

Svelte и Sapper

В ДЕЙСТВИИ

Марк Волкманн



MANNING



Svelte and Sapper in Action

R. MARK VOLKMANN



MANNING
SHELTER ISLAND

Svelte и Sapper в действии

Марк Волкманн



Санкт-Петербург • Москва • Минск

2022

ББК 32.988.02-018
УДК 004.42:004.738.5
В67

Волкманн Марк

В67 Svelte и Sapper в действии. — СПб.: Питер, 2022. — 496 с.: ил. — (Серия «Для профессионалов»).

ISBN 978-5-4461-1464-1

Многие веб-фреймворки замедляют работу приложений, загружая сотни строк кода «на всякий случай». Svelte позволяет компилировать приложения в эффективные крошечные пакеты с минимальным временем загрузки. Объединение Svelte с фреймворком Sapper облегчает маршрутизацию страниц, генерирование на стороне сервера, разработку статических сайтов и решение многих других задач.

Хотите научиться создавать быстрые и элегантные веб-приложения?

Вам помогут компоненты Svelte и качественный опыт взаимодействия. Вы научитесь использовать уникальную модель управления состояниями, освоите упрощенную маршрутизацию страниц, разбиение кода, поддержку автономного режима и работу с представлениями, генерируемыми на сервере.

16+ (В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ББК 32.988.02-018
УДК 004.42:004.738.5

Права на издание получены по соглашению с Manning Publications. Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

Информация, содержащаяся в данной книге, получена из источников, рассматриваемых издательством как надежные. Тем не менее, имея в виду возможные человеческие или технические ошибки, издательство не может гарантировать абсолютную точность и полноту приводимых сведений и не несет ответственности за возможные ошибки, связанные с использованием книги.

Издательство не несет ответственности за доступность материалов, ссылки на которые вы можете найти в этой книге. На момент подготовки книги к изданию все ссылки на интернет-ресурсы были действующими.

ISBN 978-1617297946 англ.
ISBN 978-5-4461-1464-1

© 2020 by Manning Publications Co. All rights reserved
© Перевод на русский язык ООО Издательство «Питер», 2022
© Издание на русском языке, оформление
ООО Издательство «Питер», 2022
© Серия «Для профессионалов», 2022

Краткое содержание

Краткое содержание.....	5
Предисловие.....	17
Благодарности.....	18
О книге.....	20
Об авторе.....	26
Иллюстрация на обложке.....	27

Часть 1 Первые шаги

Глава 1. Знакомство с игроками.....	30
Глава 2. Ваше первое приложение Svelte.....	50

Часть 2 Подробнее о Svelte

Глава 3. Создание компонентов.....	76
Глава 4. Блочные структуры.....	101
Глава 5. Взаимодействие между компонентами.....	121
Глава 6. Хранилища.....	148
Глава 7. Взаимодействия с DOM.....	168
Глава 8. Функции жизненного цикла.....	187
Глава 9. Маршрутизация на стороне клиента.....	199
Глава 10. Анимация.....	219

6 Краткое содержание

Глава 11. Отладка	241
Глава 12. Тестирование	250
Глава 13. Развертывание	290
Глава 14. Расширенные возможности Svelte	299

Часть 3 Подробнее о Sapper

Глава 15. Первое приложение Sapper	322
Глава 16. Приложения Sapper	331
Глава 17. Серверные маршруты Sapper	352
Глава 18. Экспортирование статических сайтов в Sapper	372
Глава 19. Поддержка автономного режима в Sapper	387

Часть 4 За пределами Svelte и Sapper

Глава 20. Препроцессоры	410
Глава 21. Svelte Native	425
Приложение А. Ресурсы	464
Приложение Б. Обращения к службам REST	471
Приложение В. MongoDB	474
Приложение Г. ESLint для Svelte	482
Приложение Д. Prettier для Svelte	484
Приложение Е. VS Code	486
Приложение Ж. Snowpack	491

Оглавление

Краткое содержание	5
Предисловие	17
Благодарности	18
О книге	20
Для кого написана эта книга.....	20
Структура.....	21
О коде.....	24
Форум liveBook.....	25
Прочие онлайн-ресурсы.....	25
От издательства.....	25
Об авторе	26
Иллюстрация на обложке	27

Часть 1 Первые шаги

Глава 1. Знакомство с игроками	30
1.1. Знакомство со Svelte.....	31
1.1.1. Почему Svelte.....	32
1.1.2. Новый подход к реактивности.....	37
1.1.3. Текущие проблемы в Svelte.....	39
1.1.4. Как работает Svelte.....	40
1.1.5. Svelte исчезает?.....	42
1.2. Знакомство с Sapper.....	42
1.2.1. Зачем использовать Sapper.....	43
1.2.2. Как работает Sapper.....	45

