Переходим на сторону сервера

Изучаем Node.js

O'REILLY®

Шелли Пауэрс

Shelley Powers

Learning Node

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

Шелли Пауэрс

Изучаем Node.js

ПИТЕР®

Москва · Санкт-Петербург · Нижний Новгород · Воронеж
Ростов-на-Дону · Екатеринбург · Самара · Новосибирск
Киев · Харьков · Минск
2014

ББК 32.988.02-018
УДК 004.737.5
П21

Пауэрс Ш.
П21 Изучаем Node.js. — СПб.: Питер, 2014. — 400 с.: ил. — (Серия «Бестселлеры O'Reilly»).

ISBN 978-5-496-00356-8

Node.js является серверной технологией, которая основана на разработанном компанией Google JavaScript-движке V8. Это прекрасно масштабируемая система, поддерживающая не программные потоки или отдельные процессы, а асинхронный ввод-вывод, управляемый событиями. Она идеально подходит для веб-приложений, которые не выполняют сложных вычислений, но к которым происходят частые обращения. По целям использования Node сходен с фреймворками Twisted на языке Python и EventMachine на Ruby. В отличие от большинства программ JavaScript этот фреймворк исполняется не в браузере клиента, а на стороне сервера.

С помощью этого практического руководства вы сможете быстро овладеть основами Node. Книга понравится всем, кто интересуется новыми технологиями, например веб-сокетами или платформами создания приложений. Эти темы раскрываются в ходе рассказа о том, как использовать Node в реальных приложениях.

12+ (В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ББК 32.988.02-018
УДК 004.737.5

ISBN 978-1449323073 англ.

ISBN 978-5-496-00356-8

Краткое содержание

Содержание

Предисловие

Это не совсем JavaScript

Вы выбрали для изучения Node весьма удачный момент.

Технологии, развивающиеся вокруг Node, довольно свежи и полны жизни, постоянно появляются новые варианты и уточнения. В то же время технологическая база достигла достаточного уровня зрелости, гарантирующего, что время на изучение Node будет потрачено не зря: установка еще никогда не была такой простой, даже под Windows «лучшие в своем классе» модули начинают выделяться среди, пожалуй, сотен других доступных модулей, инфраструктура стала достаточно надежной для ее практического использования.

При работе с Node нужно помнить о двух важных обстоятельствах. Во-первых, в основе Node лежит JavaScript, причем это почти тот же язык, который используется при разработке сценариев на стороне клиента. По правде сказать, можно применять и другие языки сценариев, например CoffeeScript, но JavaScript является для этой технологии *общепринятым*.

Во-вторых, необходимо помнить, что Node — это не просто JavaScript, это серверная технология, а значит, некоторые функциональные средства (и защитные механизмы), привычно ожидаемые в браузере, здесь не нужны, зато нужны многие новые и потенциально совершенно незнакомые способности.

Но если Node — это почти то же самое, что JavaScript в браузере, то почему Node?

Почему именно Node?

Если исследовать исходный код Node, то в нем обнаружится исходный код V8 (с технической стороны — ECMAScript), то есть JavaScript-движка, разработанного в Google и используемого в ядре браузера Google Chrome. Одно из преимуществ

Node.js заключается в возможности разработки Node-приложений только для одной реализации JavaScript, а не для полудюжины различных браузеров и их версий.

Технология Node была задумана как платформа создания приложений, ориентированных на высокую интенсивность ввода-вывода и невысокую интенсивность вычислений. Что еще важнее, Node предлагает эту функциональность в полностью готовом виде. Вам не придется беспокоиться о том, что приложение заблокирует всю остальную работу, ожидая завершения загрузки файла или обновления базы данных, поскольку большая часть функциональности по умолчанию относится к *асинхронному вводу-выводу*. Вам также не нужно волноваться по поводу программных потоков, поскольку Node-приложение выполняется в единственном программном потоке.

Асинхронный ввод-вывод означает, что приложение не ждет завершения ввода-вывода перед переходом к следующему шагу в коде приложения. Асинхронная природа Node более подробно рассматривается в главе 1.

Особую важность имеет то, что исходный код Node написан на языке JavaScript, с которым знакомы многие рядовые веб-разработчики. Возможно, вам придется изучать новые технологии, например веб-сокеты или Express, но, по крайней мере, наряду с ними не придется изучать еще и новый язык. Когда язык уже знаком, проще сосредоточиться на новом материале.

Для кого предназначена эта книга

Почему-то считается, что большинство людей, пришедших к Node-разработке, прежде имели дело с Ruby, Python или Rails. Лично я так не думаю, поэтому, рассказывая о Node-компонентах, не буду говорить что-то вроде: «А это похоже на Синатру».

В этой книге предполагается лишь то, что вы, читатель, прежде программировали на JavaScript и хорошо знакомы с этим языком. Вам не нужно быть специалистом высшей квалификации, но вы должны знать, о чем идет речь, когда я говорю о *замыканиях*, у вас должен быть опыт работы с Ajax, вы должны понимать, что такое обработка событий в клиентской среде. Кроме того, книга будет вам понятнее, если вы занимались обычной веб-разработкой и знакомы с такими понятиями, как HTTP-методы (`GET` и `POST`), веб-сеансы, cookie-файлы и т. д. Вы также должны уметь работать либо с консолью в Windows, либо с командной строкой в Mac OS X или Linux (Unix).

