
Предисловие

Еще одна восхитительная теплая калифорнийская ночь. Легкий океанский бриз лишь усиливает чувство абсолютного блаженства. Место — Лос-Анджелес, время — двухтысячные годы. Я готовлюсь отправить по FTP на свой сервер мое новое небольшое веб-приложение под названием CSSsprites.com и выпустить его в мир. Последние несколько вечеров, работая над приложением, я размышлял над одной проблемой: ну почему 20 % усилий ушло на отладку основной логики работы приложения, а 80 % — на усовершенствование пользовательского интерфейса? Сколько других инструментов я мог бы создать, если бы мне не приходилось все время пользоваться методом `getElementById()` и беспокоиться о состоянии приложения? (Подгрузил ли пользователь что-то на страницу? Что, произошла ошибка? А это диалоговое окно все еще на экране?) Почему разработка пользовательского интерфейса отнимает так много времени? И что происходит со всеми этими разными браузерами? Постепенно мое блаженство таяло, превращаясь в раздражение.

Перенесемся в март 2015 года, на Facebook-конференцию F8. Моя команда готовится сообщить о полной переработке двух наших веб-приложений: средства работы со сторонними комментариями и связанного с ним инструмента их модерации. По сравнению с моим небольшим приложением CSSsprites.com это были полноценные веб-приложения, имеющие куда более существенные возможности, гораздо бóльшую эффективность и нереальные объемы трафика. Несмотря на это, разработка доставляла огром-

ное удовольствие. Члены команды, незнакомые с приложением (а некоторые даже с JavaScript и CSS), вносили свои предложения по поводу возможностей и каких-либо улучшений его различных частей, и мы без промедления и не прилагая особых усилий внедряли их. Как сказал один из членов команды, «теперь я понимаю, что значит любить свою работу».

Что же произошло за это время? Появилась технология React.

React представляет собой библиотеку для создания пользовательского интерфейса, которая помогает вам определить его раз и навсегда. Затем при изменении состояния приложения интерфейс перестраивается, *реагируя* на изменения, и вам не нужно ничего делать дополнительно. В конце концов, вы уже определили интерфейс. Определили? Скорее, вы его *объявили*. А для создания большого полнофункционального приложения вы воспользовались небольшими управляемыми *компонентами*. Теперь уже половина внутреннего кода ваших функций не занимается поиском DOM-узлов, и вам остается лишь отслеживать состояние (state) вашего приложения (с помощью обычных объектов JavaScript), а все остальное происходит в соответствии с этим состоянием.

Изучение React стоит того — вы изучаете всего одну библиотеку и сможете использовать ее, чтобы в дальнейшем создавать:

- веб-приложения;
- приложения для работы под управлением iOS и Android;
- TV-приложения;
- обычные приложения для настольных компьютеров.

Используя ту же идею создания компонентов и пользовательских интерфейсов, вы сможете конструировать приложения, приспособленные под исходную систему с присущими ей производительностью и элементами управления (*реальными* элементами управления, а не похожими на них копиями). Речь идет не о «напиши

один раз, запусти везде» (наша индустрия постоянно терпит в этом неудачу), а о «научиться один раз, использовать везде».

Одним словом, изучение React позволяет сэкономить 80 % вашего времени и сосредоточиться на том, что важно (то есть на реальных задачах, для решения которых предназначено ваше приложение).

Об этой книге

Книга посвящена изучению React с точки зрения веб-разработки. В первых трех главах изучение начинается с использования пустого HTML-файла, на основе которого выстраивается весь остальной код. Это позволяет сосредоточиться на изучении React, не отвлекаясь на новый синтаксис или на применение дополнительных инструментальных средств.

В главе 5 больше внимания уделяется JSX — дополнительной технологии, обычно используемой в сочетании с React.