8 Оглавление

1.2.3. Когда стоит использовать Sapper	45
1.2.4. Когда не стоит использовать Sapper	46
1.3. Знакомство с Svelte Native	46
1.4. Сравнение Svelte с другими веб-фреймворками	46
1.4.1. Angular	46
1.4.2. React	47
1.4.3. Vue	47
1.5. Какие инструменты понадобятся для начала работы	48
Резюме	48
Глава 2. Ваше первое приложение Svelte	50
2.1. Svelte REPL	51
2.1.1. Использование Svelte REPL	51
2.1.2. Ваше первое приложение REPL	53
2.1.3. Сохранение приложений REPL	58
2.1.4. Распространение приложений REPL	60
2.1.5. URL-адреса REPL	61
2.1.6. Экспортирование приложений REPL	61
2.1.7. Пакеты npm	62
2.1.8. Ограничения REPL	62
2.1.9. CodeSandbox	63
2.2. Работа за пределами REPL	63
2.2.1. Использование npm degit	64
2.2.2. Файл package.json	66
2.2.3. Важные файлы	67
2.2.4. Ваше первое приложение без использования REPL	69
2.3. Еще одно приложение	70
Резюме	74

Часть 2 Подробнее о Svelte

Глава 3. Создание компонентов	76
3.1. Содержимое файлов .svelte	77

3.2. Разметка компонентов	78
3.3. Имена компонентов.....	81
3.4. Стили компонентов.....	82
3.5. Специфичность CSS.....	83
3.6. Глобальные стили и стили с ограниченной видимостью	84
3.7. Препроцессоры CSS.....	87
3.8. Логика компонентов	88
3.9. Состояние компонента.....	89
3.10. Реактивные команды	90
3.11. Контекст модуля	93
3.12. Построение нестандартных компонентов.....	94
3.13. Построение приложения Travel Packing.....	96
Резюме	99
Глава 4. Блочные структуры.....	101
4.1. Условная логика с {#if}	102
4.2. Итерации с {#each}	103
4.3. Промисы с {#await}.....	105
4.4. Построение приложения Travel Packing.....	108
4.4.1. Компонент Item	109
4.4.2. Вспомогательные функции	111
4.4.3. Компонент Category	112
4.4.4. Компонент Checklist	115
4.4.5. Компонент App	118
4.4.6. Попробуйте сами.....	119
Резюме	120
Глава 5. Взаимодействие между компонентами	121
5.1. Варианты взаимодействий между компонентами.....	122
5.2. Проп-свойства	123
5.2.1. Объявление проп-свойств ключевым словом export	123
5.2.2. Реакция на изменения проп-свойств	126
5.2.3. Типы проп-свойств.....	127

5.2.4. Директивы	128
5.2.5. Директива bind с элементами форм	128
5.2.6. bind:this	131
5.2.7. Внешнее связывание prop-свойств	133
5.3. Слоты.....	137
5.4. События.....	138
5.4.1. Отправка событий.....	139
5.4.2. Перенаправление событий.....	140
5.4.3. Модификаторы событий	141
5.5. Контекст	141
5.6. Построение приложения Travel Packing.....	143
Резюме	147
Глава 6. Хранилища	148
6.1. Хранилища для записи.....	149
6.2. Хранилища для чтения	150
6.3. Где определять хранилища	151
6.4. Использование хранилищ.....	152
6.5. Производные хранилища	159
6.6. Специальные хранилища	160
6.7. Использование хранилищ с классами	162
6.8. Долговременное хранение данных.....	166
6.9. Построение приложения Travel Packing.....	167
Резюме	167
Глава 7. Взаимодействия с DOM	168
7.1. Вставка разметки HTML.....	169
7.2. Действия.....	173
7.3. Функция tick.....	173
7.4. Реализация компонента диалогового окна	177
7.5. Перетаскивание	180
7.6. Создание приложения Travel Packing.....	183
Резюме	186