Книга также должна понравиться тем, кто интересуется новыми технологиями, например веб-сокетами или платформами создания приложений. Эти темы раскрываются в ходе рассказа о том, как использовать Node в реальных приложениях.

Самое важное при чтении книги — не бояться столкнуться с непонятным. Будьте готовы, что вам периодически придется натыкаться на препятствия в виде альфа- и бета-версий и сталкиваться с ляпами динамической технологии. В конце концов, ведь главное — это изучение Node, что действительно интересно.

Если вы не уверены, что знакомы с JavaScript в достаточной степени, можете обратиться к моей книге «Learning JavaScript», второе издание (O'Reilly).

Как получить от этой книги максимальную пользу

Вы не обязаны читать все главы по порядку, тем не менее очередность чтения зависит от того, что вы собираетесь делать потом и каков опыт вашей работы с Node.

Если работать с Node вам еще не приходилось, то лучше начать с первой главы и прочитать, как минимум, все главы по пятую включительно. В этих главах описывается, как установить Node и диспетчер Node-пакетов, как их использовать, как создать свое первое приложение, как работать с модулями. В главе 5 рассматриваются также некоторые вопросы стилевого оформления и уникальный подход к разработке асинхронных приложений.

Если вы уже кое-что знаете о Node, работали как со встроенными в Node, так и с внешними модулями, использовали REPL (интерактивная консоль, поддерживающая цикл чтения, вычисления и вывода на экран), то можете спокойно пропускать главы с первой по четвертую, но начинать чтение я все же рекомендую не далее, чем с главы 5.

Во всех примерах, описываемых в книге, я использую платформу Express и связующий модуль Connect. Если работать с Express вам еще не приходилось, то, наверное, нужно изучить главы с шестой по восьмую, где рассматриваются вопросы маршрутизации, использования прокси-серверов, веб-серверов и связующего программного обеспечения, а также дается введение в Express. В частности, если вы интересуетесь вопросами применения Express на базе MVC (Model-View-Controller — модель-представление-контроллер), то обязательно прочтите седьмую и восьмую главы.

После этих базовых глав вы можете перейти к выборочному чтению. Например, если вы преимущественно работаете с парами ключ-значение, нужно прочитать материал по Redis в главе 9, а если вас интересуют данные в виде документов, обратитесь к главе 10, где кратко рассказывается об использовании MongoDB с Node. Если же вы собираетесь работать исключительно с реляционными базами данных, то можете сразу переходить к главе 11, пропустив главы, посвященные Redis и MongoDB, хотя лучше найти время и для них, поскольку они позволяют взглянуть на использование данных по-новому.

После этих трех глав, посвященных работе с данными, мы приступим к специализированным приложениям. Глава 12 целиком посвящена доступу к графике и медиаданным, включая передачу медиаданных новому HTML5-элементу, предназначенному для воспроизведения видео, обработку PDF-документов и использование модуля Canvas. В главе 13 рассматривается весьма популярный модуль Sockets.io, специально ориентированный на новую функциональность веб-сокетов.

После разделения на два специальных варианта применения Node в главах 12 и 13 мы вернемся в общее русло повествования и не покинем его до конца книги. После того как вы уделили время примерам в других главах, вам, наверное, потребуется уделить немного времени и примерам в главе 14, более подробно изучая практику отладки и тестирования Node-приложений.

Глава 15, наверное, одна из самых сложных, но и самых важных. В ней рассматриваются вопросы безопасности и прав доступа. Я не собираюсь рекомендовать читать ее одной из первых, но перед тем как выходить в свет со своим Node-приложением, вам нужно обязательно уделить время этой главе.

Заключительную главу 16 можно смело оставить напоследок, независимо от ваших интересов и навыков. Она посвящена подготовке приложения к эксплуатации, включая развертывание Node-приложения не только на вашей системе, но и на одном из облачных серверов, предназначенных для хостинга Node-приложений. Кроме того, в ней рассматриваются вопросы развертывания Node-приложения на сервере и обеспечения его совместной работы с другим веб-сервером, таким как Apache, а также вопросы поддержания живучести вашего приложения в условиях сбоев и его перезапуска при перезагрузке системы.

Технология Node тесно связана с предлагаемыми Git приемами управления исходным кодом, и большинство (если не все) Node-модулей размещены на сайте GitHub. В приложении вы найдете руководство по выживанию в среде Git/GitHub для тех, кто с этим еще не сталкивался.

Я уже упоминал, что вам не обязательно читать главы по порядку, но я все же рекомендую поступить именно так. В основе многих глав лежит материал предыдущих, и пропуская главы, вы можете упустить важные детали. Кроме того, хотя примеры в книге достаточно автономны, я использую одно относительно простое Express-приложение под названием фабрики виджетов, код которого упоминается, начиная с главы 7, и в той или иной степени используется во всех остальных главах. Я уверен, что для вас полезнее приступить к чтению с самого начала, а затем бегло просматривать те разделы, в которых содержится уже известная вам информация, а не пропускать сразу всю главу.

Помните Алису в стране чудес? «Начни с начала, — торжественно произнес Король, — и продолжай, пока не дойдешь до конца. Тогда остановись!»

Технология

Примеры в этой книге создавались в различных выпусках Node 0.6.x. Хотя большинство из них тестировалось в среде Linux, они должны работать без всяких изменений в любой Node-среде.

