

СТАТЬЯ

УДК 004.65

**РАЗРАБОТКА КЛИЕНТСКОЙ ЧАСТИ ОДНОСТРАНИЧНОГО  
WEB-ПРИЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕКИ REACT**

**Потовиченко М.А., Шатилов Ю.Ю.**

*Донской государственный технический университет, Ростов-на-Дону, e-mail: art-web@mail.ru*

В настоящее время сеть Интернет прочно вошла в жизнь почти каждого человека, стала неотъемлемой средой поддержки бизнес-процессов. С каждым годом технологии развиваются, и возникает необходимость в web-приложениях, которые имеют быстрый отклик на действия в пользовательском интерфейсе и отлично работают на всех устройствах (как на стационарных, так и мобильных). Одновременно с увеличением скорости разработки инфраструктурных веб-приложений поднимается вопрос о выборе программных продуктов для воплощения в жизнь эффективной, современной и отказоустойчивой системы. Поэтому обрели популярность Single Page Application, особенностью реализации которых является модульная структура. В статье рассматривается разработка архитектуры клиентской части одностраничного приложения, его особенности и реализация с помощью современной JavaScript-библиотеки React, приведен сравнительный анализ использования и производительности библиотек для разработки клиентской части приложения React и AngularJS. React позволяет разработчикам моделировать состояние интерфейсов и декларативно описывать их. React – это всего лишь JavaScript, у библиотеки очень маленький API для изучения, всего несколько функций и способы их использования, что позволяет быстро изучить эту библиотеку.

**Ключевые слова:** одностраничное приложение, JavaScript, ReactJS, SPA, Frontend, архитектура, модульность, компонент, HTML, DOM

**DEVELOPMENT OF A CLIENT PART OF SINGLE-PAGE  
WEB APPLICATION USING REACT**

**Potovichenko M.A., Shatilov Yu. Yu.**

*Don State Technical University, Rostov-on-Don, e-mail: art-web@mail.ru*

Currently, the Internet has become an integral part of almost every person's life and has become an integral environment for supporting business processes. Every year, technologies are developing, and there is a need for web applications that have a quick response to actions in the user interface and work perfectly on all devices (both stationary and mobile). Simultaneously with the increase in the speed of development of infrastructure web applications, the question of choosing software products for implementing an effective, modern and fault-tolerant system is raised. This is why Single Page Application has become popular, with its modular structure being a feature of its implementation. The article describes the architecture development client side single page application, its features and implementation using modern JavaScript libraries like React, given the comparative invalid usage and performance libraries for developing client side and React.

**Keywords:** single-page application, JavaScript, ReactJS, SPA, Frontend, architecture, modularity, component, HTML, DOM

В настоящее время сеть Интернет прочно вошла в жизнь почти каждого человека, стала неотъемлемой средой поддержки бизнес-процессов [1]. С каждым годом технологии развиваются, и возникает необходимость в web-приложениях, которые имеют быстрый отклик на действия в пользовательском интерфейсе и отлично работают на всех устройствах (как на стационарных, так и мобильных). Одновременно с увеличением количества таких систем в корпоративной инфраструктуре поднимается вопрос о выборе программных продуктов для воплощения в жизнь эффективной, современной и отказоустойчивой системы [2]. Поэтому обрели популярность Single Page Application (далее SPA).

SPA-приложения – web-приложения, которые используют HTML-документ как оболочку для всех веб-страниц и организуют обмен информацией с пользова-

телем через динамически подгружаемые HTML, CSS, JavaScript.

Целью работы является проанализировать существующие фреймворки для создания веб-приложений, определить их особенности, плюсы и недостатки по различным характеристикам, выбрать подходящий фреймворк для реализации клиентской части приложения.

**Материалы и методы исследования**

Главным SPA-архитектуры является представление (View) – именно с этим взаимодействует пользователь системы. Зачастую представления основаны вокруг шаблонов (Template) – заготовок, которые преобразуются в HTML. Чтобы хранить информацию, необходима модель (Model). Моделью является набор данных и функций, необходимых для работы с событиями и данными. Вся информация модели

полностью сохраняется в памяти (Storage). Для сохранения целостности информации представление отслеживает любые изменения данных в модели. Аналогично модель откликается на оповещение представления и предоставляет непрерывное взаимодействие WEB-сервиса с сервером, исполняя запросы на передачу информации (в том числе с использованием REST) (рис. 1) [3; 4].

