

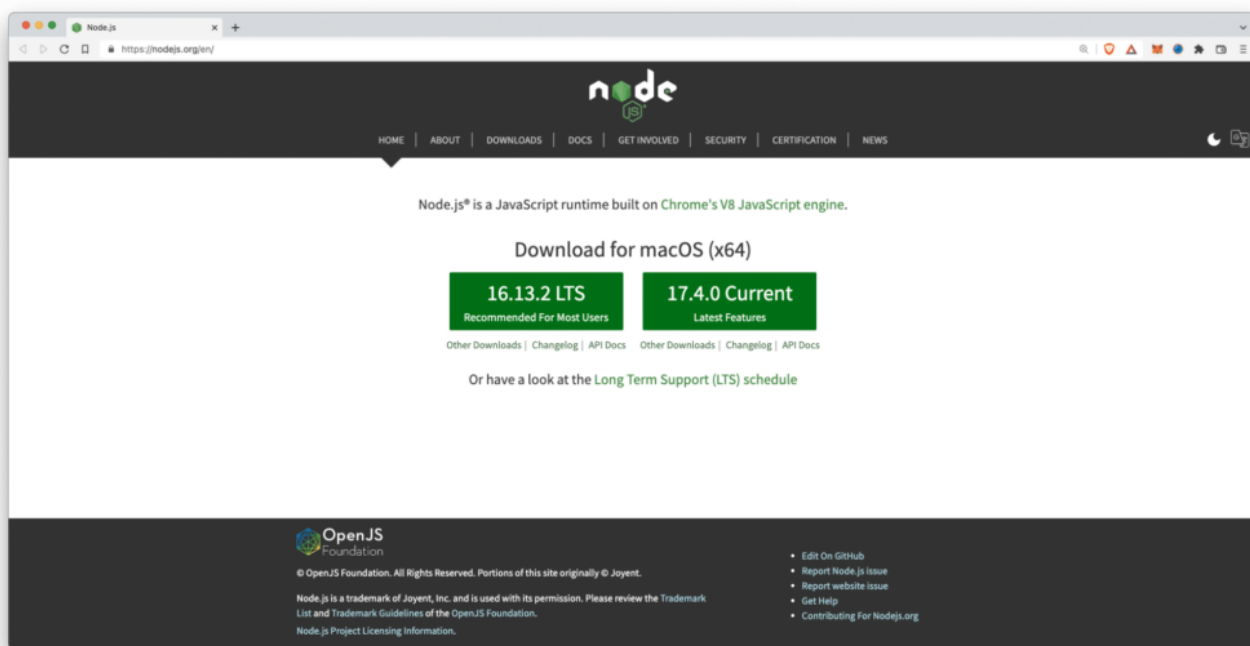
Ruby on Rails или Node.js : Сравнение лицом к лицу

06.02.2023

Выбор фреймворка или языка для вашего проекта может быть сложной задачей, особенно на серверной части, где ключевыми являются ведение пользовательских данных и [создание API](#) и библиотек. На момент написания статьи наиболее популярными технологиями веб-разработки являются Node.js и Ruby on Rails. Выбор одного из них может быть сложной задачей. Обе эти технологии имеют преимущества и недостатки, и понимание их поможет вам выбрать наилучший вариант для вашего проекта. Мы предоставим вам объективное сравнение Node.js против Rails в этой статье, чтобы помочь вам принять решение. **Почему мы сравниваем эти две технологии?** Хотя они могут показаться не совсем аналогичными, мы заметили множество пользователей, которые задавали вопрос о Ruby on Rails vs Node.js онлайн. Эта статья поможет ответить на этот вопрос, рассмотрев ключевые уточняющие функции, основные различия и варианты использования для каждой технологии.

Что такое Node.js ?

Node.js это среда выполнения с открытым исходным кодом, созданная в 2009 году на движке [JavaScript](#) Chrome V8. Это однопоточный и кроссплатформенный, который широко используется в [серверной разработке](#).



Node.js домашняя страница сайта.

[Node.js](https://nodejs.org/en/) может быть [установлен](#) на различных платформах, таких как Linux, macOS и Windows. Он используется для создания множества [приложений](#), включая приложения для чата в реальном времени, серверы REST API, приложения командной строки и многое другое.

