

## Laravel или Node.js: прямое сравнение

### Описание

Laravel и Node.js это две из самых влиятельных и эффективных [технологий веб-разработки](#) за последнее десятилетие. Их производительность и исключительные возможности решения проблем, которые выгодны для современных разработчиков, значительно повлияли на индустрию веб-разработки с момента их появления. Laravel – это PHP-фреймворк с большей прочностью и стабильностью. Поскольку PHP [поддерживает 78% веб-ресурсов](#), Laravel получает значительную долю рынка.

Node.js — также называемый Node — это среда выполнения JavaScript, созданная с использованием JavaScript, C и C ++, и она предоставляет все инструменты, необходимые разработчику для разработки с полным стеком и высокой производительностью. Оба они выдающиеся по целому ряду направлений. Вы захотите ознакомиться с каждым из них, чтобы определить правильный вариант для вашего будущего проекта.

**Почему мы сравниваем эти две технологии?** Хотя фреймворк PHP и среда выполнения JS не являются прямыми аналогами, мы заметили множество пользователей, которые поднимали вопрос о Laravel vs Node онлайн. В этой статье мы попытаемся ответить на этот вопрос, рассмотрев ключевые особенности, основные различия и варианты использования для каждой технологии.

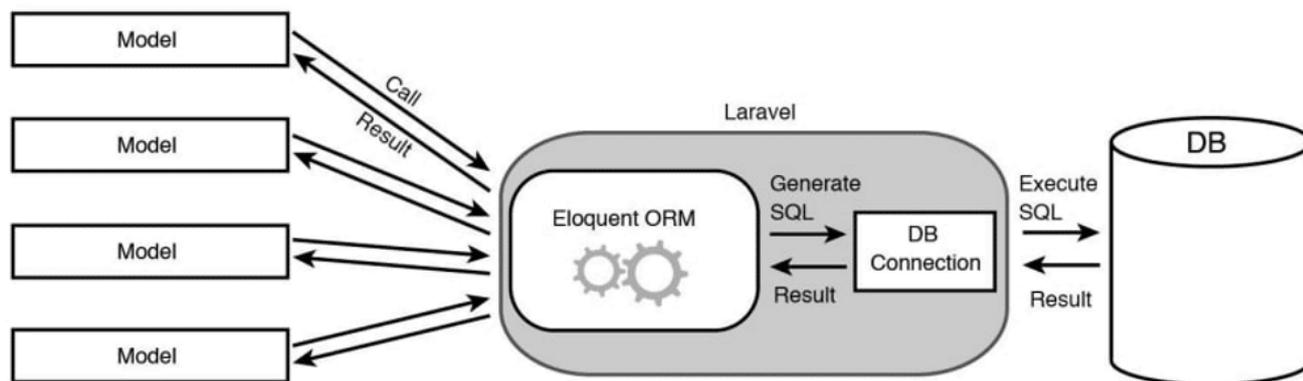
Давайте начнем!

## Что такое Laravel?



Официальный логотип Laravel. (Источник изображения: [Github](#))

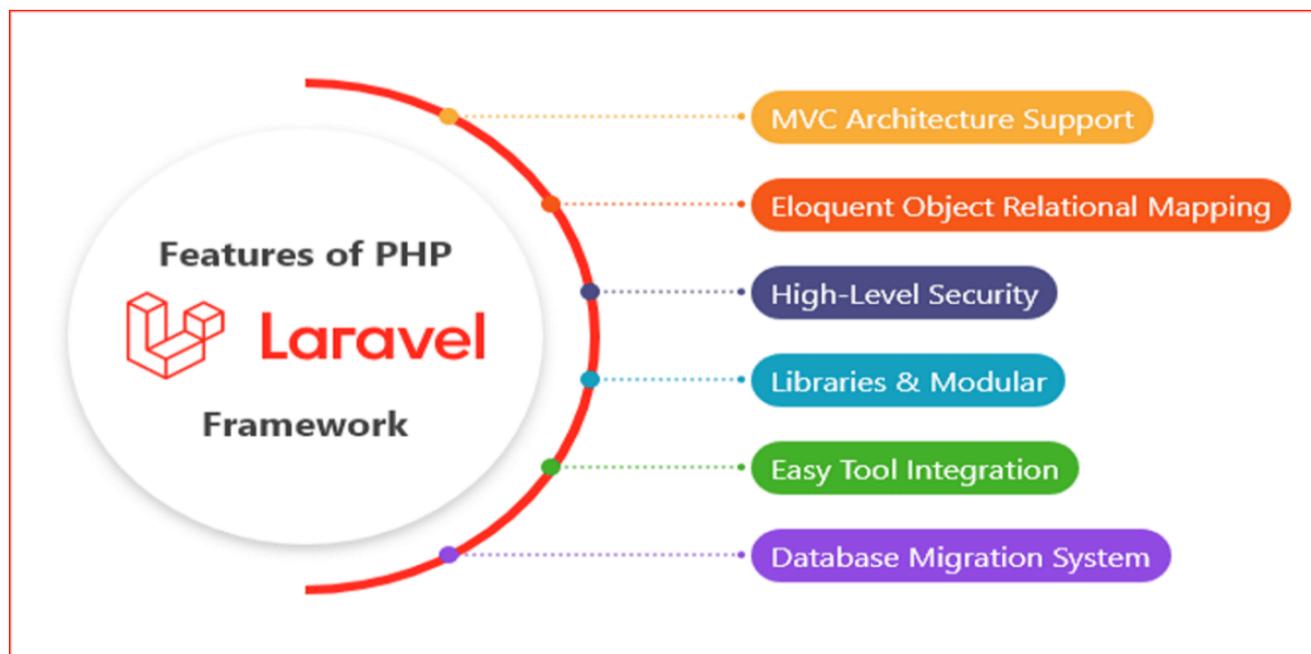
[Laravel](#) – это PHP-фреймворк, предназначенный для упрощения создания [современных PHP-приложений](#). Многие разработчики используют его для упрощения процесса разработки благодаря его надежной экосистеме, использующей встроенные возможности Laravel и множество совместимых пакетов и расширений. Laravel привел к выдающимся достижениям в разработке и неизменно оправдывает ожидания пользователей в области программирования. Это фреймворк, который [предотвратил вымирание PHP](#). Laravel облегчает веб-разработку, предлагая регулярно используемые методы или API, такие как маршрутизация или аутентификация. Вы можете использовать эти готовые функции и шаблоны для использования эффективной по времени процедуры веб-разработки, а не кодировать все с нуля.



Laravel красноречивый ORM. (Источник изображения: [TheCodework](#))

[Разработка Laravel при разработке](#) приложений следует архитектуре Model-View-Controller (MVC). Он поддерживает структуру каталогов проекта, реализует меры безопасности и содержит множество пакетов, таких как Voyager, Laravel-Breeze, Laravel-Debugbar и т. Д. В отличие от других современных платформ веб-разработки, Laravel упрощает взаимодействие с базой данных, позволяя использовать необработанный SQL, свободный конструктор запросов и Eloquent ORM с любой поддерживаемой базой данных, такой как MariaDB или MySQL.

## Основные возможности Laravel



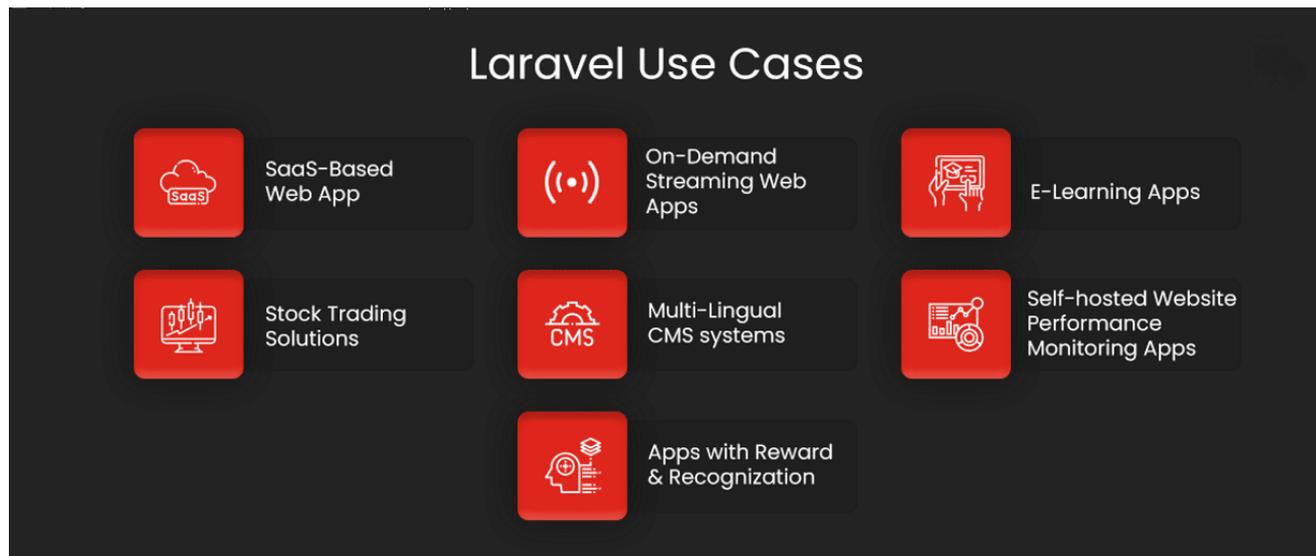
Возможности фреймворка Laravel. (Источник изображения: [MavenCluster](#))

