
  </div>
</template>
```



Parent-Child Relationships

```
<template>
  <div class="child-component">
    <input v-on:input="handleInput" v-bind:value="value" type="text">
  </div>
</template>

<script>
export default {
  name: 'Child',
  props: {
    name: String,
  },
  methods: {
    handleInput(e) {
      this.$emit('nameChange', e.target.value);
    },
  },
};
</script>
```

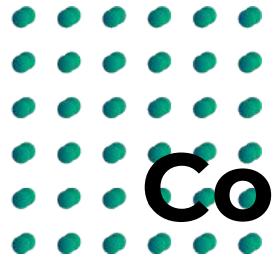


Parent-Child Relationships in React

```
class Parent extends React.Component {
  state = {
    name: ''
  };
  handleNameChange = (event) => {
    this.setState({
      name: event.target.value,
    });
  };
  render() {
    return (
      <div>
        <Child name={this.state.name}
              onNameChange={this.handleNameChange}>
        </Child>
      </div>
    );
  }
};
```

Parent-Child Relationships in React

```
class Parent extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      name: '',
    };
    this.handleNameChange = this.handleNameChange.bind(this);
  }
  handleNameChange(event){
    this.setState({
      name: event.target.value,
    });
  };
  render() {
    return (
      <div>
        <Child name={this.state.name}
              |   onNameChange={this.handleNameChange}>
        </Child>
      </div>
    );
  };
}
```



Component



Component - Single File Component

```
1 <template>
2   <div class="product">
3     <div class="product__content">
4       <div class="product__image-wrapper">
5         
6       </div>
7       <div class="product__name">{{ name }}</div>
8       <div class="product__price">{{ price }}</div>
9     </div>
10   </div>
11 </template>
12
13 <script>
14   export default {
15     name: 'ProductCard',
16     props: {
17       id: [String, Number],
18       name: String,
19       price: Number,
20       image: String,
21     },
22   };
23 </script>
24
25 <style lang="scss">
26   // here goes your CSS
27 </style>
28
29
30
```

Component - Directives

```
2 <template>
3   <div class="product">
4     <div class="product__content">
5       <div class="product__image-wrapper">
6         <img v-bind:image="image"
7               v-bind:alt="` ${name} product image`"
8               class="product__image">
9       </div>
10      <div class="product__name">{{ name }}</div>
11      <div class="product__price">{{ price }}</div>
12      <div class="product__quantity">{{ quantity }}</div>
13      <div class="product__purchase-button-wrapper">
14        <button v-if="quantity > 0" v-on:click="handleAddToCart" class="product__purchase-button">
15          Buy
16        </button>
17        <button v-else v-on:click="handlePreorder" class="product__purchase-button">
18          Preorder
19        </button>
20      </div>
21    </div>
22  </div>
23 </template>
```



Component - Directives

```
2 <template>
3   <div :id="'product-' + id" class="product">
4     <div class="product__content">
5       <div class="product__image-wrapper">
6         
9       </div>
10      <div class="product__name">{{ name }}</div>
11      <div class="product__price">{{ price }}</div>
12      <div class="product__quantity">{{ quantity }}</div>
13      <div class="product__purchase-button-wrapper">
14        <button v-if="quantity > 0" @click="handleAddToCart" class="product__purchase-button">
15          Buy
16        </button>
17        <button v-else @click="handlePreorder" class="product__purchase-button">
18          Preorder
19        </button>
20      </div>
21    </div>
22  </div>
23 </template>
```



Component - Template

```
<div :id="'product-' + id" class="product">
  <div class="product__content">
    <div class="product__image-wrapper">
```



Component - Template

Conditional rendering

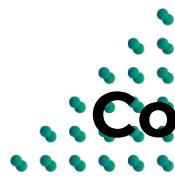
```
<div class="product__purchase-button-wrapper">
  <button v-if="quantity > 0" @click="handleAddToCart" class="product__purchase-button">
    Buy
  </button>
  <button v-else @click="handlePreorder" class="product__purchase-button">
    Preorder
  </button>
</div>
```



Component - Template

Soft conditional rendering

