

УДК 004.4'22

<https://doi.org/10.36906/AP-2022/11>

Волков В.Н.

канд. техн. наук

Новиков С.В.

ORCID: 0000-0003-1055-0113, канд. техн. наук

Герасимова Н.М.

ORCID: 0000-0002-5065-5078

Полонская Т.С.

ORCID: 0000-0003-0184-3889

*Орловский государственный университет им. И.С. Тургенева
г. Орел, Россия*

ПЕРСПЕКТИВЫ ИСПОЛЬЗОВАНИЯ ФРЕЙМВОРКА VUE.JS ДЛЯ СОЗДАНИЯ ГИБКИХ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ

Аннотация. В статье рассмотрены преимущества и недостатки использования популярного фреймворка Vue.js для создания пользовательских интерфейсов. Выполнен сравнительный анализ с фреймворком React по популярности. Также представлен график популярности по статистике запросов, которую предоставляет GoogleTrends. Дополнительно, приведены активно развивающиеся библиотеки интерфейсов «BootstrapVue» и «Vuetify», их популярность на основе запросов. Проанализирован способ создания приложений с помощью различных библиотек.

Ключевые слова: JS-фреймворк; javascript; фреймворк; компоненты; мобильные адаптивные сайты; пользовательские интерфейсы.

Volkov V.N.

Ph.D.

Novikov S.V.

ORCID: 0000-0003-1055-011, Ph.D.

Gerasimova N.M.

ORCHID: 0000-0002-5065-5078

Polonskaya T.S.

ORCHID: 0000-0003-0184-3889

*Orel State University named after I.S. Turgenev
Orel, Russia*

PROSPECTS FOR USING THE VUE FRAMEWORK.JS FOR CREATING FLEXIBLE USER INTERFACES

Abstract. The article discusses the advantages and disadvantages of using the popular Vue framework.js for creating user interfaces. A comparative analysis was performed with the React framework by popularity. There is also a graph of popularity based on query statistics provided by Google Trends. Additionally, the actively developing libraries of the "Bootstrapvue" and "Vuetify" interfaces, their popularity based on queries, are given. The method of creating applications using various libraries is analyzed.

Keywords: JS framework; javascript; framework; components; mobile adaptive sites; user interfaces.

Выбирая фреймворк для создания пользовательских интерфейсов, стоит рассмотреть основные их виды, оценить все возможные положение и отрицательные стороны. Изначально можно оценить уровень интереса других разработчиков к фреймворкам. Согласно опросам на «The State of JavaScript», результаты представлены на рисунке 1, лидирующие позиции занимает «Svelte», «Vue.js» и «React» [1, с. 70]. Большинство специалистов считает, что освоить «Vue» намного легче чем другие популярные JS-фреймворки. Документация – еще одна сильная сторона «Vue» – так считают более 60% опрошенных разработчиков [2, с. 132]. Аналогичное число респондентов (56%) отмечает производительность данного решения как одну из его самых сильных сторон. Разберемся подробнее, что же такое Vue.js?

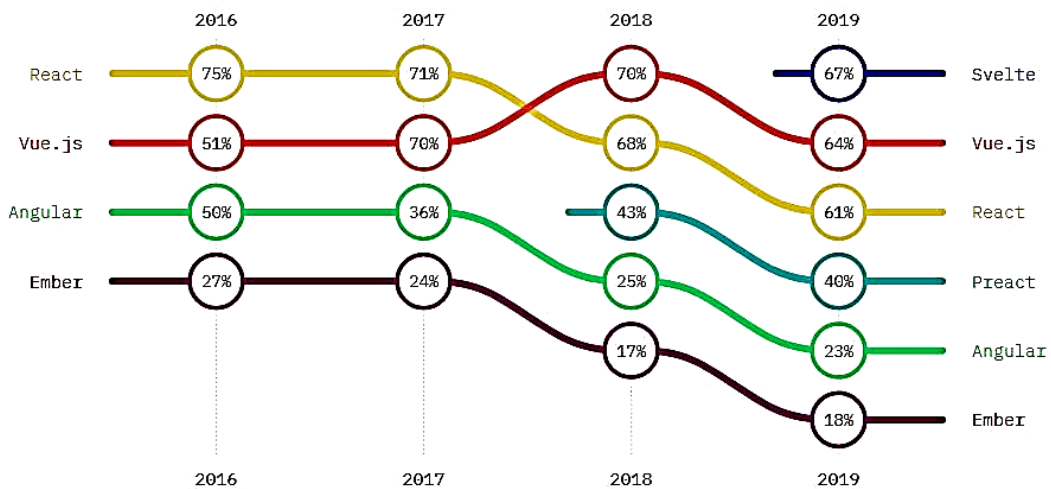


Рис. 1. Результаты опроса «The State of JavaScript»