Давайте рассмотрим некоторые из основных функций Laravel:

- **Механизм шаблонов блейда:** Blade, внутренний механизм создания шаблонов Laravel, создает представление путем интеграции нескольких шаблонов в модель данных. Он имеет собственную структуру управления, состоящую из условных операторов и циклов, что обеспечивает чистый и динамический код.
- **MVC) архитектура: Модель-представление-контроллер** (архитектура MVC Laravel обеспечивает разделение уровней представления и бизнес-логики, что ускоряет процесс разработки. Такой дизайн MVC повышает эффективность приложений и повышает безопасность и масштабируемость для разработчиков.
- **Eloquent объектно-реляционное отображение (ORM):** Laravel использует Eloquent, структуру объектно-реляционного отображения (ORM). ORM позволяет веб-разработчикам создавать запросы к базе данных на PHP, а не на SQL. Он также объединяет программистов и таблицы базы данных, назначая соответствующие модели, что приводит к более высокой производительности, чем старые [фреймворки PHP](#).
- **Встроенные пакеты:** Многочисленные предварительно настроенные пакеты в Laravel упрощают ваш код и помогают поддерживать чистоту среды. Например, пакет Socialite интегрирует аутентификацию Facebook или Google на ваш веб-сайт, позволяя пользователям регистрироваться одним щелчком мыши.
- **Безопасность:** Laravel обеспечивает внутреннюю безопасность вашего веб-сайта с помощью встроенной системы безопасности. Поскольку Laravel функционирует как промежуточная система, которая перехватывает все запросы и процессы, это предотвращает [удаленное внедрение вредоносного кода хакерами на ваш сервер](#). Более того, встроенный токен CSRF защищает вас от других угроз безопасности.
- **Artisan:** Artisan – это инструмент командной строки, который автоматизирует повторяющиеся задачи программирования в среде Laravel. Он может создавать структуру базы данных, скелет кода или перенос кода, а также создавать и поддерживать исходные файлы MVC через командную строку и управлять такими ресурсами с помощью их опций.

## Примеры использования Laravel

---



Примеры использования Laravel. (Источник изображения: [Aglowid](#))

Давайте рассмотрим несколько убедительных примеров использования Laravel, от внештатных разработчиков до компаний корпоративного уровня:

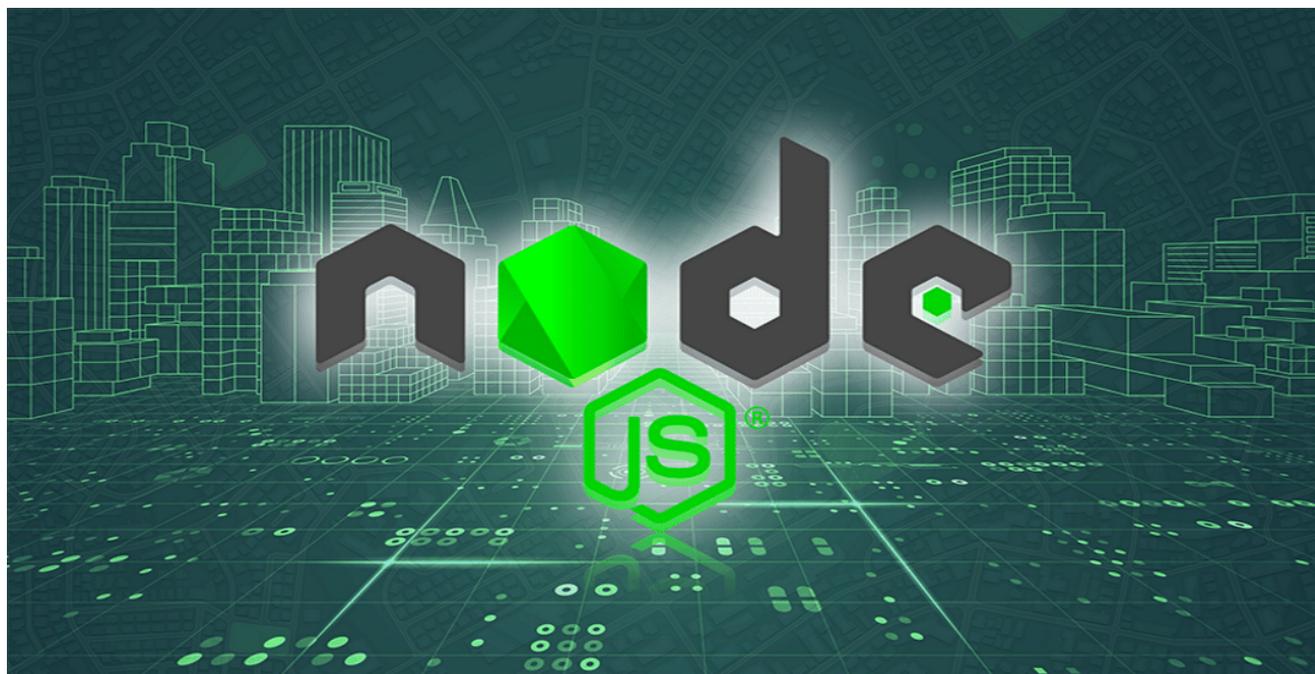
- **Высокопроизводительные веб-приложения:** разработчики используют платформу Laravel для создания высокопроизводительных и безопасных онлайн-приложений или веб-сайтов.
- **Приложения с архитектурой микросервисов:** Микропрограммный дизайн Laravel очень эффективен для приложений электронной коммерции. Он состоит из нескольких модулей, которые работают независимо и объединяют крупномасштабные приложения Laravel на основе отдельных строительных блоков. В результате становится легко разделить длинную программу на более мелкие независимые части.
- **Приложения высокого уровня безопасности:** такие функции Laravel, как хеширование секретных слов, вычисления хеширования Bcrypt и структурированный SQL, позволяют разрабатывать высокозащищенные приложения. Более того, все файлы cookie, созданные Laravel, зашифрованы и подписаны; если клиент изменит их, они станут недействительными. Следовательно, Laravel более безопасен, чем PHP.
- **Готовые приложения:** в Laravel имеется множество готовых приложений для экономии времени разработчиков и пользователей. Если вам нужно создать похожее приложение, вы можете легко использовать шаблоны, найденные на веб-сайте Laravel, чтобы получить преимущество.

## Кто использует Laravel?

Как самая популярная платформа веб-разработки, Laravel создала востребованную нишу для разработчиков. Сегодня компании всех размеров, от стартапов до корпораций, полагаются на Laravel. Вот лишь некоторые компании, которые используют Laravel:

- BBC
- Pfizer
- 9GAG
- TourRadar
- O vas
- Novell, Inc.
- Накладная Ниндзя
- Элисон
- Laracasts
- Альфа-кодеры

## Что такое Node.js ?



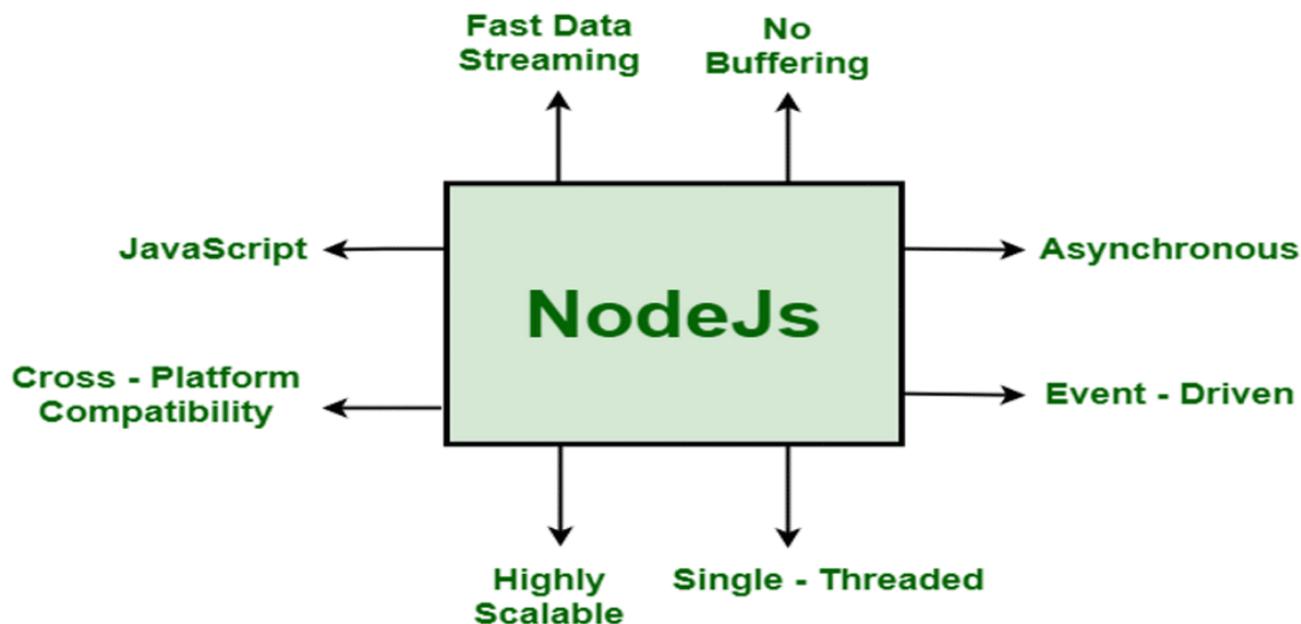
The Node.js логотип. (Источник изображения: [ZTM](#))