```
<div class="product__purchase-button-wrapper">
  <button v-show="quantity > 0" v-on:click="handleAddToCart" class="product__purchase-button">
    Buy
  </button>
  <button v-show="quantity === 0" v-on:click="handlePreorder" class="product__purchase-button">
    Preorder
  </button>
</div>
```



Component - Template



Regular JS Expressions

```
<div :class="discount ? 'product__price product__price--discounted' : 'product__price'">
  {{ price }}
</div>
```

Map definition

```
<div :class="{ 'product__price--discounted' : discount }" class="product__price">
  {{ price }}
</div>
```



Component - List rendering

```
16    data() {
17      return {
18        products: [
19          {
20            currency: 'USD',
21            id: '1',
22            image: 'http://server.com/public/images/products/blue-suede-shoes.jpg',
23            name: 'Blue Suede Shoes',
24            price: 900,
25            quantity: 10
26          },
27          {
28            currency: 'USD',
29            id: '2',
30            image: 'http://server.com/public/images/products/black-woolen-coat.jpg',
31            name: 'Black Woolen Coat',
32            price: 1400,
33            quantity: 5
34          },
35          {
36            currency: 'EUR',
37            id: '1',
38            image: 'http://server.com/public/images/products/blue-jeans.jpg',
39            name: 'Blue Jeans',
40            price: 500,
41            quantity: 50
42          },
43        ],
44      };
45    };
46  };
47 </script>
```



Component - List rendering

```
1 <template>
2   <div class="products">
3     <div v-for="(product, index) in products"
4       :key="index"
5       class="products__card-wrapper">
6       <ProductCard :id="product.id"
7                     :image="product.image"
8                     :name="product.name"
9                     :price="product.price"
10                    :quantity="product.quantity" />
11     </div>
12   </div>
13 </template>
```



Component - Object props

```
1 <template>
2   <div class="products">
3     <div v-for="(product, index) in products"
4       :key="index"
5       class="products__card-wrapper">
6       <ProductCard v-bind="product" />
7     </div>
8   </div>
9 </template>
```



Component - HTML injection

```
</div>
<p v-html="getProductDescriptionExcerpt(50, '...')" class="product__short-description"></p>
</div>
```

```
26 <script>
27 export default {
28   name: 'ProductCard',
29   props: ['id', 'name', 'image', 'price', 'discount', 'description', 'quantity', 'reviews'],
30   methods: {
31     getProductDescriptionExcerpt(length, ending = '...') {
32       return this.description.length > length
33         ? this.description.slice(0, length) + this.ending
34         : this.description;
35     },
36   }
37 }
```



Component - Event modifiers

```
<!-- prevent default -->
<a @click.prevent="subscribeToUpdates" href="#">Subscribe</a>

<form @submit.prevent="submitForm">
  <input :value="profileForm.name" type="text" name="firstName">
  <button type="button" name="button">Submit</button>
</form>
```



Component - Event modifiers

```
<!-- stopPropagation -->
<div @click="openProductPage" class="product-card">
  <!-- ... -->
  <div @click.stop="addToFavorites" class="product-card__add-to-favs-btn">
    More info
  </div>
</div>
```



Component - Event modifiers

```
<!-- only clicks from this DOM element -->
<div @click.self="closeModal" class="modal">
  <div class="modal__body">
    <div class="modal__content"></div>
    <div class="modal__footer"></div>
  </div>
</div>
```



Component - Event modifiers

```
<!-- only listen to click once -->
<button @click.once="addToFavorites">
  Add to favorites
</button>
```



Component - Event modifiers

```
<!-- click with a certain mouse key -->
<BookCard @click.left="openBookInfo"
           @click.right="openContextMenu"
           @click.middle="openBookInANewTab" />
```



Component - Event modifiers

```
<!-- click + ctrl -->
<CodeSnippet @click.ctrl="copyValueToClipboard"
             :hint="'Click while holding Ctrl key to copy value'" />

<!-- click + ONLY ctrl -->
<CodeSnippet @click.ctrl.exact="copyValueToClipboard"
             :hint="'Click while holding ONLY Ctrl key to copy value'" />
```



Component - Event modifiers

```
<!-- keydown -->
<input @keydown.enter="subscribeToUpdates"
       @input="updateSubscriptionEmail"
       :value="subscriptionForm.email"
       type="email"
       name="email">

