

REACTJS\*

## Развёрнутый комментарий Дэна Абрамова к статье «Вещи, о которых никто вам не расскажет про React»

ПЕРЕВОД

KarafiziArthur 5 мая 2017 в 20:40 👁 17,1k

Оригинал: [Dan\\_abramov](#)

**Всем привет!** Недавно Дэн Абрамов, создатель Redux, оставил довольно [массивный комментарий](#) к статье на Medium [Things nobody will tell you about React.js](#), который очень быстро разошёлся популярностью и довольно скоро набрал раза в 3 больше рекомендаций, чем сама статья :)

Собственно, текущая статья является моим переводом его комментария, так как последняя содержит ценные замечания по поводу актуального и будущего состояния React / React Router. Надеюсь, кому-то это будет полезным.

---

Привет, спасибо за обратную связь! :)

Я ценю, что вы поделились своим неприятным опытом работы с React.

Ваш пост содержит широко распространенные в React сообществе заблуждения, поэтому мне захотелось воспользоваться моментом

и разъяснить их для любого, у кого имеются те же проблемы.

Это вовсе не означает, что React для всех работает одинаково хорошо, или что затронутые вами проблемы неактуальны. Но есть несколько моментов, которые, на мой взгляд, важно обозначить для правильного понимания этих проблем.

Я начал писать своё приложение на React 15.5.0, зная, что мой код является устаревшим ещё до начала работы, потому что Facebook только что сообщил, что к следующему мажорному релизу они планируют полностью переписать фреймворк, и это скорее всего означает, что они уничтожат текущий исходный код из-за того, что он больше не может поддерживаться.

Это утверждение некорректно, потому что "полностью переписанное" имеет обратную совместимость. Мы подчёркиваем это всякий раз, когда говорим о следующем релизе, но, тем не менее, стоит повторить ещё раз.

React 16 (который ещё в процессе) *является* "переписанным", но у него такое же публичное API. Из более чем 30 000 (!) компонентов в Facebook только около дюжины нуждается в изменениях, и то, эти несколько компонентов полагались на неподдерживаемое и недокументированное поведение.

Исходя из этого, совместимость равняется буквально 99,9%. Это даёт нам уверенности в том, что React 16 будет работать и с вашим кодом.

Прочитайте больше о нашей [приверженности стабильности без стагнации](#).

Не будучи экспертом в том, как настраивать весь проект с фреймворком (и скептически относясь, чтобы доверить эту работу какому-то cli-инструменту типа [create-react-app](#))

[Create React App](#) это тонкая прослойка поверх Webpack и Babel. Она не генерирует код проекта за вас, но она конфигурирует эти инструменты рекомендуемым образом.

Жалоба по поводу бойлерплейта и стартового набора кода в следующих предложениях мне кажется не вполне справедливой, особенно учитывая, что есть решение этой проблемы, но вы намеренно решили не использовать его.

В этой области ещё много работы, но я считаю, что ситуация улучшилась за последний год, и я надеюсь, вы дадите ему шанс, если вы решите использовать React в будущем снова.

после пары часов рефакторинга я понял, что мой код продолжает расти, и он стал в 3 раза больше, чем до этого, из-за большого количества шаблонного кода, необходимого для Redux (ActionTypes, Actions, Reducers, connect...)

[Не используйте Redux, если в этом нет особой необходимости, так как в нём очевидно много бойлерплейта.](#)

React имеет [встроенную систему состояния](#), и я бы рекомендовал начинать с этого и [поднять состояние на уровень выше](#), когда вам нужно будет распределить его между компонентами.

Вы всегда можете добавить Redux позже, если почувствуете необходимость в нём, но начинать разработку с MobX или Redux — это зачастую мартышкин труд.

React-router официально не поддерживается Facebook и у его разработчиков была просто потрясающая идея выпустить аж 3 мажорные версии за каких-то 5 месяцев

Я не связан с React Router, но я думаю, что это заявление неверно. Если вы следили за релизами React Router, то вы должно быть известно следующее:

- 1.0 была выпущена в 2015.
- 2.0 была выпущена в феврале 2016 и в значительной степени обратно совместима с версией 1.0 (но в ней добавлены предупреждения об изменениях в API, которые будут в версии 3.0)
- 3.0 была выпущена в октябре 2016, и в основном, это та же самая версия 2.0, но с окончательными изменениями в API.

