



7 Лучших JavaScript Фреймворков/Библиотек для Front-End разработки

Описание

JavaScript с годами становится все более популярным, сообщество быстро растет, а разработчики постоянно развивают и создают инструменты для этого языка. Это делает его труднодоступным, когда нужно решить, какой инструмент/фреймворк/библиотеку использовать для решения конкретной задачи, потому что всегда есть несколько вариантов буквально для всего, что вы хотите сделать на JavaScript. Поначалу все еще сложно решить, какую библиотеку или фреймворк изучать. Эта статья фокусируется на демистификации преимуществ использования нескольких front-end фреймворков/библиотек JavaScript и в конечном итоге дает более четкое представление о них. Она призвана облегчить процесс принятия решения о выборе.

React



React – это не фреймворк, а библиотека JavaScript для создания пользовательских интерфейсов. Она имеет открытый исходный код и поддерживается Facebook и сообществом индивидуальных разработчиков. Изначально React был написан Джорданом Уоке как внутренний инструмент Facebook. Позже он был открыт для широкой публики в 2013 году, после чего приобрел широкую популярность.

Некоторые из особенностей включают следующее:

- Предоставляет реактивные, настраиваемые и многократно используемые компоненты
- Использует виртуальный DOM
- Чрезвычайно быстрый
- Основанный на компонентах
- Одностороннее связывание данных
- Возможность повторного использования кода
- За ним стоит энергичная, процветающая экосистема
- Удобная обработка управления состояниями

Установка/использование

React может быть использован на фронтенде двумя различными способами.

- Через CDN
- Используя Node.js

Через CDN

Вы можете обратиться на их официальный сайт, чтобы получить ссылку на скрипты, которую вы можете включить в тег header в вашей HTML-разметке. Выбирайте ссылки в зависимости от цели.

Например, если вы используете Node.JS в среде разработки, то:

```
<script crossorigin src="https://unpkg.com/react@16/umd/react.development.js">
```

```
<script crossorigin src="https://unpkg.com/react-dom@16/umd/react-dom.development.js">
```

и для производства

```
<script crossorigin src="https://unpkg.com/react@16/umd/react.production.min.js">
```

```
<script crossorigin src="https://unpkg.com/react-dom@16/umd/react-dom.production.min.js">
```

Используя Node.JS

Я предполагаю, что у вас уже установлен NodeJS. Чтобы установить React, просто введите следующую команду:

```
sudo npm i -g create-react-app --save-dev
```

После завершения установки введите следующую команду:

```
create-react-app my-first-react-application
```

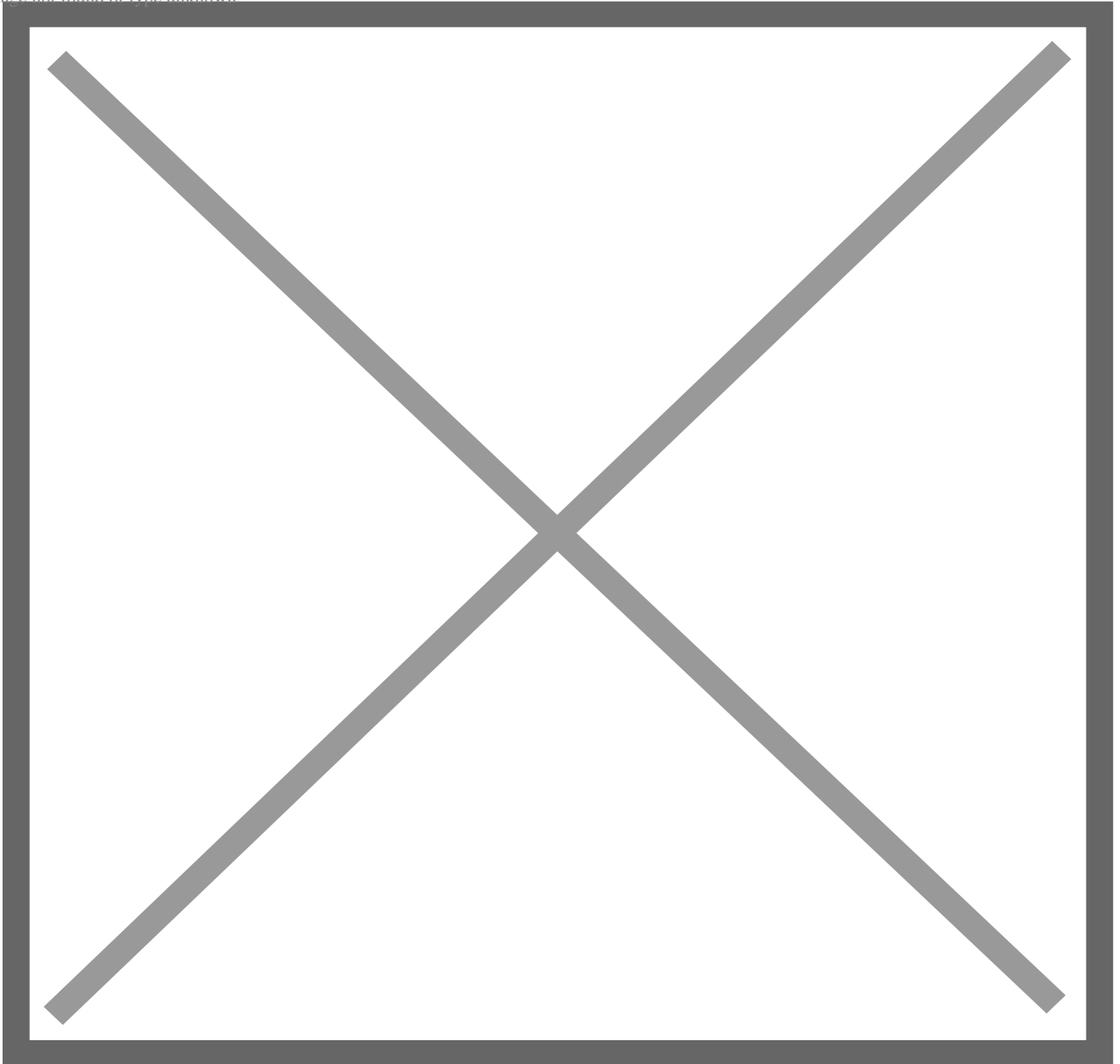
Приведенная выше команда установит все необходимые библиотеки, необходимые для правильной работы React, которые включают сервер разработки, webpack и babel.