Многие энтузиасты ИТ и разработки часто признают, что Node – это язык

программирования, что является распространенным и широко распространенным заблуждением. Node больше похож на супер-костюм для [языка программирования JavaScript](#), который наделяет его особыми возможностями, которых нет у обычного языка программирования – например, возможностями как для разработки на стороне клиента, так и на стороне сервера. [Node.js](#) это однопоточная кроссплатформенная среда выполнения с открытым исходным кодом, которая выполняет код JavaScript вне браузера для разработки на стороне сервера. Он также используется для создания сетевых приложений. Как упоминалось ранее, он использует механизм выполнения JavaScript Google V8 и неблокирующую архитектуру ввода-вывода, управляемую событиями.

Это идеальный инструмент для любого разработчика, который теряет голову, беспокоясь о создании другого конца. Learning Node делает вас разработчиком с полным стеком, который может создавать как [интерфейс, так и серверную](#) часть приложения. Node использует асинхронный, управляемый событиями подход, который идеально подходит для разработки приложений, требующих больших затрат времени и данных, которые выполняются на разнесенных устройствах. Node также обладает высокой масштабируемостью для создания приложений реального времени благодаря своей неблокирующей природе. Node может похвастаться обширной библиотекой модулей JavaScript, которые могут быть полезны при создании веб-приложений. Итак, хорошая новость заключается в том, что если вы застряли в середине разработки, помощь всегда рядом. Кроме того, [Node прост в установке](#), поэтому вы можете начать использовать его прямо сейчас.

## Основные характеристики Node



Основные характеристики Node. (Источник изображения: [GeeksforGeeks](http://GeeksforGeeks))

Давайте рассмотрим некоторые из основных функций Node:

- **Асинхронный и управляемый событиями:** все API, предоставляемые библиотекой node.js, являются асинхронными (неблокирующими). Серверы на базе node.js никогда не ожидают данных от API. Вместо этого, после посещения API, сервер переходит к следующему. Он также использует систему уведомлений Events для получения и отслеживания ответов на предыдущие запросы API.
- **Full-stack:** Node предназначен для понимания и выполнения кода JavaScript. С Node разработка на стороне клиента и сервера – это легкий бриз. Это позволило разработчикам, владеющим JavaScript, разработать приложение с полным стеком. Таким образом, интерфейс и серверная часть синхронизированы, поскольку вы можете использовать JavaScript с обеих сторон.
- **Однопоточный:** node.js восхитительно масштабируется благодаря своей однопоточной архитектуре цикла событий. В отличие от традиционных серверов, которые используют ограниченные потоки для обработки запросов, node.js механизм событий делает его свободным от блокировок и масштабируемым. По сравнению с HTTP-сервером Apache, Node использует однопоточное программное обеспечение, которое может обрабатывать больше запросов.

- 
- **Кросс-платформенная совместимость:** Node совместим с Windows, Unix, Linux, Mac OS X и мобильными платформами. Вы можете объединить его с соответствующим пакетом для создания самостоятельного исполняемого файла. Использование NW.js и Electron, разработчики могут создавать кроссплатформенные приложения реального времени без написания кода для каждой платформы отдельно.
  - **Быстрая потоковая передача данных:** Node – это легкая, быстрая и встроенная платформа потоковой передачи, поддерживающая API. Это позволяет пользователям пересылать запросы друг другу. Таким образом, он передает данные непосредственно в пункт назначения. Приложение node.js выводит данные в блоках из-за концепции функции обратного вызова. Это сокращает требуемое время обработки и обеспечивает эффективную потоковую передачу данных без буферизации.
  - **Протокол интернета вещей:** node.js не требует значительного объема памяти и ресурсов на стороне сервера. Поэтому разработчики программного обеспечения могут использовать IoT development для обеспечения одновременных соединений между несколькими устройствами. Кроме того, Node поддерживает протокол передачи телеметрии с очередью сообщений (MQTT), который используется большинством приложений Интернета вещей. Таким образом, внутренняя интеграция независимых и сторонних устройств становится легкой.
  - **Диспетчер пакетов node.js (npm):** диспетчер пакетов node.js (npm) является встроенным менеджером пакетов для node.js. npm помогает загружать и устанавливать основные пакеты приложений и позволяет использовать код из других мест, а не писать его с нуля. NPM – это крупнейший в мире реестр библиотек программного обеспечения. В дополнение к помощи в установке библиотек пакетов, он обрабатывает зависимости библиотек.
  - **Производительность:** поскольку движок JavaScript версии 8 в Google Chrome работает под управлением Node, он может выполнять код с высокой скоростью. Более того, он преобразует код JavaScript в машинный код, что упрощает и ускоряет его реализацию. Его превосходная производительность обусловлена такими концепциями, как асинхронное программирование и неблокирующие процедуры ввода-вывода. Кроме того, Node широко использует события, что способствует его необычайной скорости.
  - **масштабируемость node.js приложений** : [очень масштабируемы](#) из-за их асинхронной (неблокирующей) работы. node.js работает в одном потоке, где он начинает обработку запроса, как только он поступает, и готовится к приему
-

последующего запроса. Кроме того, после подготовки ответа он отправляется клиенту.

## Варианты использования node.js



Варианты использования node.js. (Источник изображения: [Simform](#))

Давайте рассмотрим несколько убедительных примеров использования Node для разработчиков и предприятий:

- **Приложения для веб-поточковой передачи:** Node включает собственный потоковый API, который может эффективно передавать потоки с использованием каналов Unix. Аналогично, он позволяет загружать только определенные части веб-приложения, а остальные загружаются в фоновом режиме, чтобы избежать буферизации. Таким образом, такие компании, как Netflix, используют Node, чтобы предоставить своим клиентам элегантный потоковый сервис.
- **Приложения реального времени:** приложения реального времени, возможно, являются лучшим вариантом использования для Node. node.js позволяет повторно использовать и совместно использовать пакеты библиотечного кода, что ускоряет синхронизацию данных между сервером и клиентом. Кроме того, WebSockets и Event API node.js могут обрабатывать интенсивный ввод-вывод. Таким образом, Node пользуется популярностью среди разработчиков решений для совместной работы в режиме реального

времени.

- **Масштабируемые приложения:** Node содержит несколько передовых возможностей, таких как кластерные модули, которые позволяют распределять нагрузку на несколько ядер процессора, что упрощает получение желаемых результатов с помощью небольших модулей без использования оперативной памяти. Таким образом, он стал предпочтительной альтернативой для всех приложений и платформ с постоянным увеличением пользовательской базы.
- **Прокси на стороне сервера: сторонние прокси** вызывают хаос и низкую производительность веб-приложений. Даже известные прокси, такие как [Nginx](#) и [HAProxy](#), не могут обрабатывать несколько запросов одновременно. `node.js` может поддерживать множество одновременных неблокирующих подключений, что делает его идеальным прокси-сервером на стороне сервера. Например, [BBC News](#) использует Node для управления соединением со своими сторонними информационными серверами.
- **Рендеринг на стороне сервера:** используя рендеринг на стороне сервера (SSR) в Node, ваше приложение может сгенерировать полностью отрисованную веб-страницу и отправить ее непосредственно клиенту, минуя браузер. Он также имеет поддержку библиотек и характеристики браузера вместо модели параллелизма. Недавно Airbnb перенес аналогичный сервис, который создавал бы аналогичные полностью сформированные веб-страницы, отображаемые сервером в Node.

## Кто использует Node?

С момента своего создания в 2009 году Node сыграл ключевую роль для нескольких лидеров рынка. Приложения, разработанные [Node](#), в настоящее время используются предприятиями всех размеров, от стартапов до компаний из списка Fortune 500. Давайте взглянем на некоторые из ведущих компаний отрасли, внедряющих Node:

- Netflix
- [PayPal](#)
- [Twitter](#)
- Spotify
- eBay
- [LinkedIn](#)

- Airbnb
- Uber
- Walmart
- Trello

## Laravel против node.js: прямое сравнение

Теперь, когда мы понимаем, что такое Laravel и Node, их функции и приложения, давайте сравним и сравним их по конкретным параметрам более подробно.

