

13 Библиотек Node.js для питания ваших проектов

Описание

Node.js стал основной средой выполнения для многих разработчиков, работающих над веб-приложениями. Созданный для выполнения кода, написанного на JavaScript, одном из самых популярных языков программирования в мире, Node.js делает создание приложений на стороне сервера доступным для большого сообщества разработчиков.

Node.js поддерживает возможность повторного использования кода с помощью библиотек JavaScript, но выбор лучших библиотек может оказаться сложной задачей. Полезные библиотеки могут ускорить время разработки и имеют широкий спектр преимуществ для вашего веб-приложения, включая более быстрое время загрузки и уменьшение размера пакета приложения.

При выборе библиотеки следует учитывать сложность приложения, сообщество, стоящее за библиотекой, частоту обновлений и качество документации. Библиотеки в Node.js управляются с помощью менеджера пакетов Node.js, npm, который может помочь установить множество библиотек с открытым исходным кодом. Мы определили 13 основных библиотек для Node.js, которые облегчают жизнь веб-разработчика.

Что такое Node.js?

Node.js – это среда выполнения кода JavaScript с открытым исходным кодом на стороне сервера. Она имеет асинхронную архитектуру и кросс-платформенную

совместимость, что делает ее популярной основой для веб-разработки. Node.js использует событийно-ориентированный и неблокирующий ввод-вывод, что делает его высокоэффективным в распределенных приложениях реального времени, работающих с большим количеством данных.

Что такое библиотека Node.js?

Библиотека, или модуль, – это заранее написанный код, который абстрагирует часто требуемые задачи. Вы можете использовать библиотеки для ускорения процесса кодирования и повышения возможности повторного использования кода, помогая сохранить вашу работу “DRY” (не повторяться). В отличие от фреймворков, библиотеки содержат готовые функции, которые можно включить в проект на любом этапе разработки. С другой стороны, фреймворк обычно предоставляет скелет для всего приложения, часто оказывая значительное влияние на его построение.

Библиотеки Node.js, которые работают, чтобы вам не пришлось работать

Давайте рассмотрим 13 таких библиотек Node.js и изучим их преимущества.

Sequelize

Sequelize – это основанный на обещаниях Node.js инструмент объектно-реляционного маппера (ORM), который помогает разработчикам легче работать с реляционными базами данных. Он поддерживает такие базы данных, как PostgreSQL, MySQL, MariaDB, SQLite и другие. Используя объекты JavaScript, Sequelize моделирует структуру таблиц базы данных и подключается к нужной реляционной базе данных для запроса и преобразования данных. Затем он анализирует и возвращает полученные данные в виде объекта JavaScript.



Особенности и преимущества библиотеки Sequelize

- Подключение к базам данных и выполнение операций без написания необработанных SQL-запросов
- Уменьшает уязвимости SQL-инъекций и атаки SQL-инъекций
- Совместима с GraphQL

CORS

CORS – это пакет Node.js для обеспечения кросс-оригинального разделения ресурсов (CORS) в качестве промежуточного программного обеспечения, которое опирается на Connect/Express. Пакет CORS формирует обертку вокруг промежуточного программного обеспечения маршрутизации Node.js, позволяя приложению получать доступ к ресурсам из доменов, отличных от его собственного. Он принимает множество параметров для настройки кросс-

оригинальных опций, включая происхождение, заголовки и многое другое.

```
var express = require('express')
var cors = require('cors')
var app = express()

var corsOptions = {
  origin: 'http://example.com',
  optionsSuccessStatus: 200 // some legacy browsers (IE11, various SmartTVs) choke on 204
}

app.get('/products/:id', cors(corsOptions), function (req, res, next) {
  res.json({msg: 'This is CORS-enabled for only example.com.'})
})

app.listen(80, function () {
  console.log('CORS-enabled web server listening on port 80')
})
```

Особенности и преимущества библиотеки CORS

- Сокращает объем кода, необходимого для включения CORS в веб-приложении
- Позволяет настраивать разрешенные домены и дает возможность пользователю включить CORS для выбранных источников, блокируя другие
- Обеспечивает бесшовную обработку ошибок и помогает разработчикам анализировать угрозы безопасности от подозрительных источников.

