



INTERNATIONAL SCIENCE REVIEWS

Natural Sciences and Technologies series

Has been published since 2020

№3 (3) 2022

Nur-Sultan
EDITOR-IN-CHIEF:

Doctor of Physical and Mathematical Sciences, Academician of NAS RK, Professor
Kalimoldayev M. N.

DEPUTY EDITOR-IN-CHIEF:

Doctor of Biological Sciences, Professor
Myrzagaliyeva A. B.

EDITORIAL BOARD:

- | | |
|--|--|
| Akiyanova F. Zh. | - Doctor of Geographical Sciences, Professor (Kazakhstan) |
| Seitkan A. | - PhD, (Kazakhstan) |
| Baysholanov S. S | - Candidate of Geographical Sciences, Associate professor (Kazakhstan) |
| Zayadan B. K. | - Doctor of Biological Sciences, Professor (Kazakhstan) |
| Salnikov V. G. | - Doctor of Geographical Sciences, Professor (Kazakhstan) |
| Zhukabayeva T. K. | - PhD, (Kazakhstan) |
| Urmashiev B.A | - Candidate of Physical and Mathematical Sciences, (Kazakhstan) |
| Abdildayeva A. A. | - PhD, (Kazakhstan) |
| Chlachula J. - Professor, Adam Mickiewicz University (Poland) | Redfern S.A.T. - PhD, Professor, (Singapore) |
| Cheryomushkina V.A. - Doctor of Biological Sciences, Professor (Russia) | |
| Bazarnova N. G. | - Doctor Chemical Sciences, Professor (Russia) |
| Mohamed Othman | - Dr. Professor (Malaysia) |
| Sherzod Turaev | - Dr. Associate Professor (United Arab Emirates) |

Editorial address: 8, Kabanbay Batyr avenue, of.316, Nur-Sultan,
Kazakhstan, 010000
Tel.: (7172) 24-18-52 (ext. 316)
E-mail: natural-sciences@aiu.kz

CONTENT

С.Е. Базаров СОВРЕМЕННОЕ СОСТОЯНИЕ УРОЖАЙНОСТИ МОЛОДИ ОКУНЯ И ПЛОТВЫ В ВЕРХНЕ-ЕРТИССКОМ КАСКАДЕ ВОДОХРАНИЛИЩ	5
Д.Е. Ержанов ӨСКЕМЕН СУ ҚОЙМАСЫНЫҢ КӘСІПШІЛІК МАҢЫЗЫ БАР БАЛЫҚ ТҮРЛЕРІНІҢ ЭКОЛОГИЯЛЫҚБИОЛОГИЯЛЫҚ СИПАТТАМАСЫ	11
С. Оңталапұлы, Н. Тасболатұлы МОДЕЛИ И МЕТОДЫ ОЦЕНКИ ЗАЩИЩЕННОСТИ ИНФОРМАЦИИ	21
А. Кусаинов STEM В КАЗАХСТАНЕ. ВИРТУАЛЬНАЯ РОБОТОТЕХНИКА В ШКОЛАХ	29
А.С. Муканова, А.Б. Барлыбаев, А.Е. Назырова, Л.Т. Кусепова РАЗРАБОТКА ОНТОЛОГИЧЕСКОЙ МОДЕЛИ ИЗБИРАТЕЛЬНОЙ СИСТЕМЫ.....	34
М.Ж.Калдарова, А.Н.Султангазиева МЕТОДЫ СЕГМЕНТАЦИИ ИЗОБРАЖЕНИЙ	43
Ә.Ерланұлы, Е.М.Марденов, АНАЛИЗ ПРЕИМУЩЕСТВ И НЕДОСТАТКОВ VUE И REACT	53

АНАЛИЗ ПРЕИМУЩЕСТВ И НЕДОСТАТКОВ VUE И REACT

Ә.Ерланұлы, Е.М.Марденов

Международный Университет Астана, Казахстан

Аннотация. В статье рассматриваются фреймворки Vue и React, их преимущества и недостатки в реалиях современных технологий разработки Web-приложений, а также был проведен сравнительный анализ этих инструментов. Также были описаны существующие технологии и пакеты инструментов веб-разработки, способствующие быстрому и качественному созданию модулей графического представления и загрузки данных информационных систем.