Глава 8. Функции жизненного цикла	187
8.1. Подготовка.....	188
8.2. Функция жизненного цикла onMount	189
8.2.1. Передача фокуса.....	189
8.2.2. Получение данных от служб API	190
8.3. Функция жизненного цикла onDestroy	191
8.4. Функция жизненного цикла beforeUpdate	193
8.5. Функция жизненного цикла afterUpdate	194
8.6. Вспомогательные функции	196
8.7. Создание приложения Travel Packing.....	198
Резюме	198
Глава 9. Маршрутизация на стороне клиента	199
9.1. Ручная маршрутизация.....	200
9.2. #-маршрутизация.....	208
9.3. Использование библиотеки page.js	210
9.4. Использование параметров пути и запроса с page.js	212
9.5. Построение приложения Travel Packing	216
Резюме	218
Глава 10. Анимация	219
10.1. Функции плавности.....	220
10.2. Пакет svelte/animate	221
10.3. Пакет svelte/motion.....	223
10.4. Пакет svelte/transition.....	227
10.5. Переход fade и анимация flip	229
10.6. Переход crossfade	231
10.7. Переход draw	233
10.8. Нестандартные переходы.....	235
10.9. Prop-свойства transition и in/out.....	237
10.10. События перехода	237
10.11. Построение приложения Travel Packing	238
Резюме	240

12 Оглавление

Глава 11. Отладка	241
11.1. Тер @debug.....	242
11.2. Реактивные команды	245
11.3. Svelte DevTools.....	245
Резюме	249
Глава 12. Тестирование	250
12.1. Модульное тестирование с Jest.....	251
12.1.1. Модульные тесты для приложения Todo	254
12.1.2. Модульные тесты для приложения Travel Packing.....	256
12.2. Сквозные тесты с использованием Cypress.....	261
12.2.1. Сквозные тесты для приложения Todo	263
12.2.2. Сквозные тесты для приложения Travel Packing	265
12.3. Тесты доступности.....	269
12.3.1. Компилятор Svelte.....	270
12.3.2. Lighthouse	271
12.3.3. axe	274
12.3.4. WAVE.....	277
12.4. Демонстрация и отладка компонентов с использованием Storybook	279
12.4.1. Storybook для приложения Travel Packing	282
Резюме	289
Глава 13. Развертывание	290
13.1. Развертывание на любом сервере HTTP.....	291
13.2. Использование Netlify.....	291
13.2.1. Netlify с веб-сайта	292
13.2.2. Netlify из командной строки	293
13.2.3. Планы Netlify.....	295
13.3. Использование Vercel	295
13.3.1. Vercel с веб-сайта.....	296
13.3.2. Vercel из командной строки.....	297

13.3.3. Уровни Vercel.....	297
13.4. Использование Docker.....	298
Резюме	298
Глава 14. Расширенные возможности Svelte	299
14.1. Проверка данных форм.....	300
14.2. Использование библиотек CSS	305
14.3. Специальные элементы	309
14.4. Импортирование файлов JSON.....	313
14.5. Создание библиотек компонентов.....	313
14.6. Веб-компоненты	315
Резюме	320

Часть 3 Подробнее о Sapper

Глава 15. Первое приложение Sapper	322
15.1. Создание приложения Sapper.....	324
15.2. Воссоздание приложения интернет-магазина на базе Sapper.....	326
Резюме	330
Глава 16. Приложения Sapper	331
16.1. Структура файлов Sapper	332
16.2. Маршруты страниц	336
16.3. Макеты страниц	338
16.4. Обработка ошибок.....	340
16.5. Запуск на стороне сервера и на стороне клиента	341
16.6. Обертка для Fetch API.....	341
16.7. Предварительная загрузка.....	342
16.8. Предварительная выборка.....	344
16.9. Разделение кода.....	346
16.10. Построение приложения Travel Packing	348
Резюме	351

Глава 17. Серверные маршруты Sapper.....	352
17.1. Исходные файлы серверных маршрутов.....	353
17.2. Функции серверных маршрутов	354
17.3. Пример использования операций CRUD	355
17.4. Переход на Express	364
17.5. Построение приложения Travel Packing.....	364
Резюме	371
Глава 18. Экспортирование статических сайтов в Sapper	372
18.1. Подробности Sapper	373
18.2. Когда экспортируются приложения	374
18.3. Пример приложения.....	374
Резюме	386
Глава 19. Поддержка автономного режима в Sapper	387
19.1. Знакомство с сервисными работниками	388
19.2. Стратегии кэширования.....	390
19.3. Конфигурация сервисных работников Sapper	393
19.4. События сервисных работников	395
19.5. Управление сервисными работниками в Chrome	396
19.6. Включение поддержки HTTPS на сервере Sapper.....	400
19.7. Подтверждение автономного поведения	403
19.8. Построение приложения Travel Packing.....	404
Резюме	408