<!-- Keyup: Alt + C -->
<input @keyup.alt.67="clear">
```



Available key modifiers:

- .enter
- .tab
- .delete
- .esc
- .space
- .up
- .down
- .left
- .right
- .ctrl
- .alt
- .shift
- .meta



Component - Props

```
25 <script>
26 export default {
27   name: 'ProductCard',
28   props: ['id', 'name', 'image', 'price', 'discount', 'quantity', 'reviews'],
29   methods: {
30     handleAddToCart() {
31       this.$emit('addToCart', this.id);
32     },
33     handlePreorder() {
34       this.$emit('preorder', this.id);
35     }
36   },
37 };
38 </script>
```



Component - Prop Validation

```
25 <script>
26 export default {
27   name: 'ProductCard',
28   props: {
29     id: {
30       type: [String, Number],
31       required: true,
32     },
33     name: {
34       type: String,
35       required: true,
36     },
37     image: String,
38     price: {
39       validator: val => typeof val === 'number' && val >= 0,
40       default: 0,
41     },
42     discount: {
43       validator: val => typeof val === 'number' && val >= 0,
44       default: 0,
45     },
46     quantity: {
47       type: Number,
48       default: 1,
49     },
50     reviews: {
51       type: Array,
52       default: () => [],
53     }
54   },
55 }
```

Component - Mutating data

```
1  <template>
2    <form @submit.prevent="submitFeedback" class="feedback-form">
3      <input @input="(e) => { form.email = e.target.value }" :value="form.email" type="email" name="email">
4      <input @input="(e) => { form.phone = e.target.value }" :value="form.phone" type="number" name="phone">
5
6      <textarea @input="(e) => { form.message = e.target.value }" :value="form.message" name="message">
7      </textarea>
8
9      <button type="submit">Send feedback</button>
10     </form>
11   </template>
12
13 <script>
14 export default {
15   name: 'FeedbackForm',
16   data() {
17     return {
18       form: {
19         email: '',
20         phone: '',
21         message: ''
22       },
23     };
24   },
25   methods: {
26     submitFeedback() {
27       this.$emit('submit', this.form);
28     }
29   }
30 };
31 </script>
```



Component - “2-way binding”

```
2 <template>
3   <form @submit.prevent="submitFeedback" class="feedback-form">
4     <input v-model="form.email" type="email" name="email">
5     <input v-model="form.phone" type="number" name="phone">
6
7     <textarea v-model="form.message" name="message">
8     </textarea>
9
10    <button type="submit">Send feedback</button>
11  </form>
12</template>
```



Reactivity

```
23  data() {
24    return {
25      products: [
26        {
27          currency: 'USD',
28          id: '1',
29          image: 'http://server.com/public/images/products/blue-suede-shoes.jpg',
30          name: 'Blue Suede Shoes',
31          price: 900,
32          quantity: 10
33        },
34        {
35          currency: 'USD',
36          id: '2',
37          image: 'http://server.com/public/images/products/black-woolen-coat.jpg',
38          name: 'Black Woolen Coat',
39          price: 1400,
40          quantity: 5
41        },
42        {
43          currency: 'EUR',
44          id: '1',
45          image: 'http://server.com/public/images/products/blue-jeans.jpg',
46          name: 'Blue Jeans',
47          price: 500,
48          quantity: 50
49        },
50      ],
51    };
52  },
53  methods: {
54    addProduct(productObj) {
55      this.products[this.products.length] = productObj;
56    },
57    updateProduct(productIndex, productObj) {
58      this.products[productIndex] = productObj;
59    },
60  },
61};
```



```
selectedProduct: {
    currency: 'EUR',
    id: '1',
    image: 'http://server.com/public/images/products/blue-jeans.jpg',
    name: 'Blue Jeans',
    price: 500,
    quantity: 50
},
},
methods: {
    changeSelectedProductDiscount(discount) {
        this.selectedProduct.discount = discount;
    },
}
```



```
methods: {
  changeSelectedProductDiscount(discount) {
    this.selectedProduct = Object.assign({}, this.selectedProduct, { discount });
    // OR
    this.selectedProduct = { ...this.selectedProduct, discount };
  },
  addProduct(productObj) {
    this.products.push(productObj);
  },
  updateProduct(productId, productObj) {
    this.products.splice(productId, 1, productObj);
  },
},
```



Reactivity