Когда книга уже была запущена в производство, вышла версия Node 0.8.x. Примеры, представленные в данной книге, в большинстве своем должны работать и в Node 0.8.x, а все моменты, где в код потребуется внести изменения, мною помечены, чтобы гарантировать, что приложение будет работать в самом последнем выпуске Node.

Примеры

Примеры можно найти в архивном файле на веб-странице издательства O'Reilly, посвященной этой книге (http://oreil.ly/Learning_node). После загрузки и распаковки и при

наличии установленной копии Node вы можете установить все библиотеки, от которых зависит работа примеров, перейдя в каталог examples и введя следующую команду:

```
npm install -d
```

Более подробно работа диспетчера Node-пакетов рассмотрена в главе 4.

Соглашения, используемые в этой книге

В данной книге используются следующие соглашения, связанные с типографским оформлением.

Шрифт без засечек

Служит признаком заголовков, пунктов и кнопок меню, клавиатурных комбинаций (с использованием клавиш Alt и Ctrl), а также URL-адресов, адресов электронной почты, имен файлов, расширений этих имен, путевых имен и каталогов.

Курсив

Служит признаком новых понятий.

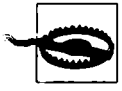
`Моноширинный шрифт`

Служит признаком команд, переменных, атрибутов, ключей, функций, типов, классов, пространств имен, методов, модулей, свойств, параметров, значений, объектов, событий, обработчиков событий, XML-тегов, HTML-тегов, макросов, контента файлов и результатов выполнения команд, а также кода, который должен набираться пользователем буквально.

`Моноширинный наклонный шрифт`

Текст, который должен быть заменен значениями, предоставляемыми пользователями.

Этими значками отмечены советы, примечания или общие замечания.



Этими значками отмечены предупреждения или предостережения.

Использование примеров кода

Эта книга призвана помочь вам в решении задач. По большей части вы можете использовать код из книги в своих программах и документации. Вам не нужно связываться с нами по поводу получения разрешения на это, если только вы не начнете копировать достаточно существенные фрагменты кода. Например, написание программы, в которой используется несколько фрагментов кода из этой

книги, не требует разрешения. А вот продажа или распространение компакт-дисков с примерами из книг издательства O'Reilly требует разрешения. Ответы на вопросы с использованием цитат из этой книги и приведением примеров не требуют получения разрешения. А вставка существенных объемов кода примеров из этой книги в документацию потребует разрешения.

Благодарности

Как всегда, я благодарен своим друзьям и семье за поддержку при работе над этой книгой. Особые благодарности моему редактору Симону Сен-Лорану (Simon St. Laurent), которому приходилось терпеть мои излияния.

Мои благодарности также коллективу издательства, помогавшему воплотить идею книги в осязаемый конечный продукт: Рэйчел Стили (Rachel Steely), Рэйчел Монохан (Rachel Monaghan), Килю Ван-Хорну (Kiel Van Horn), Аарону Хазелтону (Aaron Hazelton) и Ребекке Демарест (Rebecca Demarest).

Помните, что работая с Node, вы используете труд его разработчиков, начиная от создателя Node.js, Райана Дала (Ryan Dahl), и создателя диспетчера Node-пакетов, Исаака Шлютера (Isaac Schlueter), который теперь отвечает также и за Node.js.

Поставщиками весьма полезного кода и примеров для этой книги стали Берт Белдер (Bert Belder), Т. Дж. Холовайчук (TJ Holowaychuk), Джереми Ашкенас (Jeremy Ashkenas), Микеал Роджерс (Mikeal Rogers), Гильермо Раух (Guillermo Rauch), Джаред Хансон (Jared Hanson), Феликс Гейзендорфер (Felix Geisendörfer), Стив Сандерсон (Steve Sanderson), Мэтт Рэнни (Matt Ranney), Каолан МакМахон (Caolan McMahon), Реми Шарп (Remy Sharp), Крис О'Хара (Chris O'Hara), Мариано Иглесиас (Mariano Iglesias), Марко Аурелио (Marco Aurélio), Дамиан Суарез (Damián Suárez), Натан Райлич (Nathan Rajlich), Кристиан Амор Квалхейм (Christian Amor Kvalheim) и Джианни Чиапетта (Gianni Chiappetta). Мои извинения тем разработчикам модулей, которых я случайно пропустил.

И чем бы была технология Node без тех прекрасных людей, предоставивших нам учебники, ответы на вопросы и полезные руководства? Спасибо Тиму Касвелу (Tim Caswell), Феликсу Гейзендорферу (Felix Geisendörfer), Микато Такада (Mikato Takada), Гео Полу (Geo Paul), Мануэлю Кислингу (Manuel Kiessling), Скотту Хансельману (Scott Hanselman), Петеру Круминсу (Peter Krumins), Тому Хьюз-Кроучеру (Tom Hughes-Croucher), Бену Наделю (Ben Nadel) и всему коллективу Nodejitsu и Joyent.

От издательства

Ваши замечания, предложения, вопросы отправляйте по адресу электронной почты comp@piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На веб-сайте издательства http://www.piter.com вы найдете подробную информацию о наших книгах.