Далее мы перейдем к изучению приемов разработки реального приложения и освоению дополнительных инструментов, которые могут быть полезны. Чтобы быстро начать работу и свести обсуждение вспомогательных технологий к минимуму, в книге используется приложение `create-react-app`. Цель состоит в том, чтобы сосредоточиться прежде всего на React.

Спорным решением стало включение компонентов классов в дополнение к функциональным компонентам. Вероятно, за функциональными компонентами будущее, однако читатель может столкнуться с существующим кодом и руководствами, в которых говорится только о компонентах классов. Знание обоих синтаксисов удваивает шансы на чтение и понимание кода в реальных условиях.

Успехов вам в изучении React — пусть оно будет приятным и плодотворным!

Условные обозначения

В данной книге используются следующие обозначения.

Курсив

Курсивом выделены новые термины.

Моноширинный шрифт

Используется для листингов программ, а также внутри абзацев для обозначения таких элементов, как переменные и функции, базы данных, типы данных, переменные среды, операторы и ключевые слова, имена файлов и их расширений.

Жирный моноширинный шрифт

Показывает команды или другой текст, который пользователь должен ввести самостоятельно.

Шрифт без засечек

Используется для обозначения URL, адресов электронной почты, названий кнопок, каталогов.



Этот рисунок указывает на совет или предложение.



Этот рисунок указывает на общее примечание.

Использование примеров кода

Дополнительные материалы (примеры кода, упражнения и т. д.) доступны для скачивания по адресу <https://github.com/stoyan/reactbook2>.

Эта книга призвана помочь вам выполнять свою работу. В общем случае все примеры кода вы можете использовать в своих программах и в документации. Вам не нужно обращаться в издательство за разрешением, если вы не собираетесь воспроизводить существенные части кода, или если разрабатываете программу и используете в ней несколько фрагментов кода из книги, или если будете цитировать эту книгу или примеры из нее, отвечая на вопросы. Вам потребуется разрешение от издательства O'Reilly, если вы хотите продавать или распространять примеры из книги либо хотите включить существенные объемы программного кода из книги в документацию вашего продукта.

Мы рекомендуем, но не требуем добавлять ссылку на первоисточник при цитировании. Под ссылкой на первоисточник мы подразумеваем указание авторов, издательства и ISBN.

За получением разрешения на использование значительных объемов программного кода из книги обращайтесь по адресу permissions@oreilly.com.

Благодарности

Хочу поблагодарить всех, кто вычитывал черновые варианты этой книги и присылал отзывы и исправления.

Для первого издания: Андреа Маноле (Andrea Manole), Ильян Пейчев (Iliyan Peychev), Костадин Илов (Kostadin Ilov), Марк Дуппенталер (Mark Duppenthaler), Стефан Албер (Stephan Alber), Асен Божилов (Asen Bozhilov).

Для второго издания: Адам Ракис (Adam Rackis), Максимилиано Фиртман (Maximiliano Firtman), Четан Каранде (Chetan Karande), Кирилл Христов (Kiril Christov) и Скотт Сатоши Ивако (Scott Satoshi Iwako).

Спасибо всем сотрудникам Facebook, которые работают над (или с) React и каждый день отвечают на мои вопросы. Я также благодарен всему сообществу React, которое продолжает создавать великолепные инструментальные средства, библиотеки, публикует статьи и создает примеры применения этой технологии.

Я чрезвычайно благодарен Джордану Уалке (Jordan Walke).

Спасибо всем сотрудникам издательства O'Reilly, содействовавшим выходу этой книги: Анджеле Руфино (Angela Rufino), Дженнифер Поллок (Jennifer Pollock), Мег Фоли (Meg Foley), Киму Коферу (Kim Cofer), Джастину Биллингу (Justin Billing), Николь Шелби (Nicole Shelby), Кристен Браун (Kristen Brown) и многим другим.

Спасибо Явору Вачкову (Javor Vatchkov), дизайнеру пользовательского интерфейса примера приложения, разрабатываемого в данной книге (работу приложения можно оценить на whinepad.com).