



ОГЛАВЛЕНИЕ

| | |
|--|-----------|
| Об авторах | 11 |
| Благодарности | 12 |
| О рецензентах..... | 14 |
| Предисловие | 15 |
| О чем рассказывается в этой книге | 16 |
| Что потребуется при чтении этой книги | 17 |
| Кому адресована эта книга..... | 18 |
| Соглашения..... | 18 |
| Отзывы и пожелания | 19 |
| Поддержка клиентов | 19 |
| Загружаемые примеры программного кода | 19 |
| Ошибки и опечатки | 20 |
| Нарушение авторских прав | 20 |
| Вопросы | 20 |
| Глава 1. Дзен Angular..... | 21 |
| Знакомьтесь, AngularJS..... | 22 |
| Общие сведения о фреймворке | 22 |
| Найдите свой путь в проект | 23 |
| Сообщество..... | 23 |
| Обучающие ресурсы в Интернете | 23 |
| Библиотеки и расширения | 24 |
| Инструменты..... | 24 |
| Batarang..... | 24 |
| Plunker и jsFiddle | 25 |
| Расширения и дополнения для IDE | 25 |
| Ускоренное введение в AngularJS | 25 |
| Hello World – пример приложения на AngularJS | 25 |
| Двухнаправленное связывание данных | 27 |
| Шаблон MVC в AngularJS | 27 |
| С высоты птичьего полета | 28 |
| Подробнее о контекстах..... | 30 |
| Представление | 37 |
| Модули и внедрение зависимостей | 42 |
| AngularJS и остальной мир | 56 |
| jQuery и AngularJS | 58 |
| Взгляд в будущее | 59 |
| В заключение | 60 |

| | |
|--|------------|
| Глава 2. Сборка и тестирование | 62 |
| Введение в пример приложения..... | 63 |
| Область применения..... | 63 |
| Стек технологий..... | 65 |
| Хранилище данных | 65 |
| Система сборки | 68 |
| Принципы построения систем сборки..... | 69 |
| Инструменты | 71 |
| Организация файлов и каталогов | 73 |
| Каталоги верхнего уровня..... | 74 |
| Каталог с исходным кодом..... | 75 |
| Соглашения по именованию файлов | 79 |
| Модули и файлы AngularJS | 80 |
| Один файл, один модуль..... | 80 |
| Внутри модуля | 81 |
| Автоматическое тестирование | 84 |
| Модульные тесты..... | 86 |
| Интеграционные тесты | 93 |
| В заключение | 97 |
| Глава 3. Взаимодействие с сервером..... | 99 |
| Выполнение запросов XHR и JSONP с помощью \$http | 99 |
| Модель данных и адреса URL в MongoLab | 100 |
| Краткий обзор \$http | 100 |
| Ограничения политики общего происхождения | 104 |
| Promise API и служба \$q..... | 109 |
| Получение отложенных результатов с помощью службы \$q..... | 110 |
| Интеграция службы \$q в AngularJS..... | 119 |
| Promise API и служба \$http..... | 120 |
| Взаимодействие с конечными точками RESTful | 121 |
| Служба \$resource | 122 |
| Взаимодействия с веб-службами REST с помощью \$http | 128 |
| Дополнительные возможности \$http | 132 |
| Обработка ответов..... | 132 |
| Тестирование кода, осуществляющего взаимодействия с помощью \$http..... | 133 |
| В заключение | 136 |
| Глава 4. Отображение и форматирование данных ... | 137 |
| Знакомство с директивами | 137 |
| Отображение результатов вычисления выражений | 138 |
| Директива интерполяции..... | 138 |
| Отображение значений с помощью ngBind..... | 139 |
| Включение разметки HTML в выражения..... | 139 |
| Отображение по условию | 141 |
| Включение блоков содержимого по условию | 142 |
| Отображение коллекций с помощью директивы ngRepeat | 143 |
| Знакомство с директивой ngRepeat | 143 |