1 Установка и запуск Node.js

Node.js является серверной технологией, которая основана на разработанном компанией Google JavaScript-движке V8. Это прекрасно масштабируемая система, поддерживающая не программные потоки или отдельные процессы, а асинхронный ввод-вывод, управляемый событиями. Она идеально подходит для веб-приложений, которые не выполняют сложных вычислений, но к которым происходят частые обращения.

Если вы используете обычный веб-сервер, например Apache, то при каждом запросе веб-ресурса для обслуживания этого запроса Apache создает отдельный программный поток или вызывает новый процесс. Даже если Apache реагирует на запросы достаточно быстро, а после удовлетворения запроса все приводит в порядок, при таком подходе задействуется множество ресурсов. В результате у наиболее популярных веб-приложений возникают предпосылки для серьезных проблем производительности.

В отличие от этого Node не создает новый программный поток или процесс для каждого запроса, а прослушивает конкретные события, и когда эти события происходят, соответствующим образом на них реагирует. Node не блокирует никаких запросов, дожидаясь завершения действий, инициируемых событием, а сами события обрабатываются в относительно простом *цикле обработки событий* по принципу «первым пришел — первым обслужен».

Node-приложения создаются с помощью языка JavaScript (или альтернативных языков, компилирующихся в JavaScript), который ничем не отличается от языка, применяемого в приложениях на стороне клиента. Однако в отличие от языка JavaScript, используемого в браузере, для Node нужно создать среду разработки.

Node можно установить на платформе Unix/Linux, Mac OS или Windows. Эта глава проведет вас через всю процедуру создания среды разработки для Node в Windows 7 и Linux (Ubuntu). Установка на Mac аналогична установке на Linux. Кроме того, мы рассмотрим все требования и подготовительные действия, которые необходимо выполнить перед установкой.

Как только ваша среда разработки будет готова, я продемонстрирую простейшее Node-приложение и покажу самое главное — упомянутый ранее цикл обработки событий.

Создание среды разработки для Node

На большинстве платформ при установке Node можно использовать несколько подходов. Какой из них выбрать, зависит от имеющейся у вас среды разработки, вашей квалификации в плане работы с исходным кодом и того, как вы планируете задействовать Node в имеющихся у вас приложениях.

Установочные пакеты предоставляются как для Windows, так и для Mac OS, но вы также можете установить Node, используя копию исходного кода и скомпилировав приложение. Можно также задействовать распределенную систему управления версиям Git для *клонирования* (выгрузки) Node-*репозитория* во всех трех средах.

В этом разделе я собираюсь показать, как заставить Node работать в системе Linux на виртуальном выделенном сервере (Virtual Private Server, VPS) Ubuntu 10.04 путем непосредственного извлечения и компиляции исходного кода. Кроме этого, я покажу, как установить Node-сервер, чтобы его можно было использовать с помощью WebMatrix компании Microsoft на персональном компьютере, работающем под управлением Windows 7.

Загрузите установщики исходного и основного пакетов Node с веб-сайта http://nodejs.org/#download. Некоторые основные инструкции по установке Node в различных средах предоставлены на вики-странице https://github.com/joyent/node/wiki/Installing-Node-via-package-manager. Я также настоятельно советую вам поискать самые свежие руководства по установке Node в вашей среде, поскольку Node является очень динамичной технологией.

Установка Node на платформе Linux (Ubuntu)

Перед установкой Node на платформу Linux нужно подготовить среду. Как отмечено в документации, предоставленной на вики-странице Node, сначала убедитесь в том, установлен интерпретатор языка Python, а затем установите libssl-dev, если планируете использовать протокол SSL/TLS. В зависимости от имеющегося у вас варианта установки Linux, интерпретатор языка Python уже может быть установлен. Если это не так, можете воспользоваться установщиком пакета для установки наиболее стабильной для вашей системы версии Python, коль скоро его версия 2.6 или 2.7 (требуется для последней версии Node).

Предполагается, что у читателей книги уже есть опыт работы с JavaScript и опыт обычной веб-разработки. Учитывая это, я опускаю некоторые предупреждения и подробности при описании того, что вам нужно делать для установки Node.

Как для Ubuntu, так и для Debian вам нужно также установить другие библиотеки. Используя инструментарий Advanced Packaging Tool (APT), доступный в большинстве систем Debian GNU/Linux, вы можете обеспечить установку нужных библиотек с помощью следующих команд:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install build-essential openssl libssl-dev pkg-config
```

Команда `update` всего лишь гарантирует, что индекс пакета на вашей системе находится в актуальном состоянии, а команда `upgrade` заменяет все устаревшие пакеты на новые. Установка всех необходимых пакетов осуществляется с помощью третьей командной строки. Любые имеющиеся зависимости пакетов друг от друга обрабатываются диспетчером пакетов.

После подготовки системы загрузите на вашу систему tarball-дистрибутив Node (сжатый архивный файл исходного кода). Чтобы получить доступ к tarball-дистрибутивам, я использую команду `wget`, хотя вы можете задействовать команду `curl`. Когда я работал над книгой, самой последней версией исходного кода для Node была 0.8.2:

```
wget http://nodejs.org/dist/v0.8.2/node-v0.8.2.tar.gz
```

После загрузки файл нужно разархивировать и распаковать:

```
tar -zxf node-v0.8.2.tar.gz
```

Теперь у вас есть каталог с именем `node-v0.6.18`. Перейдите в этот каталог и для компиляции и установки Node выполните следующие команды:

```
./configure
make
sudo make install
```

Если раньше вы никогда не пользовались в Unix утилитой `make`, эти три команды подготавливают *сборочный файл* на основе варианта установки и среды вашей системы, запускают подготовительную команду `make` для проверки зависимостей, а затем выполняют завершающую команду `make` с установкой. После обработки этих команд сервер Node должен быть установлен и доступен из командной строки.

Самым забавным в программировании является то, что двух абсолютно одинаковых систем не существует. Эта последовательность действий должна привести к успеху в большинстве сред Linux, но ключевым здесь является слово «должна».

Обратите внимание, что последней командой, необходимой для установки Node, является команда `sudo`. Для установки Node этим способом требуются root-привилегии (см. следующее примечание). Но вы можете установить Node локально, используя следующие команды, с помощью которых Node устанавливается в заданный локальный подкаталог:

```
mkdir ~/working
./configure --prefix=~/working
make
make install
echo 'export PATH=~/working/bin:${PATH}' >> ~/.bashrc
. ~/.bashrc
```

Итак, здесь можно увидеть, что установка ключа конфигурации `prefix` для указания пути в вашем исходном каталоге приводит к локальной установке Node. Нужно не забыть соответствующим образом обновить переменную окружения `PATH`.

Чтобы использовать команду sudo, вы должны обладать root-привилегиями, или привилегиями суперпользователя, и ваше имя пользователя должно входить в список в специальном файле, который находится в каталоге /etc/sudoers.

Хотя Node можно установить локально, если вы собираетесь опробовать этот подход для применения Node в общей среде хостинга, то тут есть над чем задуматься. Установка Node — это еще не все, что требуется для использования Node в среде. Вам еще нужны привилегии для компиляции приложения, а также для запуска приложений посредством определенных портов (например, порта 80). Большинство общих сред хостинга не позволят вам установить собственную версию Node.

Если только у вас нет на то особых причин, я рекомендую устанавливать Node с помощью команды `sudo`.

В свое время при запуске рассматриваемого в главе 4 диспетчера Node-пакетов (Node Package Manager, npm) с root-привилегиями возникали проблемы безопасности. Но на данный момент вопросы безопасности уже решены.

Совместное использование Node и WebMatrix на платформе Windows 7

Установить Node на платформу Windows можно с помощью весьма простой последовательности действий, описанной на упомянутой ранее вики-странице. Но если вы собираетесь использовать Node в среде Windows, то, скорее всего, это будет происходить в составе инфраструктуры Windows, предназначенной для веб-разработки.

