

УДК 004.432.2

## НОВЫЕ ВОЗМОЖНОСТИ СОВРЕМЕННОГО ВЕБ-ПРОГРАММИРОВАНИЯ

А.В. ЕПАНЕШНИКОВ

(Представлено: канд. физ.-мат. наук, доц. Ю.Ф. ПАСТУХОВ)

Статья посвящена современным технологиям веб-программирования и использованию новых технических возможностей в реальных проектах по созданию программного обеспечения. В статье описаны новейшие фреймворки для создания клиентской части веб-приложения, а также рассказано о программной реализации веб-приложения на технологии React. Целью создания программной реализации является наглядная демонстрация преимуществ инновационных технологий веб-программирования в создании и внедрении веб-приложений над устаревшими технологиями.

**Введение.** В настоящее время веб-программирование развивается с космической скоростью. Новые технологии и идеологии разрабатываются не только специализированными компаниями и корпорациями, но и «рядовыми» программистами, имеющими потребность в создании новой технологии для дальнейшего использования, как в своих целях, так и для общества.

Эффективное применение современных технологий и идеологий веб-программирования способствует не только повышению качества программного продукта, но и экономии трудозатрат, в том числе финансовых ресурсов, производительности и безопасности приложения, легкой расширяемости проектов и многому другому.

**Постановка задачи.** Для исследования новых возможностей современного веб-программирования были поставлены следующие задачи:

1. Рассмотреть новейшие технологии программирования на языке Javascript (ECMAScript 6), а также зависимости клиент-серверного приложения.
2. Сравнить технологии.
3. Продемонстрировать возможности лучшей технологии, разработав программный продукт (в данном случае – интерфейс сетевого устройства).
4. Сделать выводы о преимуществах новых технологий.

**Технологии веб-программирования.** Сравнение разных подходов в веб-программировании по выбранным критериям приведено в таблице. Данные для сравнения отобраны из официальных сайтов разработчиков и информационных источников [2, 3, 4].

| Критерий для сравнения   | AngularJS | React      | Серверный рендеринг без использования шаблонов |
|--|-----------|------------|--|
| Поддержка JSX  | нет       | да         | нет  |
| Клиентский рендеринг   | да        | да         | нет  |
| Возможность переноса компиляции шаблонов на сервер                             | нет       | да         | нет  |
| Привязана к структуре и библиотекам DOM  | да        | нет        | да   |
| Размер исходного файла   | небольшой | большой    | небольшой                                      |
| Размер информации, передаваемой с сервера на клиент при каждом запросе клиента | средний   | малый      | большой  |
| Имеет внедрение зависимости  | да        | Не всегда* | нет  |
| Абстрактный клиентский роутинг**   | нет       | да         | нет  |

\* - чтобы в React подключить внедрение зависимостей, например, NPM (Node Package Manager), необходимо установить и использовать сборщик модулей. [1] Самый популярный и наиболее эффективный, на взгляд автора статьи – Webpack.

\*\* - имеется в виду абстракция клиентского роутинга от html-файлов и появление объектно-ориентированной иерархии и явной зависимости компонентов Javascript в роутинге.

AngularJS — JavaScript-фреймворк с открытым исходным кодом. Предназначен для разработки одностраничных приложений. Его цель — расширение браузерных приложений на основе MVC шаблона, а также упрощение тестирования и разработки. Фреймворк работает с HTML, содержащим дополни-

тельные пользовательские атрибуты, которые описываются директивами, и связывает ввод или вывод области страницы с моделью, представляющей собой обычные переменные JavaScript. Значения этих переменных задаются вручную или извлекаются из статических или динамических JSON-данных. [5]

React - разработанный компаниями Facebook и Instagram JavaScript-фреймворк с открытым исходным кодом для создания пользовательских интерфейсов. Большинство пользователей React считают его Представлением в идеологии MVC. React был разработан, чтобы решить большую проблему: написание больших приложений с данными, которые меняются с течением времени. [4]

Серверный рендеринг без использования шаблонов – это подход, при котором роль клиентской части веб-приложения состоит лишь в отображении HTML-файлов, посланных с сервера клиенту, в браузере пользователя. Серверная же роль заключается в генерации HTML-файлов, подходящих под запросы клиента, обработке всех данных и распределении ролей в формировании представления (Модель и Контроллер, а также частично Представление в идеологии MVC).

**Программная реализация приложения на React.** Рассмотрим реализацию интерфейса сетевого устройства с использованием React (рис. 1). В данном примере показана страница сетевых настроек. С помощью соответствующих кнопок можно добавлять и удалять строки с IP-адресами, шлюзами и статическими маршрутами.