### Сходства:

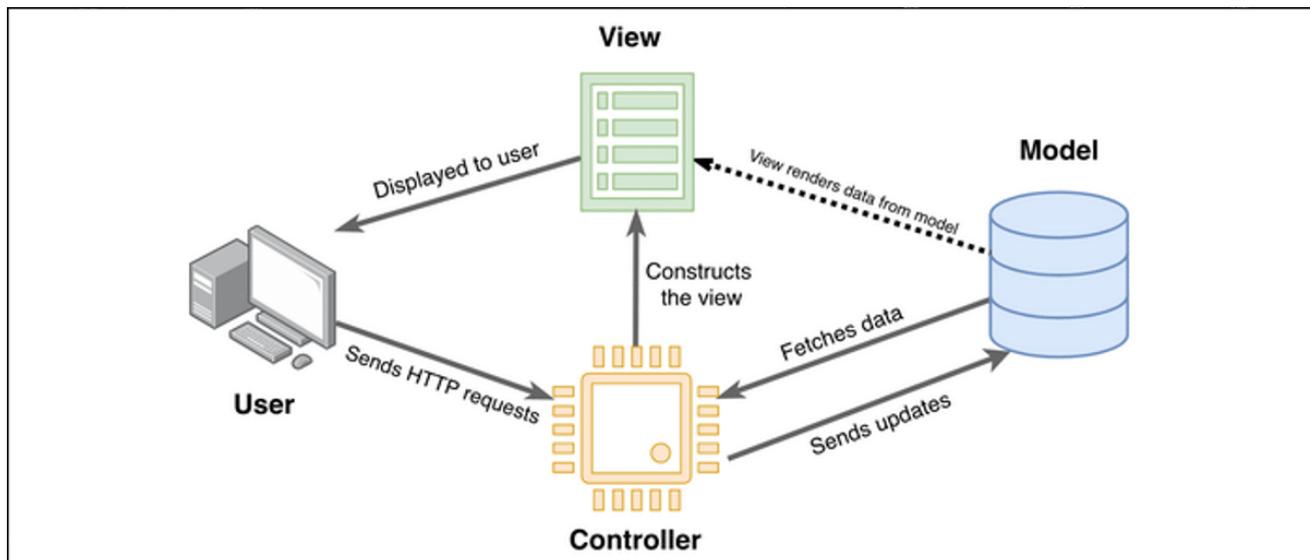
Хотя Laravel и Node выглядят непохожими, между ними есть заметное сходство. Прежде чем мы углубимся в анализ этих чудес программирования, давайте обсудим сходства между ними.

- **Открытый исходный код:** и Laravel, и Node являются приложениями с открытым исходным кодом. Node – это среда выполнения с открытым исходным кодом, в то время как Laravel – это платформа веб-разработки с открытым исходным кодом.
- **Полный стек:** и Laravel, и Node обеспечивают [разработку с полным стеком](#). Во внешнем интерфейсе Laravel использует JavaScript, а во внутреннем – PHP. JavaScript используется как во внешнем интерфейсе, так и во внутренней части Node.
- **Надежность:** Laravel и Node поддерживаются и используются крупными разработчиками программного обеспечения по всему миру. И огромное количество программных приложений построено на этих двух языках. Эти два остаются на переднем крае всех веб-технологий из-за их надежности и надежности.

### Структура и гибкость

Эффективность и производительность языка программирования зависят от его структуры, которая управляет функциями. Laravel использует архитектуру Model View Controller (MVC), тогда как Node использует однопоточную архитектуру цикла событий. Давайте сравним и сравним, чем эти два отличаются.

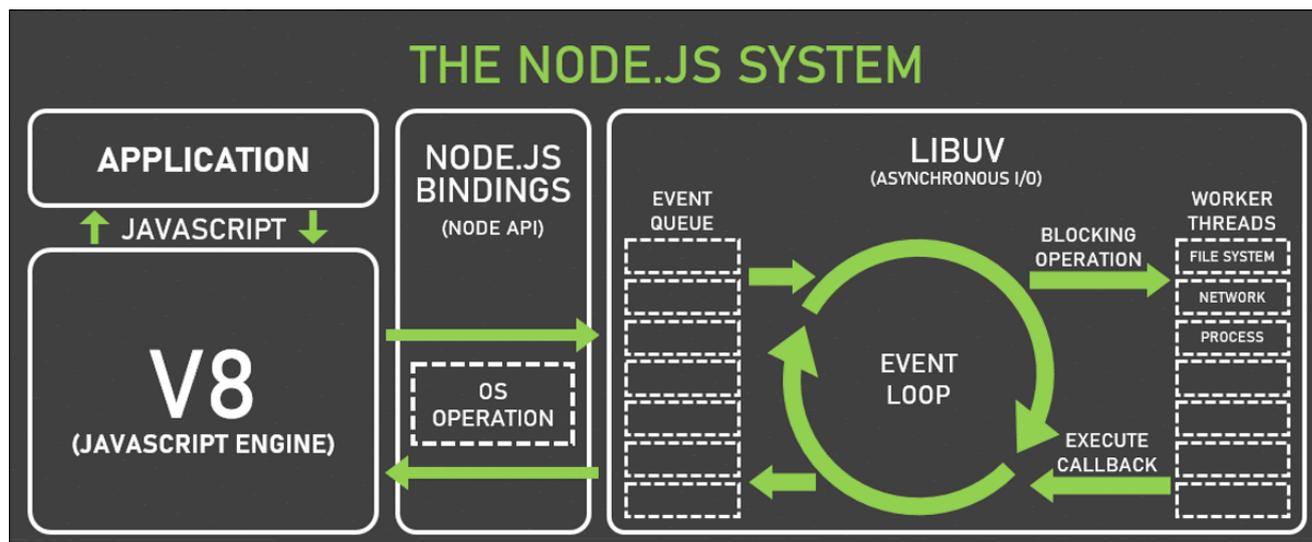
## Laravel



Рабочий процесс платформы Laravel. (Источник изображения: [How-To-Geek](#))

Laravel – это веб-платформа на основе PHP, на которую сильно влияет архитектура Model View Controller (MVC). Архитектура MVC отделяет бизнес-логику программы от ее представления и влияет на [производительность Laravel](#). Контроллер функционирует как мост, который управляет всеми запросами, связывается с моделью для получения запрошенных данных и передает данные в представления. Laravel был разработан, чтобы облегчить разработчикам PHP инициирование новых проектов. С Laravel вы меньше внимания уделяете настройке, архитектуре и зависимостям проекта, а больше – основной функциональности проекта.

## node.js



Рабочий процесс node.js. (Изображение предоставлено: [Appiness](#))

node.js придерживается однопоточной парадигмы проектирования цикла событий. Это значительно отличается от метода запроса / ответа с несколькими потоками. Однако один поток может одновременно управлять несколькими клиентами. Примечательно, что основанная на событиях модель JavaScript и механизм обратного вызова являются основными компонентами обработки node.js. node.js отправляет блокирующие и неблокирующие запросы в приложение. Затем цикл событий получает запросы один за другим. Он обслуживает и возвращает ответы на отдельные запросы, которые не требуют внешней авторизации от клиента. Если запрос сложный и требует контакта с клиентом, назначается один поток из пула потоков. Поток завершит запрос блока, включая обмен данными с базой данных.

## Расширяемость и масштабируемость

Масштабируемость в первую очередь относится к способности или способности системы обрабатывать растущий объем работы. Если вы используете систему, которая может взаимодействовать со значительным трафиком, вам следует выбрать технологию, которая может эффективно управлять им.

## Laravel

Laravel использует архитектуру MVC и несколько потоков для обработки запросов. Это становится несколько обманчивым, когда он получает несколько запросов. Поскольку Laravel требует дополнительных вычислительных усилий для

обслуживания запросов, поддержание масштабируемости может стать затруднительным. Чем больше рабочая нагрузка, тем больше логики и возможностей требуется. Таким образом, требуется большой объем памяти и функциональные возможности, которые замедляют работу композитора. С Laravel разработчики теперь могут создавать масштабируемые системы. Использование балансировщика HTTP может сделать систему на базе Laravel более масштабируемой за счет балансировки нагрузки HTTP-запросов.

## node.js

Однопоточная архитектура цикла событий Node делает его очень масштабируемым. Таким образом, ему уделяется значительное внимание как инструменту разработки. Более того, асинхронный характер работы упрощает плавную обработку больших нагрузок. Node может предоставить наиболее масштабируемое решение, необходимое для управления микросервисом в режиме реального времени. Кроме того, он обеспечивает большую [масштабируемость, чем Python](#).

