

Изучаем Vue

Основные концепции и практические паттерны
для современных и масштабируемых
пользовательских интерфейсов



Майя Шавин

Оглавление

Предисловие	11
Условные обозначения.....	11
Использование примеров кода	12
Благодарности.....	13
Глава 1. Добро пожаловать в мир Vue.js!	15
Что такое Vue.js?	15
Преимущества Vue в современной веб-разработке.....	16
Установка Node.js.....	16
NPM.....	17
Yarn.....	19
Инструменты разработчика Vue	20
Инструмент управления сборкой Vite.js.....	21
Создание нового приложения Vue	22
Файловая структура проекта	23
Резюме	24
Глава 2. Как работает Vue: основы	25
Виртуальный DOM "под капотом"	25
Проблема обновления компоновки	26
Что такое виртуальный DOM?	27
Как виртуальный DOM работает вместе с Vue	27
Экземпляр приложения Vue и Options API	29
Изучаем Options API.....	30
Синтаксис шаблонов	32
Создание локального состояния в свойстве <i>data()</i>	33
Как работает реактивность в фреймворке Vue	36
Двустороннее связывание с помощью <i>v-model</i>	38
Использование модификатора <i>v-model.lazy</i>	41
Связывание реактивных данных и передача свойств с помощью <i>v-bind</i>	42
Привязка к классу и атрибутам стиля.....	44
Перебор коллекции с помощью <i>v-for</i>	47
Перебор свойств объекта	50
Уникальная привязка элемента с помощью атрибута <i>key</i>	51
Обработка событий <i>v-on</i>	52
Использование модификаторов <i>v-on</i>	53
События клавиатуры с помощью модификаторов клавиш	56
Условный рендеринг элементов с помощью <i>v-if</i> , <i>v-else</i> и <i>v-else-if</i>	57
Условная видимость элементов с помощью <i>v-show</i>	60

Динамическая отрисовка HTML-кода с помощью <i>v-html</i>	61
Отображение текста с помощью <i>v-text</i>	62
Оптимизация рендеринга с помощью <i>v-once</i> и <i>v-memo</i>	62
Глобальная регистрация компонентов	65
Резюме	66
Глава 3. Создание более сложных компонентов	67
Структура однофайлового компонента	67
Использование <i>defineComponent()</i> для поддержки TypeScript	70
Хуки жизненного цикла компонента	71
<i>setup</i>	73
<i>beforeCreate</i>	75
<i>created</i>	75
<i>beforeMount</i>	75
<i>mounted</i>	75
<i>beforeUpdate</i>	76
<i>updated</i>	76
<i>beforeUnmount</i>	76
<i>unmounted</i>	76
Методы	80
Вычисляемые свойства	82
Наблюдатели	84
Наблюдение за изменениями вложенных свойств	87
Использование метода <i>this.\$watch()</i>	89
Возможности слотов	91
Именованные слоты в теге <i>template</i> с директивой <i>v-slot</i>	95
Использование ссылок <i>ref</i>	98
Общая конфигурация компонентов с помощью миксинов	100
Локальные стили в компонентах	104
Применение локальных CSS-стилей к дочернему компоненту	107
Применение локальных стилей к содержимому слота	108
Использование данных компонента в стиле с помощью псевдокласса <i>v-bind()</i>	108
Определение стилей компонентов с помощью CSS-модулей	110
Резюме	112
Глава 4. Взаимодействие компонентов	113
Вложенные компоненты и поток данных в Vue	113
Использование <i>props</i> для передачи данных в дочерние компоненты	114
Объявление входных параметров с валидацией и значениями по умолчанию	117
Объявление входных параметров с проверкой пользовательских типов	119
Объявление входных параметров с помощью <i>defineProps()</i> и <i>withDefaults()</i>	122
Взаимодействие компонентов через пользовательские события	123
Определение пользовательских событий с помощью <i>defineEmits()</i>	127
Взаимодействие компонентов с помощью <i>provide/inject</i>	129
Использование <i>provide</i> для передачи данных	129
Использование <i>inject</i> для получения данных	130
Teleport API	131
Реализация модального окна с помощью <i>Teleport</i> и элемента <i><dialog></i>	133
Проблема рендеринга при использовании <i>Teleport</i>	140
Резюме	141

