



VUE.JS

Прогрессивный JavaScript-фреймворк

Кротов Александр
Front – end developer, Team - Lead
alex.krotov92@gmail.com

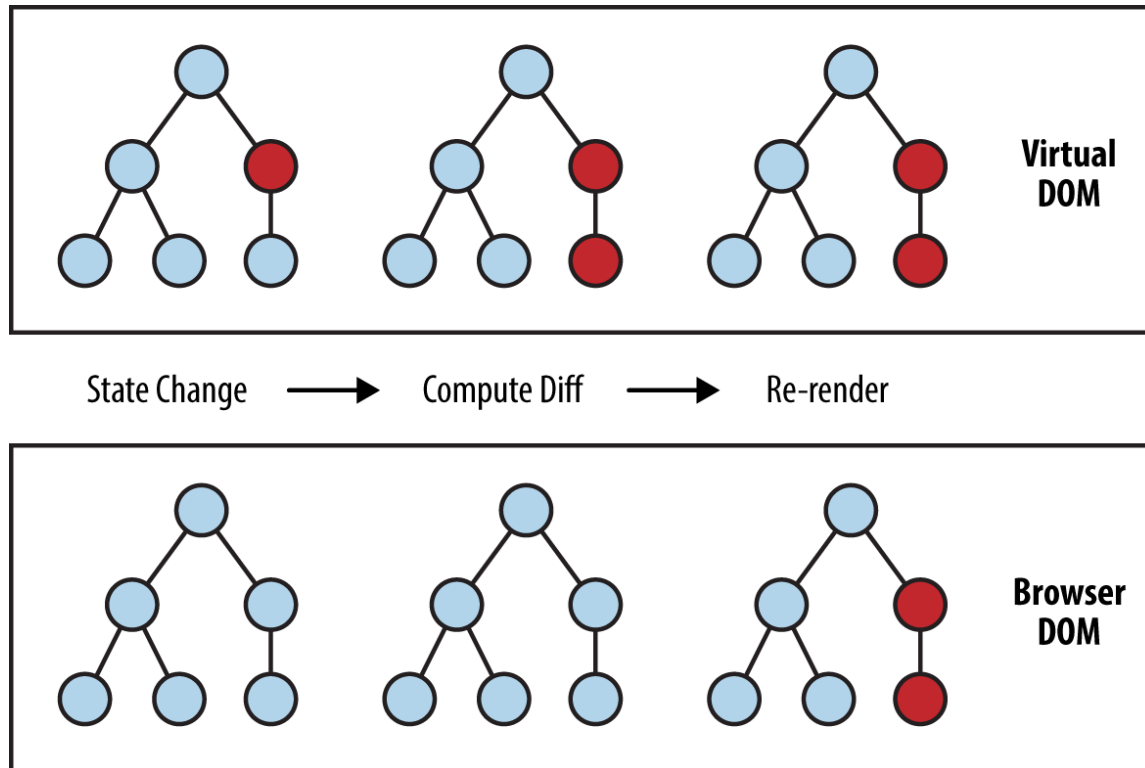
Краткое содержание

- Общий обзор фреймворка
- Основные особенности
- Базовый синтаксис
- Отличия от других фреймворков (React)
- Основные плагины для работы
- Vuex - state management pattern
- Оптимизация и особенности внедрения