Перейдите в папку my-first-react-application и выполните следующую команду:

```
npm start
```

Вышеуказанное запустит сервер разработки на порту 3000. И когда вы обратитесь к IP-адресу вашего сервера с портом 3000, вы должны увидеть что-то вроде следующего.

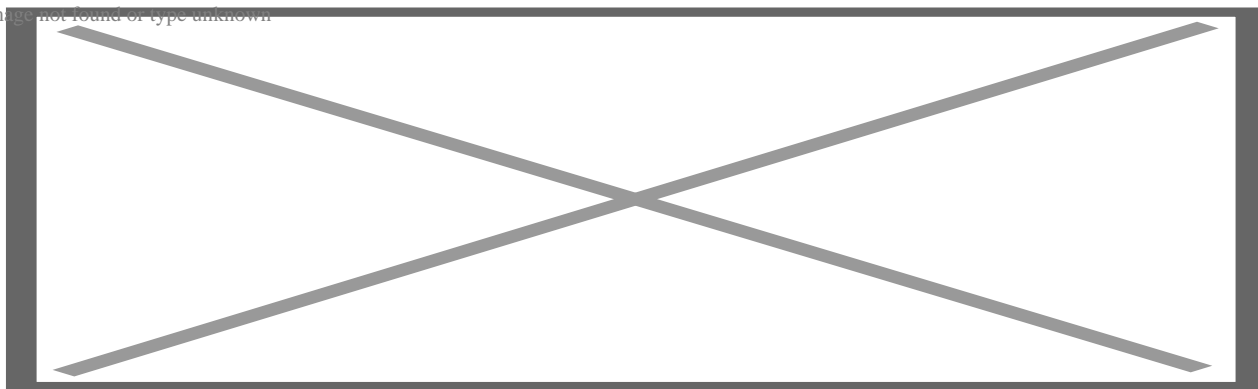
Image not found or type unknown



React набирает популярность и востребован рядом крупных организаций. Если вы заинтересованы в обучении, то я бы рекомендовал пройти этот полный курс.

Vue.js

Image not found of type unknown



Vue.js – это прогрессивный JavaScript-фреймворк для создания интерактивных пользовательских интерфейсов и одностраничных приложений. Это фреймворк типа “модель-вид”, в котором основная библиотека сосредоточена на слое представления. Vue популярен благодаря своей способности создавать одностраничные приложения. В отличие от React, Vue использует необработанный HTML, а не JSX.

Vue.js является открытым исходным кодом и изначально был создан Эваном Ю и выпущен публично в феврале 2014 года. Ниже перечислены некоторые особенности:

- Предоставляет реактивные и композитные компоненты представления.
- Использует виртуальный DOM
- Сохраняет фокус на основной библиотеке (т.е. маршрутизация и управление состоянием).
- Скопирование в CSS обрабатывается без использования CSS-In-Js.
- Одностороннее связывание внутри компонентов.
- Поддержка необходимых дополнений
- Возможность повторного использования кода

Установка/использование

Вы можете использовать Vue.js на front-end либо через CDN, либо с помощью Node.js.