Как узел обрабатывает библиотеки

Node.js имеет встроенную поддержку управления библиотеками, называемую Node Package Manager (npm). Имея более 1,3 миллиона пакетов и более миллиарда загрузок в неделю, это одна из самых популярных библиотек пакетов и, как таковая, играет жизненно важную роль в [экосистеме JavaScript](#). Эта огромная библиотека полностью бесплатна и с открытым исходным кодом. Эти библиотеки быстро растут в размерах, что делает Node.js сообщество крепнет с каждым днем. Одной из самых популярных библиотек в npm является [Express.js](#), платформа для быстрой разработки и развертывания приложений и API. Это минимальный Node.js платформа веб-приложений, которая предлагает широкий спектр возможностей как для онлайн, так и для мобильных приложений, и еженедельно получает более 22 000 000 загрузок через npm.

Вот код для простого Node.js и экспресс-API:

```
const express = require('express')
const app = express()

app.get('/', function (req, res) {
  res.send("A simple GET response")
})

app.listen(3000)
```

В приведенном выше коде мы импортируем `express.js` и создание простого GET API. В последней строке мы передаем 3000 для функции прослушивания, таким образом, наше приложение будет работать на порту 3000. Express.js упрощает создание различных типов веб-приложений за короткий промежуток времени. Для клиентских запросов платформа предоставляет простую систему маршрутизации. Он также включает промежуточное программное обеспечение, которое отвечает за принятие решений, предлагая правильные ответы на запросы клиента.

Компании, которые используют Node.js

Некоторые из популярных компаний, которые используют Node.js являются:

- **Netflix:** Netflix начал использовать Node.js чтобы обеспечить массовое потоковое вещание в Интернете для более чем 182 миллионов пользователей, они также намеревались расширить использование Node.js чтобы включить производство контента.
- **Uber:** Из-за своих асинхронных примитивов и простой однопоточной обработки основной механизм выполнения поездок Uber изначально был разработан в Node.js .
- **PayPal:** При замене Java PayPal решил использовать JavaScript из браузера вплоть до внутреннего сервера для веб-приложений.
- **НАСА:** После [неприятного происшествия в космосе](#) НАСА

решило использовать Node.js чтобы лучше защитить свои данные и даже предотвратить гибель людей.

- **LinkedIn:** Из-за его масштабируемости и эффективности LinkedIn выбрал Node.js и перестал использовать Ruby on Rails.



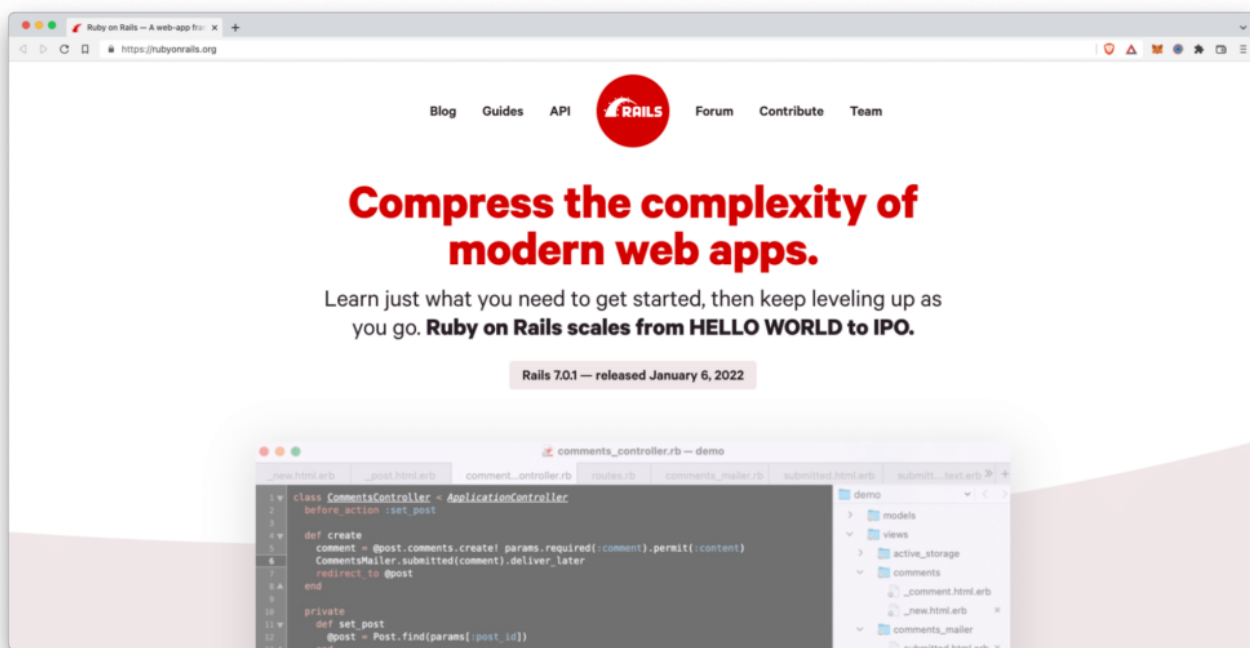
Компании, которые используют Node.js . (Источник изображения: [Coruscate](#))