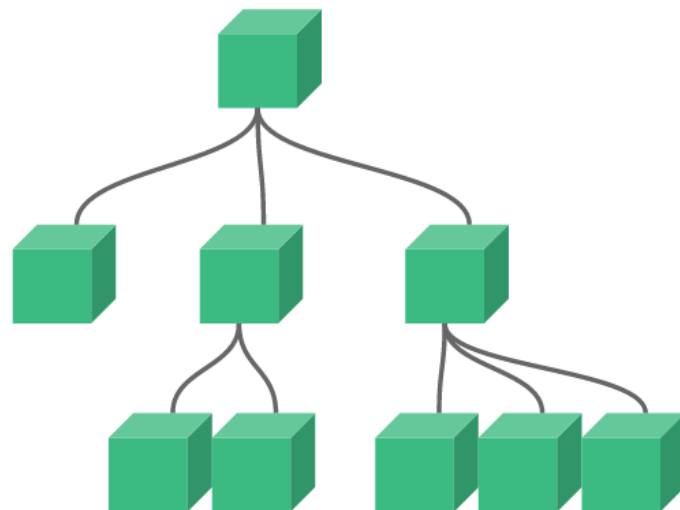
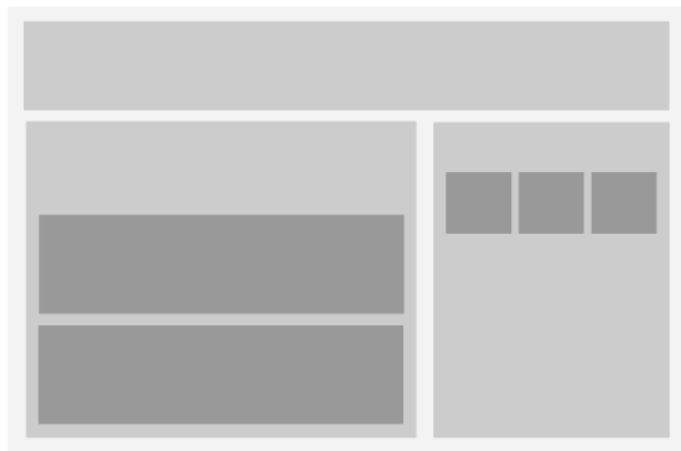
Общий обзор фреймворка

Vue - Прогрессивный фреймворк для создания пользовательских интерфейсов. В отличие от фреймворков-монолитов, Vue создан пригодным для постепенного внедрения. Его ядро в первую очередь решает задачи уровня представления (view), что упрощает интеграцию с другими библиотеками и существующими проектами. С другой стороны, Vue полностью подходит и для создания сложных одностраничных приложений (SPA, Single-Page Applications)