## Кривая обучения

При обучении программированию каждый человек воспринимает это по-разному. Изучение Laravel и Node требует знакомства с PHP и JavaScript соответственно, поэтому кривая обучения может быть сложной для начинающих программистов.

## Laravel

По слухам, у Laravel крутая кривая обучения. На протяжении всего [процесса обучения Laravel](#) вы должны иметь практические знания PHP и опыт программирования с HTML и CSS. Тем не менее, шаблоны Laravel и предварительно загруженные ресурсы ускорят разработку веб-приложений. Он также управляет некоторыми важными функциями разработки для вас. Если вы хотите работать с Laravel, вам необходимо ознакомиться со структурой кода Laravel, архитектурой MVC, библиотеками Laravel и интеграцией с базами данных, среди прочего. Однако, с такими инструментами, как Laracasts и обширной документацией, освоение фреймворка и [языка программирования PHP](#) не должно быть сложным.

## node.js

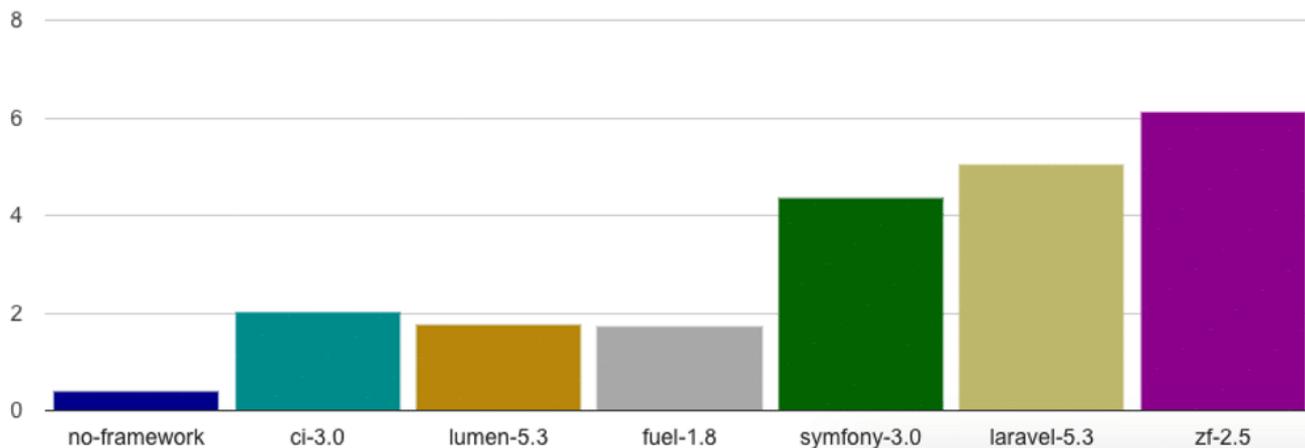
Node легко понять, если вы знакомы с JavaScript. Вам также необходимо понимать архитектуру Node, которая построена на модулях пакетов JavaScript и npm. Вы можете изучить архитектуру и установку модулей, начав с демонстрационного проекта node.js. Если вы хорошо разбираетесь в JavaScript, вы можете сразу же приступить к разработке с помощью Node.

## Производительность (скорость)

Laravel известен по множеству характеристик, но не по скорости. С другой стороны, Node невероятно быстр. Однако, поскольку Node является однопоточным, выполнение сложных численных вычислений с помощью Node должно занимать время. Node значительно быстрее, чем Laravel, но Laravel более надежен. Представьте себе Laravel как двухэтажный автобус, в то время как Node больше похож на мотоцикл. Используйте Laravel, если вы планируете выполнять многочисленные вычислительные процессы; в противном случае используйте Node.

## Laravel

Laravel имеет жесткую архитектуру, которая эффективно обрабатывает как простые, так и сложные приложения, основанные на вычислениях. Помимо прочего, он демонстрирует впечатляющую эффективность при взаимодействии с базой данных и обеспечении безопасности. Laravel предоставляет множество библиотек, которые были проверены разработчиками Laravel, обеспечивая эффективность и надежность ваших проектов

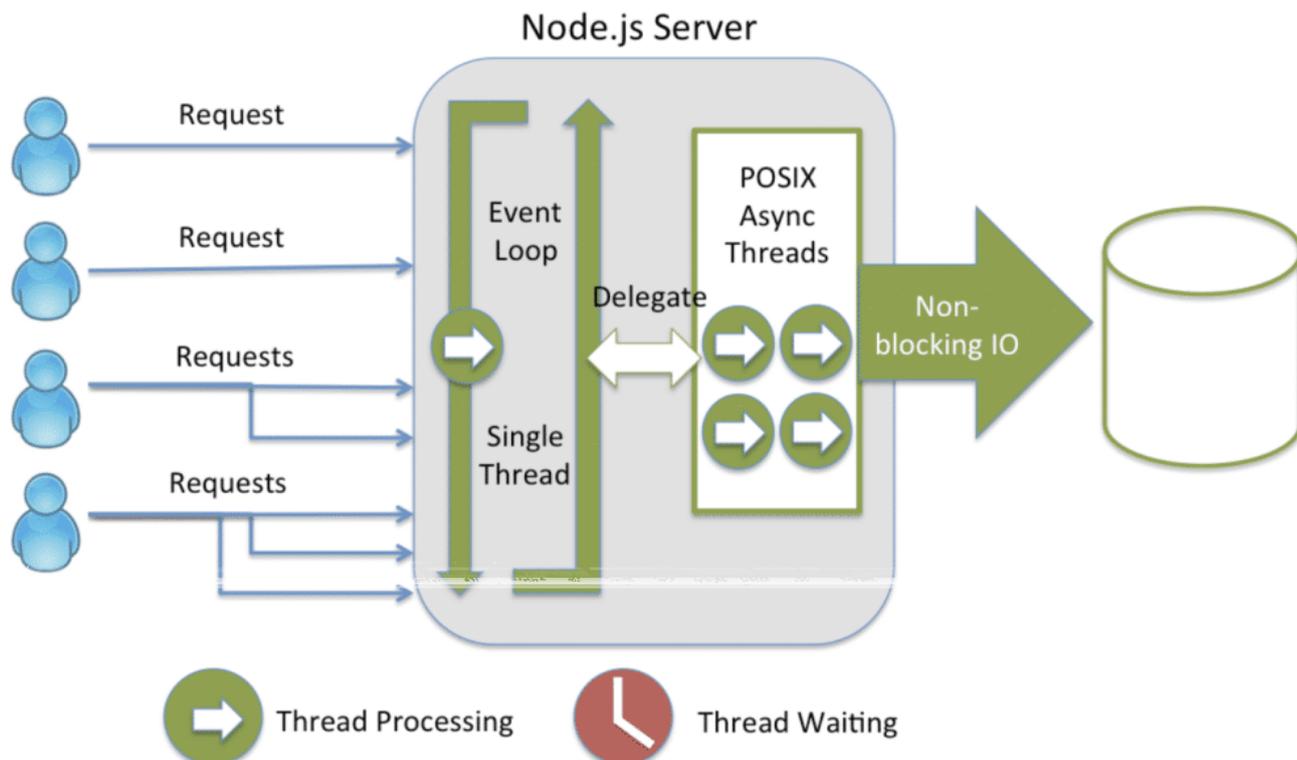


Время выполнения Laravel. (Источник изображения: [Medium](#))

Если ваш веб-сайт требует аутентификации, вы можете выбрать один из нескольких пакетов аутентификации, таких как Laravel-Auth или Breeze, для автоматического включения функции безопасности. Laravel также включает в себя готовый код для настройки. В результате вы можете сократить количество часов, затрачиваемых на разработку, и вы выиграете от улучшения качества и экономии времени.

## node.js

Однопоточная архитектура цикла событий в Node обеспечивает значительное повышение производительности. Запросы не должны ждать внешних запросов, поэтому время, необходимое для обработки указанных запросов, значительно сокращается. Это может значительно повысить производительность приложений реального времени и одностраничных приложений (SPA).



Обработка запросов node.js. (Источник изображения: [Medium](#))

Модули npm предоставляют шаблоны кода практически для каждого сценария. Однако растущее число библиотек может затруднить поиск нужной. Более того, поскольку Node позволяет создавать и повторно использовать фрагменты кода от разных пользователей, производительность каждого модуля может отличаться. Более того, Node может быть менее надежным при управлении высокопроизводительными предприятиями со сложными вычислениями и интеграцией баз данных. Тем не менее, он превосходит в управлении масштабируемостью приложений.

## Настройка

Поскольку оба имеют открытый исходный код, оба предоставляют возможности настройки. Однако, поскольку они работают по-разному, параметры различаются.