Node.js особенно хорошо подходит для систем записи, которые хранят все свои состояния в памяти.

– Крис Коваль, инженер-программист Uber

Что такое Ruby on Rails?

Ruby on Rails, также известный как Rails или RoR, является самой популярной платформой веб-приложений с открытым исходным кодом, написанной на Ruby под лицензией MIT. Разработанный в 2004 году, он известен своей эффективностью кода и высокой скоростью.



Домашняя страница ROR.

Rails был предназначен для помощи в управлении рутинной, внутренней логикой, вычислением данных и другими задачами. Это платформа веб-приложений, которая хорошо подходит для веб-приложений MVC с поддержкой баз данных и метапрограммирования. Первоначальная ересь Ruby действительно заключалась в том, чтобы возвести счастье программиста на пьедестал. ~ Дэвид Хайнмайр Хэнсон, создатель Ruby on Rails

Большинство веб-серверов, поддерживающих CGI, могут запускать Rails. [MySQL](#), [PostgreSQL](#), SQLite, [SQL Server](#), DB2 и Oracle поддерживаются этой платформой. Более того, Ruby on Rails имеет четкий, краткий и гибкий синтаксис, похожий на английский язык. Эта более низкая кривая обучения делает фреймворк особенно привлекательным для новичков.

Компании, использующие Ruby on Rails

Некоторые из популярных компаний, которые используют Ruby on Rails:

- **Airbnb:** Airbnb использует Ruby on Rails с момента своего создания. Фреймворк играет жизненно важную роль в технологическом стеке Airbnb.

- **GitHub:** серверная часть GitHub – Ruby on Rails, с более чем 200 миллионами репозиториями кода и 32 миллионами пользователей в месяц. Прошло семь лет с тех пор, как приложение было создано основателями компании.
- **Shopify** – отличный пример сложной, но хорошо структурированной и удобной платформы электронной коммерции Ruby on Rails, услугами которой пользуются более 820 000 продавцов.
- **Fiverr:** Да, Fiverr также приняла Ruby on Rails при запуске, и это все еще технология, которая поддерживает их онлайн-сервис marketplace.

Теперь, когда вы хорошо разбираетесь в этих двух технологиях, давайте обсудим их плюсы и минусы.

Ruby on Rails: плюсы и минусы

Несмотря на то, что Ruby on Rails является многофункциональным и мощным фреймворком, у него есть некоторые недостатки.

Плюсы Ruby on Rails

Использование Ruby on Rails имеет множество преимуществ. Мы перечислили лучшие из них ниже:

- **Скорость разработки:** хорошо разработанная архитектура модулей RoR, эффективная система управления пакетами, а также выразительные и компактные характеристики языка Ruby позволяют разработчикам быстро создавать приложения.
- **Большая инфраструктура:** Ruby on Rails поставляется со встроенным веб-сервером и базой данных с генераторами, что упрощает разработку веб-сайта.
- **Большое сообщество:** Еще одна замечательная особенность Ruby on Rails – это сильное и активное сообщество. Это один из популярных фреймворков на [GitHub](#), и сообщество

Ruby, вероятно, уже реализовало все функции, о которых вы можете подумать.