Virtual DOM



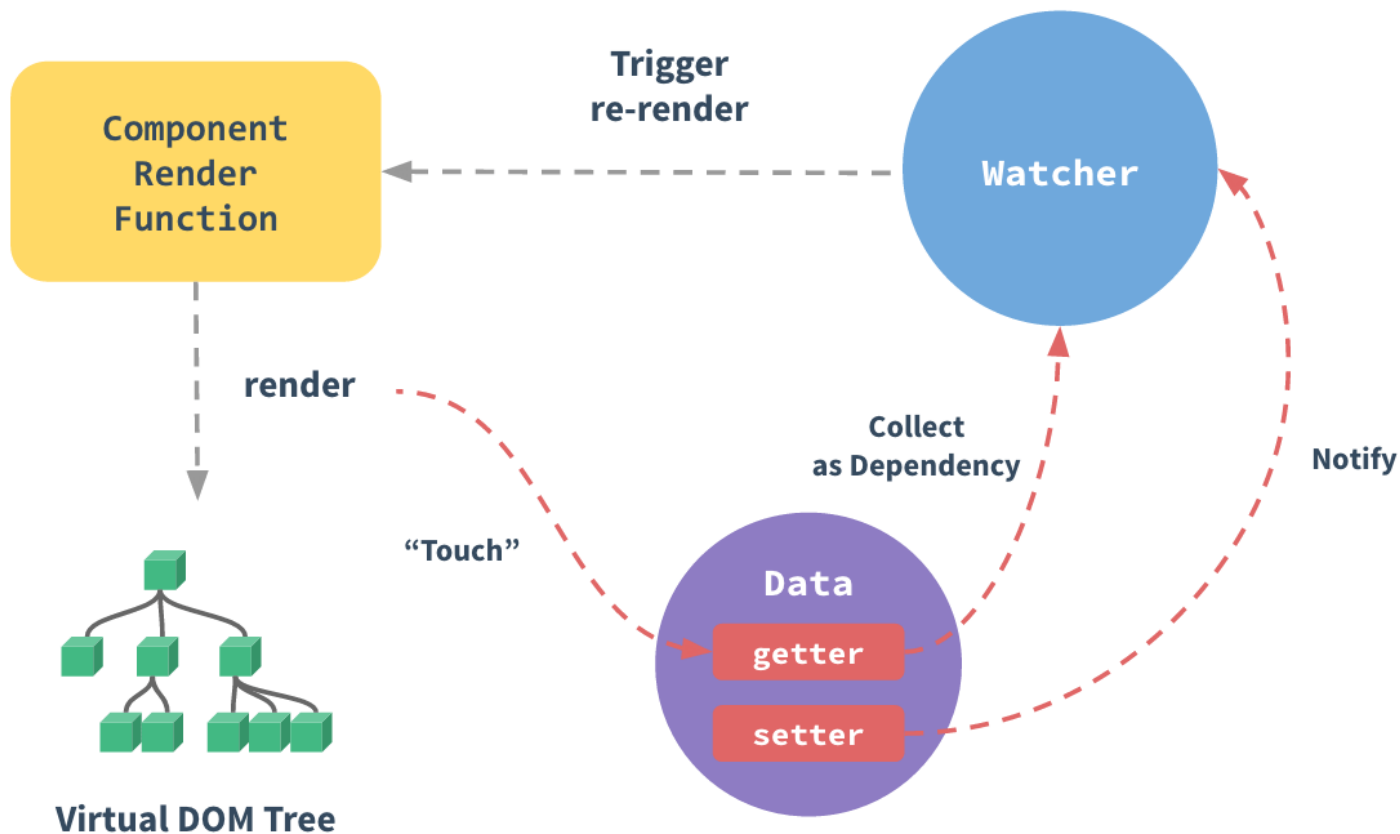
Компонентная модель



Основные особенности фреймворка

- Реактивные компоненты
- Быстрота работы
- Однофайловые компоненты
- Поддержка шаблонизаторов
- Соответствие Web Components Spec
- Поддержка слотов
- Роутинг на клиенте
- Официальный CLI

Реализация реактивности



Однофайловые компоненты



```
1 <template lang="jade">
2   div
3     p {{ greeting }} World!
4     other-component
5 </template>
6
7 <script>
8   import OtherComponent from './OtherComponent.vue'
9
10  export default {
11    data () {
12      return {
13        greeting: 'Hello'
14      }
15    },
16    components: {
17      OtherComponent
18    }
19  }
20 </script>
21
22 <style lang="stylus" scoped>
23   p
24     font-size 2em
25     text-align center
26 </style>
```

Line 27, Column 1 Spaces: 2 Vue Component

Шаблон компонента

```
<div id="app">
  <ul is="transition-group">
    <li v-for="user in users" class="user" :key="user['.key']">
      <span>{{user.name}} - {{user.email}}</span>
      <button v-on:click="removeUser(user)">X</button>
    </li>
  </ul>
  <form id="form" v-on:submit.prevent="addUser">
    <input type="text" v-model="newUser.name" placeholder="Username">
    <input type="email" v-model="newUser.email" placeholder="email@email.com">
    <input type="submit" value="Add User">
  </form>
  <ul class="errors">
    <li v-show="!validation.name">Name cannot be empty.</li>
    <li v-show="!validation.email">Please provide a valid email address.</li>
  </ul>
</div>
```

Данные и методы

```
data: {  
  name: 'Vue.js'  
},  
// определяйте методы в объекте `methods`  
methods: {  
  greet: function (event) {  
    // `this` внутри методов указывает на экземпляр Vue  
    alert('Привет, ' + this.name + '!')  
    // `event` — нативное событие DOM  
    if (event) {  
      alert(event.target.tagName)  
    }  
  }  
}
```

Стили

```
<style style="scss">
.date {
  background-color: white;
  cursor: pointer;
}
</style>
```

Поддержка слотов

```
<!-- FormHelper.vue -->
```

```
<form class="form">
```

```
<h3>{{ title }}</h3>
```

```
<slot name="help"></slot>
```

```
<slot name="elements"></slot>
```

```
<slot name="buttons"></slot>
```

```
</form>
```

```
<!-- App.vue -->
```

```
<form-helper title="Request Form">
```

```
<div slot="help">Title</div>
```

```
<div slot="elements">
```

```
<input type="text" value="">
```

```
</div>
```

```
<div slot="buttons">
```

```
<button type="button">Send</button>
```

```
</div>
```

```
</form-helper>
```