Ключевые слова: Vue.js, React, веб-приложения, разработка, JS, framework.

ВВЕДЕНИЕ

Как устроен Vue.js? Vue.js – является библиотекой JavaScript для создания и построения веб-интерфейсов с использованием шаблона архитектуры [MVVM \(Model-View-ViewModel\)](#). Vue.js был разработан с нуля и постепенно улучшается. Поэтому, знание HTML, CSS и JavaScript, вам дадут возможность легко освоить Vue.js.

Во многих случаях Vue.js заменяет Angular, Ember и React - это и отличает работу с ним. Кроме того, Vue сочетается с другими современными инструментами и поддерживающими библиотеками для создания идеального одностраничного приложения.

В основе Vue.js, так же как и React, существуют понятия компонента и состояния. Что же до составления кода во Vue.js, то, бесспорно, одним из самых популярных подходов является создание однофайловых компонентов, где каждый компонент Vue.js — это независимая переиспользуемая единица, отдельный файл с расширением .vue. Файл с данным расширением содержит в себе три секции, три обособленных друг от друга элемента: template

(представление), script (модель) и style (стили).

Фреймворк Vue.js используется в разработке:

- быстрых приложений и веб-сайтов небольшого размера;
- Высоконагруженных сайтов — интернет-магазинов, инфо порталов;
- Одностраничных приложений — микроблогов, соц. сетей, CMS и т.д.;
- Сайтов с адаптивным интерфейсом;
- Различных кабинетов пользователей и пользовательских страниц;

Плюсы Vue.js

Как и любая набирающая популярность технология, Vue.js вызывает споры в сообществе. И есть причины, по которым Vue стал вторым по популярности фреймворком в 2019 году.

И для этого есть причины:

Просто понимать и разрабатывать приложения

Простота является частью кода Vue. Одна из вероятности популярности этого фреймворка заключается в том, что его довольно легко понять. Пользователь может легко добавить Vue.js в свой веб-проект благодаря своей простой структуре. Как небольшие, так и крупномасштабные шаблоны могут быть разработаны с помощью этой структуры.

Широко известен своей гибкостью

Vue.js поставляется с идеальным балансом для быстрой записи и запуска прямо из браузера. Вы можете создать сложное приложение, используя ES6, JSX, компоненты, маршрутизацию, комплектацию и т. д. Это делает предпочтительным выбором использование Vue.js для разработки кросс-платформенных приложений. Он предоставляет вам доступ к широкому спектру различных сред.

Маленький размер фреймворка

Vue: скачанный zip с фреймворком весит 18 КБ. Будучи легким, фреймворк не только быстро загружает и устанавливает библиотеку, но и

положительно влияет на SEO и UX.

Устранение проблем AngularJS и ReactJS

Эван Ю, создатель Vue.js, представил этот популярный фреймворк JavaScript. Он создал Vue.js с целью устранения недостатков двух самых популярных фреймворков JavaScript - AngularJS и ReactJS.

Таким образом, он разработал легкое решение путем извлечения деталей AngularJS без добавления какой-либо другой концепции. Он хотел объединить лучшие функции обоих фреймворков, и именно так он справился с новым Vue.js.

Vue.js использовал компонентный подход, односторонний поток данных для иерархии компонентов, возможность виртуального рендеринга, высокую производительность, реквизит и роль, которую играет управление состоянием приложений в веб-приложениях и т. д.

Рендеринг и производительность виртуального DOM

Объектная модель документа (DOM) — это то, с чем вы, вероятно, столкнетесь при рендеринге веб-страниц.

DOM — это представление HTML-страниц с их стилями, элементами и содержимым страницы в виде объектов. Объекты, хранящиеся в виде древовидной структуры, генерируются браузером при загрузке страницы.

Когда пользователь взаимодействует со страницей, объекты меняют свое состояние, так что браузер должен обновлять информацию и отображать ее на экране. Но обновление всего DOM выглядит тяжелой задачей. Ради скорости Vue.js использует виртуальный DOM: Думайте об этом как о копии оригинального DOM, который выясняет, какие элементы обновлять, а не рендерить весь DOM. Такой подход делает рендеринг страниц довольно быстрым и повышает производительность приложений.

