

JAVASCRIPT\*

## It's the future

ПЕРЕВОД

alex\_blank 20 августа 2016 в 12:07 👁 79,8k

Оригинал: [Boopathi Rajaа](#)

*Этот пост просто шутка и не пытается выставить инструменты, упомянутые здесь, в дурном свете. Я использую их постоянно, они великолепны, и я рекомендую их использовать. По мотивам [It's the future @ CircleCI Blog](#)*

— Эй, я бы хотел научиться писать крутые веб-приложения. Слышал, у тебя есть опыт.

— Да, я как раз занимаюсь фронтендом, юзаю пару тулз.

— Круто. Я щас делаю простое приложение — обычный TODO-лист, используя HTML, CSS и JavaScript, и планирую заюзать JQuery. Это норм?

— Не-не-не. Это олдскул. Джиквери мёртв — никто не использует его теперь! Тебе нужен React. Это будущее.

— Окей, лады. А что это?

— React это новый способ разработки веб-приложений. Оно базируется на Virtual DOM и позволяет писать JSX-код.

— Virtual-что? JSX? Что это за..?

— JSX — это HTML на стероидах. Это расширенная версия JavaScript, где вы смешиваете XML и JS в одном коде. VirtualDOM означает, что у тебя есть дерево объектов представляющих часть реального DOM, и оно дает возможность манипулировать им очень быстро без необходимости работать с DOM напрямую.

— Зачем смешивать XML и JS?

— Ну как зачем. Реакт это будущее. Он позволяет создавать переиспользуемые компоненты.

— Типа как в Backbone, да?

— Нет. Бэббон мёртв. Щас теперь все будет на компонентах.

— Ну так мне не нужно заморачиваться про JSX или VirtualDOM?

— Неа. Но неплохо бы понимать как они работают, чтобы ты мог не думать о том, что у тебя что-то где-то тормозит, и оптимизировать код там, где это реально имеет смысл — типа состояния компонента.

— Окееей, я начинаю слегка теряться. Итак, у нас есть некая шняга для написания компонентов, называется React. Могу я использовать её с JQuery?

— Ну, ты можешь написать одну часть приложения на React, а другую на чём захочешь. Но я ж говорю тебе — JQuery мёртв. Кстати, тебе стоит обратить внимание на Webpack, чтобы склеивать компоненты вместе при сборке.

— ОК. Что это такое?

— Это бандлер модулей. Ты пишешь кучу файлов, как если бы это были модули Node — и затем комбинируешь их в один монолитный файл (или разбитый на части) для отправки клиенту. Тебе даже не понадобятся react-tools, можешь обойтись одним Babel для компиляции JSX.

— Babel?

— Ага, Babel. Это клевая хрень, которая транспирует твой ES6+/JSX код в читабельный ES5 код, с поддержкой sourcemaps. Это широко распространено, даже в Фейсбуке это используют.

— ES6+?

— ES6/ES2015 это следующая версия ECMAScript, там очень много новых фишек, типа классов, стрелочных функций, Maps, Sets и так далее... Щас почти все юзает ES6+.

— Это настолько хорошая вещь?

— Спрашиваешь! ES6+ + транспиляция это будущее.

— Ну ладно. Как мне это заюзать?

— Начни с установки Node...

— Установить Node? О, нет. Ты ведь говорил что-то про React и компоненты.

— Ну да. Дальше ты берешь Flux архитектуру и начинаешь создавать actions, stores и компоненты.

— Flux?

— Во Flux ты выстраиваешь свое приложение как набор stores, actions и view, где данные «текут» в одном направлении. Views вызывают actions, они проходят через dispatcher, а stores подписываются на события dispatcher и выдают change events, на которые подписан view.

— Ага. Как в MVC?

— Нет. MVC мёртв. Flux это будущее. Сейчас очень много реализаций flux.

— Что? Реализаций Flux?

— Да, ведь Flux это просто паттерн. Реализаций полно: Alt, Flumмох, marty, fluxxor, fluxible, и т.п... И даже есть фейсбучная реализация Dispatcher.

— Мне нужно это все использовать?

— Новичку довольно сложно закодировать самому весь этот шаблонный код, поэтому возьми какую-нибудь готовую реализацию.

— Ладно. Мне бы не хотелось писать эту жуть самому.

— Я ж говорю, реализаций дофига.

— А что насчёт Angular?

— Фу.

— Фу?

— Фу.

— Смотри. Мне реально не хочется возиться с установкой, настройкой и допиливанием чего-то сложного.

— Вообще-то это легко. Существуют готовые киты/репозитории для новичков, или можешь заюзать генераторы Yeoman, которые сделают это за тебя.

— Мне нужен генератор? Yeoman? Что это еще такое?

— Он может сгенерировать код за тебя, и ты можешь заюзать его для создания нескольких приложений сразу. Используй поддержку

DLL в новом Webpack, и можешь компилировать каждое приложение отдельно от других.

— У меня будет только одно приложение, одна страничка, один компонент, одно что-угодно. Только одно.

— Нет. Изучи комбинируемые компоненты. Это то, как мы делаем всё сейчас. Тебе нужно разбить всё на компоненты — по одному на каждую отдельную маленькую фигню.

— Кажется, что это избыточно.

— Это единственный способ добиться аккуратности, удобства и производительности. Ты сможешь использовать штуки вроде горячей перезагрузки (hot reload)...

— Hot Reload? Типа как livereload?

