

САМОЕ ПОЛНОЕ РУКОВОДСТВО
ДЛЯ ВЕБ-РАЗРАБОТЧИКОВ

NODE.JS



Москва
2025

УДК 004.43
ББК 32.973.26-018.1
Н85

*Последнюю версию этой книги на английском языке можно скачать с сайта:
<https://riptutorial.com/ebook/node-js>. Вы можете поделиться этим PDF-файлом
со всеми, кому он может быть полезен.*

*Книга «Learning Node.js» составлена на основе документации Stack Overflow
(<https://archive.org/details/documentation-dump.7z>), содержание которой написано мно-
гими трудолюбивыми людьми из Stack Overflow. Текстовые материалы публикуются
на условиях **Creative Commons BY-SA** (<https://creativecommons.org/licenses/by-sa/3.0/>).*

*Список авторов каждой главы приведен в разделе «Благодарности»
в конце этой книги. Изображения могут являться объектами авторского права
соответствующих владельцев, если не указано иное.*

Node.js. Самое полное руководство для веб-разработчиков в примерах
H85 от сообщества Stack Overflow. — Москва : Издательство АСТ, 2026. — 480 с. :
ил. — (Быстрый старт в программирование).

ISBN 978-5-17-162197-1.

Книга «Node.js. Самое полное руководство для веб-разработчиков в приме-
рах от сообщества Stack Overflow» представляет собой продвинутое учебное
пособие по работе на языке JavaScript в кроссплатформенной среде выполнения
Node.js — чрезвычайно популярной и важнейшей в области веб-разработки. вспомо-
гательный контент для программистов с любым опытом и уровнем знаний осно-
ван на практических примерах кодов, написанных экспертами широко известного
в профессиональных кругах сообщества Stack Overflow, где лучшие ИТ-специалисты
со всего мира делятся своими наработками, отвечая на многие технические вопросы.

Издание содержит подробное объяснение важнейших концепций и приемов
работы с Node.js, дополненное примерами их практического применения, что даст
возможность молодым разработчикам программного обеспечения быстро повысить
уровень своих компетенций, а опытным — найти актуальные решения для сложных
задач. В книге приведен мощный и современный инструментарий для создания
серверов, API, разработки веб-приложений, работы с базами данных, использования
WebSocket. Отдельные главы посвящены работе с менеджерами пакетов NPM и Yarn,
эффективной обработке событий и повышению производительности.

УДК 004.43
ББК 32.973.26-018.1

ISBN 978-5-17-162197-1

Перевод на русский язык: ООО «Интеджер»
Издание на русском языке: ООО «Издательство АСТ»

Содержание

| | |
|--|-----------|
| Глава 1. Начало работы с Node.js | 25 |
| Примечания..... | 25 |
| Версии..... | 25 |
| Примеры | 29 |
| HTTP-сервер «Hello World» | 29 |
| Пример «Hello World» в командной строке..... | 30 |
| Установка и запуск Node.js..... | 31 |
| Запуск программы Node.js | 33 |
| Развертывание вашего приложения в онлайн-среде | 33 |
| Отладка вашего Node.js-приложения | 34 |
| Отладка нативно | 34 |
| «Hello World» с использованием Express..... | 34 |
| «Hello World» с базовой маршрутизацией..... | 36 |
| TLS Socket: сервер и клиент | 37 |
| Как создать ключ и сертификат | 37 |
| Важно! | 38 |
| TLS Socket Server | 38 |
| Hello World в REPL | 40 |
| Core-модули | 40 |
| Все core-модули: краткий обзор | 41 |
| Как запустить базовый HTTPS веб-сервер?..... | 46 |
| Шаг 1: создание центра сертификации..... | 46 |
| Шаг 2: установка вашего сертификата в качестве корневого | 47 |
| Шаг 3: запуск вашего Node-сервера | 47 |
| Глава 2. Взаимодействие Arduino с Node.js | 49 |
| Введение | 49 |
| Примеры | 49 |
| Связь Node.js с Arduino через serialport..... | 49 |
| Код Node.js..... | 49 |
| Код Arduino..... | 50 |
| Запуск | 51 |