Производительность является одним из ключевых факторов, которые могут предопределить выбор структуры. Фактические тесты представлены на

странице сравнения Vue. Например, при тестировании компонентов DOM, связанных с обновленными данными, в них Vue.js кажется более производительным, чем React.

Как отмечают в своей работе Д. А. Данилин, Я. В. Зиновьев, К. М. Кузьмин, “Анализ технологий web-программирования для создания модулей визуализации и загрузки” – “ Производя сравнение Vue.js с другими популярными решениями, такими как React.js и Angular, стоит отметить, что именно Vue.js наиболее прост в изучении. Одной из особенностей Vue.js является предоставление возможности изменения приложения без разрушения его архитектуры, благодаря чему достигается легкое внедрение и расширение функционала ПО. Для построения стандартного высокопроизводительного интерфейса может быть целесообразен выбор Vue.js в качестве замены сложных и объемных фреймворков.”, что подтверждает высказывания выше о высокой производительности Vue.

ДЕКЛАРАТИВНОСТЬ, ПРОГРЕССИВНОСТЬ И РЕАКТИВНОСТЬ:

Vue.js сконструирован на основе декларативного отображения данных в DOM с помощью простых шаблонов. Декларативный подход стоит в основании всех фреймворков и нередко оказывается одной из причин его создания.

Декларативность — это когда человек печатает, что он хочет сделать, а фреймворк сам выполняет нужную задачу. Декларативность как правило противопоставляют императивному стилю «ванильного» JavaScript, когда мы невольно должны пошагово прописывать отдельное действие.

В JavaScript нам приходится не просто описывать суть задачи, но и точно, пошагово, прописывать всю логику выполнения:

- работать с html-элементами, обращаясь к ним через id, классы, css-селекторы;
- навешивать и удалять обработчики события через addEventListener /

removeEventListener;

- при изменении данных изменять что-либо в конкретных DOM-узлах (например, менять текст через textContent) и тому подобное.

- Vue.js преуспевает в декларативности. Например, у него есть набор директив, которые довольно просто и легко описывают основные операции с html-элементами: v-bind, v-for, v-if, v-show, v-html и другие.

Декларативность Vue.js против декларативности React

React - это библиотека JavaScript с открытым исходным кодом, которую можно использовать для создания веб-приложений со сложными пользовательскими интерфейсами. React позволяет разработчикам создавать многоразовые индивидуальные компоненты, что позволяет ускорить разработку. Кроме того, его способность быстро загружать веб-страницу делает ее еще более удобной для поисковых систем. В целом, это отличная библиотека, которая поощряет создание как простых, так и передовых корпоративных приложений.

Сильные стороны React

- поддержка Facebook;
- высокая скорость работы;
- большое community;
- кроссплатформенность;
- разработка UI на основе отдельных компонентов;
- технология Virtual DOM (высокая производительность).

Что подтверждается в исследовании Сергачевой М. А., Михалевской К. А., "Анализ Фреймворков для разработки современных web-приложений". В статье были рассмотрены все современные фреймворки с их преимуществами и недостатками.

React так же преуспевает в философии декларативности, но, как мы видим, декларативность — понятие относительное. Рассмотрим на примере

с двумя Vue-директивами: v-for и v-show — и сравним с той же самой опцией в React.

Предположим, где-то в коде у нас есть массив объектов articles. И наша задача — вывести у каждого объекта свойство title, а также, в зависимости от логики приложения, показать или скрыть параграф.

Сконцентрируемся на логике представления.

Vue-код:

```
<template>
  <div>
    <ul>
      <li v-for="article in articles" :key="article.key">
        {{ article.title }}
      </li>
    </ul>
    <p v-show="isShow">Данные статьи доступны для скачивания</p>
  </div>
</template>
```

Чтобы отобразить массив, во Vue.js мы используем директиву v-for в теге . В качестве значения указываем "article in articles", где articles — это наш массив, доступный в конкретном компоненте, а article — отдельная статья, элемент на каждой итерации цикла.

Таким образом, мы проходим по всему массиву и выводим свойство "title" для каждой статьи ("article.title"). Значение в двойных фигурных скобках называется интерполяцией и связывает модель с представлением.