Поэтому, если вы обновлялись только до стабильной версии (учитывая, что вы могли не знать что делаете), то переход с версии 1.0 до 2.0 должен был быть практически без происшествий. Вы должны были видеть кучу предупреждений, и у вас было 8

месяцев, чтобы исправить их прежде, чем следующий мажорный релиз (3.0) удалит эти предупреждения.

Вы абсолютно правы, что версия 4.0 появилась сразу после 3.0. 4.0 это полностью переписанная версия, и она немного отличается своей философией (это сделано для того, чтобы исправить множество ошибок и проблем, о которых сообщалось в Issues React Router на протяжении этих лет). Вы правы, что версия 4.0 вышла сразу после 3.0. Но тут такая вещь — вам совершенно не обязательно переходить на неё. Всё ещё существует множество приложений, использующих 3.0, и её ветка всё ещё активно поддерживается. Поэтому, с вами ничего не случится, если вы останетесь на версии 3.0, и команда разработчиков работает над созданием миграционного патча, чтобы сделать переход на 4.0 версию более простым для тех, кто в этом заинтересован.

Как итог: если вы использовали только стабильные версии релизов, React Router дал вам целый год, чтобы перейти на новый API. Они выпустили 4.x версию для исправления давних проблем, но обязуются поддерживать ветку 3.x сколько потребуется. Звучит не так уж и плохо, на мой взгляд!

Однажды я понял, в чём была проблема, меня затянуло в кроличью нору, заставляя добавлять повсюду пустые `div`, только ради того, чтобы позволить приложению нормально функционировать

вы можете отрендерить функциональные компоненты, просто используя JS функции, но вам нужно заворачивать всю вашу

разметку в какой-нибудь тег, чтобы Virtual DOM создавался корректно

Рад, что вы обратили на это внимание! React 16 будет поддерживать возврат массива из функции `render()`, поэтому вы можете удалить все эти достающие вас `div` и обёрточные теги. Мы усердно работаем, чтобы сделать вашу жизнь лучше.

React кажется фреймворком расширенного рендеринга, который рождён, чтобы заменить использование файлов на серверной стороне (например, PHP) и смешан с HTML, что подтверждает этот твит

Я не совсем понял, что именно вы имеете под этим ввиду. Но если вас интересует, почему React имеет для нас смысл (и ребятам из Airbnb, Twitter, Pinterest и другим компаниям со своим продуктом), то, как я обычно говорю, React, благодаря своему явному потоку данных и отсутствию ручного управления DOM деревом, помогает нам создавать приложения с меньшим количеством багов. И это здорово, если вы нашли другие стратегии для решения этих проблем, ведь все мы учимся друг у друга.

[Эта статья](#) раскрывает подробности того, почему некоторые разработчики считают React полезным, поэтому я рекомендую взглянуть на неё! И также здорово, если React не решает вашей проблемы, или если для решения ваших задач вы попросту предпочли альтернативный фреймворк, например, Ember или Angular.

его кажущаяся простота скрывается за целой вереницей инструментов, которые вам нужно настроить прежде, чем вы приступите к написанию всего одной-единственной строчки кода

Это заявление некорректно, так как при желании вы можете использовать React **без ES6** и **без JSX**. Это так же просто, как прописать один тег скрипта в вашем приложении.

Но мы действительно считаем, что инструменты сборки полезны (учитывая то, что вы упомянули о использовании вами линтера — вы, вероятно, придерживаетесь того же мнения). Хотя их непросто настроить, но мы всё же надеемся, что такие решения, как Create React App сделают их более доступными, и фундаментальные проекты (такие как Webpack) со временем станут более удобными для пользователей (Webpack 2 делает хорошие успехи в этой области со своей проверкой конфигурации). Поэтому я оптимистично смотрю на эти проекты, хотя и понимаю почему люди могли бы захотеть использовать React без них.

вам нужно импортировать в ваши скрипты `react-dom` и `react`, без использования когда-либо последнего, по какой-то дикой причине, известной только самой команде React

Причина в том, что преобразование JSX кода требует наличия React в области видимости, потому что именно так JSX работает с глобальными переменными браузера. И это как раз связано с вашей предыдущей жалобой: мы хотим, чтобы JSX работал даже без использования какого-либо сборщика!