Чтобы использовать способ CDN, вы можете добавить следующий скрипт в раздел заголовка вашей HTML-страницы.

```
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
```

Приведенный выше сценарий подходит для целей разработки, поскольку он включает важное консольное сообщение. Однако для производства следует

использовать приведенный ниже сценарий.

```
<script src="https://cdn.jsdelivr.net/npm/vue"></script>
```

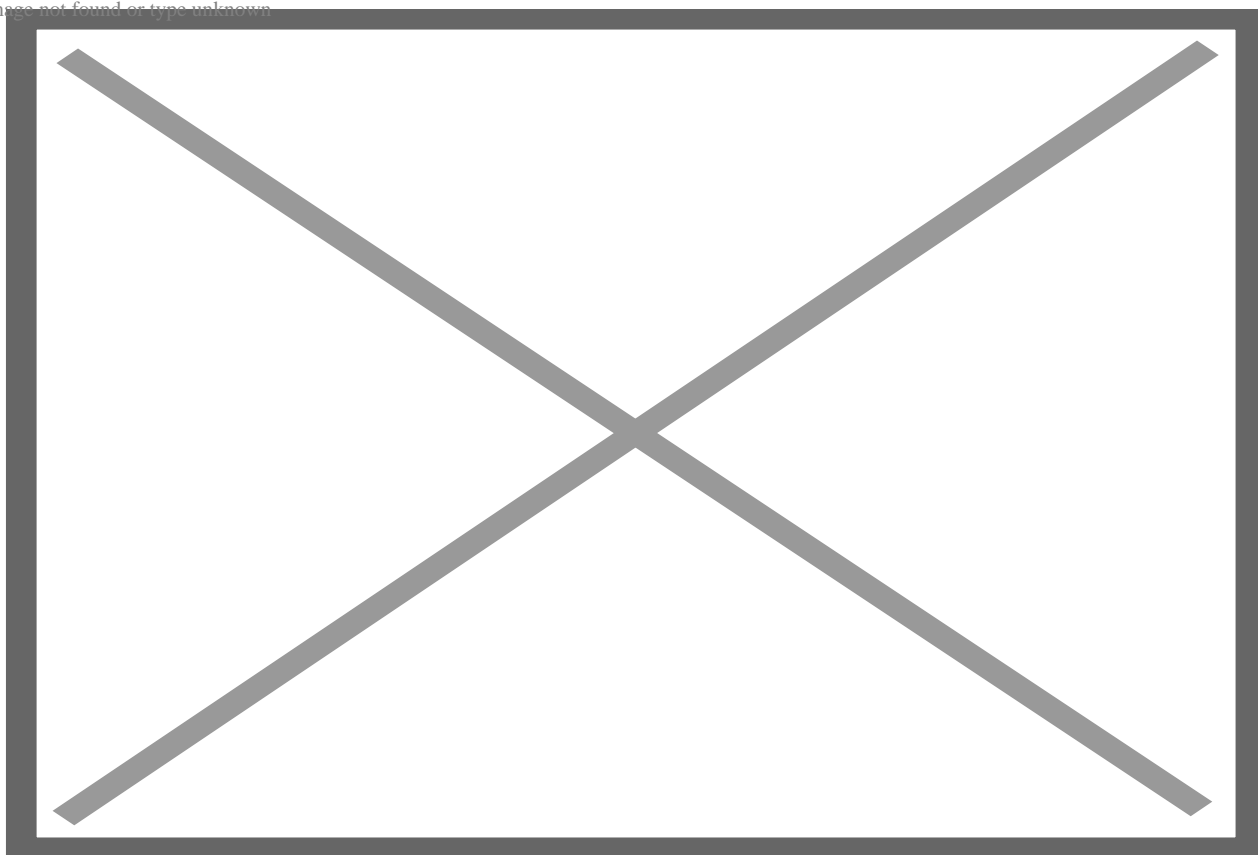
А для использования с Nodejs вы можете установить его с помощью команды npm.

```
npm install vue
```

Я настоятельно советую вам прочитать официальную документацию Vue JS, чтобы узнать больше или рассмотреть возможность пойти по этому пути.

Angular

Image not found or type unknown



Angular – это структурный JavaScript-фреймворк для динамических страниц. Он позволяет использовать HTML в качестве языка шаблонов и позволяет использовать синтаксис HTML для четкого и краткого выражения компонентов приложений. Это проект с открытым исходным кодом, поддерживаемый Google и другими участниками.

Установка

Убедитесь, что у вас установлена последняя версия Nodejs. Первое, что нам понадобится установить, это инструмент Angular CLI.

```
npm install -g @angular/cli
```

После установки мы можем создать новый проект с помощью следующей команды.

```
ng new my-first-angular-app
```

Следуйте инструкциям на экране. Это создаст некоторые файлы и папки и использует модуль npm для загрузки сторонних библиотек, необходимых для правильной работы Angular.