Шаблоны React vs Vue

```
render () {  
  let { items } = this.props  
  let children  
  if (items.length > 0) {  
    children = (  
      <ul>  
        {items.map(item =>  
          <li key={item.id}>{item.name}</li>  
        )}  
      </ul>  
    )  
  } else {  
    children = <p>Ничего не найдено.</p>  
  }  
  return (  
    <div className='list-container'>  
      {children}  
    </div>  
  )  
}
```

```
<template>  
  <div class="list-container">  
    <ul v-if="items.length">  
      <li v-for="item in items">  
        {{ item.name }}  
      </li>  
    </ul>  
    <p v-else>Ничего не найдено.</p>  
  </div>  
</template>
```

Скорость React vs Vue

	Vue	React
Fastest	23ms	63ms
Median	42ms	81ms
Average	51ms	94ms
95th Perc.	73ms	164ms
Slowest	343ms	453ms

Стили React vs Vue

```
<style scoped>
  @media (min-width: 250px) {
    .list-container:hover {
      background: orange;
    }
  }
</style>
```

Опциональный атрибут `scoped` автоматически ограничивает область видимости CSS текущим компонентом, добавляя элементам уникальные атрибуты (такие как `data-v-1-21e5b78`), и компилируя `.list-container:hover` во что-нибудь вроде `.list-container[data-v-1-21e5b78]:hover`.



awesome

Vue router

```
<div id="app">
  <h1>Hello App!</h1>
  <p>
    <router-link to="/foo">Go to Foo</router-link>
    <router-link to="/bar">Go to Bar</router-link>
  </p>
  <router-view></router-view>
</div>
```

```
const Foo = { template: '<div>foo</div>' }
const routes = [
  { path: '/foo', component: Foo }
]
const router = new VueRouter({
  routes // short for routes: routes
})
```

Vue resource

```
{  
  // POST /someUrl  
  this.$http.post('/someUrl', {foo: 'bar'}).then(response => {  
  
    // get status  
    response.status;  
  
    // get status text  
    response.statusText;  
  
    // get 'Expires' header  
    response.headers.get('Expires');  
  
    // get body data  
    this.someData = response.body;  
  
  }, response => {  
    // error callback  
  });  
}
```

Vue - i18n

```
var locales = {  
  en: {  
    message: {  
      hello: 'hello world'  
    }  
  },  
  ja: {  
    message: {  
      hello: 'こんにちは、世界'  
    }  
  }  
}
```

