

Node.js

Разработка серверных веб-приложений на JavaScript



Дэвид Хэррон

ОМК
ИЗДАТЕЛЬСТВО

Дэвид Хэррон

Node.js

Разработка серверных веб-приложений на JavaScript

David Herron

Node Web Development

**A practical introduction to Node,
the exciting new server-side
JavaScript web development stack**

[PACKT]
PUBLISHING

BIRMINGHAM – MUMBAI

open source 
community experience distilled

Дэвид Хэррон

Node.js

Разработка серверных веб-приложений на JavaScript

2-е издание, электронное



Москва, 2023

УДК 004.738.5:004.45Node

ББК 32.973.202-018.2

X99

Хэррон, Дэвид.

X99 Node.js. Разработка серверных веб-приложений на JavaScript / Д. Хэррон ; пер. с англ. А. А. Слинкина. — 2-е изд., эл. — 1 файл pdf : 145 с. — Москва : ДМК Пресс, 2023. — Систем. требования: Adobe Reader XI либо Adobe Digital Editions 4.5 ; экран 10". — Текст : электронный.

ISBN 978-5-89818-632-6

Книга посвящена разработке веб-приложений в Node.js — платформе, которая выводит язык JavaScript за пределы браузера и позволяет использовать его в серверных приложениях. В основе платформы лежит исключительно быстрый движок JavaScript, заимствованный из браузера Chrome, к которому добавлена быстрая и надежная библиотека асинхронного сетевого ввода/вывода. Основной упор в Node.js делается на создании высокопроизводительных, хорошо масштабируемых клиентских и серверных приложений.

На практических примерах вы научитесь пользоваться серверным и клиентским объектами HTTP, каркасами Connect и Express, освоите алгоритмы асинхронного выполнения и узнаете, как работать с базами данных на основе SQL и с MongoDB.

Начав с практических рекомендаций по установке и настройке Node.js в режиме разработки и эксплуатации, вы научитесь разрабатывать клиентские и серверные HTTP-приложения; познакомитесь с применением в Node.js системой организации модулей на основе спецификации CommonJS, позволяющей реализовать подмножество технологии объектно-ориентированного проектирования.

Издание предназначено для программистов, знакомых с основами JavaScript и веб-разработки.

УДК 004.738.5:004.45Node

ББК 32.973.202-018.2

Электронное издание на основе печатного издания: Node.js. Разработка серверных веб-приложений на JavaScript / Д. Хэррон ; пер. с англ. А. А. Слинкина. — Москва : ДМК Пресс, 2016. — 144 с. — ISBN 978-5-97060-397-0. — Текст : непосредственный.

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

Материал, изложенный в данной книге, многократно проверен. Но поскольку вероятность технических ошибок все равно существует, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несет ответственности за возможные ошибки, связанные с использованием книги.

В соответствии со ст. 1299 и 1301 ГК РФ при устранении ограничений, установленных техническими средствами защиты авторских прав, правообладатель вправе требовать от нарушителя возмещения убытков или выплаты компенсации.

ISBN 978-5-89818-632-6

© Packt Publishing

© Издание, Оформление, ДМК Пресс



Содержание

Об авторе	8
Благодарности	9
О рецензентах	10
Предисловие	11
О содержании книги	11
Что необходимо для чтения этой книги	12
На кого рассчитана эта книга	13
Графические выделения	13
Отзывы	14
Поддержка клиентов	14
Исходный код примеров	14
Опечатки	14
Нарушение авторских прав	15
Вопросы	15
Глава 1. ЧТО ТАКОЕ NODE?	16
Что позволяет делать Node?	17
Серверный JavaScript	18
Почему имеет смысл использовать Node?	18
Архитектура: потоки или асинхронный ввод/вывод с управлением по событиям	19
Производительность и использование процессора	21
Использование серверов, экономия затрат и экологичный Интернет	23
Как правильно: Node, Node.js или Node.JS?	24
Резюме	24
Глава 2. НАСТРОЙКА NODE	25
Системные требования	25
Установка в POSIX-совместимых системах (Linux, Solaris, Mac и т. п.)	26
Предварительная установка инструментария	26
Установка средств разработки в Mac OS X	26
Установка в свой домашний каталог	27
Зачем устанавливать в домашний каталог?	28
Установка в системный каталог	29
Установка в Mac OS X с помощью MacPorts	29
Установка в Mac OS X с помощью homebrew	30
Установка в Linux с помощью систем управления пакетами	30
Установка одновременно нескольких экземпляров Node	31
Выполним несколько команд для проверки установки	31