Директива v-show фактически «переключает» свойство display с "none" в "initial" и наоборот, в зависимости от значения. В нашем примере предполагается, что, если в наших данных isShow = true, значит, элемент будет отображён, если false — элемент окажется скрытым.

Важно помнить, что это не рендеринг элемента по условию (за это отвечает директива Vue.js `v-if`), директива `v-show` именно «переключает» свойство `display`, то есть при рендере компонента наш элемент `<p>` в любом случае будет присутствовать и займёт своё место в DOM-дереве, а будет ли этот элемент отображён — будет решать значение переменной `isShow`.

А ниже продемонстрировано, как реализуется та же функциональность на React:

```
return (
  <div>
    <ul>
      {articles.map(article => (
        <li key={article.key}> {article.title} </li>
      ))}
    </ul>
    <p style={{ display: isShow ? 'initial' : 'none' }}>
      Данные статьи доступны для скачивания
    </p>
  </div>
);
```

React также является декларативным фреймворком, но мы видим, что в нём куда меньше синтаксического сахара. Да, нам не нужно взаимодействовать с html-элементами напрямую, и мы всё так же имеем реактивную связь данных и отображения, довольно много логики инкапсулировано.

Однако цикл — это известный каждому перебор элементов при помощи функции `map`, а условное отображение элемента использует тернарный оператор. Как минимум, выглядит всё так, будто мы в чистую html-разметку вставляем чистые кусочки JavaScript, в общем ничего дополнительного и нового

для JavaScript-разработчика.

В этом сравнительном примере можно наблюдать, в какой степени Vue.js декларативен. Если речь идёт о связывании отображения с данными, мы, как разработчики, лишь декларативно «отдаём приказы» (пишем директивы), и Vue выполняет за нас много работы — как и React. Тем не менее Vue идёт ещё дальше и инкапсулирует даже JS-методы перебора массива и изменения css-свойства `"display"`, изменяя их простым и лаконичным синтаксисом специальных директив.

Самые большие недостатки Vue включают в себя:

- Проблемы с двусторонним связыванием
- Проблемы в мобильной поддержке
- Ограниченные плагины
- Ограниченная масштабируемость
- Программисты с ограниченным опытом
- Слишком большая гибкость в программировании
- Использование сообщества имеет некоторые ограничения

Самые большие недостатки React включают в себя:

• Из-за быстрых темпов роста окружающая среда постоянно меняется, что затрудняет адаптацию программистов.

• При таком быстром обновлении и ускорении хорошую документацию трудно найти.

• ReactJS просто обрабатывает уровни пользовательского интерфейса приложения, поэтому вам нужно будет использовать другие инструменты для других частей разработки.

- Если есть много шаблонов, и они пересекаются, JSX может сбить с толку.

ЗАКЛЮЧЕНИЕ

В итоге довольно тяжело упоминать Vue.js без сравнения с React. Эти два

фреймворка похожи, но в деталях имеется много различий.

Первое различие:

- Vue в большей степени строг и декларативен. Строгость можно увидеть в синтаксисе (определённые директивы для каждой задачи), а также в том, как выстраиваются сами компоненты (чёткая структура разделения модели, представления и стилей, в особенности при использовании однофайловых компонентов с расширением .vue).

Второе различие:

- Vue имеет более «крепкую» инфраструктуру — выбор самых важных библиотек зачастую предопределён.

Такой подход приводит к тому, что новым разработчикам проще писать читабельный код с самых первых дней знакомства с Vue.js. В примере выше код на Vue был написан без знания о существовании тернарного оператора и даже без знания о том, как перебирать элементы в массиве. Понадобилось лишь изучить пару простейших директив (v-for и v-show). Остальную работу Vue сделал сам.

Vue - это очень мощный инструмент для создания обогащающего опыта в Интернете, и структура очень зрелая и хорошо поддерживается сообществом.

Он очень популярен среди разработчиков, и из опроса 89% разработчиков в 2017 году заявили, что снова будут использовать фреймворк.

Vue превосходит своих конкурентов с простотой и дополнительными преимуществами:

- легкость обучения,
- отличная документация,
- очень высокая скорость кодирования для разработчика.