Чтобы запустить только что созданное приложение, выполните следующую команду из папки apps.

```
ng server
```

Это должно автоматически запустить сервер на порту 4200.

```
[root@lab my-first-angular-app]# ng serve
10% building 3/3 modules 0 active? ?wds?: Project is running at http://localhost:4200/
? ?wds?: webpack output is served from /
? ?wds?: 404s will fallback to //index.html
```

```
chunk {main} main.js, main.js.map (main) 47.8 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 264 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.15 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 10 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 3.81 MB [initial] [rendered]
Date: 2019-12-28T12:08:20.138Z - Hash: 5d4b93c7bf9e61979c4d - Time: 12864ms
** Angular Live Development Server is listening on localhost:4200, open your
browser on http://localhost:4200/ **
```

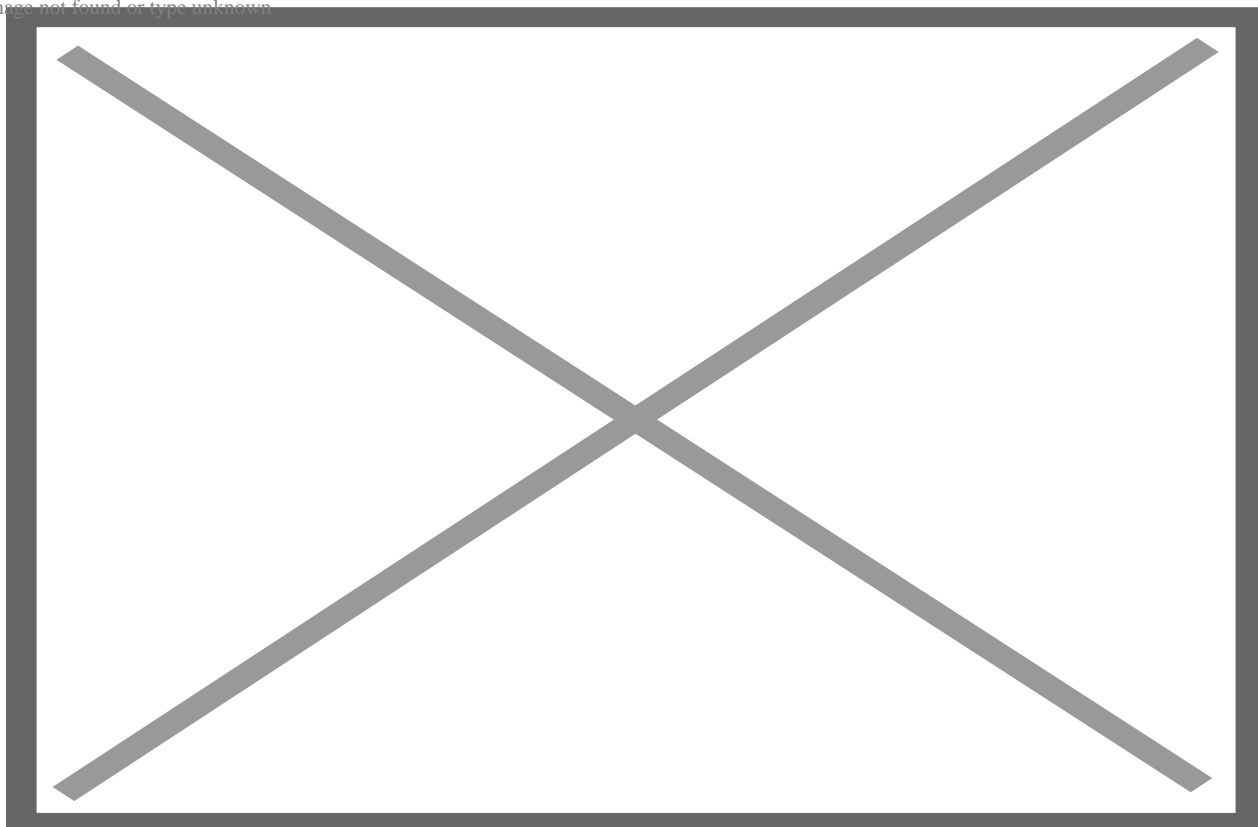
```
? ?wdm?: Compiled successfully.
```

Ember

Ember представляет себя как “продуктивный, проверенный в боях” JS-фреймворк для создания веб-приложений. Фреймворк предлагает готовые инструменты, которые помогут вам создать масштабируемое приложение. EmberJS поставляется с Ember CLI, позволяя пользователям создавать новые сущности с помощью генератора кода и обеспечивая правильную структуру файлов для вашего

приложения.

Image not found or type unknown



EmberJS известен следующими особенностями:

- **Встроенная среда разработки:** Эта среда предлагает тестовый запуск, автозагрузку и быструю перестройку.
- **Встроенный маршрутизатор:** Комплексная система маршрутизации Ember сочетает асинхронную загрузку данных с параметрами запроса и динамическими сегментами url.
- **Поставляется с Ember Data:** Ember позволяет использовать любой слой данных. Однако вы можете пойти еще дальше и получить доступ к данным из нескольких источников одновременно, благодаря библиотеке Ember Data.
- **Поддерживает тестирование:** Тесты автоматически генерируются каждый раз, когда вы создаете новую сущность для приложения Ember. Платформа поддерживает три уровня тестирования, гарантируя, что ваше приложение будет готово для потребителей.