Командные утилиты Node.....	31
Запуск скрипта в Node	33
Запуск сервера в Node	34
Установка npm – менеджера пакетов для Node.....	35
Запуск Node-серверов на этапе инициализации системы	36
Использование всех процессорных ядер в многоядерной системе	40
Резюме	42
Глава 3. МОДУЛИ NODE.....	43
Что такое модуль?	43
Модули Node.....	44
Как Node ищет модули, затребованные в require('module')?.....	44
Идентификаторы модулей и пути	44
Локальные модули внутри приложения	45
Комплектация приложения с внешними зависимостями	46
Системные модули в каталогах, перечисленных в массиве require.paths	48
Составные модули – модули-каталоги	49
Менеджер пакетов для Node (npm).....	50
Формат npm-пакета	50
Поиск npm-пакетов	52
Команды npm	53
Версии и диапазоны версий пакета.....	61
Спецификация CommonJS	63
Резюме	64
Глава 4. ВАРИАЦИИ НА ТЕМУ ПРОСТОГО ПРИЛОЖЕНИЯ	65
Разработка учебной программы по математике	65
Использовать ли каркас?.....	65
Реализация Math Wizard в Node (без каркасов)	66
Маршрутизация запросов в Node.....	66
Обработка параметров запроса	67
Умножение чисел	69
Вычисление других математических функций	70
Обобщение Math Wizard	73
Продолжительные вычисления (числа Фибоначчи)	74
Чего не хватает до «настоящего веб-сервера»?.....	77
Использование каркаса Connect для реализации Math Wizard	78
Установка и настройка Connect	79
Знакомство с Connect	80
Реализация Math Wizard с помощью Express	82
Реализация Express Math Wizard	82
Обработка ошибок	87
Параметризованные URL и службы данных	88
Резюме	93
Глава 5. ПРОСТОЙ ВЕБ-СЕРВЕР, ОБЪЕКТЫ EVENTEMITTER И HTTP-КЛИЕНТЫ	95
Отправка и получение событий с помощью объектов EventEmitter.....	95
Теоретические основы EventEmitter.....	97
HTTP Sniffer – прослушивание обмена данными по протоколу HTTP	97

Реализация простого веб-сервера	100
Реализация Basic Server	101
Типы MIME и прм-пакет MIME	110
Обработка куков	111
Отправка HTTP-запросов клиентом	112
Резюме	114
Глава 6. ХРАНЕНИЕ И ВЫБОРКА ДАННЫХ	115
Движки сохранения данных для Node	115
SQLite3 – облегченная встраиваемая база данных на основе SQL	115
Установка	116
Реализация приложения Notes с помощью SQLite3	116
Использование других СУБД на основе SQL на платформе Node	129
Mongoose – интерфейс между Node и MongoDB	130
Установка Mongoose	130
Реализация приложения Notes с помощью Mongoose	131
Отображение заметок на консоли – show.js	135
Другие продукты, поддерживающие MongoDB	137
Краткий обзор средств аутентификации пользователей	138
Резюме	140
Предметный указатель	141



Об авторе

Дэвид Хэррон вот уже больше 20 лет занимается созданием программного обеспечения, работая в Кремниевой долине в роли разработчика и инженера по контролю качества. Его последнее место работы – архитектор по организации контроля качества в компании Yahoo!, где ведутся работы по созданию новой платформы для веб-приложений на основе Node.

В компании Sun Microsystems Дэвид занимал должность архитектора по организации контроля качества Java SE и работал главным образом над средствами автоматизации тестирования, в том числе написанным на базе AWT классом Robot, который ныне широко применяется в программах автоматизации тестирования графического интерфейса пользователя (ГИП). Он принимал участие в запуске проектов OpenJDK и JDK-Distros, а также отвечал за организацию конкурса Mustang Regressions Contest, идея которого состояла в том, чтобы обратиться к обществу разработчиков на Java с просьбой поискать ошибки в версии 1.6.

До перехода в Sun Дэвид работал в компании VXtreme, где занимался разработкой программных средств потокового видео. После покупки этой компании корпорацией Microsoft созданный ей продукт лег в основу Windows Media Player. В компании The Wollongong Group Дэвид работал над клиентским и серверным ПО электронной почты и принимал участие в деятельности нескольких рабочих групп IETF, направленной на совершенствование протоколов электронной почты.

Дэвида интересуют транспортные средства с электрическими двигателями, мировые запасы энергии, изменение климата и вопросы охраны окружающей среды. Он сооснователь компании Transition Silicon Valley. В качестве сетевого журналиста он ведет раздел Green Transportation Examiner на сайте examiner.com, пишет на темы экологии в блоге на сайте 7gen.com, организовал дискуссионный форум по электромобилям на сайте visforvoltage.org, а также обсуждает различные программы, в том числе Node.js, Drupal и Doctor Who на сайте davidherron.com.



Благодарности

Я благодарен многим людям.