| | |
|---|-----------|
| Глава 3. async.js | 52 |
| Синтаксис | 52 |
| Примеры | 52 |
| Параллельное выполнение: многозадачность | 52 |
| Вызов <code>async.parallel()</code> с объектом | 53 |
| Обработка нескольких значений | 54 |
| Последовательное выполнение: независимый монотаскинг | 55 |
| Вызов <code>async.series()</code> с объектом | 55 |
| Водопад (waterfall): зависимый монотаскинг | 56 |
| <code>async.times</code> (для более удобной обработки циклов) | 57 |
| <code>async.each</code> (для эффективной обработки массива данных) | 57 |
| <code>async.series</code> (для обработки событий по одному) | 58 |
| Вывод | 59 |
| Глава 4. Async/Await | 60 |
| Введение | 60 |
| Примеры | 60 |
| Асинхронные функции с обработкой ошибок через <code>try-catch</code> | 60 |
| Сравнение Promises и Async/Await | 61 |
| Переход от обратных вызовов | 62 |
| Остановка выполнения на <code>await</code> | 63 |
| Глава 5. Асинхронное программирование | 64 |
| Введение | 64 |
| Синтаксис | 64 |
| Примеры | 64 |
| Функции обратного вызова в JavaScript | 64 |
| Синхронные функции обратного вызова | 64 |
| Асинхронные функции обратного вызова | 66 |
| Функции обратного вызова (callbacks) в Node.js | 67 |
| Пример кода | 68 |
| Вывод | 68 |
| Обработка ошибок в асинхронном коде | 69 |
| Обработчики событий | 69 |
| Домены | 69 |
| Callback hell | 70 |
| Как избежать callback hell | 71 |
| Нативные Promise (Промисы) | 71 |
| Глава 6. Автоматическая перезагрузка при изменениях | 73 |
| Примеры | 73 |
| Автоперезагрузка при изменении исходного кода с использованием <code>nodemon</code> | 73 |

| | |
|---|-----------|
| Установка nodemon глобально | 73 |
| Установка nodemon локально | 73 |
| Использование nodemon..... | 73 |
| Browsersync..... | 74 |
| Обзор..... | 74 |
| Установка | 74 |
| Пользователи Windows | 75 |
| Основное использование..... | 75 |
| Продвинутое использование | 75 |
| Grunt.js..... | 75 |
| Gulp.js..... | 75 |
| API..... | 76 |
| Глава 7. Избегайте ада обратных вызовов (callback hell) | 77 |
| Примеры | 77 |
| Модуль Async..... | 77 |
| Еще о модуле Async | 78 |
| Пример использования Async waterfall..... | 78 |
| Глава 8. Bluebird Promises (промисы) | 79 |
| Примеры | 79 |
| Преобразование библиотеки nodeback в Promises..... | 79 |
| Функциональные Promises | 79 |
| Корутины (coroutines)..... | 80 |
| Автоматическое освобождение ресурсов (Promise.using) | 80 |
| Последовательное выполнение..... | 80 |
| Глава 9. Преобразование функций обратного вызова в Promises (промисах) | 81 |
| Примеры | 81 |
| Промисификация функций обратного вызова..... | 81 |
| Ручная промисификация функции обратного вызова | 82 |
| Промисификация setTimeout | 83 |
| Глава 10. Интеграция с Cassandra..... | 84 |
| Примеры | 84 |
| Hello world | 84 |
| Глава 11. Интерфейс командной строки (CLI) | 86 |
| Синтаксис | 86 |
| Примеры | 86 |
| Опции командной строки..... | 86 |
| Глава 12. Взаимодействие «клиент — сервер» | 91 |
| Примеры | 91 |
| Использование с Express, jQuery and Jade | 91 |