Часть 4

За пределами Svelte и Sapper

Глава 20. Препроцессоры	410
20.1. Нестандартная препроцессорная обработка	411
20.1.1. Webpack	413
20.2. Пакет svelte-preprocess	414
20.2.1. Режим автоматической препроцессорной обработки.....	414
20.2.2. Внешние файлы.....	415

20.2.3. Глобальные стили	417
20.2.4. Sass	417
20.2.5. TypeScript.....	418
20.2.6. Совет по поводу VS Code.....	420
20.3. Markdown	421
20.4. Использование нескольких препроцессоров	423
20.5. Сжатие графики	424
Резюме	424
Глава 21. Svelte Native.....	425
21.1. Компоненты NativeScript.....	426
21.1.1. Компоненты вывода данных.....	427
21.1.2. Компоненты форм	428
21.1.3. Компоненты действий.....	428
21.1.4. Компоненты диалоговых окон	428
21.1.5. Компоненты макета	429
21.1.6. Навигационные компоненты.....	432
21.2. Знакомство с Svelte Native	433
21.3. Локальная разработка приложений Svelte Native.....	434
21.4. Стилизовое оформление NativeScript	436
21.5. Предопределенные классы CSS в NativeScript	437
21.6. Темы NativeScript.....	439
21.7. Расширенный пример	439
21.8. Библиотека компонентов NativeScript UI	457
21.9. Проблемы Svelte Native.....	462
Резюме	463
Приложение А. Ресурсы.....	464
А.1. Презентации Svelte	464
А.2. Ресурсы Svelte	465
А.3. Сравнение фреймворков.....	465
А.4. Ресурсы Sapper	465
А.5. Ресурсы Svelte Native.....	466

A.6. Ресурсы Svelte GL.....	466
A.7. Инструменты Svelte	466
A.8. Библиотеки Svelte.....	467
A.9. Ресурсы VS Code	467
A.10. Учебные ресурсы, не относящиеся к Svelte	468
A.11. Инструменты, не относящиеся к Svelte	469
A.12. Библиотеки, не относящиеся к Svelte.....	470
A.13. Ресурсы, не относящиеся к Svelte	470
Приложение Б. Обращения к службам REST.....	471
Б.1. Заголовки	473
Приложение В. MongoDB	474
В.1. Установка MongoDB	475
В.1.1. Установка MongoDB в системе Windows.....	475
В.1.2. Установка MongoDB в системе Linux	476
В.1.3. Установка MongoDB в macOS	476
В.2. Запуск сервера базы данных.....	477
В.3. Использование оболочки MongoDB	477
В.4. Использование MongoDB из JavaScript.....	479
Приложение Г. ESLint для Svelte.....	482
Приложение Д. Prettier для Svelte	484
Приложение Е. VS Code.....	486
Е.1. Настройка VS Code.....	487
Е.2. Расширение Svelte for VS Code	488
Е.3. Расширение Svelte 3 Snippets.....	488
Е.4. Расширение Svelte Intellisense	489
Приложение Ж. Snowpack.....	491
Ж.1. Использование Snowpack со Svelte	491

Предисловие

Я профессионально занимаюсь программированием уже 37 лет, а веб-разработкой — около десяти. В своих проектах я использовал разные технологии и фреймворки, включая низкоуровневые манипуляции с DOM, jQuery, Ruby, Angular 1, React, Polymer, Angular 2+, Vue, Svelte... и, вероятно, с какими-то еще, о которых я уже позабыл.

Большое внимание я уделяю производительности труда разработчика. Избыточная сложность кода противоречит этой цели. Хотя в Svelte и Sapper мне нравится очень многое, самой привлекательной стороной я считаю простоту этих технологий по сравнению с другими методологиями веб-разработки. Svelte и Sapper помогают сделать работу намного более продуктивной.

О Svelte я узнал, посмотрев выступление Rethinking Reactivity Рича Харриса (Rich Harris) — создателя Svelte. Это была исключительно увлекательная речь, которая определенно отвечала моему стремлению к сокращению сложности веб-разработки. В итоге я занялся дальнейшим изучением темы, написал длинную статью о Svelte, участвовал в пользовательских группах и наконец стал выступать с докладами на конференциях. Следующий логичный шаг — написать книгу!

На страницах издания изложены практически все темы, связанные с Svelte и Sapper, а также некоторые вопросы, имеющие косвенное отношение к этим темам. После прочтения книги вы будете хорошо подготовлены к использованию этих средств в вашем следующем веб-проекте.

