**Создание PWA на языке Java Script**

**Writing a PWA in Java Script**

Ошурков А. Ю.

*Нижнетагильский государственный социально-педагогический институт, филиал Российского государственного профессионально-педагогического университета, РФ, г. Нижний Тагил*

Oshurkov A. Y.

Nizhny Tagil State Socio-Pedagogical Institute, branch of the Russian State Vocational Pedagogical University, Russia, NizhnyTagil

**Аннотации.**

Типичный сайт состоит из множества HTML-страниц. Нажимая на ссылки, браузер загружает новые страницы по этим ссылкам, вы чувствуете движение с одной страницы на другую. Страницы могут храниться в виде файлов на каком-либо сервере или сгенерированы по вашему запросу какой-либо серверной программой. Каждая страница сайта — это отдельная техническая сущность, отдельный документ, между которыми вы перемещаетесь. С другой стороны, мобильные приложения, похоже, стоят на месте. В них загружаются данные, меняются экраны, но кажется, что мы всегда внутри этого приложения. В начале 2010-х годов появилась новая концепция - нечто среднее между веб-сайтом и приложением. Эта архитектура называется SPA.

**Annotation.**

A typical site consists of many HTML pages. By clicking on the links, the browser loads new pages from these links, you get the feeling of movement from one page to another. Pages can be stored as files on some server or generated for your request by some server program. Each page of the site is a separate technical entity, a separate document, between which you move. Mobile applications, on the other hand, seem to stand still. Data is loaded into them, screens change, but it seems that we are always inside this application. In the early 2010s, a new concept emerged - a cross between a website and an application. This architecture is called SPA.

**Ключевые слова:** SPA, PWA, React, Vue, Angular, JS, web-programming, Ajax.

**Актуальность.**

SPA — Single Page Application. SPA или Single Page Application — это одностраничное веб-приложение, которое загружается на одну HTML-страницу. Благодаря динамическому обновлению с помощью JavaScript, во время использования не нужно перезагружать или подгружать дополнительные страницы. На практике это означает, что пользователь видит в браузере весь основной контент, а при прокрутке или переходах на другие страницы, вместо полной перезагрузки нужные элементы просто подгружаются. Пользователи VK.com или Facebook.com работают с такими продуктами.

Для написания SPA нужно знать JS. Часто для этого используют фреймворки — React, Angular, Vue и другие. SPA работает так: пользователь открывает страницу, браузер загружает сразу весь код приложения написанный на JavaScript. Но показывает только конкретный модуль — часть сайта, которая нужна пользователю. Когда пользователь переходит в другую часть приложения, браузер берёт уже загруженные данные и показывает ему. И, если нужно, динамически подгружает с сервера нужный контент без обновления страницы. С одной стороны, такие приложения работают быстро и меньше нагружают сервер. С другой стороны, они требуют большей загрузки на старте.

В процессе работы пользователю может показаться, что он запустил не веб-сайт, а десктопное приложение, так как оно мгновенно реагирует на все его действия, без задержек.

**Инструменты разработки.**

Самыми распространенными по количеству пользователей на Github и по количеству скачиваний инструментами для разработки веб-приложений являются: React, Angular и Vue. Но также с 2019 года набирает популярность Svelte (рис. 1).

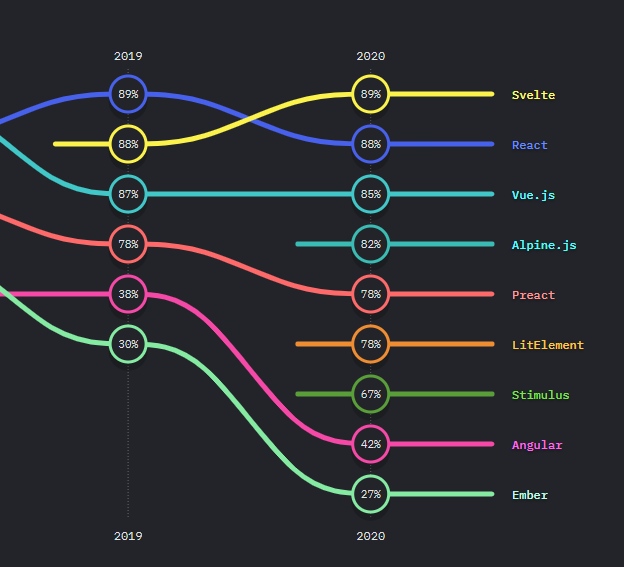


Рис. 1. Рейтинг инструментов для веб-разработки за 2019-2020 гг.