```
↳ ▼ {...} ⓘ  
  currency: (...)  
  id: (...)  
  image: (...)  
  name: (...)  
  price: (...)  
  quantity: (...)  
► __ob__: Observer {value: {...}, dep: Dep, vmCount: 0}  
► get currency: f reactiveGetter()  
► set currency: f reactiveSetter(newVal)  
► get id: f reactiveGetter()  
► set id: f reactiveSetter(newVal)  
► get image: f reactiveGetter()  
► set image: f reactiveSetter(newVal)  
► get name: f reactiveGetter()  
► set name: f reactiveSetter(newVal)  
► get price: f reactiveGetter()  
► set price: f reactiveSetter(newVal)  
► get quantity: f reactiveGetter()  
► set quantity: f reactiveSetter(newVal)  
► __proto__: Object
```



Component - Reactivity

```
> $vm.products[0].discount = 10;
<- 10
> $vm.products[0]
<- ▼ {...} ⓘ
  currency: (...)

  discount: 10
  id: (...)

  image: (...)

  name: (...)

  price: (...)

  quantity: (...)

  ▶ __ob__: Observer {value: {...}, dep: Dep, vmCount: 0}
  ▶ get currency: f reactiveGetter()
  ▶ set currency: f reactiveSetter(newVal)
  ▶ get id: f reactiveGetter()
  ▶ set id: f reactiveSetter(newVal)
  ▶ get image: f reactiveGetter()
  ▶ set image: f reactiveSetter(newVal)
  ▶ get name: f reactiveGetter()
  ▶ set name: f reactiveSetter(newVal)
  ▶ get price: f reactiveGetter()
  ▶ set price: f reactiveSetter(newVal)
  ▶ get quantity: f reactiveGetter()
  ▶ set quantity: f reactiveSetter(newVal)
  ▶ __proto__: Object
```



Reactivity



```
> $vm.products.splice(0, 1, Object.assign({}, $vm.products[0], { discount: 10 }));
< ▶ [{}]
> $vm.products[0];
< ▼ {...} ⓘ
  currency: ....
  discount: ....
  id: ....
  image: ....
  name: ....
  price: ....
  quantity: ....
  ▶ __ob__: Observer {value: {}, dep: Dep, vmCount: 0}
  ▶ get currency: f reactiveGetter()
  ▶ set currency: f reactiveSetter(newVal)
  ▶ get discount: f reactiveGetter()
  ▶ set discount: f reactiveSetter(newVal)
  ▶ get id: f reactiveGetter()
  ▶ set id: f reactiveSetter(newVal)
  ▶ get image: f reactiveGetter()
  ▶ set image: f reactiveSetter(newVal)
  ▶ get name: f reactiveGetter()
  ▶ set name: f reactiveSetter(newVal)
  ▶ get price: f reactiveGetter()
  ▶ set price: f reactiveSetter(newVal)
  ▶ get quantity: f reactiveGetter()
  ▶ set quantity: f reactiveSetter(newVal)
  ▶ __proto__: Object
```

Computed properties

Search by name:



Blue Suede Shoes

Price: **900 USD**
Quantity: 10

Buy



Black Woolen Coat

Price: **1400 USD**
Quantity: 5

Buy



Blue Jeans

Price: **500 EUR**
Quantity: 50

Buy



Computed properties

```
<template>
  <div class="products">
    <div class="products__filters">
      <div class="products__filter-wrapper">
        <label for="name-search" class="products__filter-label">Search by name:</label>
        <input v-model="nameSearch" id="name-search" type="text" class="products__filter-input">
      </div>
    </div>
    <div class="products__cards">
      <div v-for="(product, index) in filteredProducts"
           :key="index"
           class="products__card-wrapper">
        <Product v-bind="product" />
      </div>
    </div>
  </div>
</template>
```



Source data for computing