Благодарности

Многие авторы благодарят своих супругов за терпение, которое они проявляли, пока их мужья работали над книгами. Этот опыт наглядно показал мне, на какие жертвы им приходится идти. Моя жена Тэми великодушно подбадривала меня и помогала найти время для завершения проекта. Огромное спасибо, Тэми, за то, что помогла мне достичь цели и все-таки дописать книгу!

Спасибо Дженнифер Стаут (Jennifer Stout), руководителю проекта из издательства Manning, — ее правки, советы, стимулы и комплименты позволяли мне не сбиться с правильного пути. Благодаря ее комментариям «Я тебя обожаю!» вкупе с «Лучше объяснить, почему кто-нибудь станет это делать» процесс написания книги стал намного более сносным.

Спасибо Алену Кунио (Alain Couniot), научному редактору издательства Manning, который раз за разом указывал, где я изъяснялся недостаточно ясно или не приводил убедительных примеров. Он также следил за тем, чтобы я упоминал TypeScript везде, где это возможно. Его критика сильно улучшила эту книгу!

Спасибо научному редактору Manning Эрику Вуллингсу (Erik Vullings). Он опробовал ряд неочевидных подходов в коде, которые мне даже не приходили в голову, и предложил много улучшений текста. Его педантичность была невероятно уместной!

Спасибо Пирю Рейндерсу (Peer Reynders), добровольному рецензенту из программы MEAP, который рассмотрел код примеров под увеличительным стеклом и указал на многие возможности его совершенствования.

Я благодарен всем рецензентам, которые поделились своим мнением о способах улучшить книгу. Это Адейл Ретамал (Adail Retamal), Эмит Ламба (Amit Lamba), Клайв Харбер (Clive Harber), Дэмьен Эстебан (Damian Esteban), Дэвид Кабреро Сото (David Cabrero Souto), Дэвид Пакку (David Paccoud), Деннис Рейл (Dennis Reil), Герд Кливесаат (Gerd Klevesaat), Густаво Филипе Рамос Гомес (Gustavo Filipe Ramos Gomes), Джонатан Кук (Jonathan Cook), Келум Сенанаякэ (Kelum Senanayake), Константинос Леймонис (Konstantinos Leimonis), Маттео Гилдоне (Matteo Gildone), Потито Колучелли (Potito Coluccelli), Роберт Уолш (Robert Walsh), Родни Вайс (Rodney Weis), Сандер Зегвельд (Sander Zegveld), Сергио

Арбео (Sergio Arbeo) и Таня Вилке (Tanya Wilke). Широта взглядов и мнений определенно пошла на пользу!

Спасибо Чарльзу Шарпу (Charles Sharp) из Object Computing, Inc. Он чаще всего занимался редактированием моих предыдущих текстов. За последние десять лет Чарльз потратил немало времени на то, чтобы повысить мою литературную квалификацию... Серийные запятые и все такое!

Спасибо Элдону Аролду (Eldon Ahrold) из Object Computing, Inc. за рецензирование главы о Svelte Native. Элдон — чрезвычайно опытный разработчик мобильных и веб-приложений, и я рад возможности воспользоваться его опытом.

Наконец, спасибо доктору Эбрахиму Мошири (Ebrahim Moshiri), который 24 года назад принял меня на работу в Object Computing, Inc. Он привел меня в среду, где учиться приходилось постоянно, чем я и занимался. Если бы не карьера, открывшаяся передо мной, то, скорее всего, я бы никогда не написал эту книгу.

ДЛЯ КОГО НАПИСАНА ЭТА КНИГА

Книга «Svelte и Sapper в действии» предназначена для веб-разработчиков, которые хотят сделать свою работу более продуктивной. Вероятно, вас терзает мысль о том, что должен существовать более простой способ разработки веб-приложений. Могу вас обрадовать — он действительно существует и вы о нем узнаете в этой книге!

На многочисленных примерах кода вы научитесь использовать Svelte и Sapper для реализации многих стандартных аспектов функциональности веб-приложений.

Книга предполагает, что читатель хотя бы на базовом уровне владеет HTML, CSS и JavaScript.

- HTML — читатель должен знать основные элементы: `html`, `head`, `link`, `style`, `script`, `body`, `div`, `span`, `p`, `ol`, `ul`, `li`, `input`, `textarea` и `select`.
- CSS — читатель должен понимать синтаксис правил CSS и смысл определения «каскадный», знать часто используемые свойства CSS (включая `color`, `font-family`, `font-size`, `font-style` и `font-weight`) и блочную модель CSS (`content`, `padding`, `border` и `margin`).
- JavaScript — читатель должен разбираться в переменных, строках, массивах, объектах, функциях, классах, промисах, деструктуризации, экспортировании и импортировании.