В настоящее время Node можно использовать с двумя различными инфраструктурами Windows. Одной из них является новая облачная платформа Windows Azure, позволяющая разработчикам размещать приложения на удаленной службе

(называемой *облаком*). Инструкции по установке Windows Azure SDK для Node предоставляются компанией Microsoft, поэтому я не хочу давать описание этого процесса в данной главе (об SDK мы поговорим чуть позже).

Windows Azure SDK для Node и инструкции по установке можно найти по адресу https://www.windowsazure.com/en-us/develop/nodejs/.

Еще одним подходом к использованию Node на платформе Windows (в данном случае Windows 7) является включение Node в разработанное Microsoft средство WebMatrix, поддерживающее объединение технологий с открытым кодом для веб-разработчиков. Чтобы установить и запустить Node вместе с WebMatrix в Windows 7, нужно выполнить следующие действия:

1. Установите WebMatrix.
2. Установите Node, используя самый последний установочный пакет для Windows.
3. Установите iisnode для IIS Express 7.x, что позволит Node-приложениям работать с IIS под управлением Windows.
4. Установите Node-шаблоны для WebMatrix, что позволит упростить Node-разработку.

Установка WebMatrix производится, как показано на рис. 1.1, с помощью Microsoft Web Platform Installer. Это средство также установит программу IIS Express, являющуюся версией веб-сервера Microsoft для разработчика. Загрузить WebMatrix можно с веб-сайта http://www.microsoft.com/web/webmatrix/.

Когда установка WebMatrix будет завершена, установите самую последнюю версию Node, используя установщик, предоставляемый основным сайтом Node (http://nodejs.org/#download). Установка производится одним щелчком, и когда она завершится, вы сможете, как показано на рис. 1.2, открыть окно командной строки и набрать команду node, чтобы проверить работоспособность приложения.

Чтобы система Node работала с IIS в Windows, установите iisnode, исходный модуль IIS 7.x, созданный и поддерживаемый Томашем Янчуком (Tomasz Janczuk). Его установка, как и установка Node, проходит с помощью одного щелчка с использованием предварительно собранного пакета установки, доступного по адресу https://github.com/tjanczuk/iisnode. Имеются варианты установки для x86 и для x64, но для x64 придется провести обе установки.

В процессе установки iisnode может появиться окно, показанное на рис. 1.3 и сообщающее об отсутствии пакета Microsoft Visual C++ 2010 Redistributable Package. В таком случае нужно установить этот пакет, убедившись в том, что он соответствует версии устанавливаемого модуля iisnode, это будет либо пакет x86 (доступный по адресу http://www.microsoft.com/download/en/details.aspx?id=5555), либо пакет x64 (доступный по адресу http://www.microsoft.com/download/en/details.aspx?id=14632), либо тот и другой. После установки необходимого пакета повторите установку iisnode.