вы должны использовать `className` вместо `class` чтобы определить CSS класс для DOM

Вы полностью правы, это раздражает. Это одно из ранних дизайнерских решений в соответствии с DOM API, которое сбивает с толку. Мы можем изменить это в будущем!

а что вы ожидали от фреймворка, у которого в Github Issues более 1000 обращений

React Native имеет куда более широкий API, и он очень популярен, поэтому естественно, что он получает множество обращений в issues. Многие из них это запросы на поддержку, но мы признательны любой помощи в исправлении багов и помощи людям в решении их проблем.

что позволит вам установить альфа-зависимости по умолчанию (React@16.0.0-alpha.6) чтобы разработать ваше нативное приложение?!?

Версия пакета React в основном не подходит для пользователей React Native, поскольку содержит очень мало кода (`Component` и `createElement`). Код `reconciler` синхронизируется с React Native отдельно. Таким образом, это артефакт различных циклов релиза RN и React, но это совсем не означает, что приложения RN используют нестабильную версию React. Это точно такая же версия, которую мы используем в Facebook для production. Хотя я согласен, это сбивает с толку, и мы надеемся исправить процесс



выпуска новых версий между React и React Native в ближайшем будущем.

Подводя итог, я благодарен, что вы подняли эти проблемы в своей статье. Они появляются очень часто, и подумал, что это важно — отделить зёрна от плевел. Но для нас и для экосистемы React определённо гораздо больше работы, чтобы предоставить больше удобства для работы пользователям React, и я рад, что вы смогли сформулировать эти общие проблемы, чтобы мы смогли лучше их решить.

**Отредактировано:** я бы хотел поблагодарить всех, кто указал на пассивную агрессивность первой версии этого комментария. Я сильно извиняюсь за это! Я отредактировал его в более профессиональном тоне, что я и должен был изначально сделать перед публикацией. Я также удалил все смайлики, но вы всё ещё можете найти их в [моём Twitter](#).

Проголосовать:



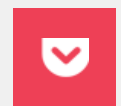
+28



Поделиться:



Сохранить:



Комментарии (3)

## Похожие публикации

---

### Архитектура модульных React + Redux приложений

marshinov • 14 апреля 2017 в 03:27

41

### Эволюция на React+Redux

Fen1kz • 7 апреля 2017 в 10:21

28

### Упрощаем универсальное/изоморфное приложение на React + Router + Redux + Express

dfuse • 9 марта 2017 в 09:36

22

## Популярное за сутки

---

### Яндекс открывает Алису для всех разработчиков. Платформа Яндекс.Диалоги (бета)

BarakAdama • вчера в 10:52

69

### Почему следует игнорировать истории основателей успешных стартапов

ПЕРЕВОД

m1rko • вчера в 10:44

20

### Как получить телефон (почти) любой красотки в Москве, или интересная особенность MT\_FREE

24

cab404 • вчера в 20:27

## Java и Project Reactor

zealot\_and\_frenzy • вчера в 10:56

10

## Пользовательские агрегатные и оконные функции в PostgreSQL и Oracle

erogov • вчера в 12:46

6

## Лучшее на Geektimes

### Как фермеры Дикого Запада организовали телефонную сеть на колючей проволоке

NAGru • вчера в 10:10

31

### Энтузиаст сделал новую материнскую плату для ThinkPad X200s

alizar • вчера в 15:32

49

### Кто-то посылает секс-игрушки с Amazon незнакомцам. Amazon не знает, как их остановить

Pochtoycom • вчера в 13:06

85

### Илон Маск продолжает убеждать в необходимости создания колонии людей на Марсе

marks • вчера в 14:19

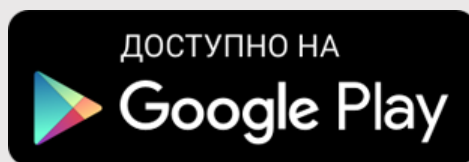
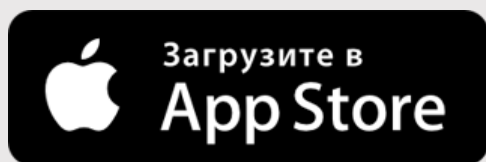
140

# Дела шпионские (часть 1)

TashaFridrih • вчера в 13:16

16

Мобильное приложение



Полная версия

2006 – 2018 © TM