| | |
|---|------------|
| Глава 13. Модуль Cluster | 94 |
| Синтаксис | 94 |
| Примечания..... | 94 |
| Примеры | 95 |
| Hello World | 95 |
| Пример кластера..... | 96 |
| Глава 14. Подключение к MongoDB | 97 |
| Введение | 97 |
| Синтаксис | 97 |
| Примеры | 97 |
| Простой пример подключения к MongoDB из Node.js..... | 97 |
| Простой способ подключения к MongoDB с использованием чистого Node.js..... | 98 |
| Глава 15. Создание библиотеки Node.js, поддерживающей как Promises, так и функции обратного вызова с первым аргументом-ошибкой | 99 |
| Введение | 99 |
| Примеры | 99 |
| Пример модуля и соответствующей программы с использованием Bluebird | 99 |
| Глава 16. Создание API с помощью Node.js | 102 |
| Примеры | 102 |
| GET API с использованием Express | 102 |
| POST API с использованием Express | 102 |
| Глава 17. Парсер CSV в Node.js | 104 |
| Введение | 104 |
| Примеры | 104 |
| Использование FS для чтения CSV..... | 104 |
| Глава 18. Работа с базой данных (MongoDB с Mongoose) | 105 |
| Примеры | 105 |
| Подключение Mongoose | 105 |
| Модель | 105 |
| Вставка данных..... | 106 |
| Чтение данных..... | 106 |
| Глава 19. Отладка Node.js-приложения | 108 |
| Встроенный отладчик Node.js и node inspector..... | 108 |
| Использование встроенного отладчика..... | 108 |
| Команды отладки..... | 108 |
| Использование встроенного инспектора Node | 110 |

| | |
|---|------------|
| Глава 20. Доставка HTML или других типов файлов | 111 |
| Синтаксис | 111 |
| Примеры | 111 |
| Доставка HTML по указанному пути..... | 111 |
| Структура папок..... | 111 |
| server.js..... | 111 |
| Глава 21. Внедрение зависимостей | 113 |
| Почему следует использовать внедрение зависимостей | 113 |
| Глава 22. Развертывание приложения Node.js без простоя..... | 114 |
| Примеры | 114 |
| Развертывание с использованием PM2 без простоя..... | 114 |
| Глава 23. Развертывание приложений Node.js в продакшене | 116 |
| Примеры | 116 |
| Установка NODE_ENV="production" | 116 |
| Флаги выполнения | 116 |
| Зависимости | 116 |
| Менеджер процессов PM2 | 118 |
| Развертывание с использованием PM2 | 118 |
| Полезные команды при работе с PM2..... | 119 |
| Forever..... | 119 |
| Использование разных свойств/конфигураций для разных окружений, таких как dev, qa, staging и т. д. | 120 |
| Использование преимуществ кластеров | 121 |
| Глава 24. ECMAScript 2015 (ES6) с Node.js | 123 |
| Примеры | 123 |
| Объявления const/let..... | 123 |
| Стрелочные функции | 123 |
| Пример стрелочной функции | 124 |
| Деструктуризация | 124 |
| Flow | 125 |
| Класс ES6 | 125 |
| Глава 25. Окружение | 127 |
| Примеры | 127 |
| Доступ к переменным окружения | 127 |
| Аргументы командной строки process.argv | 127 |
| Использование различных свойств/конфигураций для разных окружений, таких как dev, qa, staging и т. д. | 128 |
| Глава 26. Источники событий | 131 |
| Примечания..... | 131 |
| Примеры | 132 |

| | |
|--|------------|
| Аналитика HTTP через источник событий | 132 |
| Основы..... | 133 |
| Получение имен событий, на которые подписаны | 134 |
| Получение количества слушателей, зарегистрированных для определенного события | 134 |
| Глава 27. Event loop (цикл событий) | 136 |
| Введение | 136 |
| Примеры | 136 |
| Event loop на псевдокоде..... | 136 |
| Пример однопоточного HTTP-сервера без event loop | 136 |
| Пример многопоточного HTTP-сервера без event loop | 137 |
| Пример HTTP-сервера с event loop | 137 |
| Глава 28. Обработка исключений | 139 |
| Примеры | 139 |
| Обработка исключений в Node.js | 139 |
| Управление необработанными исключениями | 141 |
| Тихая обработка исключений..... | 141 |
| Возврат к начальному состоянию | 142 |
| Ошибки и Promises (промисы) | 143 |
| Глава 29. Выполнение файлов или команд с помощью дочерних процессов | 144 |
| Синтаксис | 144 |
| Замечания | 144 |
| Примеры | 144 |
| Создание нового процесса для выполнения команды | 144 |
| Создание оболочки для выполнения команды | 145 |
| Создание процесса для запуска исполняемого файла | 146 |
| Глава 30. Экспорт и использование модулей | 148 |
| Примечания..... | 148 |
| Примеры | 148 |
| Загрузка и использование модуля..... | 148 |
| Создание модуля hello-world.js | 149 |
| Признание недействительным кеша модуля | 151 |
| Создание собственных модулей | 151 |
| Каждый модуль загружается только один раз | 152 |
| Загрузка модуля из node_modules | 153 |
| Папка (директория) как модуль | 153 |
| Глава 31. Экспорт и импорт модулей в Node.js..... | 155 |
| Примеры | 155 |
| Использование простого модуля в Node.js | 155 |