| | |
|---|------------|
| Специальные переменные | 144 |
| Итерации по свойствам объекта | 145 |
| Приемы использования директивы ngRepeat | 145 |
| Обработчики событий DOM | 149 |
| Увеличение эффективности с помощью шаблонов на основе DOM | 150 |
| Избыточный синтаксис | 150 |
| Применение директивы ngRepeat к множеству элементов DOM | 151 |
| Элементы и атрибуты не могут изменяться во время выполнения | 152 |
| Нестандартные элементы HTML и старые версии IE | 153 |
| Преобразование моделей с помощью фильтров | 153 |
| Применение встроенных фильтров | 154 |
| Создание собственных фильтров – реализация постраничного вывода | 161 |
| Доступ к фильтрам из кода на JavaScript | 163 |
| Правила использования фильтров | 164 |
| В заключение | 168 |
| Глава 5. Создание улучшенных форм | 169 |
| Сравнение традиционных форм с формами AngularJS | 169 |
| Введение в директиву ngModel | 171 |
| Создание формы с информацией о пользователе | 172 |
| Директивы ввода | 173 |
| Добавление проверки обязательного наличия значения | 174 |
| Текстовые элементы ввода | 174 |
| Кнопки-флажки | 175 |
| Радиокнопки | 175 |
| Элементы выбора из списка | 176 |
| Использование скрытых полей ввода | 181 |
| Устройство механизма связывания данных в ngModel | 182 |
| ngModelController | 182 |
| Проверка форм в AngularJS | 184 |
| ngFormController | 184 |
| Добавление динамического поведения в форму с информацией о пользователе | 185 |
| Вывод сообщений об ошибках | 186 |
| Отключение процедуры проверки, встроенной в браузер | 188 |
| Вложенные формы | 188 |
| Вложенные формы как компоненты многократного пользования | 189 |
| Повторение вложенных форм | 189 |
| Проверка повторяющихся полей ввода | 191 |
| Отправка традиционной формы HTML | 192 |
| Непосредственная отправка форм на сервер | 192 |
| Обработка события отправки формы | 193 |
| Сброс формы в исходное состояние | 194 |
| В заключение | 195 |
| Глава 6. Организация навигации | 196 |

| | |
|---|-----|
| Адреса URL в одностраничных веб-приложениях | 197 |
| Адреса URL с решеткой до появления HTML5 | 198 |
| HTML5 и интерфейс истории посещений | 199 |
| Служба \$location | 200 |
| Знакомство с интерфейсом службы \$location и адресами URL | 201 |
| Адреса фрагментов, навигация внутри страницы и \$anchorScroll | 202 |
| Настройка режима HTML5 интерпретации адресов URL | 203 |
| Навигация вручную с помощью службы \$location | 205 |
| Служба \$route | 208 |
| Определение основных маршрутов | 208 |
| Гибкое сопоставление маршрутов | 210 |
| Повторное использование шаблонов разметки с разными контроллерами | 212 |
| Предотвращение «мерцания» пользовательского интерфейса при изменении маршрута | 213 |
| Предотвращение изменения маршрута | 215 |
| Ограничения службы \$route | 216 |
| Один маршрут соответствует одной области на экране | 217 |
| Распространенные приемы использования, советы и рекомендации | 220 |
| Обработка ссылок..... | 220 |
| Организация определений маршрутов | 222 |
| В заключение | 224 |

Глава 7. Безопасность приложений 226

| | |
|---|-----|
| Аутентификация и авторизация на стороне сервера..... | 227 |
| Обработка неавторизованного доступа | 227 |
| Реализация прикладного интерфейса аутентификации на стороне сервера | 228 |
| Безопасность шаблонов разметки | 228 |
| Противостояние нападениям | 229 |
| Предотвращение перехвата cookie («атака через посредника») | 230 |
| Предотвращение нападений вида «межсайтовый скриптинг» | 231 |
| Предотвращение внедрения данных в формате JSON | 233 |
| Предотвращение подделки межсайтовых запросов | 234 |
| Обеспечение безопасности на стороне клиента | 235 |
| Служба security | 236 |
| Отображение формы аутентификации | 236 |
| Создание меню и панелей инструментов с поддержкой системы безопасности..... | 238 |
| Поддержка аутентификации и авторизации на стороне клиента | 240 |
| Обработка ошибок авторизации | 241 |
| Перехват ответов | 242 |
| Создание службы securityInterceptor | 242 |
| Создание службы securityRetryQueue | 244 |
| Предотвращение переходов по защищенным маршрутам..... | 246 |
| Использование функций в свойстве resolve маршрутов | 247 |
| Создание службы authorization | 248 |