```
data() {
  return {
    nameSearch: '',
    products: [
      {
        currency: 'USD',
        id: '1',
        image: '/static/images/shopping-cart-placeholder.png',
        name: 'Blue Suede Shoes',
        price: 900,
        quantity: 10,
      },
      {
        currency: 'USD',
        id: '2',
        image: '/static/images/shopping-cart-placeholder.png',
        name: 'Black Woolen Coat',
        price: 1400,
        quantity: 5,
      },
      {
        currency: 'EUR',
        id: '3',
        image: '/static/images/shopping-cart-placeholder.png',
        name: 'Blue Jeans',
        price: 500,
        quantity: 50,
      },
    ],
  };
},
```



Computed getter

```
computed: {
  filteredProducts() {
    if (this.nameSearch) {
      return this.products.filter(pr => pr.name.toLowerCase().includes(this.nameSearch.toLowerCase()));
    }
    return this.products;
  },
},
```

Computed properties

Search by name:



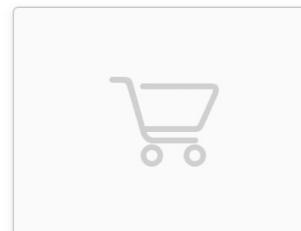
Blue Suede Shoes

Price: **900 USD**

Discounted: **810 USD**

Quantity: **10**

Buy

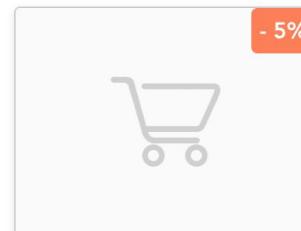


Black Woolen Coat

Price: **1400 USD**

Quantity: **0**

Preorder



Blue Jeans

Price: **500 EUR**

Discounted: **475 EUR**

Quantity: **50**

Buy



Computed properties

```
<div class="product__price">
  <span>Price:</span>
  <strong :class="{ 'product__price-value--discounted': discount }"
    class="product__price-value">
    {{ price }} {{ currency }}
  </strong>
</div>
<div v-if="discount" class="product__price product__price--discount">
  Discounted: <strong class="product__price-value">{{ discountedPrice }} {{ currency }}</strong>
</div>
```



Computed properties

```
computed: {  
  discountedPrice() {  
    return this.price * (1 - (this.discount / 100));  
  },  
},
```



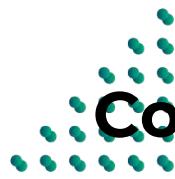
Component Transitions

```
<div class="products__filter-wrapper">
  <transition name="fade" appear>
    <span v-if="filteredProducts.length !== products.length" class="products__filter-label">
      Filtered products: {{ filteredProducts.length }}
    </span>
  </transition>
</div>
```



Component Transitions

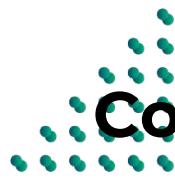
```
<transition-group name="fade" tag="div" class="products__cards">
  <div v-for="product in filteredProducts"
    :key="product.id"
    class="products__card-wrapper">
    <Product v-bind="product" />
  </div>
</transition-group>
```



Component Slots



```
<!-- component with slots -->
<div class="container">
  <header>
    <slot name="header"></slot>
  </header>
  <main>
    <slot></slot>
  </main>
  <footer>
    <slot name="footer"></slot>
  </footer>
</div>
```



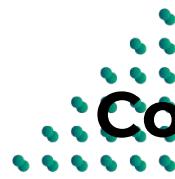
Component Slots



```
<!-- inserting slots -->
<base-layout>
  <template slot="header">
    <h1>Here might be a page title</h1>
  </template>

  <p>A paragraph for the main content.</p>
  <p>And another one.</p>

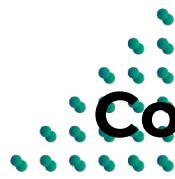
  <template slot="footer">
    <p>Here's some contact info</p>
  </template>
</base-layout>
```



Component Slots



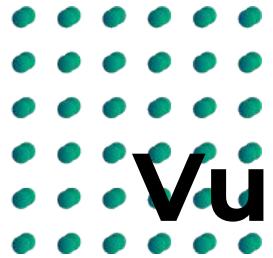
```
<!-- scoped slots inside -->
<ul>
  <li v-for="todo in todos"
      v-bind:key="todo.id">
    >
      <slot v-bind:todo="todo">
        {{ todo.text }}
      </slot>
    </li>
</ul>
```



Component Slots