## Laravel

Будучи открытым исходным кодом, Laravel предоставляет вам работающий код и позволяет экспериментировать сколько угодно. Если вы хорошо разбираетесь в PHP, вы даже можете изменить эти коды в соответствии со спецификациями

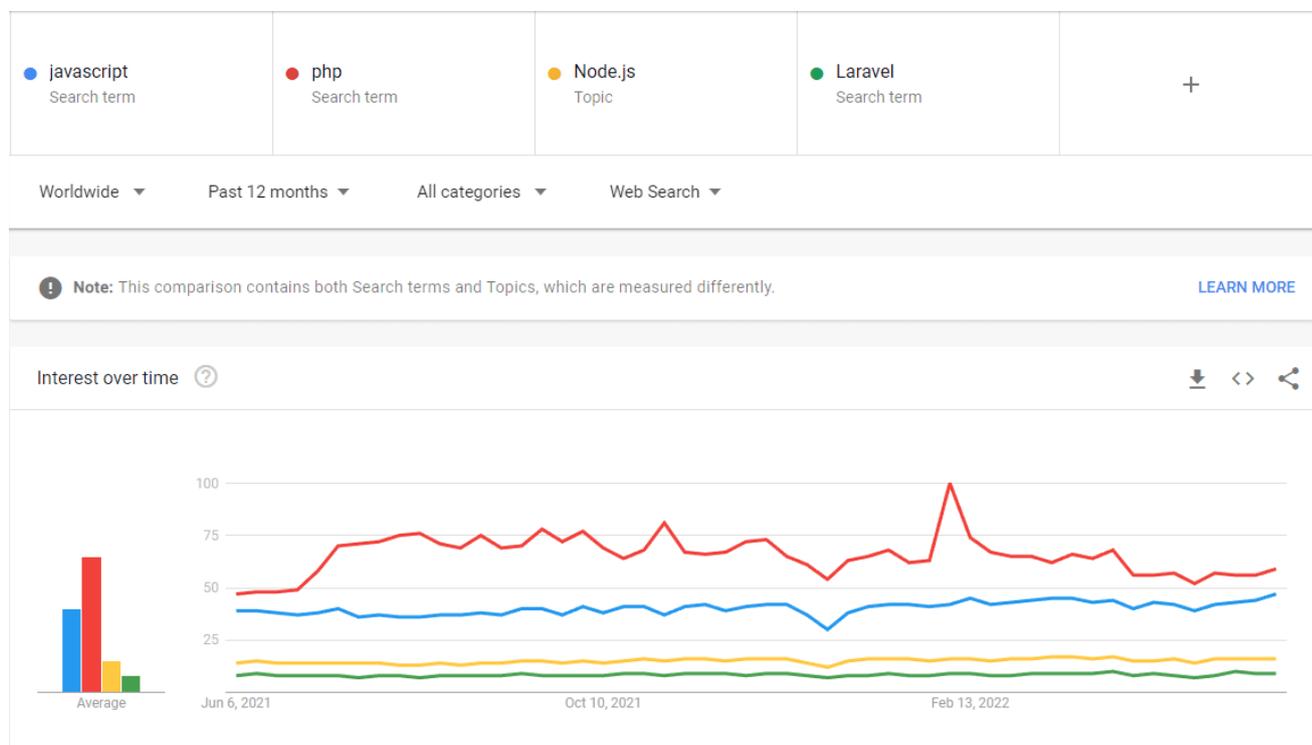
вашего приложения. Кроме того, Laravel-Auth и Laravel-Breeze предоставляют вашему приложению систему аутентификации внешнего интерфейса. Вы можете изменить функциональность и внешний вид как интерфейсных, так и внутренних кодов аутентификации, используя библиотеки пакетов Laravel.

## node.js

Как и в Laravel, среда выполнения Node имеет открытый исходный код. Вы можете адаптировать модули пакета и управлять версиями зависимостей в соответствии с вашей спецификацией разработки. Node также предоставляет шаблон, который вы можете преобразовать в полнофункциональное приложение и поделиться им с другими, если захотите.

## Популярность и рынок труда

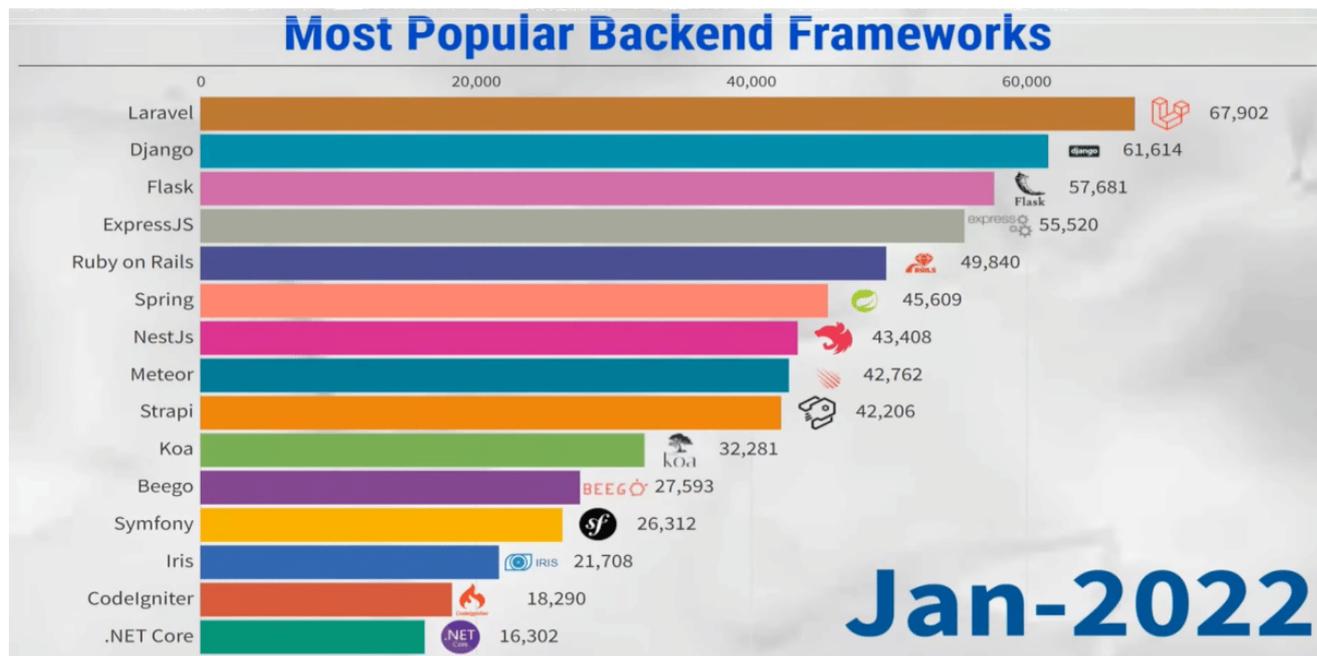
Популярность Laravel обусловлена распространённостью PHP, в то время как популярность Node является результатом популярности JavaScript. Просмотрев приведенный ниже отчет Google Trends, мы можем увидеть, как часто пользователи искали эти веб-технологии.



Популярность Laravel и Node в Google.

## Laravel

Laravel – самый популярный фреймворк PHP из-за его чистого синтаксиса, легкой кривой обучения, надежной безопасности и высокой производительности. Согласно опросу, проведенному JetBrains в 2020 году, 50% разработчиков PHP регулярно используют Laravel, а рейтинг Github показывает, что Laravel является самым популярным фреймворком PHP со значительным отрывом.



Самые популярные серверные фреймворки. (Источник изображения: [Статистика и данные](#))

Спрос на [разработчиков Laravel](#) огромен, что означает, что разработчики Laravel имеют право требовать адекватной компенсации. Средняя [зарплата разработчиков Laravel](#) в Соединенных Штатах составляет примерно 92 000 долларов в год, или 44 доллара в час. Поскольку большая часть интернета построена на PHP, Laravel и аппетит к его разработчикам вряд ли исчезнут в ближайшее время.

## node.js

JavaScript был самым популярным языком программирования за последнее десятилетие, и 68% профессиональных разработчиков так или иначе использовали его. И Node позволяет разработчикам JavaScript управлять как интерфейсом, так и бэкэндом, что оправдывает его экспоненциальный рост популярности. Согласно [StackOverflow](#), 36% профессиональных разработчиков в настоящее время

используют Node в качестве основного языка или фреймворка. У Node также 87 000 звезд на [Github](#).



Самые популярные технологии программирования 2021 (источник изображения: [Stack Overflow](#))

Node – это футуристический язык программирования, поэтому маловероятно, что спрос на рабочие места в Node снизится в обозримом будущем. Благодаря этому разработчики node.js получают щедрую компенсацию. Зарплата разработчика удаленного [node.js](#) составляет примерно 114 000 долларов в год, или 55 долларов в час. Если вы заинтересованы в карьере в Node, сейчас самое время начать.

## Безопасность

Что касается безопасности, Laravel стоит выше, чем Node, поскольку Laravel защищает от распространенных уязвимостей. Если вы не используете Laravel с необработанными запросами, он очищает все данные. Тем не менее, ядро Node.js является безопасным, но сторонние пакеты часто требуют дополнительной безопасности.