| | |
|---|------------|
| Использование Imports в ES6 | 156 |
| Экспорт с использованием синтаксиса ES6 | 157 |
| Глава 32. Загрузка файлов | 158 |
| Примеры | 158 |
| Загрузка одного файла с использованием multer | 158 |
| Как фильтровать загрузку по расширению | 159 |
| Использование модуля formidable | 160 |
| Глава 33. Файловый ввод/вывод (Filesystem I/O) | 162 |
| Примечания | 162 |
| Примеры | 162 |
| Запись в файл с использованием writeFile или writeFileSync | 162 |
| Асинхронное чтение из файлов | 163 |
| С использованием кодировки | 163 |
| Без использования кодировки | 164 |
| Относительные пути | 164 |
| Список содержимого каталога с использованием readdir или readdirSync | 164 |
| Использование генератора | 165 |
| Синхронное чтение из файла | 165 |
| Чтение строки | 165 |
| Удаление файла с использованием unlink или unlinkSync | 166 |
| Чтение файла в Buffer с использованием потоков | 166 |
| Проверка прав доступа к файлу или директории | 167 |
| Асинхронно | 167 |
| Синхронно | 168 |
| Избегание состояния гонки (race condition) при создании или использовании существующего каталога | 168 |
| Проверка, существует ли файл или директория | 169 |
| Асинхронно | 169 |
| Синхронно | 169 |
| Клонирование файла с использованием потоков | 170 |
| Копирование файлов с использованием соединения потоков | 170 |
| Изменение содержимого текстового файла | 171 |
| Определение количества строк в текстовом файле | 171 |
| Чтение файла построчно | 172 |
| Глава 34. Начало работы с профилированием Node.js | 173 |
| Введение | 173 |
| Замечания | 173 |
| Примеры | 173 |
| Профилирование простого приложения на Node.js | 173 |

| | |
|--|------------|
| Глава 35. Хороший стиль кодирования | 176 |
| Примечания..... | 176 |
| Примеры | 176 |
| Базовая программа для регистрации | 176 |
| Глава 36. Корректное завершение..... | 181 |
| Примеры | 181 |
| Корректное завершение — SIGTERM | 181 |
| Глава 37. Grunt..... | 182 |
| Примечания..... | 182 |
| Примеры | 183 |
| Введение в Grunt.js..... | 183 |
| Установка плагинов Grunt | 184 |
| Глава 38. Совет по упрощению работы..... | 186 |
| Добавление новых расширений в require() | 186 |
| Глава 39. Обработка POST-запросов в Node.js..... | 188 |
| Замечания | 188 |
| Примеры | 189 |
| Глава 40. Как загружаются модули | 190 |
| Примеры | 190 |
| Режим глобальной установки..... | 190 |
| Загрузка модулей..... | 190 |
| Загрузка модуля из папки (директории) | 191 |
| Глава 41. HTTP | 192 |
| Примеры | 192 |
| HTTP-сервер | 192 |
| HTTP-клиент | 193 |
| Глава 42. Установка Node.js..... | 195 |
| Примеры | 195 |
| Установка Node.js на Ubuntu | 195 |
| Использование пакетного менеджера apt | 195 |
| Использование последней или конкретной версии (например LTS 6.x) непосредственно из nodesource..... | 195 |
| Установка Node.js на Windows..... | 196 |
| Использование Node Version Manager (nvm) | 196 |
| Установка Node.js из исходных кодов с помощью пакетного менеджера АРТ..... | 198 |
| Установка Node.js на Mac с помощью пакетного менеджера | 199 |
| Homebrew..... | 199 |
| Macports | 199 |