Хочу сказать спасибо матери Эвелин за... ну, в общем, за все, отцу Джиму, сестре Пэтти и брату Кену. Даже не представляю, как жил бы без вас!

Спасибо моей подруге Мэгги за то, что она всегда рядом, готовая поддержать меня, за ее веру в меня, за мудрость и чувство юмора и за то, что устраивает мне головомойки, когда это необходимо. Пусть так будет и дальше.

Хочу поблагодарить доктора Кена Кубота из Университета штата Кентукки, который поверил в меня и принял на первую работу, связанную с компьютерами. Шесть лет я учился у него не только обслуживанию компьютерных систем, но и многому другому.

Спасибо моим бывшим работодателям, математическому факультету Университета штата Кентукки, компаниям The Wollongong Group, MainSoft, VXtreme, Sun Microsystems и Yahoo!, а также всем, с кем мне довелось работать в разных компаниях. Я благодарен своей бывшей начальнице Тине Су, которая побуждала меня выступать на публичных мероприятиях и писать, хотя это совсем не свойственно инженеру-программисту с интровертным складом ума. Особенно я благодарен компании Yahoo!, которая предоставила мне возможность поработать над внутренним проектом на базе Node.js и понять, что существует потребность в этой книге.

Я признателен издательству Packt Publishing за предоставленную возможность написать эту книгу. Благодаря сотрудникам издательства, которые помогли мне на протяжении всей работы, я понял, что писать книги – моя мечта.

Я благодарю Райана Дала, Айзека Шлютера и других разработчиков ядра Node за их мудрость и провидческий дар, необходимый для создания такой наполняющей сердце радостью подвижной платформы для разработки ПО. Есть платформы, с которыми работать откровенно трудно, но эта не из их числа. И чтобы реализовать ее настолько хорошо, необходимы проницательность и дальновидность.



О рецензентах

Благовест Дачев, пишет программы для веб с 2002 года. Он прошел все стадии: начал с HTML, CSS и JavaScript, а затем перебрался в мир серверов и баз данных. Благовест одним из первых принял платформу Node.js и внес вклад в развитие нескольких проектов с открытым исходным кодом. В настоящее время он работает инженером-программистом в компании Dow Jones & Company, где занимается разработкой системы виджетов, позволяющей третьим сторонам искать и публиковать новости на своих сайтах.

Благовест учился в Массачусетском университете в Амхерсте, где участвовал в исследованиях по информационно-поисковым системам, завершил проекты в рамках двух программ Google Summer of Code подряд и был соавтором нескольких статей.

Я хотел бы поблагодарить свою маму Татьяну за любовь, неустанную заботу и силу духа, которые вдохновляли меня на протяжении многих лет, а также отца Йордана за счастливые детские воспоминания.

Мэтт Рэнни, одним из первых принял платформу Node.js и внес свой вклад в ее развитие. Он один из основателей компании Voxel, использующей Node на своих серверах.



Предисловие

Добро пожаловать в мир разработки ПО на базе Node (другое название – Node.js). Node – это недавно появившаяся платформа, которая выводит язык JavaScript за пределы браузера и позволяет использовать его в серверных приложениях. В основе платформы лежит исключительно быстрый движок JavaScript, заимствованный из браузера Chrome, V8, к которому добавлена быстрая и надежная библиотека асинхронного сетевого ввода/вывода. Основной упор в Node делается на создании высокопроизводительных, хорошо масштабируемых клиентских и серверных приложений для «веб реального времени».

Эту платформу разработал Райан Дал (Ryan Dahl) в 2009 году, после двух лет экспериментирования с созданием серверных веб-компонентов на Ruby и других языках. В ходе своих исследований он пришел к выводу, что вместо традиционной модели параллелизма на основе потоков следует обратиться к событийно-ориентированным системам. Эта модель была выбрана за простоту (хорошо известно, что многопоточные системы трудно реализовать правильно), за низкие накладные расходы, по сравнению с идеологией «один поток на каждое соединение», и за быстродействие. Цель Node – предложить «простой способ построения масштабируемых сетевых серверов». При проектировании за образец были взяты такие системы, как Event Machine (Ruby) и каркас Twisted (Python).

В настоящей книге рассматривается в первую очередь вопрос о построении веб-приложений с помощью Node. Мы познакомимся с важными концепциями, которые необходимо понимать, чтобы повысить быстродействие приложения. Для этого мы будем писать реальные приложения, подробно анализировать их составные части и обсуждать, как применить эти идеи в своих программах. Мы установим Node и npm и научимся устанавливать и разрабатывать npm-пакеты и модули Node. Мы создадим несколько приложений, изучим, как отражаются на отзывчивости цикла обработки событий продолжительные вычисления, расскажем о двух способах распределения нагрузки между серверами, поработаем с каркасом Express и прочее.