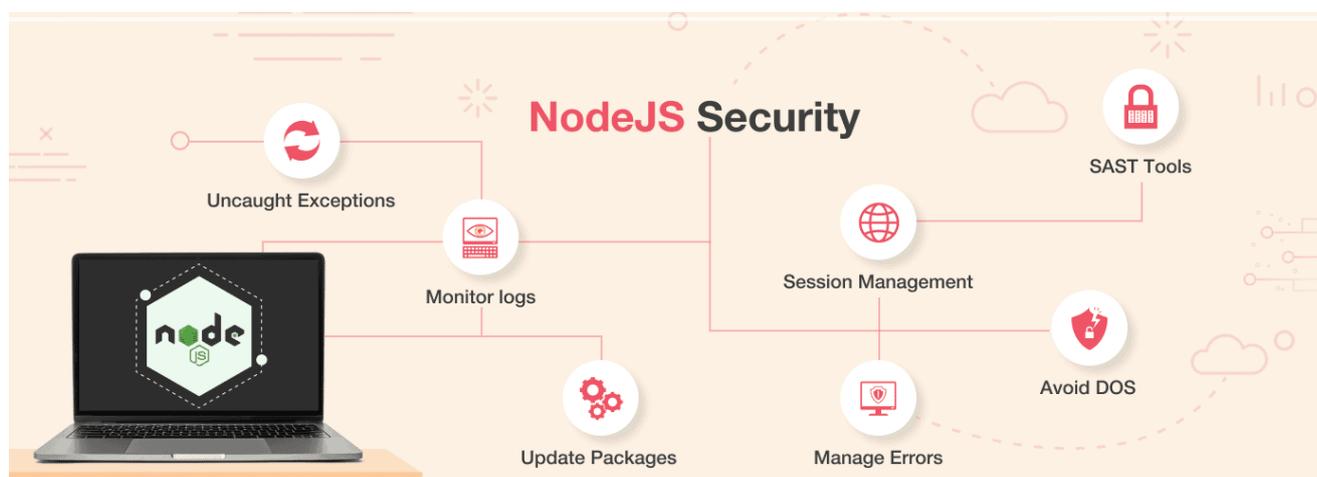
## Laravel



Функции безопасности Laravel (источник изображения: [Webnexus](#))

Функции безопасности Laravel являются наиболее важным аспектом. Он использует безопасный алгоритм хеширования bcrypt для шифрования наших паролей и токена CSRF для защиты данных вашей формы. Laravel защищает от SQL-инъекций с помощью своего красноречивого ORM. Класс объектов данных PDO или PHP является основной причиной защиты Eloquent ORM. Laravel также позволяет запрашивать HTTPS для передачи конфиденциальных данных.

## node.js



Потенциальные проблемы с безопасностью node.js. (Источник)

изображения: [Simform](#))

Node также обеспечивает высокий уровень безопасности при использовании своих лицензионных компонентов. Однако это не учитывает проблемы безопасности и лицензирования программного обеспечения с открытым исходным кодом.



Атака “машина в середине node.js” (MIDM). (Источник изображения: [Medium](#))

Node до отказа набит сторонними модулями, и многие из них имеют недостатки в безопасности. Однако в Node реализован токен защиты от CSRF, который проверяет аутентификацию, и пользователи также могут использовать модули сеанса cookie. Кроме того, автоматическое [сканирование уязвимостей](#) может помочь в выявлении типичных уязвимостей безопасности node.js.

## Экосистема (поддержка и сообщество)

### Laravel

Laravel – одна из самых популярных платформ веб-разработки во всем мире, и разработчики по всему миру создают на ней значительное количество приложений. В результате Laravel имеет обширное сообщество пользователей. Большинство разработчиков помогают друг другу, делятся своими пакетами и наставляют новых разработчиков. Laravel.io или Laracast являются примерами значительных сообществ технической поддержки фреймворка. Кроме того, в Laravel используется PHP, а это означает, что любой, кто сталкивается с трудностями с PHP-кодом, может обратиться за помощью к сообществу Laravel. Кроме того, такие технологические гиганты, как [Facebook](#), Github, [LinkedIn](#), другие социальные сети и онлайн-форумы пользователей, содержат многочисленные группы поддержки пользователей Laravel.

### node.js

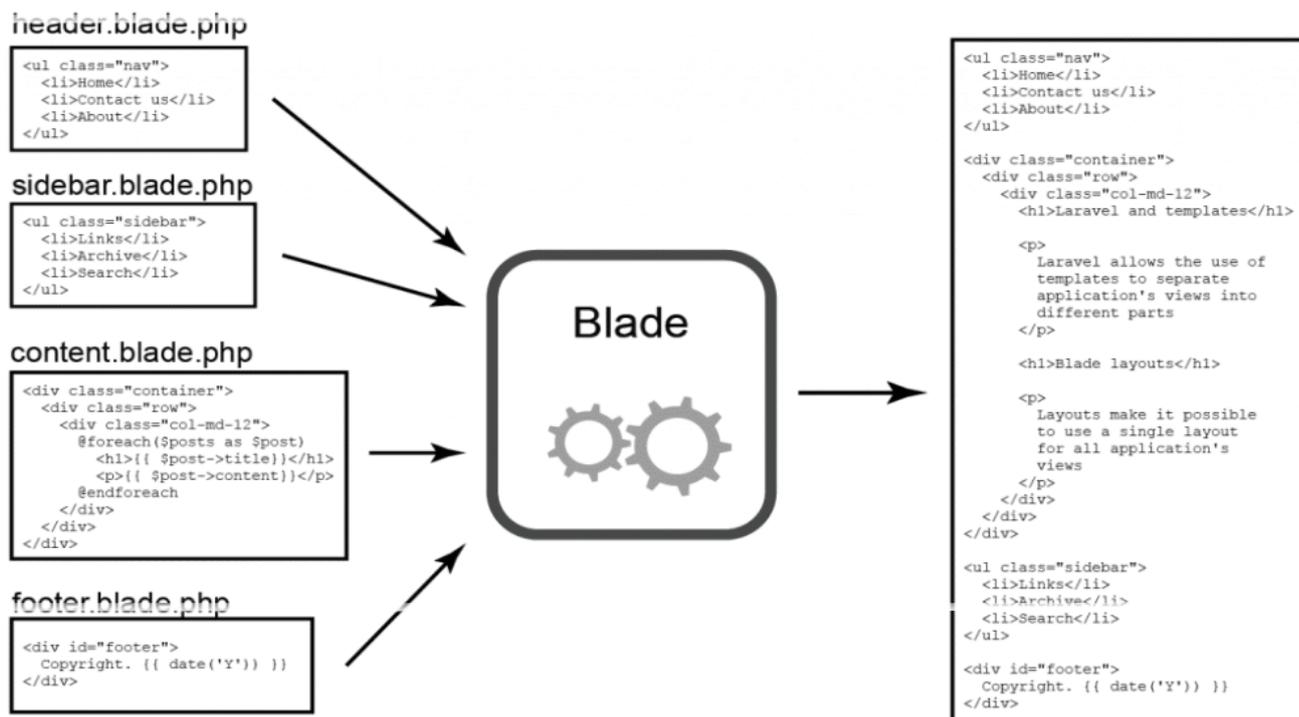
Node может похвастаться большим количеством пользовательских форумов для технической помощи и консультаций. На официальном веб-сайте Node есть собственное сообщество пользователей, а веб-сайт Nodejs.dev предлагает дополнительное сообщество поддержки пользователей Node. Пользователи могут найти специальные места для обсуждения проблем и решений, связанных с узлом, на [Facebook](#), Github, StackOverflow, LinkedIn и других социальных сетях. По мере того, как с помощью Node разрабатывается все больше веб-сайтов и приложений, поддержка платформы сообществом продолжает расти в геометрической прогрессии.

## Синтаксисы

Laravel использует синтаксис блейда, в то время как Node использует синтаксис JavaScript. Давайте обсудим каждый из них более подробно.

### Лезвие Laravel

Laravel Blade – это надежный механизм создания шаблонов, который позволяет пользователям Laravel быстро разрабатывать синтаксис и без труда использовать механизм шаблонов. Он обеспечивает существенную структуру, включая условные выражения и циклы. Вы можете создать шаблон блейда, создав файл представления и сохранив его с помощью **.blade.php** расширение вместо **.php**.