**Svelte** — это принципиально новый подход к созданию пользовательских интерфейсов. Когда традиционные фреймворки, такие как React и Vue, выполняют основную часть своей работы в браузере, Svelte переносит эту работу на этап компиляции, который происходит при сборке приложения. Вместо использования таких тяжелых приёмов, как сравнение состояний виртуального DOM, Svelte пишет код, который точечно обновляет DOM при изменении состояния вашего приложения.

**React. React** уже третий год подряд занимает первую строчку большинства рейтингов. Некоторые разработчики вовсе не имеют нареканий в сторону этого фреймворка, поскольку он стремительно развивается и становится более стабильным. Примечательно, что на самом деле, React — это не фреймворк, а библиотека. Но она имеет настолько широкий функционал, что зачастую инструмент можно использовать без дополнительных интеграций. Frontend в результате будет не менее эффектным.

**React** создан командой Facebook в 2013 году с целью разделения пользовательского интерфейса на набор компонентов, чтобы упростить процесс разработки.

Одна из ключевых причин популярности React — поддержка со стороны авторитетной компании Facebook. На базе этой среды разработки сегодня работают такие сервисы, как Instagram, Whatsapp и Twitter.

**Vue.** Vue стал открытием последних лет. Внезапно, из рядовой среды разработки он стал одной из самых любимых среди профессионалов.

Примечательно, что популярность выросла без поддержки крупных компаний. Гиганты рынка обратили внимание на Vue только после резкого скачка. Именно поэтому он оказался третьим фреймворком в длительном противостоянии **React и Angular.**

Vue сейчас выделяется тем, что он имеет огромное количество особенностей. Некоторые даже могут выступать в качественедостатков. Например, Vue.js очень гибкий. Но в случае работы над большими проектами в большой команде это может поспособствовать большему количеству ошибок.

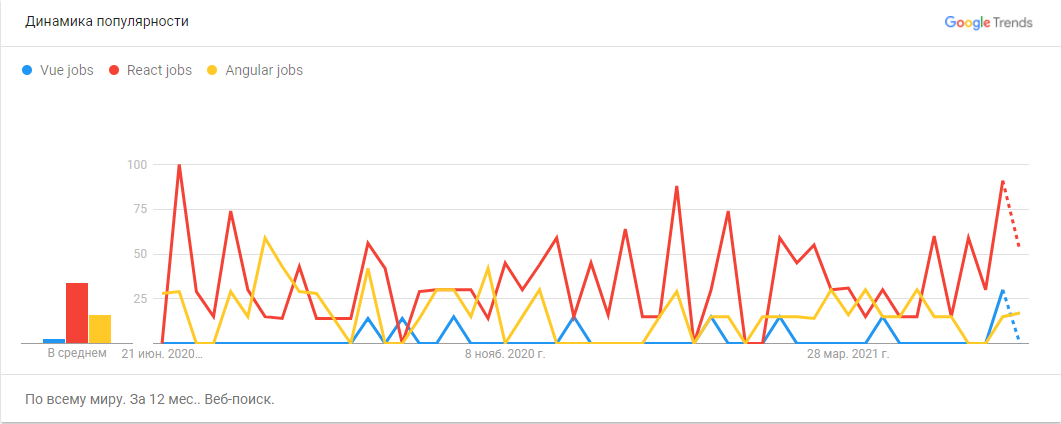
После того, как Vue.js начал проявлять свои особенности, на него обратили внимание многие гиганты рынка, среди которых Gitlab, EuroNews. полностью созданы на базе Vue, а Alibaba и Xiaomi в 2018 году объявили о полном переходе на Vue.