Если у вас возникнут вопросы по материалу книги, со мной можно легко связаться в интернете. Надеюсь, после чтения книги вы убедитесь, что Svelte и Sapper заслуживают вашего внимания, и этим технологиям найдется место в вашем следующем проекте.

СТРУКТУРА

Книга делится на четыре части и состоит из 21 главы.

Часть 1 знакомит читателя с Svelte и Sapper.

- Глава 1 объясняет, почему новые технологии веб-разработки заслуживают вашего внимания. Глава завершается кратким введением в Svelte Native, сравнением Svelte с другими популярными веб-фреймворками и описанием инструментов, которые понадобятся вам для начала работы.
- В главе 2 описывается процесс создания вашего первого приложения Svelte в сетевой среде (REPL). Приложения, построенные в этой среде, можно сохранять, передавать другим разработчикам и экспортировать для локального продолжения разработки. Далее описываются основные этапы локальной разработки приложений Svelte.

В части 2 Svelte рассматривается более подробно, а материал поясняется множеством примеров кода.

- Глава 3 научит вас строить компоненты Svelte. Здесь описываются основные части этих компонентов: логика, разметка и стилевое оформление. Далее рассматривается управление состоянием компонентов, использование реактивных команд и контекста модуля. Глава завершается примером разработки и использования нестандартного компонента.
- В главе 4 рассматриваются блочные структуры Svelte. В этих структурах условная логика, итерации и обработка промисов заключаются вместе с разметкой (чаще всего HTML). Условная логика реализуется структурой `{#if}`, итерации — структурой `{#each}`, а обработка промисов — структурой `{#await}`.
- В главе 5 исследуются некоторые способы взаимодействия между компонентами. В их числе — проп-свойства, двустороннее связывание, слоты, события и контекст.
- В главе 6 описано использование хранилищ для совместного доступа к состоянию между компонентами. Существует четыре разновидности хранилищ: для записи, для чтения, производные и специальные. Затем описываются методы использования хранилищ с классами JavaScript и долгосрочное хранение информации.
- В главе 7 продемонстрированы различные подходы к взаимодействию с DOM в компонентах Svelte. В частности, рассматриваются такие методы, как вставка HTML, использование «действий» для получения доступа к элементам DOM и применение функции `tick` для ручного изменения модели DOM после ее обновления Svelte. Глава завершается описанием способов реализации диалоговых окон и перетаскивания.

- В главе 8 дается обзор функций жизненного цикла. Они позволяют регистрировать функции, которые должны вызываться в ключевых точках жизненного цикла компонентов. К ним относятся функции `onMount`, `beforeUpdate`, `afterUpdate` и `onDestroy`. Наконец, описан способ реализации нестандартных функций жизненного цикла на основе имеющихся.
- Глава 9 демонстрирует три разных подхода к маршрутизации страниц в приложениях Svelte: ручная маршрутизация, `#`-маршрутизация и использование библиотеки `page.js`. Мы разработаем приложение интернет-магазина, чтобы рассмотреть каждый из этих вариантов. Еще один популярный подход основан на использовании `Sapper`. Маршрутизация `Sapper` описана в главе 16.
- В главе 10 рассматривается всесторонняя поддержка анимации, встроенная в Svelte. Подробно разбираются пакеты `svelte/animate`, `svelte/motion` и `svelte/transition`. Описаны два подхода к анимации перемещений элементов между двумя списками: сочетание переходов и `flip`-анимации¹ и плавный переход. Глава завершается разбором создания нестандартных анимаций и использования событий переходов.
- В главе 11 представлены основные методы устранения багов в приложениях Svelte. К ним относятся тег `@debug`, реактивные команды с методами `console` и расширение для браузера `svelte-devtools`.
- Глава 12 описывает различные подходы к тестированию приложений Svelte. Для реализации модульных тестов используется Jest и `svelte-testing-library`. Сквозные тесты реализуются на базе Cypress. Компилятор Svelte обеспечивает некоторую степень проверки доступности. Расширенное тестирование доступности осуществляется в `Lighthouse`, `axe` и `WAVE`. Наконец, `Storybook` используется для демонстрации и ручного тестирования компонентов.
- В главе 13 исследуются несколько вариантов развертывания приложений Svelte: ручное развертывание на сервере HTTP, использование `Netlify`, `Vercel` и `Docker`.
- В главе 14 рассматриваются дополнительные темы, относящиеся к Svelte. Здесь описаны проверка форм, использование библиотек CSS, «специальные элементы», библиотеки компонентов Svelte и генерирование веб-компонентов из компонентов Svelte.