— Не. Webpack поддерживает эту замечательную фицу, она называется горячей перезагрузкой модулей, и есть плагин react-hot-loader для реакта, так что ты сможешь менять код отдельного компонента без перезагрузки всей страницы. А вместе с Flux ты сможешь делать ваще башнесрывные вещи типа прокрутки истории изменений в данных туда-обратно — одновременно с горячим редактированием кода.

— Итак. Теперь у меня десятки различных инструментов и библиотек для упаковки, сборки, транспилирования и чего угодно. Еще что-то?

— Как я уже говорил, Flux дает возможность выстраивать хорошую архитектуру приложений. Но с Observables и Immutable данными всё становится намного лучше и интереснее.

— Observable? Мне нужен Immutable?

— Тебе нужны observables чтобы удобно работать с событиями и асинхронностью, а Immutable.js нужен для персистентности, эффективности и простоты. Observables это типа как массивы, только асинхронные. Они возвращают значения с помощью async-генераторов из ES2016.

— Что за async generator?

— Ну смотри, обычный генератор дает возможность функции возвращать серию значений, а с модификатором async ты можешь возвращать значения в будущее. Правда, автор этого, Джафар Хусейн, уже отозвал свой proposal и работает с Кевином Смитом над более узкоспециализированным es-observable для ES7.

— Кхм. Я просто хочу запустить простое приложение. Насколько же глубока кроличья нора?

— Ну, ты можешь использовать RxJS, который ближе к оригинальным observable. Оно широко распространено и годится для продакшена.

— RxJS? Чем же он полезен?

— Работает с твоими уже написанными promises и событиями. Ты можешь создать Observable из любого асинхронного кода, и работать с ним как с обычным значением. Но вообще-то, если ты ищешь что-то по-настоящему реактивное и интересное, позырь фреймворк Cycle.js, разработанный Andre Staltz.

— WTF. Мне всего лишь нужно написать и запустить простое приложение. Смогу я сделать уже это или нет?

— Конечно, хотя деплоинг это отдельный интересный вопрос, но мы уже почти закончили.

— Окей, «сегодня мы многое поняли». Спасибо за объяснения.

— Отлично! Нет проблем.

— Давай я повторю тогда, чтобы мы удостоверились, что я всё правильно понял. Итак, мне нужно разбить код своего приложения на actions, stores и компоненты, выстроить однонаправленный dataflow, писать ES6+ код чтобы занять все последние языковые фишки, позволяющие мне писать чистый код, затем использовать Babel для транспиляции этого ES6+ кода в ES5 код, пригодный для всех браузеров, использовать webpack, чтобы склеить все части моего приложения, написанные как модули node в один файл, использовать ImmutableJS для представления моих данных и подключить RxJS для описания событий и других асинхронных функций.



— Да. Разве это не восхитительно?

— И, да, я еще кажется забыл про статику и минификацию.

— Не проблема вообще. Webpack умеет это. Все что тебе нужно, это настроить несколько загрузчиков, плагинов — и вот уже почти и всё. Ты можешь импортировать CSS и картинки. Кстати, есть еще альтернативы CSS, которые позволяют описывать стили в JS...

— Я возвращаюсь к JQuery.

Проголосовать:



+156



Поделиться:



Сохранить:



Комментарии (283)

## Похожие публикации

Использование React JSX вместе с TypeScript 1.6

ИЗ ПЕСОЧНИЦЫ

5

## Мастер-классы по профессиональному Javascript, AJAX/COMET, jQuery

31

iliakan • 25 августа 2010 в 13:31

## Мастер-классы по javascript, AJAX/COMET, jQuery в Киеве и Москве

23

iliakan • 4 июня 2010 в 12:47

## Популярное за сутки

### Наташа — библиотека для извлечения структурированной информации из текстов на русском языке

14

alexkuku • вчера в 16:12

### Unit-тестирование скриншотами: преодолеваем звуковой барьер. Расшифровка доклада

4

lahmatiy • вчера в 13:05

### Люди не хотят чего-то действительно нового — они хотят привычное, но сделанное иначе

25

ПЕРЕВОД

Smileek • вчера в 10:32

## Руководство по SEO JavaScript-сайтов. Часть 2. Проблемы, эксперименты и рекомендации

ПЕРЕВОД

ru\_vds • вчера в 12:04

2

## Как адаптировать игру на Unity под iPhone X к апрелю

P1CACHU • вчера в 16:13

0

## Лучшее на Geektimes

### Стивен Хокинг, автор «Краткой истории времени», умер на 77 году жизни

HostingManager • вчера в 13:49

33

### Обзор рынка моноколес 2018

lozga • вчера в 06:58

70

### «Битва за Telegram»: 35 пользователей подали в суд на ФСБ

alizar • вчера в 15:14

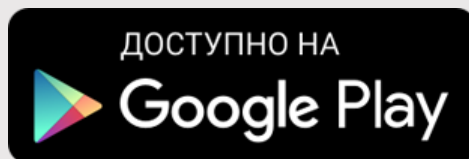
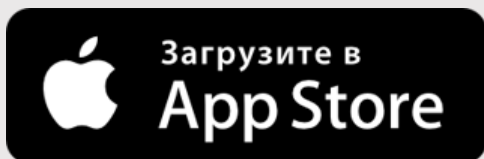
40

### Стивен Хокинг и его работа — что дал ученый человечеству?

marks • вчера в 14:46

8

Мобильное приложение



Полная версия

2006 – 2018 © TM