- **Лучшая практика:** Rails был создан с целью создания лучших практик для веб-разработки, и он поставляется со всеми необходимыми библиотеками и модулями для следования этим стандартам в проектах разработки.
- **Качество кода:** качество стороннего кода Ruby значительно выше, чем на других языках.

Минусы Ruby on Rails

Некоторые из недостатков Ruby on Rails можно увидеть ниже:

- **Меньшая гибкость:** настройка Ruby on Rails для создания единственного в своем роде приложения с уникальными функциями может быть затруднена.
- **Скорость и производительность во время выполнения:** одним из наиболее распространенных аргументов против Ruby on Rails является низкая скорость выполнения, что затрудняет масштабирование ваших приложений. Это также стало причиной того, что Twitter перестал использовать Ruby on Rails для своей поисковой системы.
- **Скорость загрузки:** Большинство разработчиков жалуются на скорость загрузки Rails. Начало работы может занять много времени, в зависимости от количества ваших зависимостей и файлов gem.
- **Сложная отладка:** может быть сложно устранить неполадки в приложении Rails из-за сложности Ruby on Rails и различных уровней. Поиск ошибки в миксе может занять много времени.

Плюсы и минусы Node.js

Node.js известно, что это гибкая структура, которая может быть сформирована в соответствии с потребностями практически любого

человека. Однако у него тоже есть свои недостатки.

Плюсы Node.js

Мы перечислили наши лучшие варианты Node.js преимущества:

- **Легко учиться:** потому что Node.js [основан на JavaScript](#), его может быть намного [легче освоить](#), если у вас уже есть хороший опыт программирования и вы знакомы с JavaScript. Доступно множество курсов и руководств, которые сделают обучение приятным.
- **Сообщество:** Заинтересованное сообщество означает большую поддержку и обратную связь. Node.js находится в окружении большого сообщества разработчиков. npm, менеджер пакетов узла, является одним из самых популярных и быстро расширяющихся реестров программного обеспечения. В нем представлены многочисленные библиотеки и повторно используемые шаблоны, которые вы можете использовать в своем проекте.
- **Бесшовная поддержка JSON:** в то время как другие серверные технологии, такие как Ruby on Rails, могут взаимодействовать с использованием формата JSON, Node.js делает это без преобразования между двоичными моделями и вместо этого использует JavaScript. Это чрезвычайно полезно при создании RESTful API с базой данных NoSQL, [такой как MongoDB](#).
- **Высокая расширяемость:** Node.js он известен своей высокой расширяемостью, что означает, что разработчики могут настраивать и расширять его в соответствии с потребностями своих проектов.

Минусы Node.js

Вот некоторые из Node.js недостатки, которые могут сделать его менее подходящим выбором для вашего проекта:

- **Нестабильный API:** Node.js страдает от несовместимых изменений API на регулярной основе. Эти несовместимости приводят к значительным изменениям кода, что является основным недостатком использования Node.js .
- **Незрелость инструментария:** хотя ядро Node.js модули достаточно надежны, многие пакеты в репозитории прм низкого качества и плохо документированы. В результате может быть сложно найти подходящий пакет для ваших нужд.
- **Не подходит для задач с интенсивным использованием процессора:** невозможность Node.js обработка операций, связанных с процессором, является другим серьезным недостатком, который у него есть прямо сейчас. Это только для задач, связанных с вводом-выводом (например, веб-серверов).
- **Проблема с обратным вызовом:** обратные вызовы, функции, которые выполняются при завершении каждой задачи в очереди, являются большой частью Node.js . На качество кода напрямую влияет сохранение нескольких процессов в очереди в фоновом режиме, каждый из которых имеет свой собственный обратный вызов.

Node.js vs Ruby on Rails: сравнение лицом к лицу

Теперь, когда вы хорошо разбираетесь в этих двух технологиях, давайте углубимся и сравним их параллельно.

Производительность

Хотя производительность может и не быть фактором для небольших проектов, ее очень важно учитывать при создании больших и сложных проектов. Node.js выигрывает, когда дело доходит до производительности. Node.js поставляется с двигателем V8, разработанным Google, и он намного быстрее, особенно при интенсивных операциях ввода-вывода. Он также известен созданием чрезвычайно быстрых и масштабируемых программ,

поскольку использует архитектуру, управляемую событиями, и неблокирующие (асинхронные) процессы, которые выполняются в одном потоке. Node.js приложения также могут обрабатывать больше запросов благодаря способности платформы управлять большими рабочими нагрузками сервера.