Nodemailer

Nodemailer упрощает отправку электронной почты с сервера Node.js. Он использует транспортный объект, который опирается на Simple Mail Transfer Protocol (SMTP), среди других поддерживаемых транспортов. Этот транспортный объект принимает от, до, тему, тело и другие параметры в качестве входных данных для создания сообщения.



Nodemailer

Особенности и преимущества библиотеки Nodemailer

- Единый модуль, поддерживающий множество транспортов, включая SMTP, Amazon Simple Email Service (SES), Sendmail и stream
- Поддерживает текстовый и HTML-контент в качестве тела письма
- Настройка уведомлений о статусе доставки и поддержка массовых рассылок электронной почты

Passport

Passport – это модульное промежуточное ПО аутентификации для Node.js. Passport поддерживает более 500 стратегий аутентификации, включая Google, Facebook, Twitter и другие пользовательские провайдеры и провайдеры единой регистрации (SSO). Стратегии включают обычную аутентификацию по имени пользователя и паролю, делегированную аутентификацию с использованием OAuth для сайтов социальных сетей и OpenID для федеративной аутентификации.



Особенности и преимущества библиотеки Passport

- Встроенная SSO-аутентификация для сайтов социальных сетей с минимальным кодом
- Настраивает постоянную информацию для входа в систему в течение нескольких сеансов
- Позволяет избежать установки новых маршрутов в приложении благодаря использованию беспрепятственной конфигурации с промежуточным ПО Express и Connect

Async

Async – это мощный утилитный модуль Node.js, который помогает разработчикам работать с асинхронным JavaScript путем работы с “асинхронными” или принимающими обратные вызовы функциями JavaScript. Если вы передаете модулю Async массив обратных вызовов, он выполняет и оборачивает их, чтобы вернуть обещание.



Особенности и преимущества библиотеки Async

- Предоставляет около 70 вспомогательных функций для легкой разработки асинхронного потока управления
- Предлагает “параллельную” функцию для обработки нескольких запросов к хосту (что в противном случае потребовало бы много кода для реализации)
- Помогает устранить вложенный “ад обратных вызовов” в JavaScript.

Winston

Winston – это библиотека протоколирования Node.js, которая поддерживает универсальное протоколирование с несколькими транспортом. Эти транспорты хранят и настраивают журналы в соответствии с потребностями вашего приложения. Помимо стандартных, функция createLogger помогает создавать пользовательские логгеры, которые используют доступные варианты транспорта, такие как консоли, файлы и базы данных. Пользовательские регистраторы также можно использовать с пользовательскими транспортом.

```
const { createLogger, format, transports } = require('..');

const logger = createLogger({
  level: 'info',
  format: format.combine(
    format.timestamp({
      format: 'YYYY-MM-DD HH:mm:ss'
    }),
    format.errors({ stack: true }),
    format.splat(),
    format.json()
  ),
  defaultMeta: { service: 'your-service-name' },
  transports: [
    //
    // - Write to all logs with level `info` and below to `quick-start-combined.log`.
    // - Write all logs error (and below) to `quick-start-error.log`.
    //
    new transports.File({ filename: 'quick-start-error.log', level: 'error' }),
    new transports.File({ filename: 'quick-start-combined.log' })
  ]
});
```

Особенности и преимущества библиотеки Winston

- Централизует контроль над ведением журнала через единый конфигурационный файл
- Позволяет настраивать форматы журналов, например, хранить журнал в формате JSON или в виде текста
- Предоставляет настраиваемые уровни протоколирования, которые можно настроить в соответствии с потребностями вашего приложения

Mongoose

Mongoose – это инструмент объектного моделирования для MongoDB на базе Node.js, также известный как библиотека объектного моделирования данных (ODM), который предлагает множество функций, таких как крючки, проверка модели,

подключение и запросы. Mongoose обеспечивает основанное на схемах решение для прикладных данных путем принудительного применения единой схемы на прикладном уровне, которая создает коллекцию в MongoDB. Каждая схема связана с моделью Mongoose, которая позволяет выполнять запросы к коллекции MongoDB, такие как получение, обновление и удаление данных.