```
<!-- scoped slots outside -->
<todo-list v-bind:todos="todos">
  <template slot-scope="slotProps">
    <!-- Define a custom template for todo items, using -->
    <!-- `slotProps` to customize each todo. -->
    <span v-if="slotProps.todo.isComplete">/</span>
    {{ slotProps.todo.text }}
  </template>
</todo-list>
```

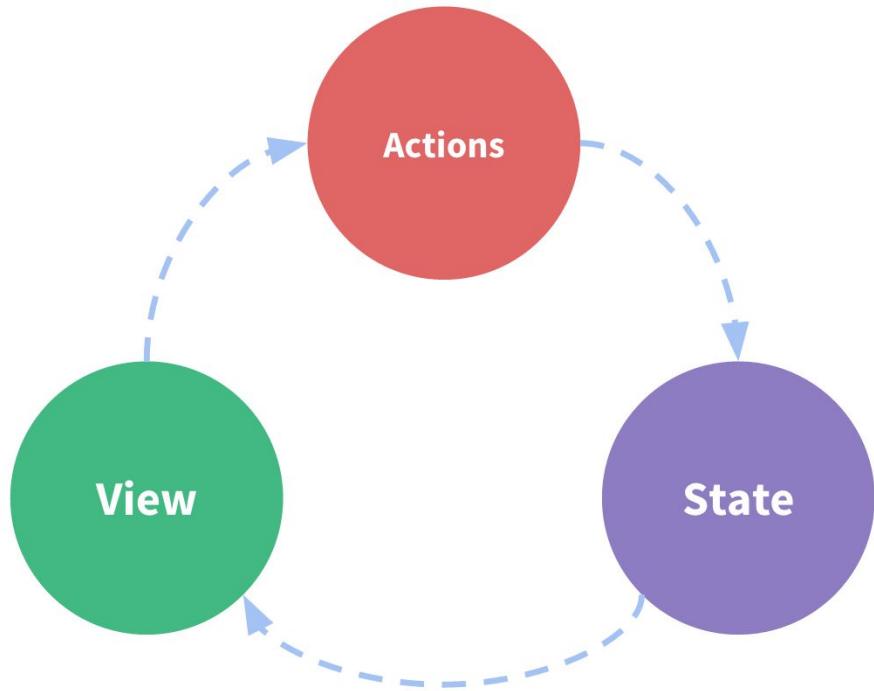


Vuex Store

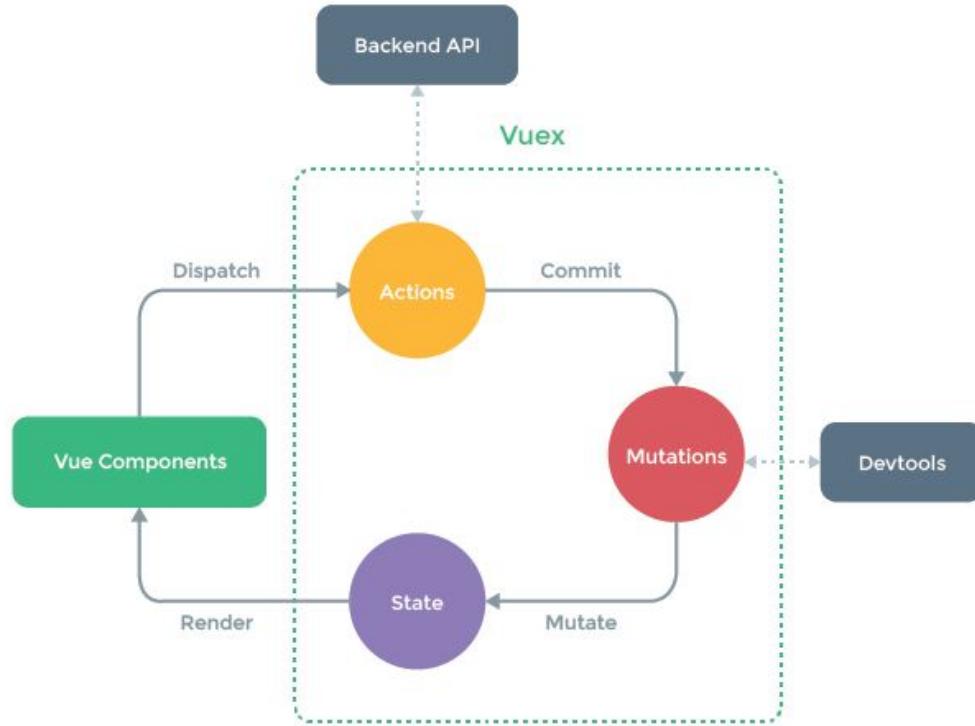


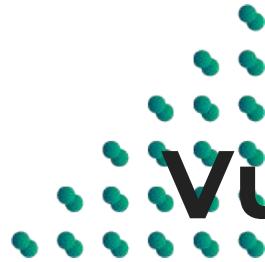


Vuex Store



Vuex Store





Vuex consists of:

- State
- Getters
- Actions
- Mutations



Vuex Store