Особенностью реализации SPA является модульная структура. Это необходимо для того, чтобы упростить рефакторинг и тестирование, также улучшить и упростить обслуживание приложения.

Модульная архитектура подразумевает разделение приложения на отдельные пакеты с явными зависимостями. Каждый пакет – файл и модуль [4].

JavaScript – это язык Всемирной паутины. Изначально он представлял собой средство управления определенными типами элементов веб-страниц (например, изображения и поля форм), но с тех пор этот язык быстро развивается. На данный момент по-

мимо создания клиентских сценариев, выполняемых браузером, JavaScript применяется для разработки кроссплатформенных программ [5]. JavaScript входит в список основных технологий современностей, которые необходимо знать и уметь применять веб-программистам: HTML (язык разметки), CSS (язык стилей) и язык программирования JavaScript, определяющий поведение веб-страниц [6].

Для реализации клиентской части программы используется JavaScript-библиотека ReactJS. ReactJS – мощный инструмент для реализации интерфейсов для пользователей. Цель данного инструмента – предоставить легкость и масштабируемость приложения, обеспечение вывода на экран всего того, что свойственно веб-страницам с высокой скоростью [7]. React позволяет разработчикам моделировать состояние интерфейсов и декларативно описывать их. React – это всего лишь JavaScript, у библиотеки очень маленький API для изучения, всего несколько функций и способы их использования, что позволяет быстро изучить ее [8].

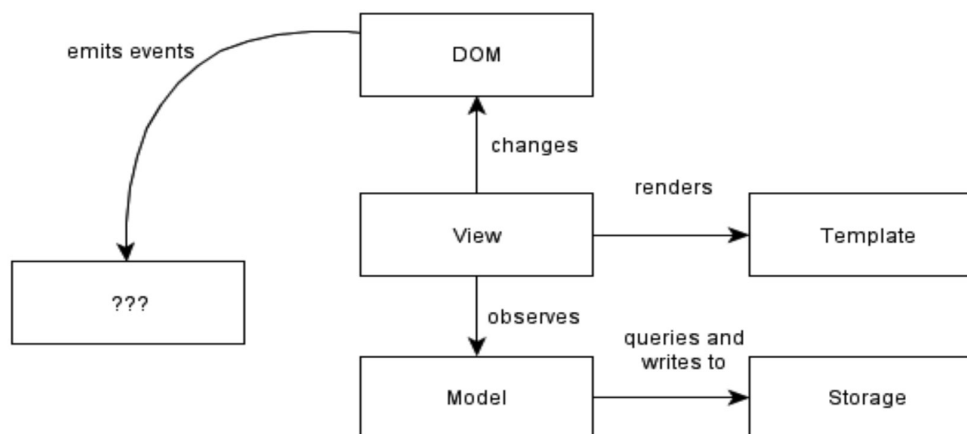


Рис. 1. Архитектура SPA-приложения

Таблица 1

Разница между модульным и немодульным приложением

Немодульное приложение	Модульное приложение
Каждый участок кода является глобальным	Пакеты предоставляют единственный публичный интерфейс
Переменные являются глобальными	Переменные локальные в каждом пакете
Очередность загрузки имеет значение, потому что что-либо может перезаписать или изменить что-то	Очередность загрузки не имеет значения благодаря разделению по пакетам
Неявные зависимости от чего-либо глобального	Детали реализации недоступны за пределами пакета
Файлы и модули не имеют связи по смыслу	Каждый файл соответствует одному модулю
	Явно объявленные зависимости

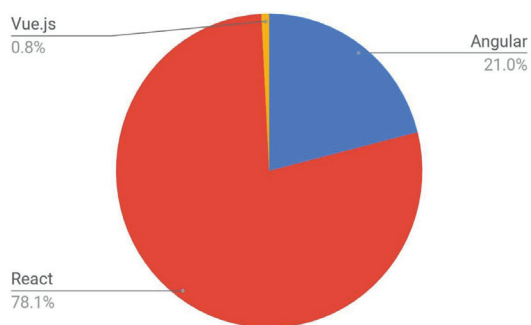


Рис. 2. Соотношение React и AngularJS

**Основные концепции:**

Элементы – объекты JavaScript, которые представляют HTML-элементы.