| | |
|---|------------|
| Установка с помощью MacOS X Installer | 200 |
| Проверка установки Node..... | 200 |
| Установка Node.js на Raspberry PI | 201 |
| Установка с помощью Node Version Manager в Fish Shell с Oh My Fish | 201 |
| Установка Node.js из исходников на CentOS, RHEL и Fedora..... | 202 |
| Установка Node.js с помощью n | 203 |
| Глава 43. Взаимодействие с консолью..... | 205 |
| Синтаксис | 205 |
| Примеры | 205 |
| Логирование..... | 205 |
| Модуль Console..... | 205 |
| console.log..... | 205 |
| console.error | 205 |
| console.time, console.timeEnd | 206 |
| Модуль Process | 206 |
| Форматирование | 206 |
| Общее | 207 |
| Цвет шрифта..... | 207 |
| Цвет фона..... | 207 |
| Глава 44. Постоянное выполнение Node.js-приложения | 208 |
| Примеры | 208 |
| Использование PM2 в качестве менеджера процессов | 208 |
| Запуск и остановка daemon Forever | 209 |
| Непрерывная работа с помощью nohup..... | 209 |
| Управление процессами с помощью Forever | 210 |
| Глава 45. Фреймворк Коа v2..... | 211 |
| Примеры | 211 |
| Пример Hello World..... | 211 |
| Обработка ошибок с использованием middleware (промежуточного ПО) | 211 |
| Глава 46. Lodash | 212 |
| Введение | 212 |
| Примеры | 212 |
| Фильтрация коллекции | 212 |
| Глава 47. Loorback — REST-соединитель..... | 213 |
| Введение | 213 |
| Примеры | 213 |
| Добавление веб-соединителя | 213 |
| Глава 48. metalsmith | 215 |
| Примеры | 215 |
| Создание простого блога..... | 215 |

| | |
|---|------------|
| Глава 49. Интеграция с MongoDB | 217 |
| Синтаксис | 217 |
| Параметры | 217 |
| Примеры | 218 |
| Подключение к MongoDB | 218 |
| Метод Connect() для MongoClient..... | 219 |
| Добавление документа..... | 219 |
| Метод коллекции insertOne() | 219 |
| Чтение коллекции | 220 |
| Метод коллекции find() | 220 |
| Обновление документа | 221 |
| Метод коллекции updateOne()..... | 221 |
| Удаление документа..... | 222 |
| Метод коллекции deleteOne()..... | 222 |
| Удаление нескольких документов..... | 222 |
| Метод коллекции deleteMany()..... | 223 |
| Простое подключение | 223 |
| Простое подключение с использованием промисов..... | 224 |
| Глава 50. Интеграция MongoDB для Node.js/Express.js..... | 225 |
| Введение | 225 |
| Примечания..... | 225 |
| Примеры | 225 |
| Установка MongoDB | 225 |
| Создание модели Mongoose | 226 |
| Запрос к вашей базе данных Mongo..... | 227 |
| Глава 51. Библиотека Mongoose | 228 |
| Примеры | 228 |
| Подключение к MongoDB с использованием Mongoose | 228 |
| Сохранение данных в MongoDB с использованием Mongoose и маршрутов Express.js..... | 229 |
| Настройка | 229 |
| Код..... | 229 |
| Использование..... | 230 |
| Поиск данных в MongoDB с использованием Mongoose и маршрутов Express.js..... | 231 |
| Настройка | 231 |
| Код..... | 231 |
| Использование..... | 232 |
| Поиск данных в MongoDB с использованием Mongoose, маршрутов Express.js и оператора \$text..... | 233 |
| Настройка | 233 |
| Код..... | 233 |
| Использование..... | 235 |

| | |
|---|------------|
| Индексы в моделях | 236 |
| Полезные функции Mongoose | 238 |
| Поиск данных в MongoDB с использованием промисов | 238 |
| Настройка | 238 |
| Код | 239 |
| Использование | 240 |
| Глава 52. Интеграция MSSQL | 242 |
| Введение | 242 |
| Примечания | 242 |
| Примеры | 243 |
| Подключение к SQL через модуль <code>pm mssql</code> | 243 |
| Глава 53. Многопоточность | 246 |
| Введение | 246 |
| Замечания | 246 |
| Примеры | 246 |
| Cluster | 246 |
| Child Process | 247 |
| Глава 54. Пул соединений MySQL | 249 |
| Примеры | 249 |
| Использование пула соединений без указания базы данных | 249 |
| Глава 55. Интеграция MySQL | 251 |
| Введение | 251 |
| Примеры | 251 |
| Запрос объекта подключения с параметрами | 251 |
| Использование пула соединений | 252 |
| Выполнение нескольких запросов одновременно | 252 |
| Подключение к MySQL | 252 |
| Запрос объекта соединения без параметров | 253 |
| Выполнение нескольких запросов с одним соединением из пула | 253 |
| Возврат запроса при возникновении ошибки | 254 |
| Экспорт пула соединений | 255 |
| Глава 56. N-API | 256 |
| Введение | 256 |
| Примеры | 256 |
| Приветствие в N-API | 256 |
| Глава 57. Локализация Node.js | 259 |
| Введение | 259 |
| Примеры | 259 |
| Использование модуля <code>i18n</code> для поддержания локализации в приложении на Node.js | 259 |