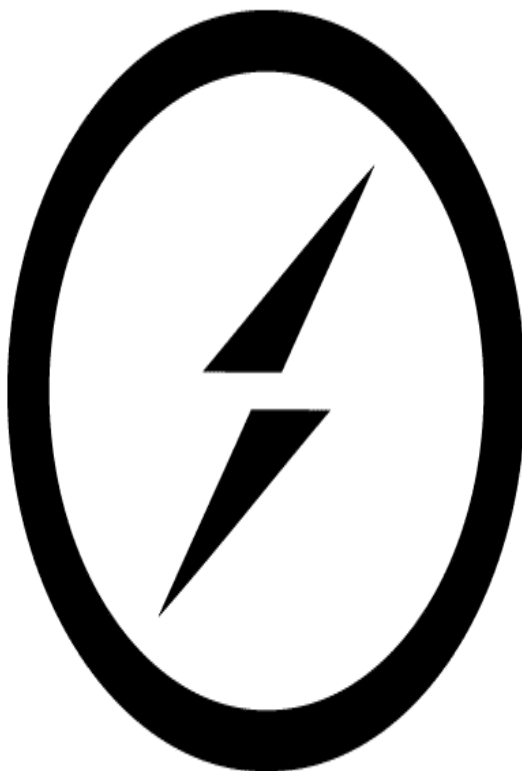
Особенности и преимущества библиотеки Mongoose

- Обеспечивает легкую абстракцию запросов, позволяя разработчикам писать меньше кода для транзакций MongoDB
- Встроенная проверка данных для определения правил о том, какие данные могут быть добавлены или обновлены в базе данных
- Реализует предопределенную структуру для коллекции MongoDB, которая предоставляет разработчикам типовой экземпляр MongoDB
- Позволяет создавать цепочки запросов для работы с несколькими запросами

Socket.IO

Socket.IO – это коммуникационная библиотека Node.js, которая устанавливает

двунаправленную, основанную на событиях коммуникацию в реальном времени между клиентским браузером и сервером. Она использует Engine.IO для установления низкоуровневого соединения между сервером и клиентом, используя цифровое рукопожатие через HTTP long-polling. После установления соединения связь между клиентом и сервером происходит в режиме реального времени по протоколу TCP.

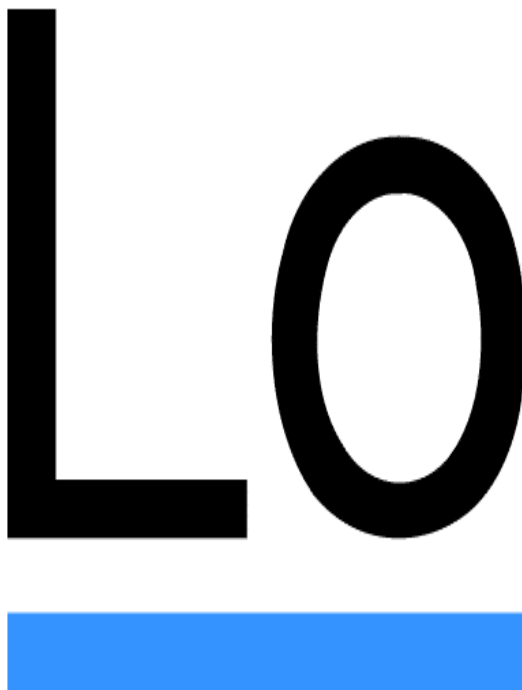


Особенности и преимущества библиотеки Sockets.IO

- Обеспечивает низкозатратный канал связи с использованием WebSocket и позволяет использовать HTTP long-polling в качестве запасного варианта
- Масштабируется и позволяет серверам легко транслировать события множеству клиентов
- Поддерживает мультиплексирование через пространства имен, что минимизирует количество используемых TCP-соединений и сокращает порты сокетов на сервере

Lodash

Lodash – это библиотека утилит, которая помогает разработчикам писать лаконичный и удобный в обслуживании код JavaScript. Она содержит более 200 полезных функций для упрощения повседневных задач программирования, включая проверку типов, простые математические операции и многое другое.