Компоненты – элементы React, разработанные программистом приложения. Зачастую являются частями интерфейса пользователя, содержащими функциональность и структуру. React значительно облегчает создание интерфейсов благодаря компонентам.

JSX – метод реализации элементов и компонентов React. Например, это React-элемент, написанный на JSX:

```
<h1>Hello World</h1>
```

Реализация аналогичного элемента возможна и на JavaScript:

```
React.DOM.h1(null, 'Hello World');
```

С JSX необходимо намного меньше усилий, он преобразовывается в JavaScript перед запуском в браузере.

Сравним производительность. Создаётся ассоциация между данными домена и элементом DOM, назначая «ключ». При изменении данных элемент DOM с соответствующим ключом будет обновлен. Любые изменения элемента в массиве данных вызывает соответствующее изменение в DOM (табл. 2) [9]. Также одним из ключевых показателей является выделение памяти (табл. 3).

**Результаты исследования и их обсуждение**

Рассмотрим плюсы и минусы React и Angular 5 в табл. 4 и 5 [9].

**Таблица 2**

**Сравнение производительности**

Наименование	angular-v4.1.2-keyed	react-v15.5.4-mobX-v3.1.9	vue-v2.3.3-keyed
Создавать строки. Продолжительность создания 1000 строк после загрузки страницы	193.17.9 (1.2)	243.99.4 (1.5)	166.78.6 (1.0)
Заменить все строки. Продолжительность обновления всех 1000 строк таблицы (с 5 итерациями прогрева)	197.45.3 (1.2)	229.212.2 (1.4)	168.55.0 (1.0)
Частичное обновление. Время обновлять текст каждой 10-й строки (с 5 итерациями разминки)	13.04.5 (1.0)	16.01.8 (1.0)	17.32.9 (1.1)
Выберите строку. Длительность выделения строки в ответ на щелчок по строке (с 5 итерациями разминки)	3.42.3 (1.0)	10.13.8 (1.0)	9.31.7 (1.0)
Поменять строки. Время поменять местами 2 строки в таблице 1К (с 5 итерациями разминки)	13.41.0 (1.0)	18.01.2 (1.1)	18.31.5 (1.1)
Удалить строку. Продолжительность удаления строки (с 5 итерациями разминки)	46.13.2 (1.0)	53.72.1 (1.2)	52.62.7 (1.1)
Создать много строк. Продолжительность создания 10 000 строк	1946.041.8 (1.2)	2217.371.5 (1.4)	1587.533.9 (1.0)
Добавить строки в большую таблицу. Продолжительность добавления 1000 строк в таблицу из 10 000 строк	324.610.1 (1.0)	459.847.2 (1.4)	399.511.0 (1.2)
Очистить строки. Продолжительность очистки таблицы, заполненной 10 000 строк	379.911.3 (1.5)	495.128.8 (1.9)	254.55.0 (1.0)

Таблица 3

## Использование памяти приложений

Наименование	angular-v4.1.2-keyed	react-v15.5.4-mobX-v3.1.9	vue-v2.3.3-keyed
Использование памяти после загрузки страницы	4.80.0 (1.3)	5.40.1 (1.4)	3.80.0 (1.0)
Использование памяти после добавления 1000 строк	10.90.1 (1.4)	14.30.1 (1.9)	7.50.1 (1.0)