Касаемо популярности данных фреймворков в России складывается ситуация представленная на рисунке 2, согласно статистике из «Google Trends» (код доступа: <https://clck.ru/YF8BE>). Следовательно, основными конкурентами при выборе фреймворка для разработки остаются «React» и «Vue.js».

Основываясь на статистике выше, можно отметить, что большинство разработчиков отдают предпочтение «React.js», однако это не фреймворк, а библиотека с широким функционалом, где инструмент можно использовать без дополнительной интеграции.

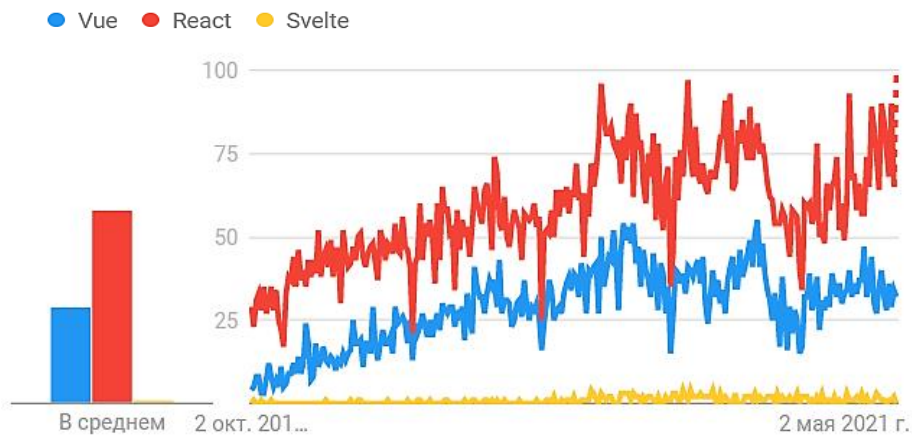


Рис. 2. Статистика «Google Trends» по запросам «Svelte», «Vue.js» и «React»

В свою очередь «Vue.js» – это именно JS-фреймворк. Если сравнивать «React» и «Vue.js» по производительности, то «Vue.js» отличается более высокой производительностью и стабильностью, в силу использования виртуальной DOM.

Одним из основных критериев сравнения выступает размерность. Относительно этого показателя «React» (около 100КБ) уступает «Vue.js» (около 80КБ). Хотя следует отметить, что для «React» необходимо использование сторонних библиотек, а «Vue.js» же – проект с обширным набором стандартных возможностей.

Говоря о гибкости, можно отметить, что ядро «React» дает только основной функционал, что позволяет разработчику выбрать дополнительный инструмент самостоятельно, «Vue.js» же дает большой набор стандартных инструментов, которые позволяют решать задачи, где при использовании «React» необходимо было бы подключение дополнительных библиотек.

«Vue.js» является исключительным инструментом для любого разработчика, так как все его функции находятся в свободном доступе. Для удобства использования разработчики могут просто назвать функцию так, как они хотят. Каждая деталь может иметь дискретную функцию, что значительно упрощает модификацию веб-приложения в соответствии с конкретными требованиями [3, с. 210].

«Vue.js» – гибкая технология, которую можно использовать при разработке приложений без использования многочисленных ресурсов. «Vue.js» имеет гибкий дизайн, который подчеркивает состав компонентов и декларативный рендеринг.

Архитектурный стиль фреймворка определяется моделью MVVM (Модель-Представление-Модель представления). Прогрессивные характеристики, необходимые в многогранных приложениях: управление состоянием, маршрутизация - представлены через пакеты и библиотеки, где «Nuxt.js» стал наиболее популярным решением.