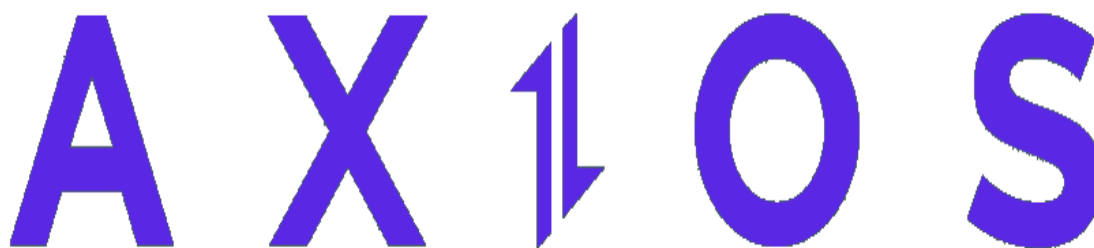


Особенности и преимущества библиотеки Lodash

- Поддерживает совместимость между браузерами с помощью полифиллов
- Обеспечивает встроенные решения при работе с массивом объектов, позволяя выполнять такие операции, как filter, find и flatMap
- Помогает разработчикам избегать повторений и поддерживать чистоту кода

Axios

Axios – это основанный на обещаниях HTTP-клиент для браузеров и Node.js. Он также обрабатывает преобразование данных запроса и ответа от браузера или Node.js по мере необходимости. Axios изоморфен, что означает, что он может работать на сервере и клиенте с одной и той же кодовой базой. Для HTTP-коммуникации Axios использует собственный модуль HTTP на стороне сервера и XMLHttpRequest на стороне клиента.

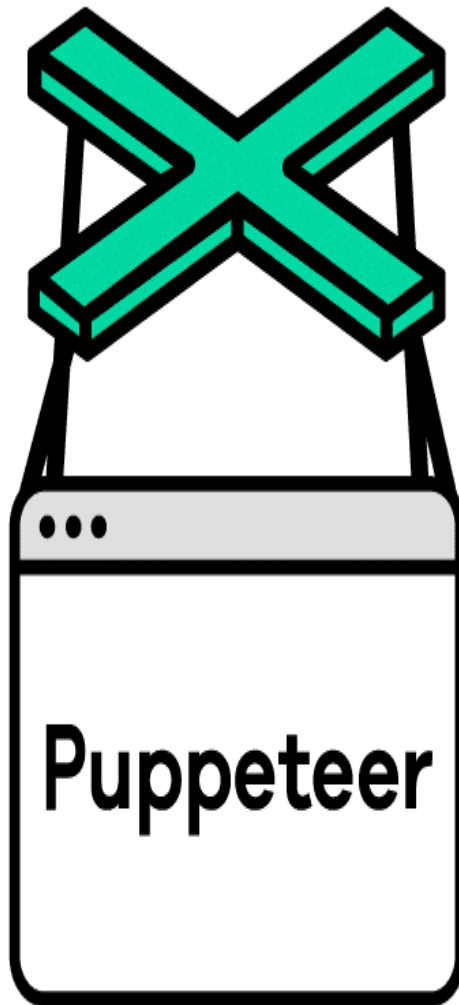


Особенности и преимущества библиотеки Axios

- Предлагает встроенные API-функции для распространенных типов данных HTTP, таких как GET, PUT, POST и DELETE.
- Обеспечивает большую безопасность благодаря защите от межсайтовой подделки запросов (CSRF) при выполнении HTTP-запросов через Интернет
- Легко преобразует данные ответа в JSON благодаря автоматическому преобразованию данных JSON.