Использование/установка

Вы можете установить Ember через Node Package Manager (npm):

```
npm install -g ember-cli
```

Затем вы можете создать новое приложение с помощью этой команды:

```
ember new ember-quickstart --lang en
```

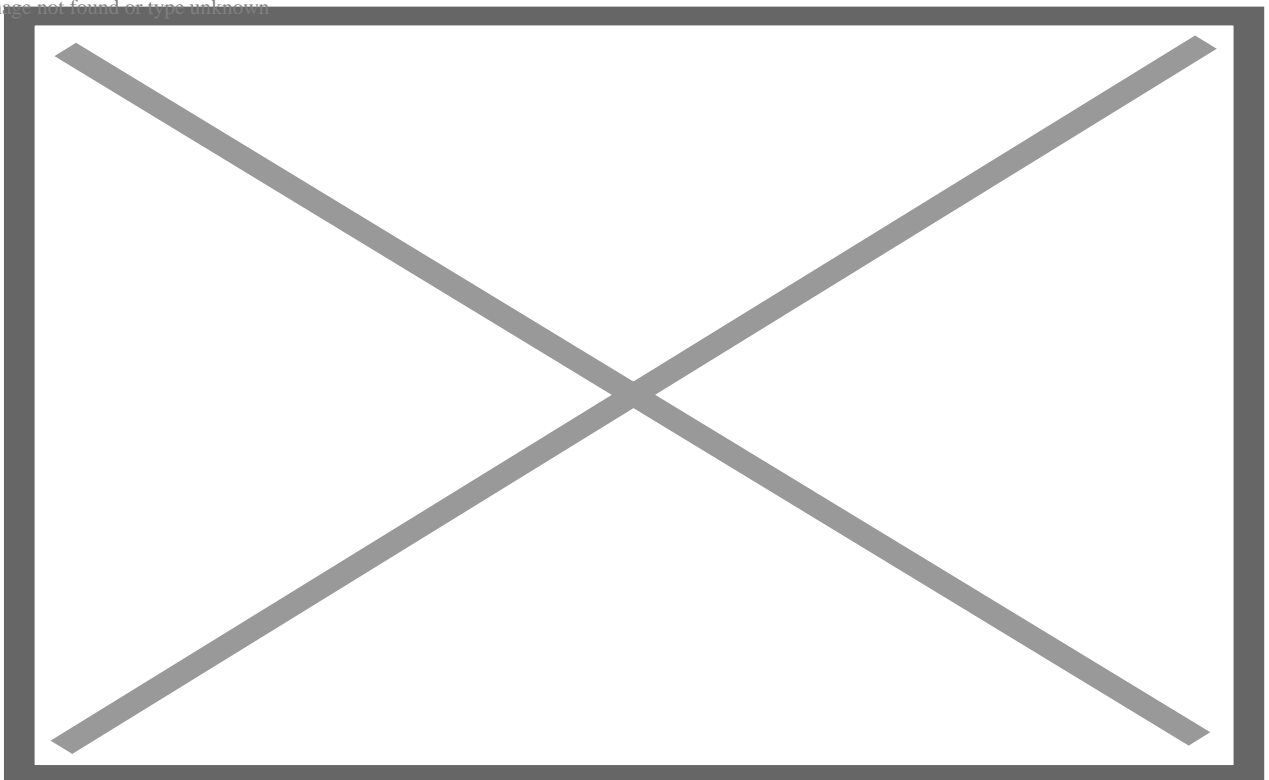
Наконец, вы можете запустить сервер разработки с помощью этих двух команд:

```
cd ember-quickstart
```

```
ember serve
```

Теперь ваше приложение готово, и когда вы посетите сайт <http://localhost:4200/>, вы получите следующий результат:

Image not found or type unknown

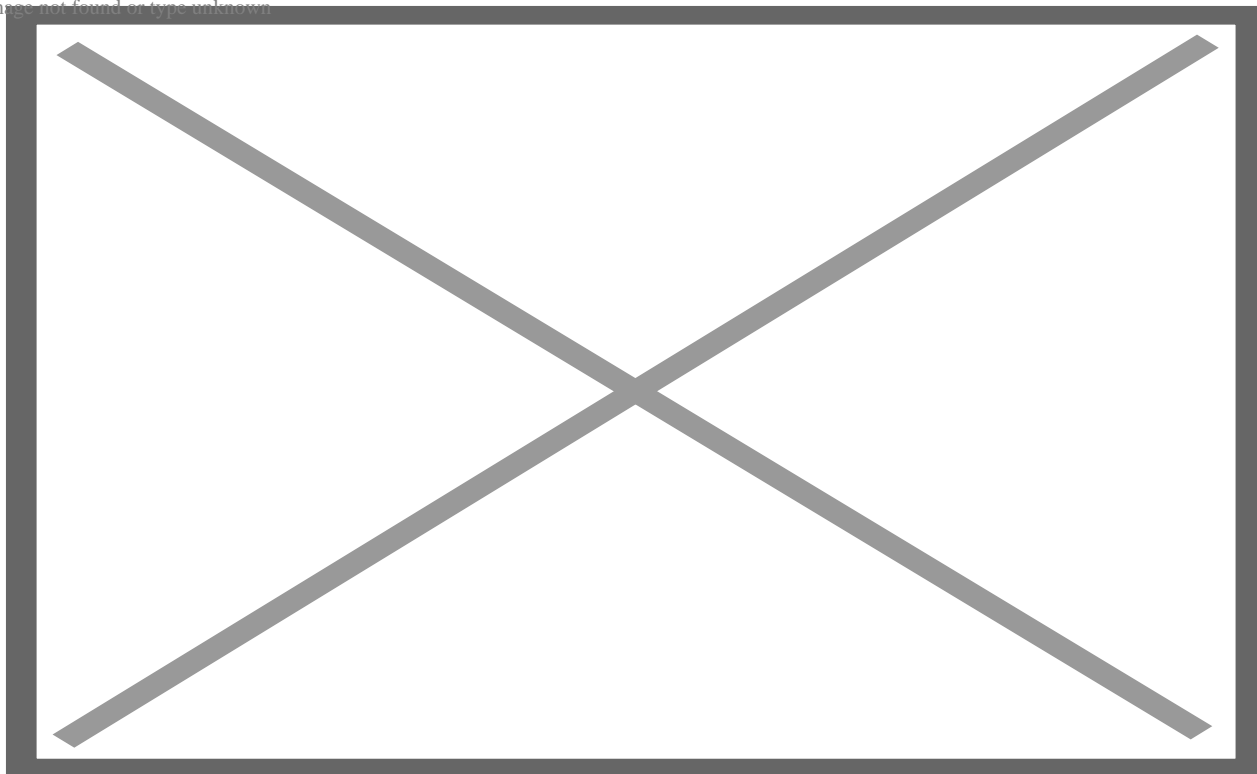


Ознакомьтесь с официальной документацией по созданию приложения в соответствии с вашими потребностями.

Svelte

Svelte – это еще один фреймворк JavaScript, призванный помочь разработчикам создавать элегантные интерактивные пользовательские интерфейсы. Одним из основных отличий Svelte от других JavaScript frontend библиотек/фреймворков, таких как Vue и React, является время компиляции.

Image not found or type unknown



React и Vue полагаются на среду выполнения для интерпретации и выполнения вашего кода. С другой стороны, Svelte преобразует ваш код в идеальный JavaScript во время сборки. Эта особенность позволяет создавать более быстрые и компактные приложения.

Svelte имеет следующие особенности:

- **Архитектура, основанная на компонентах:** Svelte позволяет разделить ваше приложение на небольшие многократно используемые компоненты. Такой подход позволяет легко изменять простые компоненты, не переписывая весь код.
- **Полностью реактивный подход:** Svelte позволяет создавать интерактивные и динамичные приложения, поскольку он позволяет декларативно определить, как пользовательский интерфейс (UI) будет реагировать на изменения состояния.

- **Подход на основе компилятора:** Платформа преобразует ваш код в JavaScript во время сборки, в результате чего получаются небольшие и быстрые приложения.

Использование/установка

Вы можете создать свое первое приложение, используя Node Package Manager (npm).

```
npm create svelte@latest myapp
```

Измените каталог на созданное приложение с помощью этой команды:

```
cd myapp
```

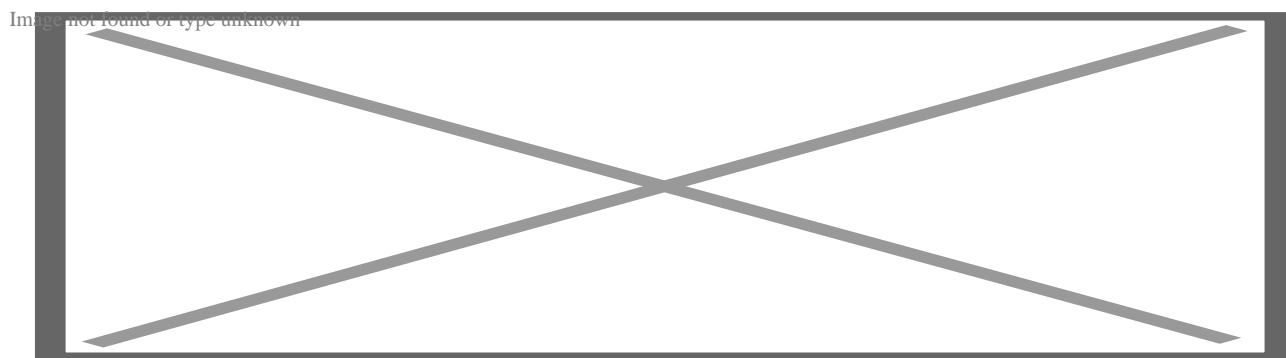
Установите все необходимые зависимости:

```
npm install
```