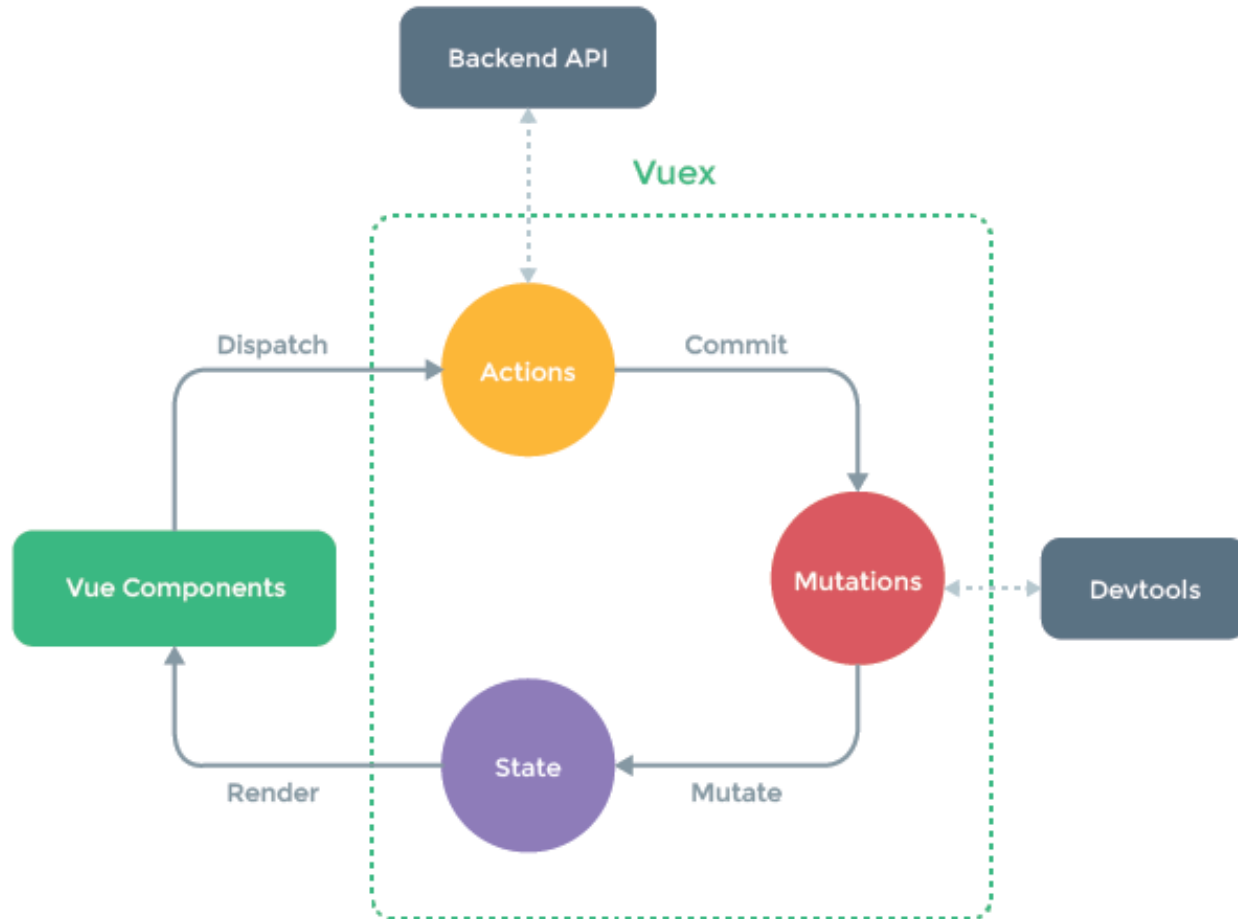
```
// set lang  
Vue.config.lang = 'ja'  
  
// set locales  
Object.keys(locales).forEach(function  
(lang) {  
  Vue.locale(lang, locales[lang])  
})  
  
// template  
<p>{{ $t("message.hello") }}</p>
```

Плагины JQuery

`<input type="text" class="form-control" :value="value" readonly>`

```
methods: {  
  initPlugin () {  
    $(this.$el).clockpicker(this.options);  
  },  
  addListeners () {  
    let vm = this;  
    $(this.$el).find('input').change(function () {  
      // vm.curValue = this.value;  
      vm.$emit('input', this.value);  
    });  
  },  
  mounted () {  
    // init ClockPicker  
    this.initPlugin();  
    // set listener  
    this.addListeners();  
  },  
}
```

Vuex (state management pattern)



Vuex

```
const store = new Vuex.Store({  
  state: {  
    count: 0  
  },  
  mutations: {  
    increment (state) {  
      state.count++  
    }  
  }  
})
```

```
store.commit('increment')  
console.log(store.state.count) // -> 1
```