В части 3 в подробностях рассматривается `Sapper`. Построенный на базе Svelte, он добавляет еще больше возможностей.

- В главе 15 рассматривается создание вашего первого приложения `Sapper`. Приложение интернет-магазина, разработанное в главе 9, создается заново с использованием `Sapper`.

¹ Здесь *flip* — сокращение от First, Last, Invert, Play. — Примеч. пер.

- В главе 16 исследуются многие аспекты Sapper. Сначала объясняется стандартная структура файлов в приложениях. Далее описываются некоторые возможности Sapper, включая маршруты страниц, макеты страниц, предварительную загрузку, предварительную выборку и разделение кода.
- В главе 17 рассматриваются серверные маршруты Sapper. Они позволяют реализовать службы API в одном проекте с клиентской стороной веб-приложения. Далее представлен пример реализации службы CRUD.
- Глава 18 показывает, как «экспортировать» приложения Sapper и генерировать на их основе статические сайты. Например, эта возможность может пригодиться приложениям, у которых разметка HTML каждой страницы генерируется во время построения. В конце главы есть пример такого приложения со страницами для вывода информации об игре «камень-ножницы-бумага» и о разных собаках, живущих в моей семье.
- В главе 19 описано, как приложения Sapper поддерживают автономную работу с помощью сервисных работников. Представлены различные стратегии кэширования и стандартные сервисные работники Sapper, а также события сервисных работников `install`, `activate` и `fetch`. Далее я рассказываю, как активизировать использование HTTPS на серверах Sapper. Глава завершается приемами проверки автономного поведения приложений Sapper.

Часть 4 не ограничивается Svelte и Sapper.

- В главе 20 рассматривается предварительная обработка исходных файлов с целью поддержки альтернативных вариантов синтаксиса (таких как Sass, TypeScript и Markdown). Приводятся примеры использования этих альтернатив.
- В главе 21 читатель знакомится с Svelte Native. Эта технология объединяет Svelte и NativeScript для построения мобильных приложений для Android и iOS. Две сетевые среды REPL помогут вам взяться за создание приложений Svelte Native без установки каких-либо дополнительных инструментов. Описываются предоставляемые компоненты для вывода данных, форм, действий, диалоговых окон, макетов и навигации, а также подробно описывается стилевое оформление компонентов Svelte Native. В главе приводится пример приложения, демонстрирующий большинство этих возможностей. В конце описывается библиотека NativeScript UI и пример приложения, использующего один из ее компонентов — `RadSideDrawer`.

Последняя глава — это еще не все! За ней идут семь приложений с полезной информацией.

- В приложении А собраны ссылки на ресурсы по темам Svelte, Sapper, Svelte Native и т. д.

- Приложение Б объясняет, как использовать Fetch API для работы со службами REST.
- В приложении В изложены основы работы с базой данных MongoDB, которая используется в главе 17.
- В приложении Г рассматривается настройка и применение ESLint для выявления проблем в приложениях Svelte и Sapper.
- Приложение Д посвящено настройке и использованию Prettier для форматирования кода в приложениях Svelte и Sapper.
- В приложении Е рассказано об использовании некоторых расширений VS Code при редактировании приложений Svelte и Sapper.
- В приложении Ж описано применение Snowpack для построения приложений Svelte. Snowpack реализует более эффективный способ построения веб-приложений по сравнению с подходом, который обычно используется такими упаковщиками модулей, как Webpack, Rollup и Parcel.

В книге мы будем постепенно разрабатывать приложение Travel Packing. Большинство глав дополняет приложение новыми функциями в соответствии с рассматриваемой темой.

Читателям без опыта использования Svelte стоит прочитать главы 1–8, прежде чем переходить к другим частям книги. В них рассматриваются основные принципы Svelte. Опытные читатели могут сразу переходить к главам, которые представляют для них интерес.

О КОДЕ

Большая часть кода книги доступна в GitHub-репозитории автора по ссылке <https://github.com/mvolkmann>. В частности, см. <https://github.com/mvolkmann/svelte-and-sapper-in-action> и <https://github.com/mvolkmann/svelte-native-components>.

Для запуска кода необходимо установить новейшую версию Node.js. Если она еще не установлена в вашей системе, перейдите на сайт <https://nodejs.org/> и нажмите кнопку **LTS** или **Current** для загрузки.

В книге приводится множество примеров исходного кода — как в нумерованных листингах, так и в тексте. В обоих случаях исходный код оформляется моноширинным шрифтом, чтобы он отличался от обычного текста. Иногда код также выделяется жирным шрифтом для обозначения частей, изменившихся по сравнению с предыдущими разделами главы — например, при добавлении новой возможности к существующему фрагменту кода.