О содержании книги

Глава 1 «Что такое Node?» содержит введение в платформу Node. Мы поговорим о том, где она применяется, об архитектурных решениях, принятых в Node, о ее истории и об истории использования JavaScript на стороне сервера, а также о том, почему JavaScript не должен быть замурован в браузере.

В главе 2 «Настройка Node» речь пойдет о настройке среды разработки для Node, в том числе о нескольких способах сборки и установки из исходного кода. Мы также коснемся вопроса о развертывании Node на производственных серверах.

Глава 3 «Модули Node» посвящена модулям – единицам модульной структуры приложений для Node. Мы разберемся в том, что такое модули, и разработаем несколько. Затем познакомимся с программой `npm`, Node Package Manager (менеджер пакетов Node) и рассмотрим несколько способов использования `npm` для управления установленными пакетами, а также для разработки и распространения `npm`-пакетов.

В главе 4 «Вариации на тему простого приложения» мы, уже вооруженные знанием основ, приступим к изучению процесса разработки приложений на платформе Node. Точнее, мы разработаем простое приложение, применяя саму Node, систему промежуточного уровня `Connect` и веб-каркас `Express`. Хотя приложение несложное, оно даст нам возможность изучить цикл обработки событий в Node, узнать о его адаптации к продолжительным вычислениям, познакомиться с синхронными и асинхронными алгоритмами и вынести громоздкие вычисления на вспомогательный сервер.

В главе 5 «Простой веб-сервер, объекты `EventEmitter` и HTTP-клиенты» объясняется, что в Node клиентские и серверные объекты – это соответственно передний план и центр. Мы подробно рассмотрим обе стороны протокола HTTP, для чего разработаем клиентское и серверное приложения на основе HTTP.

В главе 6 «Хранение и выборка данных» речь пойдет о том, что в большинстве приложений необходимо какое-то надежное хранилище для длительного хранения данных. Мы рассмотрим применение для этой цели СУБД на основе SQL и MongoDB. Попутно поговорим о применении каркаса `Express` для аутентификации пользователей и создания более привлекательной страницы с сообщением об ошибке.

Что необходимо для чтения этой книги

В настоящее время мы обычно собираем Node из исходного кода. Лучше всего платформа работает в системах на базе Unix или совместимых со стандартом POSIX. Предъявляемые Node требования достаточно скромны, а самым важным инструментом является то, что находится у вас между ушами.

Для установки из исходного кода необходима система на базе Unix/POSIX (Linux, Mac, FreeBSD, OpenSolaris и т. д.), современный компилятор C/C++, библиотеки `OpenSSL` и `Python` версии не ниже 2.4.

Программы для Node можно набирать в любом текстовом редакторе, но лучше использовать такой, который понимает синтаксис JavaScript, HTML, CSS и т. д.

Хотя эта книга посвящена разработке веб-приложений, иметь веб-сервер обязательно. В состав Node входит собственный веб-сервер.

На кого рассчитана эта книга

Данная книга рассчитана на любого программиста, ищущего приключений, сопутствующих новой программной платформе, построенной на основе новой парадигмы программирования.

Разработчики серверных компонентов, возможно, придут к выводу, что свежие идеи позволили им по-новому взглянуть на процесс создания веб-приложений. JavaScript – мощный язык, а асинхронная природа Node подчеркивает его сильные стороны.

Разработчикам, имеющим опыт работы с JavaScript в браузере, будет интересно применить свои знания на новой территории и посмотреть, что значит писать на JavaScript без доступа к DOM. (Раз нет браузера, то нет и DOM, если только вы не установите JSDom.)

Хотя главы зависят друг от друга, порядок чтения выбираете вы сами.

Мы предполагаем, что вы уже умеете писать программы и знакомы с современными языками программирования, такими как JavaScript.

Графические выделения

В этой книге используются различные шрифты для обозначения типа информации. Ниже приведено несколько примеров с пояснениями.

Фрагменты кода внутри абзаца выделяются следующим образом: «Объект `http` инкапсулирует протокол HTTP, а его метод `http.createServer` создает полноценный веб-сервер, прослушивающий порт, указанный в методе `.listen`».

Блок кода выделяется следующим образом:

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(8124, "127.0.0.1");
console.log('Server running at http://127.0.0.1:8124/');
```

Если мы хотим привлечь внимание к конкретной части кода, то выделяем ее полужирным шрифтом:

```
var util = require('util');
var A = "a different value A";
var B = "a different value B";
var m1 = require('./module1');
util.log('A='+A+ ' B='+B+ ' values='+util.inspect(m1.values()));
```

Команды и выводимая ими информация записываются так:

```
$ sudo /usr/sbin/update-rc.d node defaults
```