Vue CLI

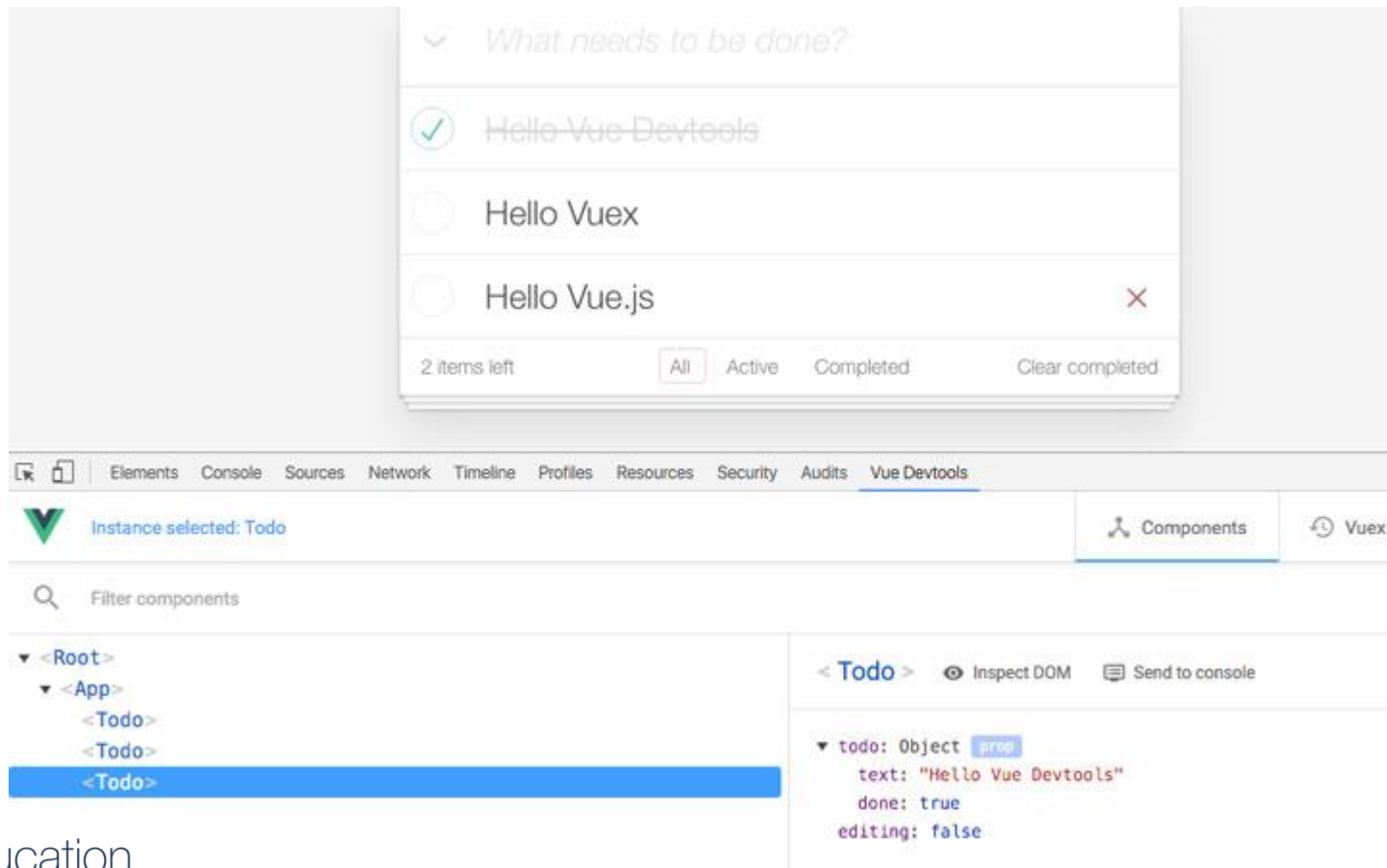
- Modularization
- Transpilers
- Pre-processors,
- Hot-reload
- Linting
- Testing.

```
npm install -g vue-cli  
vue init webpack my-project  
# answer prompts  
cd my-project  
npm install  
npm run dev # tada!
```

Vue + Laravel



Vue devtools for Chrome



Советы по оптимизации и внедрению

- Проверяйте наличие данных перед использованием их в компонентах, чтобы избежать ошибок
- Разбивайте шаблоны на мелкие компоненты для удобства тестирования и переиспользования
- При создании SPA используйте одну точку входа
- Используйте сервисы для работы с API



ВОПРОСЫ?

Спасибо за внимание

Кротов Александр
Front – end developer, Team - Lead
alex.krotov92@gmail.com