Во многих случаях исходный код переформатируется; я добавляю разрывы строк и изменяю отступы, чтобы эффективно использовать площадь страниц. В редких случаях этого оказывается недостаточно и в листинги включаются маркеры продолжения строк (➡). Кроме того, комментарии исходного кода часто удаляются из листингов, если сам код разбирается в тексте. Многие листинги снабжаются аннотациями, подчеркивающими важные концепции.

ФОРУМ LIVEBOOK

Приобретая книгу, вы получаете бесплатный доступ к закрытому веб-форуму Manning, на котором можно публиковать комментарии по поводу книги, задавать технические вопросы и получать помощь от автора и других пользователей. Чтобы получить доступ к форуму, откройте страницу <https://livebook.manning.com/#!/book/svelte-and-sapper-in-action/discussion>. За информацией о форумах Manning и правилах поведения обращайтесь по ссылке <https://livebook.manning.com/#!/discussion>.

В рамках своих обязательств перед читателями издательство Manning предоставляет ресурс для проведения диалога между читателями и автором. Эти обязательства не подразумевают конкретную степень участия со стороны автора, участие которого в работе форума остается добровольным (и неоплачиваемым). Задавайте автору интересные вопросы, чтобы он не терял интереса к происходящему! Форум и архивы предшествующих обсуждений доступны на веб-сайте издателя, пока книга находится в печати.

ПРОЧИЕ ОНЛАЙН-РЕСУРСЫ

В приложении А приводится список онлайн-ресурсов. Большинство из них имеет прямое отношение к Svelte и Sapper, но некоторые темы применимы ко всем разновидностям веб-разработки.

ОТ ИЗДАТЕЛЬСТВА

Ваши замечания, предложения, вопросы отправляйте по адресу comp@piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На веб-сайте издательства www.piter.com вы найдете подробную информацию о наших книгах.

Об авторе



Марк Волкманн (Mark Volkmann) — партнер компании Object Computing, Inc. (OCI) в Сент-Луисе. Там он занимался консультациями и обучением с 1996 года. Помогал многим компаниям в решении проблем с JavaScript, Node.js, Svelte, React, Vue, Angular и т. д. Марк создал и проводил учебные курсы по многим темам, включая React, Vue, AngularJS, Node.js, jQuery, JavaScript, HTML5, CSS3, Ruby, Java и XML. Он часто выступает с докладами на встречах пользовательских групп в Сент-Луисе. Также он выступал на многих конференциях, включая Nordic.js,

Jfokus, NDC Oslo, Strange Loop, MidwestJS, No Fluff Just Stuff и XML DevCon. Марк часто пишет статьи по различным вопросам, связанным с разработкой. Эти статьи доступны по ссылке <https://objectcomputing.com/resources/publications/mark-volkmann>.

В свободное время Марк любит бегать. На момент написания этой книги он участвовал в 49 марафонах в 39 штатах.

Иллюстрация на обложке

Иллюстрация на обложке *Femme Corfiote* («Женщина с Корфу») позаимствована из коллекции костюмов различных стран, созданной Жаком Грассе де Сен-Совер (Jacques Grasset de Saint-Sauveur) (1757–1810) под названием *Costumes civils actuels de tous les peuples connus*, опубликованной во Франции в 1788 году. Каждая иллюстрация этой коллекции была нарисована и раскрашена от руки. Богатое разнообразие коллекции Грассе де Сен-Совер наглядно напоминает нам, насколько культурно разрознены были города и регионы всего мира еще 200 лет назад. Будучи изолированными друг от друга, люди говорили на различных диалектах и языках. На сельских же улицах было достаточно взглянуть на одяние жителя, чтобы распознать, откуда он родом, кем работает и в каком жизненном положении находится.

Манера одеваться с тех пор сильно изменилась, и некогда богатое региональное разнообразие практически угасло. Сегодня стало сложно различить жителей разных континентов, не говоря уже о разных странах, регионах или городах. Можно предположить, что мы променяли культурное разнообразие на более разностороннюю личную жизнь, но при этом точно на более богатую и быстро меняющуюся жизнь мира технологий.

Во времена, когда стало тяжело отличить одну компьютерную книгу от другой, Manning выделяется изобретательностью и инициативой в компьютерном бизнесе, создавая обложки на основе богатого разнообразия жизни регионов двухвековой давности, которое было повторно явлено миру благодаря работе Грассе де Сен-Совер.