Таблица 4

## Плюсы Angular 5 и React

Angular 5	React
Современные функции, доработанный RXJS, более быстрая компиляция, запуск HttpClient	Проще в изучении из-за его простоты с точки зрения синтаксиса. Нет необходимости глубоко обучаться TypeScript, по сравнению с Angular
Доступная документация, предоставляющая актуальную информацию	Максимальная гибкость и отзывчивость
Двусторонняя привязка данных. Это позволяет уменьшить риск потенциальных ошибок	Виртуальный DOM, который позволяет упорядочивать документы в форматах HTML, XHTML или XML в дерево, из которого лучше подходят веб-браузеры при анализе различных элементов веб-приложения
MVVM (Model-View-ViewModel), позволяет программистам разрабатывать одновременно один и тот же раздел программного продукта с помощью одинакового набора данных	В совмещении с ES6 / 7 ReactJS система имеет возможность работать под высокой нагрузкой
Внедрение зависимостей от компонентов, связанных с модулями и модульностью в целом	Нисходящая привязка данных, что означает, что с этим типом потока данных дочерние элементы не могут влиять на родительские данные

Таблица 5

## Минусы Angular 5 и React

Angular 5	React
Достаточно сложный синтаксис, в основу которого положена первая версия Angular	Нет легкодоступной документации, т.к. разрабатывался ReactJS очень быстро
Могут возникнуть проблемы при переходе от старой версии к более новой	Долгое время обучения. React JS требует глубоких знаний о том, как использовать инфраструктуру MVC

## Заключение

Исходя из сравнения производительности AngularJS и React, можно сделать вывод о том, что React не уступает AngularJS по скорости выполнения операций с DOM-узлами, и основное отличие заключается в архитектуре приложения. AngularJS представляет собой реализацию паттерна MVVM, React же ставит View на первое место, его основная цель – это рендеринг данных, а остальные компоненты он оставляет на усмотрение разработчика. В силу такого подхода вполне можно объединить

в одном приложении AngularJS и React для повышения производительности проблемных компонентов.

## Список литературы

1. Натальченко И.А. Анализ механизмов передачи крупных массивов данных через сеть интернет с помощью технологии веб-сервиса // Инженерный вестник Дона. 2008. № 4 [Электронный ресурс]. URL: [ivdon.ru/magazine/archive/n4y2008/98](http://ivdon.ru/magazine/archive/n4y2008/98) (дата обращения: 17.01.2020).
2. Прошчаева А.А., Синелобова С.В. Обзор программного обеспечения для построения распределенных веб систем // Инженерный вестник Дона. 2017. № 4. [Электронный ресурс]. URL: [ivdon.ru/uploads/article/pdf/IVD\\_130\\_proshchaeva\\_sinelobova.pdf](http://ivdon.ru/uploads/article/pdf/IVD_130_proshchaeva_sinelobova.pdf). (дата обращения: 17.01.2020).

3. Разработка одностраничных клиентских приложений для CRM-систем. 1 часть. [Электронный ресурс]. URL: [habr.com/company/qbs/blog/243545](http://habr.com/company/qbs/blog/243545) (дата обращения: 17.01.2020).
4. Бондаренко С.О. Современные интерактивные веб-приложения – построение пользовательского интерфейса с React // Вестник науки и образования. 2018. № 5 (41). С. 46-48.
5. Карышев А.А., Афанасьев В.Р. Разработка web-сервиса для авто-матизированной генерации документов на основе docx-шаблонов // Известия ТулГУ. Технические науки. 2017. № 5. С. 47–54.
6. Клочков Д.В. Общие компоненты при кроссплатформенной разработке для web-и мобильных приложений с использованием react и react-native // Молодой ученый. 2018. № 36. С. 1–5.
7. Скороходов И.С., Тихомирова А.Н. Исследование и сравнение современных реализаций Flux-архитектур разработки веб-приложений // Наука, техника и образование. 2016. № 6 (24). С. 290–297.
8. ReactJS vs Angular5 vs Vue.js-What to choose in 2018? [Электронный ресурс]. URL: [medium.com/@TechMagic/reactjs-vs-angular5-vs-vue-js-what-to-choose-in-2018-b91e028fa91d](https://medium.com/@TechMagic/reactjs-vs-angular5-vs-vue-js-what-to-choose-in-2018-b91e028fa91d) (дата обращения: 17.01.2020).
9. Results for js web frameworks benchmark – round 6. [Электронный ресурс]. URL: [stefankrause.net/js-frameworks-benchmark6/webdriver-ts-results/table.html](https://stefankrause.net/js-frameworks-benchmark6/webdriver-ts-results/table.html) (дата обращения: 17.01.2020).