Сообщество

Когда дело доходит до поиска поддержки и вклада, обе эти технологии имеют обширное сообщество разработчиков. Согласно [опросу разработчиков, проведенному Stack Overflow](#), Node.js является одной из лучших технологий, используемых разработчиками по всему миру. Некоторые технологические гиганты, такие как Google, Facebook и Amazon, внесли значительный вклад в Node.js окружающая среда. Доступно несколько форумов сообщества для Node.js, такие как Github, [Stack Overflow](#) и [Reddit](#).

Аналогично, Ruby on Rails также имеет огромное и сильное сообщество с огромными репозиториями на [GitHub](#). У него более 4000 активных участников на GitHub и множество других активных форумов сообщества. Разработчики Rails активно занимаются улучшением существующих функциональных возможностей и разработкой новых функций. Они часто обновляют исходный код, исправляют ошибки и устраняют уязвимости в системе безопасности. В Rails также есть множество форумов сообщества, включая GitHub, Slack и Stack Overflow. Их [сообщество Slack](#) насчитывает более 17 000 участников и 27 каналов со всего мира, в том числе заядлых разработчиков OSS, [инженеров полного стека](#), основателей стартапов, бэкенд-инженеров и людей, только изучающих Ruby on Rails.

Вакансии

Поскольку обе эти технологии имеют большие сообщества пользователей, существует множество возможностей для трудоустройства Node.js как и разработчики Rails. [Средняя зарплата](#) Node.js разработчик в Соединенных Штатах стоит от 71

000 до 92 000 долларов, в то время как в Европе он составляет от 24 000 до 94 000 долларов. И Node.js фрилансеры могут зарабатывать в среднем около 80-100 долларов в час. [Согласно Indeed](#), разработчик Ruby on Rails может зарабатывать около 116 000 долларов в год в Соединенных Штатах; средняя зарплата фрилансера Rails составляет около 49 долларов в час.

Масштабируемость

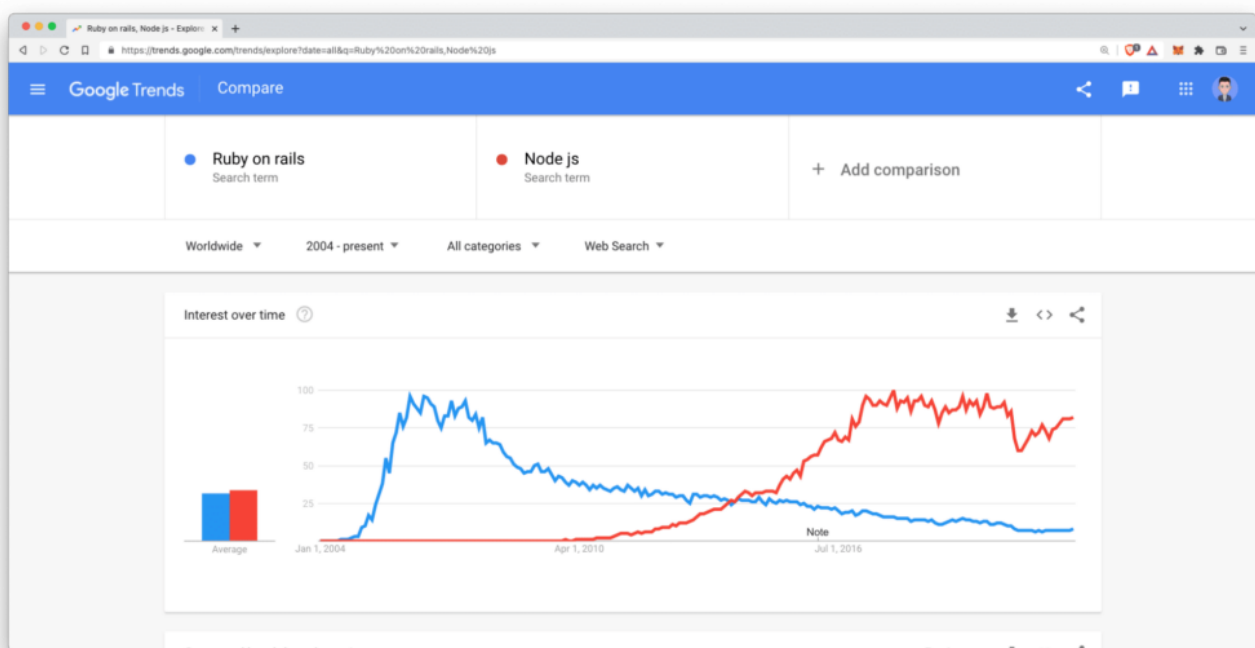
Node.js используется для создания приложений, которые очень масштабируемы. Несколько одновременных запросов обрабатываются с помощью неблокирующей парадигмы ввода-вывода и событийной парадигмы. Node.js в конечном счете он более масштабируем, чем Ruby, благодаря кластерному модулю. Процесс создается в кластерах с абстракциями, основанными на рабочей нагрузке программы с наименьшим количеством процессоров. Масштабирование Ruby on Rails достижимо, но для этого требуется больше ресурсов, чем для некоторых других популярных серверных платформ. Однако это не должно мешать вам использовать Ruby on Rails в вашем проекте.