**Angular. Angular** долгое время был лучшим выбором для разработки пользовательского интерфейса. Так, он стал частью популярного стека MEAN. MEAN — это стек для веб-разработки с открытым исходным кодом, который в основном используется для создания облачных приложений. Приложения на основе стекаMEAN являются гибкими, масштабируемыми и расширяемыми, то есть превосходно подходят для размещения в облаке.

Тем не менее, за последние несколько лет разработчики все больше жаловались на неполноценность фреймворка в сравнении с конкурентами. Как результат, многие компании постепенно отказываются от использования **Angular.**

Во многом, фреймворк потерял свою популярность из-за появления новых трендов веб-разработки. Команда Angular не реализовала необходимый функционал в новых версиях среды разработки. Поэтому сегодня мы наблюдаем, что Vue и React становятся более предпочтительным выбор. Если выходить за рамки этих трех фреймворков, то Angular иногда уступает и другим инструментам (например, Svelte).

Исходя из статистики, результатов опросов и разнообразных отчетов можно сделать вывод, что наиболее предпочтительный выбор в 2021 – это React. Второе место занимает Vue, а Angular останется на третьем с большим отставанием (рис. 2).

Рис. 2. Динамика популярности фреймворков

**Достоинства и недостатки SPA.**

Одностраничные приложения чаще всего используют в сервисах, где пользователь проводит на одной странице много времени или совершает с ней какие-то действия. Например, веб-версии Gmail, Facebook Netflix, AirBnB и Pinterest — одностраничные приложения. Технология настолько распространена, что её используют даже для сайтов компаний.

SPA может обмениваться данными с сервером без перезагрузки страницы, с помощью AJAX-запросов. Благодаря этому наполнение страницы может меняться динамически. Например, раньше в социальной сети нужно было перезагрузить страницу, чтобы проверить, нет ли новых сообщений, теперь они появляются автоматически.

Преимущества SPA

**SPA быстрые.** Переход между модулями в приложении происходит быстрее: нужные ресурсы уже загружены, нужно просто подставить данные, которые запросил пользователь. Часто при этом сервер возвращает не тяжеловесный HTML, а лёгкий JSON или XML.

**SPA гибкие**. Раз пользователь всё время работает с одной страницей, проще делать интересные переходы и анимацию элементов. Можно работать с состоянием кнопок, вкладок и переключателей. Таким образом, интерфейс SPA больше похоже на полноценное приложение, а не на сайт.

**SPA работают везде**. Всё, что нужно для SPA — поддержка JavaScript. Такие сайты хорошо работают и на десктопе, и в вебе, могут отчасти заменить полноценные мобильные приложения.

Производительность при первой инициализации сайта на стороне клиента будет уступать классическим сайтам, но при роутинге (переходам по страницам) выбору фильтров, оформления покупки и т.д. (AJAX), то SPA в десятки раз выигрывает по производительности у MPA приложений, т.к. время на отрисовку при каждом действии всей страницы требуется гораздо больше, нежели чем SPA приложению (т.к. перерисовывается только то, что изменилось и не более того). Стоит помнить, что самая дорогая операция в web — это рендер. Так же, с помощью SSR (Server side rendering) мы можем реализовывать следующую технику: загружать только те части JavaScrip или CSS, которые необходимы для работы конкретного компонента

Недостатки SPA

Проблемы с SEO. Поисковая оптимизация (англ. *search engine optimization, SEO*) — комплекс мероприятий по внутренней и внешней оптимизации для поднятия позиций сайта в результатах выдачи поисковых систем по определённым запросам пользователей, с целью увеличения сетевого трафика [1]. По умолчанию у приложений напряжённые отношения с поисковыми машинами: те натренированы индексировать отдельные страницы, у каждой из которых есть заголовок, описание и остальные метатеги. Для SEO в мире SPA существуют SSR фреймворки: Nuxt.js для Vue, Next.js для React и Universal для Angular. Когда пользователь впервые обращается к сайту, то на стороне сервера делаются все необходимые запросы к API для получения данных, «раскрывается» весь HTML, в итоге на клиент улетает уже развернутая HTML, со всеми необходимыми данными, а дальше веб-приложение начинает работать как обычное SPA.