Запустите сервер:

```
npm run dev
```

После выполнения всех этих команд вы увидите что-то вроде этого:



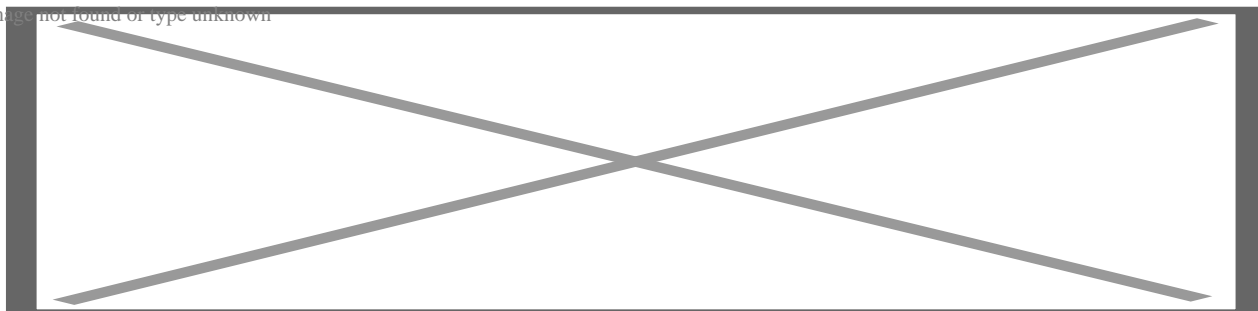
Вы можете прочитать официальную документацию, чтобы понять, как начать работу с Svelte и создать свое первое приложение.

Backbone.js

Backbone.js – это легкий JavaScript-фреймворк для фронтенда, который придает структуру веб-приложениям. Этот фреймворк поставляется с представлениями, поддерживающими декларативную обработку событий, коллекциями с богатым API

перечислимых функций и моделями с пользовательскими событиями, а также привязкой к ключу и значению.

Image not found of type unknown



Backbone.js обладает следующими особенностями:

- **Расширяемость и модульность:** Backbone.js позволяет разделить ваше приложение на небольшие, многократно используемые компоненты. Таким образом, вы можете добавлять или удалять определенные функции из вашего приложения, не переписывая весь исходный код.
- **Открытый исходный код:** Backbone.js – это бесплатный фреймворк, исходный код которого доступен на GitHub.
- **Событийно-ориентированная архитектура:** При использовании Backbone.js вы можете вызывать и связывать пользовательские события,
- **Совместимость с RESTful API:** Вы можете легко интегрировать серверные технологии в приложение Backbone.js благодаря его совместимости с REST API.

Использование/установка

Сначала создайте приложение Backbone.js и войдите в него. Вы можете выполнить следующие команды:

```
mkdir backbone-app
```

```
cd backbone-app
```

Инициализируйте ваш новый проект npm с помощью этой команды:

```
npm init
```

Установите Backbone.js с его зависимостями (jQuery и underscore.js). Используйте эту команду:

```
npm install backbone underscore jquery --save
```

Теперь вы можете следовать официальной документации и настраивать свои модели, представления и коллекции, готовясь к созданию своего первого приложения.

Mithril.js

Mithril.js – это небольшой клиентский JavaScript-фреймворк для создания великолепных одностраничных приложений. Он имеет размер всего 9,17 КБ в gzipped формате и используется такими крупными компаниями, как Nike и Vimeo.

Mithril.js обладает следующими возможностями:

- **Встроенная маршрутизация:** Перемещение по страницам и создание различных маршрутов в Mithril.js очень просто благодаря надежной функции маршрутизации.
- **Легкость в освоении:** Mithril.js подходит даже для начинающих JavaScript благодаря простому синтаксису.
- **Быстрота:** Благодаря небольшому размеру эта библиотека загружается за миллисекунды и поэтому подходит для приложений, где скорость работы критична.
- **Совместимость с большинством браузеров:** Вы можете запускать приложения на Mithril.js в современных браузерах, таких как Chrome, Firefox, Edge и Safari.
- **Встроенная библиотека XHR:** Эта библиотека позволяет легко выполнять AJAX-запросы.

Использование/установка

Создайте новый проект.