Puppeteer

Puppeteer – это библиотека Node.js, которая автоматизирует Chrome, предоставляя высокоуровневый API для управления Chrome/Chromium через протокол DevTools. Она автоматизирует фронтенд-тестирование, включая тесты обработки запросов, поиск и сравнение элементов пользовательского интерфейса, тесты производительности и многое другое. Разработчики могут импортировать пакет Puppeteer в свой код для создания экземпляра Chromium. Затем этот экземпляр может автоматизировать тесты, взаимодействуя с движком браузера.



Особенности и преимущества библиотеки Puppeteer

- Нулевая установка, легко настраивается и не требует дополнительных драйверов
- Просматривает веб-сайты для создания предварительно отрендеренного контента
- Совместимость с популярными фреймворками для тестирования, такими как Jest и Mocha.

Multer

Multer – это библиотека промежуточного ПО для Node.js. Она написана поверх

парсера HTML-форм busboy и работает с многокомпонентными и многоформенными данными. После инициализации экземпляра Multer он принимает объект dest в качестве одного из параметров, чтобы указать, где загруженный файл будет храниться на сервере. Multer отправляет объект file вместе с запросом на загрузку, который затем анализируется и отправляется в место назначения с помощью Multer API.

```
<form action="/profile" method="post" enctype="multipart/form-data">
  <input type="file" name="avatar" />
</form>

const express = require('express')
const multer = require('multer')
const upload = multer({ dest: 'uploads/' })

const app = express()

app.post('/profile', upload.single('avatar'), function (req, res, next) {
  // req.file is the `avatar` file
  // req.body will hold the text fields, if there were any
})

app.post('/photos/upload', upload.array('photos', 12), function (req, res, next) {
  // req.files is array of `photos` files
  // req.body will contain the text fields, if there were any
})

const cpUpload = upload.fields([
  { name: 'avatar', maxCount: 1 },
  { name: 'gallery', maxCount: 8 }
])
app.post('/cool-profile', cpUpload, function (req, res, next) {
  // req.files is an object (String -> Array) where fieldname is the key, and the value is array of
  //
  // e.g.
  // req.files['avatar'][0] -> File
  // req.files['gallery'] -> Array
  //
  // req.body will contain the text fields, if there were any
})
```

Особенности и преимущества библиотеки Multer

- Делает необработанные данные HTTP-запроса более доступными для хранения благодаря встроенному синтаксическому анализу

- Позволяет указать тип кодировки для файла, что добавляет дополнительный уровень безопасности к загружаемому файлу
- Фильтры и возможность ограничения вариантов загрузки по типу и размеру файла

Dotenv

Dotenv – это библиотека утилит Node.js, которая управляет переменными окружения в приложении и защищает конфиденциальные переменные конфигурации. Dotenv также помогает приложению следовать методологии двенадцати факторного приложения при хранении переменных окружения. Конфигурирование библиотеки Dotenv на ранних этапах автоматически вводит переменные окружения из .env в process.env.



Особенности и преимущества библиотеки Dotenv

- Помогает отделить секреты – такие как ключи API и учетные данные для входа – от исходного кода и позволяет каждому разработчику создать файл .env для

собственного использования

- Не увеличивает размер приложения благодаря модулю нулевой зависимости

Заключение

Node.js имеет множество полезных библиотек, но выбрать лучшую для вашего проекта может быть непросто. Некоторые из библиотек Node.js, которые мы рассмотрели здесь, могут стать “обязательными” для вашего следующего приложения. Например, если вы работаете преимущественно с MongoDB, использование Mongoose может стать спасением. CORS поможет вам доставлять контент из нескольких доменов, а Dotenv может быть очень удобен, если вам нужно поделиться кодом – но не секретами – внутри команды. Теперь, когда вы ознакомились с некоторыми из наиболее важных пакетов Node.js, пришло время создать свое собственное приложение Node.js.

Дата Создания

05.07.2023