| | |
|---|------------|
| Глава 58. Сервер Node без фреймворка | 261 |
| Примечания..... | 261 |
| Примеры | 261 |
| Сервер на Node без фреймворка | 261 |
| Решение проблем с CORS..... | 262 |
| Глава 59. Node.js (express.js) с Angular.js | 264 |
| Введение | 264 |
| Примеры | 264 |
| Создание проекта | 264 |
| Как создать каркас проекта на express?..... | 264 |
| Как работает Express? | 265 |
| Установка Pug и обновление шаблонизатора Express | 266 |
| Как AngularJS вписывается во все это?..... | 266 |
| Глава 60. Node.js и MongoDB..... | 269 |
| Примечания..... | 269 |
| Примеры | 269 |
| Подключение к базе данных..... | 269 |
| Установка Mongoose | 269 |
| Создание новой коллекции | 270 |
| Вставка документов..... | 271 |
| Чтение..... | 271 |
| Обновление | 272 |
| Update()..... | 273 |
| UpdateOne..... | 273 |
| UpdateMany..... | 273 |
| ReplaceOne..... | 274 |
| Удаление | 274 |
| Глава 61. Архитектура и внутренняя работа Node.js | 275 |
| Примеры | 275 |
| Node.js — «под капотом»..... | 275 |
| Node.js — «в движении»..... | 275 |
| Глава 62. Код Node.js для STDIN и STDOUT без использования библиотек ... | 276 |
| Введение | 276 |
| Примеры | 276 |
| Программа..... | 276 |
| Глава 63. Основы проектирования Node.js | 277 |
| Примеры | 277 |
| Малое ядро, малые модули..... | 277 |
| Шаблон Reactor..... | 277 |
| Неблокирующий движок ввода-вывода Node.js — libuv | 277 |

| | |
|---|------------|
| Глава 64. Управление ошибками в Node.js | 278 |
| Введение | 278 |
| Примеры | 278 |
| Создание объекта ошибки (Error) | 278 |
| Выброс ошибки | 279 |
| Блок try...catch | 279 |
| Глава 65. Производительность Node.js | 281 |
| Примеры | 281 |
| Цикл событий | 281 |
| Пример блокирующей операции | 281 |
| Пример неблокирующей операции ввода-вывода (IO-bound) | 281 |
| Рекомендации по производительности | 282 |
| Увеличение maxSockets..... | 283 |
| Основы | 283 |
| Установка собственного агента | 283 |
| Полное отключение пула сокетов | 283 |
| Подводные камни..... | 283 |
| Включение gzip | 283 |
| Глава 66. Новые функции и улучшения Node.js v6..... | 285 |
| Введение | 285 |
| Примеры | 285 |
| Параметры по умолчанию для функций | 285 |
| Остаточные параметры | 285 |
| Оператор расширения (Spread Operator)..... | 286 |
| Стрелочные функции (Arrow Functions)..... | 286 |
| this в стрелочных функциях..... | 286 |
| Глава 67. Node.js с CORS | 289 |
| Примеры | 289 |
| Включение CORS в express.js | 289 |
| Глава 68. Node.js с ES6..... | 291 |
| Введение | 291 |
| Примеры | 291 |
| Поддержка ES6 в Node.js и создание проекта с Babel..... | 291 |
| Использование ES6 в вашем Node.js-приложении | 293 |
| Предварительные условия..... | 293 |
| Глава 69. Использование Node.js с Oracle..... | 297 |
| Примеры | 297 |
| Подключение к Oracle DB..... | 297 |
| Выполнение запроса на объекте соединения без параметров | 298 |
| Использование локального модуля для упрощения запросов..... | 299 |