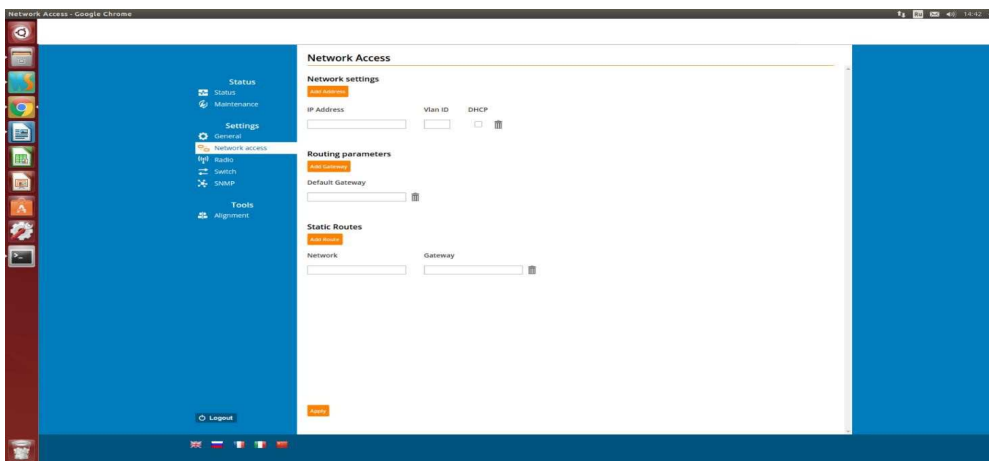


Рисунок 1. – Интерфейс сетевого устройства, реализованный с помощью React

При добавлении или удалении строки перерисовывается только соответствующие элементы, а количество самих строк и данные внутри них хранятся в состоянии родительского React-компонента страницы (NetworkAccess). Для отображения, эти данные передаются далее дочерним компонентам – в данном случае, строкам NetworkSettingsRow (рис. 2).

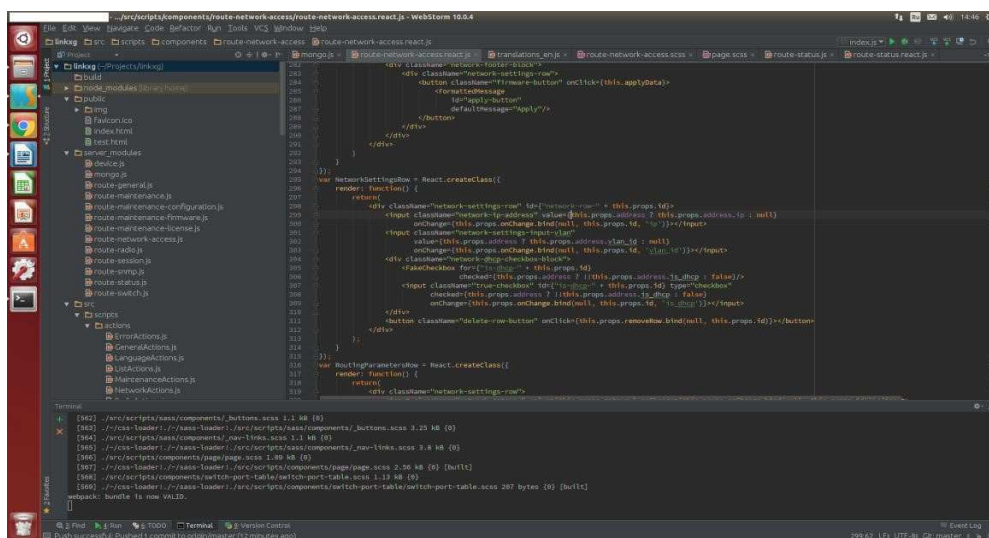


Рисунок 2. – Программный код React-компонента NetworkSettingsRow

Необходимо отметить, что хоть дочерние компоненты рендерятся как DOM-элементы, но внутри кода они полностью контролируются за счёт состояний родительского компонента, свойств, переданных дочерним элементам, а также обработчикам их изменений. Таким образом, не обращаясь к DOM-иерархии напрямую из javascript, можно управлять любыми компонентами в веб-представлении любого веб-приложения, использующего React, не осуществляя бесполезную и неявную ручную выборку элементов из DOM (например, итерирование по массиву, возвращенному методом `document.getElementsByClassName(...)`).

**Заключение.** Рассмотренные современные технологии веб-программирования и программная реализация приложения на React дают понять, что использование современных и быстро развивающихся технологий веб-программирования необходимо при разработке конкурентоспособного программного обеспечения. Устаревшие технологии в настоящее время подходят для очень узкого круга задач, в большинстве случаев они применяются в компаниях с большим объемом предыдущих разработок на старых технологиях, вследствие чего переход на новые технологии означает переписывание практически всего кода «с нуля». Однако, новые технологии позволяют писать код для много более масштабируемой разработки, которая к тому же не замыкается на уникальных компетенциях в непопулярных или устаревших фреймворках, или же просто на коде с большой вложенностью зависимостей и непродуманной оптимизацией. Более того, использование, например, React или AngularJS позволяет улучшить отзывчивость интерфейса, ускорить переход между страницами благодаря клиентскому роутингу, а кастомизация и шаблонизация интерфейса значительно упрощает доработку интерфейса.

#### ЛИТЕРАТУРА

1. Документация React [Электронный ресурс]. – Режим доступа: <https://facebook.github.io/react/docs/getting-started.html>.
2. Versus. Сравнение AngularJS и React [Электронный ресурс]. Режим доступа: <http://versus.com/ru/react-vs-angularjs>.
3. AngularJS [Электронный ресурс]. – Режим доступа: <https://angularjs.org/>
4. Краткое руководство по React JS [Электронный ресурс]. – Режим доступа: <http://habrahabr.ru/post/248799/>
5. AngularJS – Википедия, свободная энциклопедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/AngularJS>.