СПИСОК ЛИТЕРАТУРЫ

1. Skalka, J.; Drlik, M.; Benko, L.; Kapusta, J.; Rodríguez del Pino, J.C.; Smyrnova-Trybulska, E.; Stolinska, A.; Svec, P.; Turcinek, P. Conceptual Framework for Programming Skills Development Based on Microlearning and Automated Source Code Evaluation in Virtual Learning Environment. Sustainability 2021, 13, 3293. <https://doi.org/10.3390/su13063293>
2. “Проектирование и разработка архитектуры масштабируемых веб-приложений”, Автор: Кувакин А. Е., Научный руководитель: Мокина Е. Е., URL:(“<http://earchive.tpu.ru/handle/11683/40202>”)
3. Байдыбеков А.А., Гильванов Р. Г., Молодкин И. А., “Современные Фреймворки для разработки web-приложений”, URL:(“<https://cyberleninka.ru/article/n/sovremennye-freymvorki-dlya-razrabotki-web-prilozheniy>”)
4. Skalka, J.; Drlik, M.; Obonya, J. Automated Assessment in Learning and Teaching Programming Languages using Virtual Learning Environment. In Proceedings of the IEEE Global Engineering Education Conference (EDUCON2019), Dubai, United Arab Emirates, 8–11 April 2019. [Google Scholar] [CrossRef]
5. Сергачева М. А., Михалевская К. А., “Анализ Фреймворков для разработки современных web-приложений”, URL: (“<https://cyberleninka.ru/article/n/analiz-freymvorkov-dlya-razrabotki-sovremennyh-veb-prilozheniy>”)
6. Зиатдинов А. Р., “Сравнение эффективности использования различных современных фронтенд-фреймворков в корпоративных приложениях.” (“<https://cyberleninka.ru/article/n/sravnenie-effektivnosti-ispolzovaniya-razlichnyh-sovremennyh-frontend-freymvorkov-v-korporativnyh-prilozheniyah>”)
7. Atelge, M.R.; Senol, H.; Djaafri, M.; Hansu, T.A.; Krisa, D.; Atabani, A.; Eskicioglu, C.; Muratçobanoğlu, H.; Unalan, S.; Kalloum, S.; Azbar, N.; Kıvrak, H.D. A Critical Overview of the State-of-the-Art Methods for Biogas Purification and Utilization Processes. Sustainability 2021, 13, 11515. <https://doi.org/10.3390/su132011515>
8. Бондаренко С. О., “Современные интерактивные веб-приложения - построение пользовательского интерфейса с React”, URL: (“<https://cyberleninka.ru/article/n/sovremennye-interaktivnye-veb-prilozheniya-postroenie-polzovatelskogo-interfeysa-s-react>”)
9. Газизуллин Н. И., Плещинская И. Е., “Разработка клиентской части веб-приложения с использованием технологий spa”, URL: (“<https://cyberleninka.ru/article/n/razrabotka-klientskoy-chasti-veb-prilozheniya-s-ispolzovaniem-tehnologiy-spa>”)
10. Данилин Д. А., Зиновьев Я. В., Кузьмин К.М., “Анализ технологий веб-

программирования для создания модулей визуализации и выгрузки данных информационных систем”, URL: (<https://cyberleninka.ru/article/n/analiz-tehnologiy-veb-programmirovaniya-dlya-sozdaniya-moduley-vizualizatsii-i-vygruzki-dannyh-informatsionnyh-sistem>)

11. Скороходников И. С., Тихомирова А. Н., “Исследование и сравнение современных реализаций Flux-архитектур разработки веб-приложений”, URL: (<https://cyberleninka.ru/article/n/issledovanie-i-sravnenie-sovremennyh-realizatsiy-flux-arhitektur-razrabotki-veb-prilozheniy>)

12. Сукиасян В. М., Придус Е. С., “Современные принципы и подходы к Frontend архитектуре веб-приложений”, URL: (<https://cyberleninka.ru/article/n/sovremennye-printsipy-i-podhody-k-frontend-arhitecture-veb-prilozheniy>)

13. Kordaki, M. A drawing and multi-representational computer environment for beginners’ learning of programming using C: Design and pilot formative evaluation. Comput. Educ. 2010

14. Anindyaputri, N.A.; Yuana, R.A.; Hatta, P. Enhancing Students’ Ability in Learning Process of Programming Language using Adaptive Learning Systems: A Literature Review. Open Eng. 2020