Зависимость от интернета. Для запуска веб-приложения нужна связь с сервером, так что в большинстве случаев без интернета не обойтись, как и с обычными сайтами. Этим SPA проигрывают обычным приложениям, но есть исключение — если во время первой загрузки браузер получает все данные и больше ничего подгружать не нужно, то можно работать и без интернета.

Так же минусом можно выделить долгую загрузку веб-приложения на стороне клиента при первом запуске, так как сервер сразу отправляет все данные о приложении.

**Краткая информация о PWA.**

Так же стоит выделить прогрессивные приложения или Progressive Web Application взаимодействуют с пользователем, как приложение. Они могут устанавливаться на главный экран смартфона, отправлять push-уведомления и работать в офлайн-режиме, например, Google Docs. PWA это следующая ступень развития SPA, т.е. еще более функциональная. PWA умеет все то, что умеет SPA, и еще в придачу пару новых функций: PWA-приложение можно установить иконкой в телефоне и запускать с рабочего стола как приложение. Оно даже будет выглядеть как приложение, и его можно продвигать в Google Play и Apple Store. PWA умеет отправлять push-уведомления (в браузер и на телефон). Самая интересная функция PWA — это возможность работать в приложении без подключения к интернету. То есть у вас не сайт, а кроссплатформенное мобильное приложение.

Cамая интересная функция PWA — это возможность работать в приложении без подключения к интернету. То есть у вас не сайт, а кроссплатформенное мобильное приложение. Главными преимуществами являются:

* кроссплатформенность — могут работать сразу с несколькими операционными системами;
* высокая скорость работы и возможность запуска и отображения данных в офлайн-режиме с моментальной загрузкой;
* высокая скорость установки;
* быстрая разработка — для создания PWA, не нужен отдельный сайт, достаточно изменить уже существующий.

Недостатки:

* не все браузеры поддерживают основные функции таких приложений (например, Firefox и Edge).

SPA и PWA — это веб-сайты, которые постепенно смещают со своих позиций классические сайты. Так происходит из-за того, что они более простые в разработке, быстрее работают.

**Используемые технологии**

Для реализации PWA нужно продвинутое понимания Java Script, языка программирования высокого уровня, который даёт возможность реализовывать сложное поведение веб-страницы. Считается, что JavaScript сложнее изучить, чем связанные с ним технологии, наподобие HTML и CSS. Поэтому, перед изучением JavaScript, настоятельно рекомендуется сначала ознакомиться хотя бы с этими двумя технологиями. Следующим, что нужно знать, это node.js, программная платформа, основанная на движке V8, превращающая JavaScript из узкоспециализированного языка в язык общего назначения [2]. Так же понадобится пакетный менеджер npm, который идет в комплекте с Node.js. Альтернативой для npm может служить пакетный менеджер Yarn. Yarn — это новый менеджер пакетов, совместно созданный Facebook, Google, Exponent и Tilde [3]. Как можно прочитать в официальной документации, его целью является решение целого ряда проблем, с которыми столкнулись разработчики при использовании npm, а именно:

* быструю и последовательную установку пакетов;
* наличие проблем с безопасностью, так как npm позволяет пакетам запускать код при установке.

**Итоги**

SPA и PWA — это веб-сайты, которые постепенно смещают со своих позиций MPA сайты. Так происходит из-за того, что они более простые в разработке, быстрее работают и нравятся пользователям. Однако у них есть слабое место — SEO-оптимизация. Пока еще не все браузеры могут с ними нормально работать, поэтому, чтобы сделать такие приложения дружественными для SEO, нужно прибегать к ряду ухищрений. MPA-сайты в этом плане более простые и надежные.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

**Ресурсы Интернет**

1. Статья про SEO: сайт — URL: https://seo.ru/. Режим доступа: свободный. — Текст: электронный.
2. Официальная документация Node.js: сайт — URL: <https://nodejs.org/en/about/>. Режим доступа: свободный. — Текст: электронный.
3. Официальная документация пакетного менеджера Yarn: сайт — URL: <https://classic.yarnpkg.com/en/docs>. Режим доступа: свободный. — Текст: электронный.