```
1 import Vue from 'vue';
2 import Vuex from 'vuex';
3 Vue.use(Vuex);
4
5 const store = new Vuex.Store({
6   state: {
7     user: {
8       id: '',
9       token: '',
10      firstName: '',
11      lastName: '',
12      email: '',
13      phone: ''
14    },
15  },
16  mutations: {
17    UPDATE_USER(context, updateObj) {
18      context.user = Object.assign({}, context.user, updateObj);
19    },
20  },
21  actions: {
22    fetchUser() {
23      return new Promise((resolve, reject) => {
24        axios.get('/api/me').then((response) => {
25          if (response.data) {
26            context.commit('UPDATE_USER', response.data);
27          }
28          resolve(response);
29        }).catch(reject);
30      });
31    },
32  },
33  getters: {
34    userAuthenticated: state => !!state.user.token,
35    userFullName: `${state.user.firstName} ${state.user.lastName}`,
36  },
37});
38
39 export default store;
```



Vuex Store



```
state: {
  user: {
    id: '',
    token: '',
    firstName: '',
    lastName: '',
    email: '',
    phone: ''
  },
},
```



Vuex Store

```
actions: {
  fetchUser() {
    return new Promise((resolve, reject) => {
      axios.get('/api/me').then((response) => {
        if (response.data) {
          context.commit('UPDATE_USER', response.data);
        }
        resolve(response);
      }).catch(reject);
    });
  },
},
```



Vuex Store



```
getters: {
  userAuthenticated: state => !!state.user.token,
  userFullName: `${state.user.firstName} ${state.user.lastName}`,
},
```

```
mutations: {
  UPDATE_USER(context, updateObj) {
    context.user = Object.assign({}, context.user, updateObj);
  },
},
```



Vuex Store

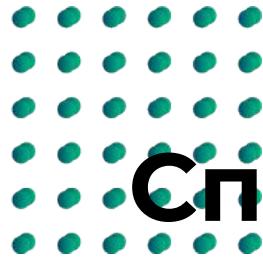
```
<script>
export default {
  name: 'user',
  data() {
    return {
      fetchingData: false,
    };
  },
  computed: {
    isAuthenticated() {
      return !!this.$store.state.user.token;
    },
    fullName() {
      return `${this.store.state.user.firstName} ${this.store.state.user.lastName}`;
    },
  },
  methods: {
    fetchMe() {
      this.fetchingData = true;
      this.$store.dispatch('fetchUser')
        .catch((error) => {
          alert(error.message);
        })
        .always(() => {
          this.fetchingData = false;
        });
    },
  },
  mounted() {
    this.fetchMe();
  },
}
</script>
```



Vuex Store

```
<script>
import { mapGetters, mapActions } from 'vuex';

export default {
    name: 'user',
    data() {
        return {
            fetchingData: false,
        };
    },
    computed: {
        ...mapGetters('user', ['userAuthenticated', 'userFullName']),
    },
    methods: {
        ...mapActions('user', ['fetchUser']),
        fetchMe() {
            this.fetchingData = true;
            this.fetchUser()
                .catch((error) => {
                    alert(error.message);
                })
                .always(() => {
                    this.fetchingData = false;
                });
        },
    },
    mounted() {
        this.fetchMe();
    },
};
</script>
```



Спасибо за внимание!

Telegram: @zilberstein

Facebook: [Andy Grishchenko](#)