Основными концепциями «Vue.js» являются, компоненты, конструкторы, директивы и переходы. Изначально необходимо создать новый экземпляр «Vue.js», где указываются элемент для рендеринга, данные, вычисляемые свойства и методы [4, с. 49].

При разработке следует придерживаться идеи разбиения сложных вещей на части (компоненты) и связывать их наиболее удобным образом. Таким образом, компоненты помогают расширить основные html элементы и повторно использовать часть UI. Во время

проектирования, приложение разбивается на независимые части, которые в дальнейшем формируют древовидную структуру компонентов.

Применение директив даёт возможность дополнительной функциональности html-элементам. С помощью префиксов «v-» можно добавлять обработчики событий, управлять отображением элементов и передавать данные. Так, появляется возможность использовать условия и циклы следующих видов, представленных на рисунке 3.

```
<p v-if="x > 5 ">X больше 5</p>
<p v-else> X меньше 5 </p>

<div>
  <span v-for="n in 10">{{ n }} </span>
</div>
```

Рис. 3. Разработка с использованием префиксов «v-»

Так же при обновлении объектов DOM и их добавлении или удалении «Vue.js» даёт возможность сопровождать этот процесс анимацией.

В связи со стремительным ростом популярности «Vue.js», появляются эффективные для создания интерфейсов библиотеки, как, например, «BootstrapVue», «Vuetify», «Vue Material» и «Vux», статистика Google Trends» по запросам которых представлена на рисунке 3. Самой популярной является «Vuetify», так как она нацелена на создание чистых, семантических компонентов с использованием принципов «Google Material Design». В состав входит более 80 компонентов, которые построены на простых и чётких именах свойств для более простого запоминания.

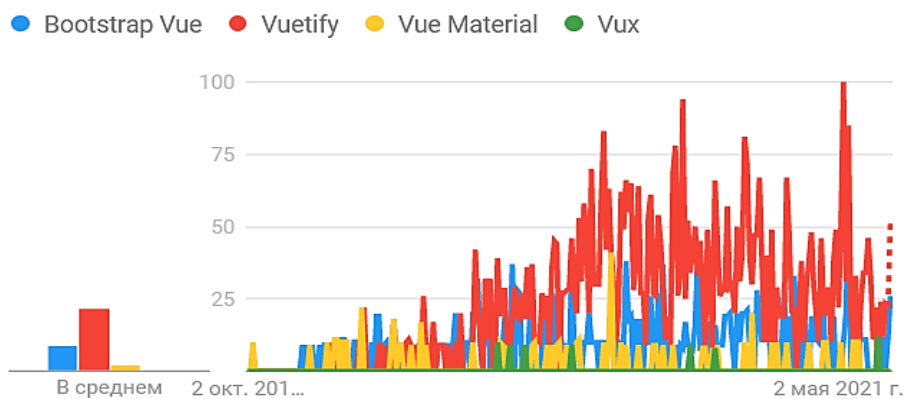


Рис. 4. Статистика «Google Trends» по запросам «Bootstrap Vue», «Vuetify», «Vue Material» и «Vux»

Вторым по популярности стал «Bootstrap Vue», так как «Bootstrap» стал популярным фреймворком HTML / CSS для создания мобильных адаптивных сайтов. Однако для своих основных функций он, как правило, использует «jQuery», а также обширный список компонентов, таких как оповещения и модальные элементы. Поэтому использование «Bootstrap» с «Vue.js», даёт возможность удалить необходимость в «jQuery» (код доступа: <https://2019.stateofjs.com/front-end-frameworks>). Обладая более чем 85 компонентами, более чем 45 доступными плагинами, несколькими директивами и более 1200 значками, «BootstrapVue» предоставляет одну из наиболее полных реализаций компонентов «Bootstrap v4» и системы сетки.

Для сравнения их преимуществ рассмотрим создание одинаковых компонентов «Card» с применением «Bootstrap Vue» и «Vuetify» соответственно. Использование «Bootstrap Vue» представлено на рисунке 5. Пример использования «Vuetify» представлен на рисунке 6.