Синтаксис блейда Laravel. (Источник изображения: [Medium](#))

Давайте рассмотрим некоторые синтаксисы блейда Laravel:

- **Отображение данных:** чтобы вывести значение переменной, заключите его в фигурные скобки: `{{ $variable }}`;
- **оператор: троичный** Синтаксис троичного оператора в шаблоне блейда выглядит следующим образом: `{{ $variable or 'default value' }}`
- **Блейд-циклы:** механизм создания шаблонов блейда предлагает директивы цикла, включая `@for`, `@endfor`, `@foreach`, `@endforeach`, `@while`, и `@endwhile`:  
`@for ($i = 0; $i < 10; $i++) The current value is {{ $i }} @endfor`  
`@foreach ($users as $user) <p>This is user {{ $user->id }}</p>`  
`@endforeach @forelse ($users as $user) <li>{{ $user->name }}</li>`  
`@empty <p>No users</p> @endforelse @while (true) <p>I'm looping forever.</p> @endwhile`
- **Операторы If:** используя директивы `@if`, `@elseif`, `@else`, и `@endif`, вы можете создавать операторы if. Эти директивы имеют ту же функциональность, что и их эквиваленты на PHP: `@if (count($records) === 1) I have one record!`  
`@elseif (count($records) > 1) I have multiple records! @else I don't have any records! @endif`
- **Необработанный PHP:** иногда полезно добавить PHP-код в свои представления. Используя директиву `Blade@php`, вы можете выполнить простой PHP-блок в своем шаблоне: `@php $counter = 1; @endphp`
- **Комментарии:** Blade также позволяет добавлять [комментарии Laravel](#) к вашим

представлениям. Блейд-комментарии, в отличие от HTML-комментариев, не включаются в HTML-код, предоставляемый приложением: `{--The resulting HTML will not contain this comment. --}}`

- **Поле CSRF:** когда вы определяете HTML-форму в своем приложении, включите скрытое поле токена CSRF, чтобы промежуточное программное обеспечение могло проверить запрос. Вы можете сгенерировать поле токена, используя директиву @csrfблейда: `<form method="POST" action="/profile"> @csrf ... </form>`
- **Стеки:** блейд позволяет переходить к именованным стекам, которые вы можете отображать в другом месте в другом представлении или макете. Это полезно для указания библиотек JavaScript, необходимых для представлений вашего дочернего элемента: `@push('scripts') <script src="/example.js"></script> @endpush`

## node.js

Node и JavaScript используют один и тот же синтаксис. Однако между API-интерфейсами есть различия. node.js не поддерживает обычный DOM браузера, хотя дополнительные API доступны на [nodejs.org](https://nodejs.org). Однако причуды браузера объясняют различия в синтаксисе. Давайте рассмотрим некоторые синтаксисы Node:

- **Примитивные типы:** node.js поддерживает следующие примитивные типы данных:
- **Строка**
- **Количество**
- **Не определено**
- **Логическое**
- **Null**
- **Регулярное выражение**
- **Свободный ввод текста:** JavaScript в Node допускает свободный ввод текста, аналогичный браузерному JavaScript. Вы можете объявить любую переменную с ключевым словом `var`.
- **Объектный литерал:** синтаксис объектного литерала идентичен JavaScript браузера: `var obj = { authorName: 'Zadhid Powell', language: 'Node' }`
- **Функции:** node.js рассматривает функции как первоклассные граждане. Кроме того, функция может также обладать признаками и свойствами. Его также можно рассматривать как класс JavaScript: `function Display(x) { console.log(x); } Display(100);`
- **Отладчик node.js:** node.js имеет простой протокол TCP и [отладочный клиент](#)

. Вы можете использовать `debug`, за которым следует имя файла `.js`, для отладки JavaScript: `node debug [script.js | -e "script" | <host>:<port>]`

- **Файловая система node.js (FS):** node.js предоставляет файлы ввода-вывода с помощью обычных методов POSIX, завернутых в простые оболочки. Вы можете импортировать [файловую систему node.js](#), используя следующий синтаксис: 

```
var fs = require("fs")
```
- **События node.js:** после завершения задачи поток node.js выпускает связанное событие, которое выполняет функцию прослушателя событий: 

```
// Import events module var events = require('events'); // Create an eventEmitter object var eventEmitter = new events.EventEmitter();
```
- **Node.js V8:** Тот Node.js Модуль V8 содержит специфичные для V8 интерфейсы и события. `v8.getHeapStatistics()` С помощью функций `v8.getHeapSpaceStatistics()` и он предлагает доступ к статистике кучи памяти: 

```
const v8 = require('v8');
```
- **Node Punycode:** Punycode преобразует строки Unicode (UTF-8) в строки ASCII. Используется Punycode, поскольку имена хостов понимают только ASCII. Punycode.js поставляется с Node.js версии 0.6.2 и более поздние: 

```
punycode = require('punycode');
```

## Laravel vs Node: сравнительная таблица

Давайте посмотрим на параллельное сравнение Laravel и Node и рассмотрим некоторые основные параметры программирования:

	Laravel	node.js
Создано	2011	2009
Категория	Языковая (PHP) платформа	Среда выполнения JavaScript
Простота кодирования	Краткое	Долго
Популярность	0,35% веб-сайтов	4,24% веб-сайтов
Движок	Движок шаблонов блейда	JavaScript версии 8 от Google
Менеджер пакетов	Менеджер пакетов Composer	Диспетчер пакетов node.js (npm)
Выполнение	Синхронный	Асинхронный
Скорость выполнения	Мощный и легкий	Быстрее и легче

	Laravel	node.js
Параллелизм	Многопоточный блокирующий ввод- вывод	Управляемый событиями неблокирующий ввод- вывод
Производительность	Медленнее	Быстрее
Веб-сервер	Не требует	Apache и IIS
База данных	4 (MySQL, PostgreSQL, SQLite, SQL)	Реляционные и обычные
JSON	json_encode	JSON.stringify() и json.parse
Последняя версия	<a href="#">Laravel 9</a>	<a href="#">node.js 18.3.0</a>
Сообщество	Небольшой, но растущий; разделяет сообщество PHP	Обширное онлайн- сообщество

## Laravel vs Node: что выбрать?

Теперь, когда мы подробно изучили эти два чуда разработки, вы должны лучше их понять. Эти две платформы предлагают множество возможностей, которые сделают вашу разработку более доступной, стабильной, безопасной и масштабируемой. Наш анализ показывает, что у обоих есть преимущества и недостатки. С этой целью заявление о превосходстве, основанное на объективных доказательствах, было бы бесполезным. В конце концов, вы должны выбрать технологию, которая соответствует набору функций, требуемому вашим проектом..

### Когда использовать Laravel

Вы должны использовать Laravel, если ваш проект требует следующего:

- **Быстрая разработка:** благодаря готовым шаблонам и поддержке сообщества Laravel является одним из самых быстрых фреймворков для разработки приложений.
- **Полномасштабная система:** Laravel предлагает надежную систему для веб-сайтов на основе [CMS](#) любого размера. Используя PHP в качестве основы, Laravel может эффективно управлять большими сайтами.
- **rade: - g индустрия** Laravel процветает в профессиональной,

---

образовательной, торговой, коммерческой и других основных отраслях, за исключением индустрии искусства и развлечений, в которой доминирует Node.

- **Высокая степень безопасности:** Laravel запрещает хранение паролей открытого текста в базе данных, поскольку для этого требуются хэшированные и зашифрованные пароли. Кроме того, он содержит интегрированную систему безопасности и авторизации и алгоритм хеширования bcrypt для шифрования паролей.

## Когда использовать Node

Вы должны использовать Node для проектов по разработке, которые требуют следующего:

- **Полный стек:** выберите node.js для одновременной разработки на стороне сервера и на стороне клиента. node.js устраняет головную боль, связанную с координацией отдельных [интерфейсных и серверных](#) фреймворков или команд.
- **Скорость и производительность:** Node запускает программы быстрее, чем большинство языков программирования, что приводит к значительному увеличению производительности и стабильности. Выберите Node, если вы хотите, чтобы ваше приложение обрабатывало множество команд и запросов одновременно.
- **Приложение для чата:** Node – лучший выбор для разработки приложения реального времени. Другой сопоставимой технологии не существует. Используйте Node, если вы создаете многопользовательское приложение реального времени.
- **Потоковая передача данных:** на обычных платформах HTTP-запросы являются независимыми событиями. node.js предоставляет аналогичные возможности и позволяет осуществлять потоковую передачу (обработку файлов во время загрузки). Создавайте приложения для потоковой передачи аудио / видео с помощью Node.
- **Мониторинг** панели мониторинга: [Node – отличный инструмент для разработки панелей мониторинга, особенно панелей мониторинга приложений и систем](#). Функциональность node.js в режиме реального времени, двусторонние сокеты и возможности цикла событий помогают в разработке надежных возможностей мониторинга.

## **Краткие сведения**

И Laravel, и Node уникальны по своим функциям и назначению, и нет никаких сомнений в том, что в последние годы они пользуются огромной популярностью у разработчиков. Однако окончательное решение зависит больше от ваших пользовательских требований, чем от возможностей, которые может предоставить Laravel vs Node. Когда дело доходит до вашей разработки, одна технология может включать в себя все необходимые вам вещи, а другая – нет. В любом случае, вы можете рассчитывать на то, что и Laravel, и Node останутся жизнеспособными вариантами разработки в обозримом будущем.

### **Дата Создания**

08.02.2023