Глава 5. Composition API	143
Создание компонентов в Composition API	143
Управление данными с помощью <i>ref()</i> и <i>reactive()</i>	144
Использование <i>ref()</i>	144
Использование <i>reactive()</i>	149
Хуки жизненного цикла в Composition API	151
Наблюдатели в Composition API	154
Использование <i>computed()</i>	157
Создание многократно используемых композаблов	159
Резюме	163
Глава 6. Загрузка внешних данных	165
Что такое Axios?	165
Установка Axios	166
Загрузка данных с помощью хуков жизненного цикла и Axios	166
Проблема асинхронных запросов в режиме выполнения	170
Создание повторно используемого компонента для выполнения запросов	173
Хранение данных на стороне браузера	176
Резюме	177
Глава 7. Продвинутый рендеринг, динамические компоненты и плагины	179
Функция рендеринга и JSX	179
Использование функции рендеринга	179
Использование функции <i>h</i> для создания VNode	180
JavaScript XML в функции рендеринга	182
Функциональный компонент	183
Определение <i>props</i> и <i>emits</i> для функционального компонента	185
Пользовательская функциональность глобального уровня с помощью плагинов Vue	185
Динамический рендеринг с помощью тега <i><component></i>	188
Кеширование экземпляра неактивного компонента с помощью <i><keep-alive></i>	189
Резюме	191
Глава 8. Маршрутизация	193
Что такое маршрутизация?	193
Использование Vue Router	194
Установка Vue Router	194
Определение маршрутов	196
Создание экземпляра маршрутизатора	199
Подключение экземпляра маршрутизатора к приложению Vue	200
Рендеринг текущей страницы с помощью <i>RouterView</i>	201
Панель навигации с помощью компонента <i>RouterLink</i>	202
Передача данных между маршрутами	204
Передача параметров маршрута с помощью <i>props</i>	208
Навигационные хуки	209
Глобальные навигационные хуки	209
Навигационные хуки уровня маршрута	210
Навигационные хуки уровня компонента	211
Создание вложенных маршрутов	213
Создание динамических маршрутов	215

Переход назад и вперед с помощью экземпляра маршрутизатора	218
Обработка неопределенных маршрутов	219
Резюме	221
Глава 9. Управление состояниями с помощью Pinia.....	223
Что такое управление состояниями в Vue?	223
Что такое Pinia	225
Создание хранилища <i>pizzas</i> для Pizza House	226
Создание хранилища <i>cart</i> для Pizza House	230
Использование хранилища <i>cart</i> в компоненте	231
Добавление элементов в корзину из галереи	232
Отображение элементов корзины с помощью действий.....	235
Удаление товаров из хранилища <i>cart</i>	239
Модульное тестирование хранилищ Pinia.....	241
Подписка на изменения хранилища и добавление побочных эффектов	241
Резюме	244
Глава 10. Переходы и анимация в Vue.....	245
Что такое CSS-переходы и CSS-анимация?	245
Компонент <i>transition</i> в Vue	246
Использование атрибутов с пользовательскими классами переходов	249
Эффект перехода при первом рендеринге с помощью <i>appear</i>	250
Создание перехода для группы элементов	251
Создание переходов для маршрутов	252
Использование событий перехода для управления анимацией.....	253
Резюме	255
Глава 11. Тестирование в Vue.....	257
Общие сведения о модульном и сквозном тестировании	257
Vitest как инструмент для модульного тестирования	259
Настройка Vitest с помощью параметров и файла конфигурации	259
Создание вашего первого теста.....	261
Тестирование комposable, не использующих хуки жизненного цикла	266
Тестирование комposable с хуками жизненного цикла	269
Тестирование компонентов с помощью Vue Test Utils	274
Тестирование событий компонента	277
Пользовательский интерфейс Vitest	279
Использование Vitest с инструментами для оценки покрытия кода.....	280
Сквозное тестирование с помощью PlaywrightJS	284
Отладка сквозных тестов с помощью расширения Playwright Test для VSCode	291
Резюме	293
Глава 12. Непрерывная интеграция и непрерывное развертывание	
Vue-приложений	295
CI/CD в разработке программного обеспечения.....	295
Непрерывная интеграция.....	296
Непрерывная доставка	296
Непрерывное развертывание.....	296
Конвейер CI/CD с помощью GitHub Actions.....	297
Непрерывное развертывание с помощью Netlify.....	301

Развертывание с помощью Netlify CLI.....	303
Резюме	304
Глава 13. Рендеринг на стороне сервера в фреймворке Vue.....	305
Рендеринг на стороне клиента в Vue	305
Рендеринг на стороне сервера (SSR)	306
Рендеринг на стороне сервера с помощью Nuxt.js	311
Генератор статических сайтов (SSG).....	319
Напоследок.....	320
Предметный указатель.....	323
Об авторе.....	331
Об изображении на обложке.....	333