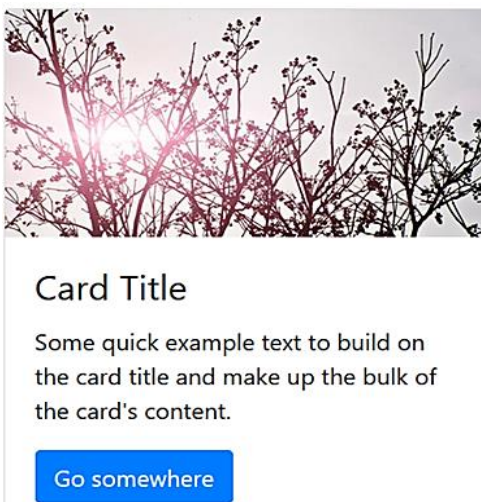
```
<b-card
  title="Card Title"
  img-src="https://picsum.photos/600/300/?image=25"
  img-alt="Image"
  img-top
  tag="article"
  style="max-width: 20rem;"
  class="mb-2">
  <b-card-text>
    Some quick example text to build on the card title and make up the bulk of the card's content.
  </b-card-text>
  <b-button href="#" variant="primary">Go somewhere</b-button>
</b-card>
```

Рис. 5. Пример разработки с использованием «Bootstrap Vue»

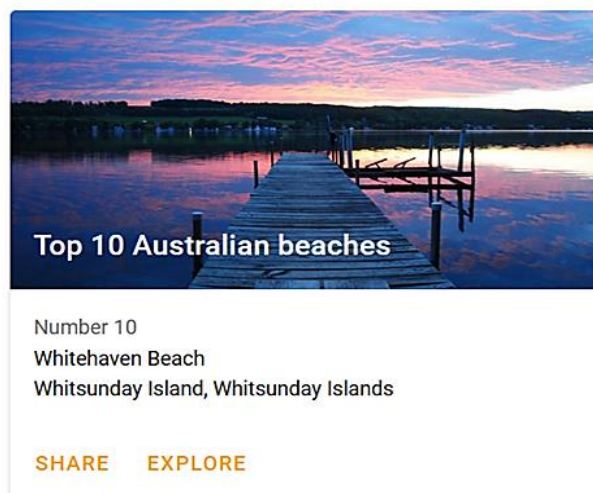
```
<v-card class="mx-auto" max-width="400">
  <v-img class="white--text align-end" height="200px" src="https://cdn.vuetifyjs.com/images/cards/docks.jpg"/>
  <v-card-title>Top 10 Australian beaches</v-card-title>
  </v-img>
  <v-card-subtitle class="pb-0">Number 10</v-card-subtitle>
  <v-card-text class="text--primary">
    <div>Whitehaven Beach</div>
    <div>Whitsunday Island, Whitsunday Islands</div>
  </v-card-text>
  <v-card-actions>
    <v-btn color="orange" text>Share</v-btn>
    <v-btn color="orange" text>Explore</v-btn>
  </v-card-actions>
</v-card>
```

Рис. 6. Пример разработки с использованием «Vuetify»

Ниже представлены результаты разработки – визуальное представление компонентов «Card» на рисунке 7 соответственно.



а)



б)

Рис. 7. «Card»: а) «Bootstrap Vue»; б) «Vuetify»

Как видно из рисунка 7, визуально компоненты имеют много общего, у каждого есть изображение, заголовок, некая информация и кнопки для переходов на дальнейший контент. Если сравнивать скрипты, то в целом их структура и названия элементов схожа.

Таким образом, выбор библиотеки не сильно влияет на отображение элементов на странице, однако выбор фреймворка может сильно повлиять на процесс создания гибких пользовательских интерфейсов, в некоторых случаях усложнив его, а в некоторых способен помочь разработчику сократить время работы над проектом.

Литература

1. Nelson B. Getting to Know Vue.js: Learn to Build Single Page Applications in Vue from Scratch. Minnesota: Apress, 2018. 280 p. <https://doi.org/10.1007/978-1-4842-3781-6>
2. Macrae C. Vue.js: Up and Running: Building Accessible and Performant Web Apps. O'Reilly Media, Inc., 2018. 174 p.
3. Hanchett E., Listwon B. Vue.js in Action. Manning, 2018. 375 p.
4. Шкарбан, Ф.В, Халилова З.Э., Абдуллаев А.Н. Javascript-фреймворки и библиотеки // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2017. № 3(17). С. 48-52.

© Волков В.Н., Новиков С.В., Герасимова Н.М., Полонская Т.С., 2022