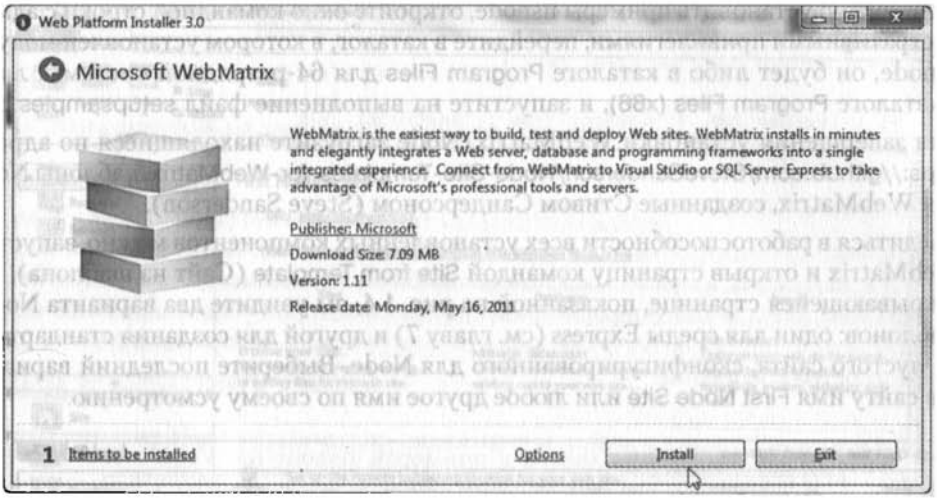


Рис. 1.1. Установка WebMatrix в Windows 7

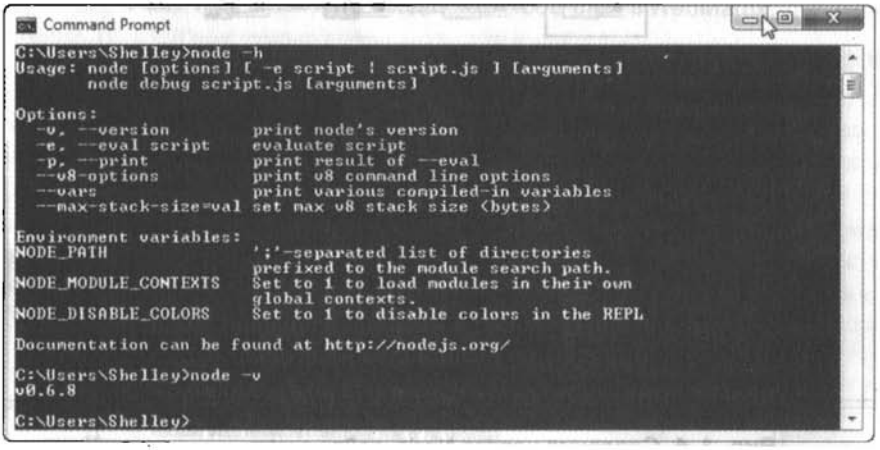


Рис. 1.2. Проверка с помощью окна командной строки правильности установки Node

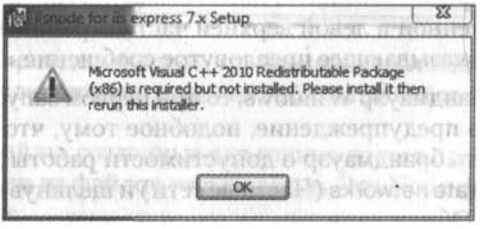


Рис. 1.3. Уведомление о необходимости установки свободно распространяемого пакета Microsoft Visual C++ 2010 Redistributable Package

Если нужно установить примеры iisnode, откройте окно командной строки с административными привилегиями, перейдите в каталог, в котором установлен модуль iisnode, он будет либо в каталоге Program Files для 64-разрядной системы, либо в каталоге Program Files (x86), и запустите на выполнение файл setupsamples.bat.

Для завершения установки WebMatrix/Node загрузите находящиеся по адресу https://github.com/SteveSanderson/Node-Site-Templates-for-WebMatrix шаблоны Node для WebMatrix, созданные Стивом Сандерсоном (Steve Sanderson).

Убедиться в работоспособности всех установленных компонентов можно, запустив WebMatrix и открыв страницу командой Site from Template (Сайт из шаблона). На открывающейся странице, показанной на рис. 1.4, вы увидите два варианта Node-шаблонов: один для среды Express (см. главу 7) и другой для создания стандартного, пустого сайта, сконфигурированного для Node. Выберите последний вариант, дав сайту имя First Node Site или любое другое имя по своему усмотрению.

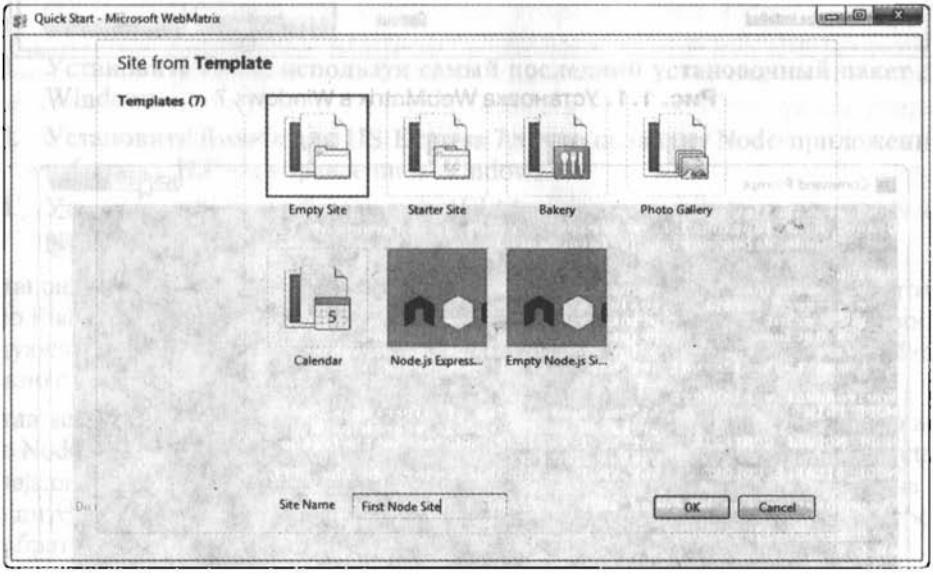


Рис. 1.4. Создание нового Node-сайта с использованием шаблона в WebMatrix

На рис. 1.5 показана среда WebMatrix после создания сайта. Щелкните на кнопке Run (Пуск), расположенной в левой верхней части страницы. В результате откроется окно браузера, показывающее пресловутое сообщение «Hello, world!».

Если у вас работает брандмауэр Windows, то при первом запуске Node-приложения может быть получено предупреждение, подобное тому, что показано на рис. 1.6. Вы должны оповестить брандмауэр о допустимости работы данного приложения, установив флажок Private networks (Частные сети) и щелкнув на кнопке Allow access (Разрешить доступ). Обмен данными на машине, используемой для разработки, следует ограничить рамками своей частной сети.

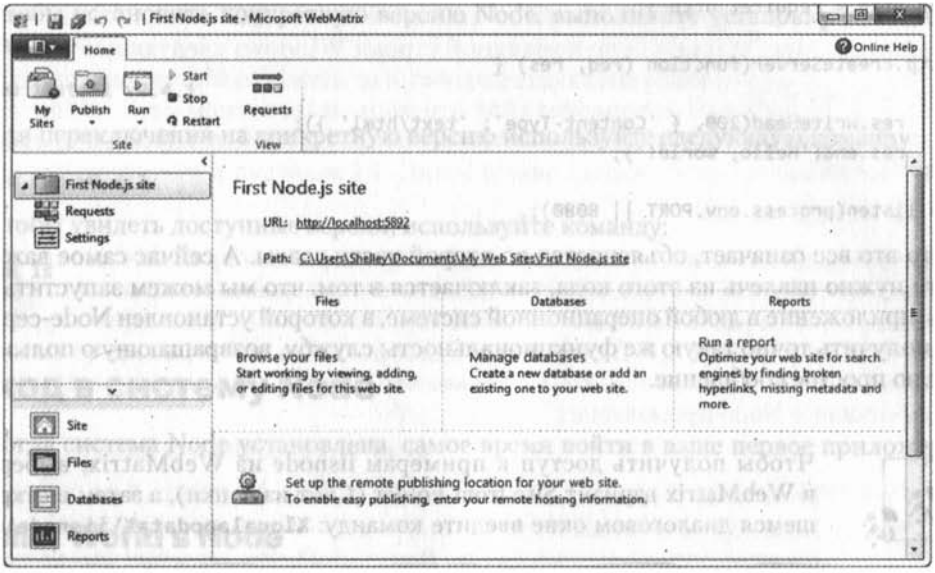


Рис. 1.5. Только что созданный Node-сайт в WebMatrix

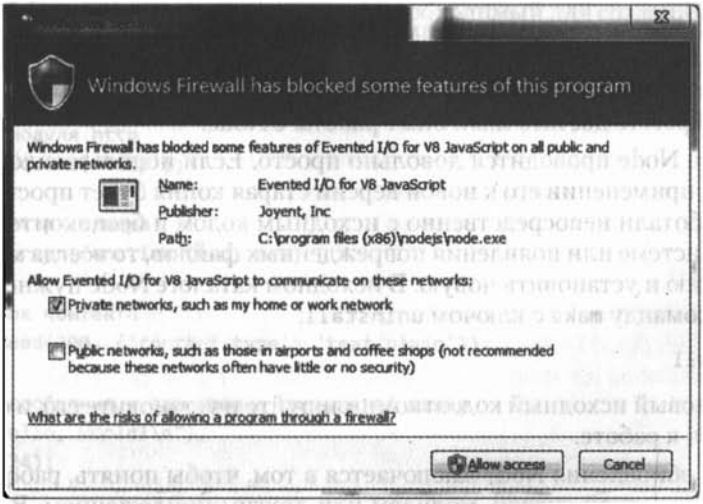


Рис. 1.6. Предупреждение о том, что брандмауэр Windows заблокировал Node-приложение, и вариант обхода этой блокировки

Если посмотреть на файлы, созданные для вашего нового проекта WebMatrix Node, то вы увидите, что один из файлов назван app.js. Это Node-файл, в котором содержится следующий код:

```
var http = require('http');

http.createServer(function (req, res) {

    res.writeHead(200, { 'Content-Type': 'text/html' });
    res.end('Hello, world!');

}).listen(process.env.PORT || 8080);
```

Что это все означает, объясняется во второй части главы. А сейчас самое важное, что нужно извлечь из этого кода, заключается в том, что мы можем запустить это же приложение в любой операционной системе, в которой установлен Node-сервер, и получить точно такую же функциональность: службу, возвращающую пользователю простое сообщение.

Чтобы получить доступ к примерам iisnode из WebMatrix, выберите в WebMatrix вариант Site from Folder (Сайт из папки), а затем в открывшемся диалоговом окне введите команду: **%localappdata%\iisnode\www**.

Обновление Node

Все стабильные выпуски Node имеют четные номера (текущий выпуск 0.8.x), а разрабатываемые выпуски — нечетные (текущий выпуск 0.9.x). Я рекомендую останавливать свой выбор только на стабильных выпусках, по крайней мере, пока вы не приобретете достаточный опыт работы с Node.

Обновление Node проводится довольно просто. Если использовался установщик пакета, при применении его к новой версии старая копия будет просто переписана. Если вы работали непосредственно с исходным кодом и беспокоитесь насчет беспорядка в системе или появления поврежденных файлов, то всегда можно удалить старую копию и установить новую. В исходном каталоге Node нужно будет просто запустить команду `make` с ключом `uninstall`:

```
make uninstall
```

Загрузите новый исходный код, откомпилируйте и установите его, и система снова будет готова к работе.

Сложность обновления Node заключается в том, чтобы понять, работает ли новая версия с конкретной средой, модулем или другим приложением. В большинстве случаев проблемы возникнуть не должны. Однако если они все же возникнут, есть приложение, которым можно воспользоваться для «переключения» версий Node. Это приложение называется диспетчером версий Node (Node Version Manager, Nvm).

Nvm можно загрузить с GitHub (https://github.com/creationix/nvm). Как и Node, Nvm нужно откомпилировать и установить в вашей системе.

Чтобы установить конкретную версию Node, выполняйте установку с помощью Nvm:

```
nvm install v0.4.1
```

Для переключения на конкретную версию используйте следующую команду:

```
nvm run v0.4.1
```

Чтобы увидеть доступные версии, используйте команду:

```
nvm ls
```

Вход в систему Node

Когда система Node установлена, самое время войти в ваше первое приложение.

Hello, World в Node

Как обычно, для тестирования любой новой среды разработки, языка или инструментария первым создаваемым приложением становится «Hello, World» — простая программа, выводящая приветствие всем, кто к ней обращается.

В листинге 1.1 показан весь исходный код, необходимый для создания приложения Hello, World в Node.

Листинг 1.1. Hello, World в Node

```
// загрузка модуля http
var http = require('http');

// создание http-сервера
http.createServer(function (req, res) {

  // заголовок контента
  res.writeHead(200, {'content-type': 'text/plain'});

  // запись сообщения и завершение сигнальной связи
  res.end("Hello, World!\n");
}).listen(8124);

console.log('Server running on 8124');
```

Код сохраняется в файле helloworld.js. С точки зрения функциональности на стороне сервера в этом Node-приложении нет ни большого объема кода, ни какой-либо таинственности, так что любому, даже тому, кто ничего не понимает в Node, под силу предположить, что произойдет. А самое лучшее в этом коде то, что он нам знаком, поскольку написан на языке JavaScript, который нам хорошо известен.

Чтобы запустить приложение, нужно набрать в командной строке Linux, в окне терминала MacOS или в окне командной строки Windows следующую команду:

```
node helloworld.js
```

После успешного запуска программы в командной строке будет выведена следующая строка:

```
Server running at 8124
```

Теперь обратитесь к сайту, используя любой браузер. Если приложение запущено на вашей локальной машине, нужно воспользоваться адресом `localhost:8124`. Если оно запущено на удаленной системе, нужно воспользоваться URL-адресом удаленного сайта с портом 8124. В результате будет показана веб-страница со словами «Hello, World!». Получается, что вы только что создали свое первое полноценное и работающее Node-приложение.

Если Node устанавливается в системе Fedora, следует иметь в виду, что Node переименовывается из-за конфликта имен с уже существующими там программными конструкциями. Дополнительные сведения можно найти по адресу http://nodejs.tchol.org/.

Поскольку после команды `node` не был указан амперсанд (`&`), предписывающий запуск приложения в фоновом режиме, приложение запускается, но не возвращает управление командной строке. Вы можете продолжать обращения к приложению, и при этом будут выводиться те же самые слова. Работа приложения продолжится до тех пор, пока вы не нажмете комбинацию клавиш Ctrl+C, чтбы прекратить его выполнение, или не прервете процесс каким-нибудь другим способом.

Если требуется запустить приложение в фоновом режиме на Linux-системе, нужно воспользоваться следующей командой:

```
node helloworld.js &
```

Но затем вам придется найти идентификатор процесса, используя команду `ps -ef`, и вручную прекратить выполнение нужного процесса — в данном случае процесса, которому присвоен идентификатор 3747, используя команду `kill`:

```
ps -ef | grep node
kill 3747
```

Работа процесса будет также прервана при выходе из окна терминала.

Вопросу создания надежного Node-приложения посвящена глава 16.

Запустить еще одно Node-приложение, слушающее тот же самый порт, не удастся: для одного и того же порта можно одновременно запустить только одно Node-приложение. Если у вас запущен сервер Apache, использующий порт 80, то Node-при-