Вы можете использовать эти команды:

```
mkdir mithril-app
```

```
cd mithril-app
```

Используйте эту команду для установки Mithril.js через npm:

```
npm install mithril --save
```

Инициализируйте ваше приложение, чтобы создать файл package.json, в котором будут находиться ваши зависимости. Используйте эту команду:

```
npm init --yes
```

Добавьте стартовые скрипты в бит scripts в файле package.json:

```
{  
  "name": "my-project",  
  "scripts": {  
    "start": "webpack src/index.js --output bin/app.js -d --watch"  
  }  
}
```

Установите Webpack с помощью этой команды:

```
npm install webpack webpack-cli --save-dev
```

Теперь у вас есть базовая структура приложения Mithril.js. Обратитесь к официальной документации, чтобы продолжить создание своего приложения.

Заключение

Итак, что вы выберете для изучения – это скорее личное предпочтение, чем вопрос “что лучше”.

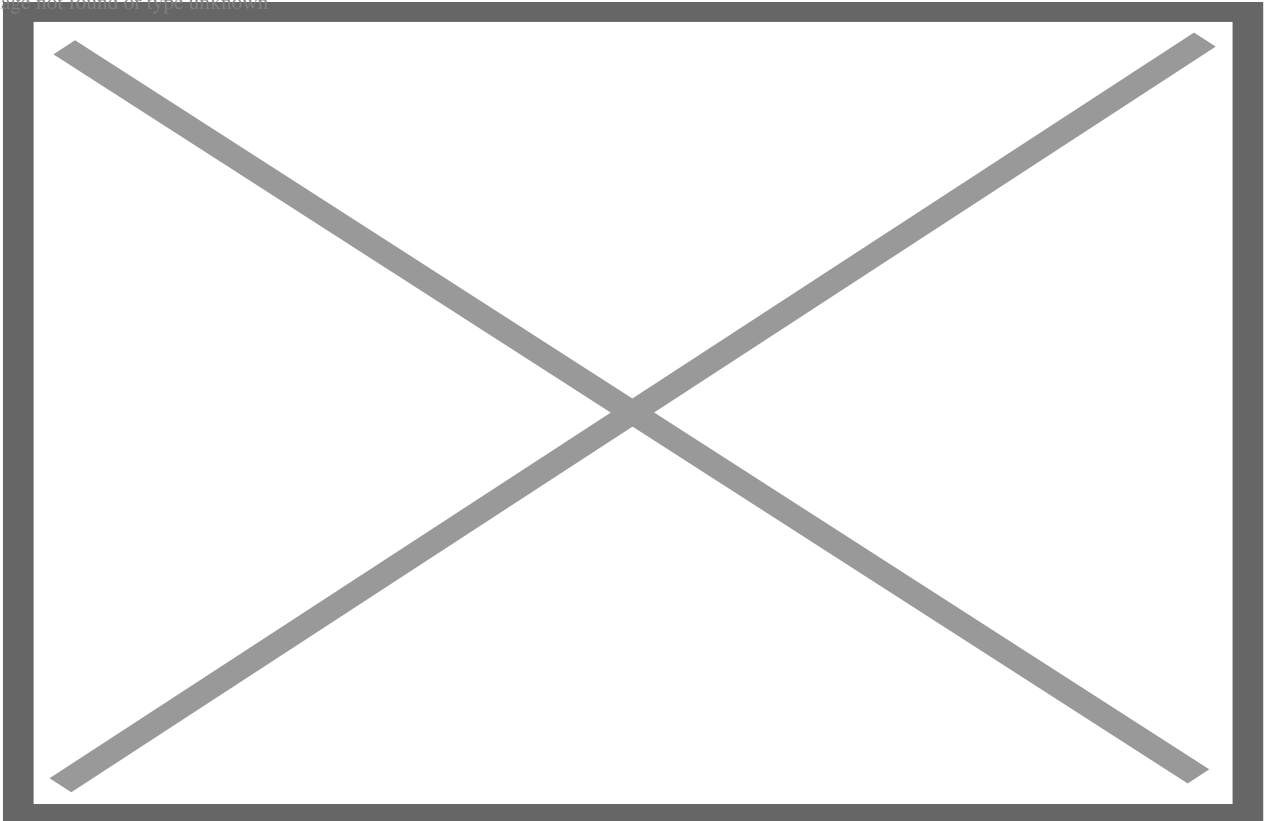
Все перечисленные выше фреймворки/библиотеки великолепны. Вот краткий обзор:

- Вы должны изучать Angular, если вам нужен фреймворк, на который вы хотите полагаться без необходимости иметь дело с внешними зависимостями.
- Вам следует изучать React, если вы хотите быстро создавать PWA, одностраничные приложения, и вам удобно работать с JSX.
- React имеет самое активное сообщество и больше возможностей для трудоустройства благодаря своему большому сообществу.
- С React относительно легко начать работу.

- React обладает высокой степенью настраиваемости и рассматривает каждый элемент пользовательского интерфейса как компонент.
- Vue обладает теми же преимуществами, что и React, но без JSX.
- Рынок вакансий для Vue постоянно растет.

Вот график Google Trends, показывающий сравнение популярности этих трех продуктов.

Image not found or type unknown



Дата Создания

04.05.2023