Вот несколько советов по масштабированию вашего приложения Ruby on Rails:

- Встроенное в Rails кэширование действий, страниц и фрагментов может быть использовано по максимуму. Вы также можете использовать Memcache для кэширования результатов из вашей базы данных, которые в противном случае были бы получены.
- Вы также можете использовать сторонние инструменты, такие как Docker и Kubernetes, которые помогут вам плавно расширять ваше приложение.
- Сделайте нагрузочные тесты стандартным элементом вашей процедуры предварительного развертывания. Это позволит вам увидеть, сколько времени занимает каждая процедура и где есть возможности для улучшения.

Популярность

Популярность вашего стека – еще один фактор, который следует учитывать при создании современного продукта или приложения. В целом, чем популярнее он, тем лучше он будет поддерживаться. [Согласно Google Trends](#), Ruby on Rails был довольно популярен с 2005 по 2009 год, но с тех пор он постепенно начал терять популярность. Node.js, с другой стороны, набирает популярность с 2017 года, а это значит, что есть *много* людей, которые хотят использовать и узнать о нем больше.



График, сравнивающий популярность Node.js и Ruby on Rails в поисковой системе Google.

Если вы ищете самый популярный вариант на момент написания, Node.js это ваш победитель.

Node.js vs Ruby on Rails: что вы должны использовать?

Теперь, когда мы обсудили детали Ruby on Rails vs Node.js пришло время принимать решение! Давайте в последний раз посмотрим, какой фреймворк лучше всего подходит для каких проектов.

Когда использовать Node.js

Node.js это хороший выбор, если вы хотите создать приложение реального времени, которому необходимо управлять большим количеством одновременных запросов и данных между [клиентом и сервером](#), например, приложения для чатов или видеозвонков. Если вас беспокоит производительность или масштабируемость, вам следует сначала обратиться Node.js прежде чем перейти к Ruby on Rails.

Когда использовать Ruby on Rails

Ruby on Rails – отличное решение для приложений с интенсивным использованием процессора, где требуется быстрая разработка. Rails – особенно хороший выбор, если вы хотите создать платформу электронной коммерции или сайт социальной сети, способный обрабатывать большой трафик.

Краткие сведения

Оба Node.js а Ruby on Rails – это мощные фреймворки для веб-сайтов и приложений. Может быть очень сложно выбрать кого-то в качестве победителя. Разработчики и владельцы стартапов должны рассмотреть свои бизнес-модели и проанализировать плюсы и минусы обеих технологий. Если вас беспокоит производительность или масштабируемость, Node.js это может быть очень хорошим выбором. С другой стороны, если вы хотите создать приложение, способное обрабатывать большой объем трафика и задачи с интенсивным использованием СРІ, Ruby on Rails может быть лучшим фреймворком для вас. Планируете ли вы использовать Node.js или Ruby on Rails для вашего следующего проекта? Как вы выбирали между ними? Дайте нам знать в разделе комментариев ниже!