| | |
|---|------------|
| В заключение | 250 |
| Глава 8. Создание собственных директив | 251 |
| Что такое директива AngularJS? | 251 |
| Встроенные директивы | 252 |
| Использование директив в разметке HTML | 253 |
| Тестирование директив | 255 |
| Определение директивы | 257 |
| Оформление кнопок с помощью директив..... | 258 |
| Создание директивы button | 258 |
| Директивы-виджеты..... | 261 |
| Создание директивы постраничного просмотра | 261 |
| Тест для директивы постраничного просмотра списков | 262 |
| Использование шаблонов с разметкой HTML в директивах | 263 |
| Изолирование директивы от родительского контекста | 264 |
| Реализация виджета | 267 |
| Добавление в директиву функции обратного вызова selectPage | 268 |
| Создание директивы проверки..... | 269 |
| Внедрение контроллера другой директивы | 270 |
| Взаимодействие с контроллером ngModelController | 271 |
| Тестирование директивы проверки | 272 |
| Реализация директивы проверки | 274 |
| Асинхронная проверка модели..... | 275 |
| Имитация службы Users | 275 |
| Тестирование директивы асинхронной проверки | 276 |
| Реализация директивы асинхронной проверки | 278 |
| Директива-обертка для виджета выбора даты из библиотеки jQueryUI | 279 |
| Тестирование директив-оберток..... | 281 |
| Реализация директивы datePicker | 282 |
| В заключение | 283 |
| Глава 9. Создание продвинутых директив | 285 |
| Включение | 285 |
| Использование включения в директивах..... | 286 |
| Включение в директивах с изолированным контекстом..... | 286 |
| Директива вывода предупреждения на основе приема включения..... | 286 |
| Контекст включения | 288 |
| Создание и использование функций включения | 291 |
| Создание функции включения с помощью службы \$compile | 291 |
| Использование функций включения в директивах | 292 |
| Создание директивы if, использующей включение | 294 |
| Контроллеры директив..... | 296 |
| Внедрение специальных зависимостей в контроллеры директив..... | 297 |
| Создание директивы постраничного просмотра на основе контроллера | 298 |
| Различия между контроллерами директив и функциями связывания | 299 |
| Комплект директив виджета «аккордеон»..... | 301 |
| Управление процессом компиляции..... | 305 |

| | |
|---|------------|
| Создание директивы field | 306 |
| Использование службы \$interpolate | 308 |
| Динамическая загрузка шаблонов | 310 |
| Настройка шаблона директивы field | 310 |
| В заключение | 312 |
| Глава 10. Создание интернациональных веб-приложений..... | 313 |
| Использование национальных наборов символов и настроек..... | 314 |
| Модули с национальными настройками | 314 |
| Использование доступных национальных настроек..... | 315 |
| Поддержка переводов..... | 317 |
| Перевод строк в шаблонах AngularJS | 318 |
| Перевод строк в коде JavaScript..... | 321 |
| Шаблоны проектирования, советы и рекомендации..... | 322 |
| Инициализация приложений с учетом выбранных национальных настроек | 322 |
| Переключение между национальными настройками | 325 |
| Нестандартное форматирование дат, чисел и валют | 326 |
| В заключение | 328 |
| Глава 11. Создание надежных веб-приложений на основе AngularJS | 330 |
| Внутренние механизмы AngularJS | 331 |
| Это не механизм строковых шаблонов | 331 |
| Настройка производительности – определить требования, измерить, настроить и повторить | 343 |
| Настройка производительности приложений на основе AngularJS | 345 |
| Оптимизация использования процессора..... | 346 |
| Оптимизация потребления памяти | 356 |
| Директива ng-repeat | 358 |
| В заключение | 360 |
| Глава 12. Подготовка и развертывание веб-приложений на основе AngularJS..... | 362 |
| Повышение производительности сетевых операций | 363 |
| Минификация статических ресурсов..... | 363 |
| Предварительная загрузка шаблонов | 368 |
| Оптимизация начальной страницы | 373 |
| Избегайте отображения шаблонов в необработанном виде | 373 |
| AngularJS и подключение прикладных сценариев..... | 376 |
| Поддержка браузеров | 379 |
| Поддержка Internet Explorer..... | 380 |
| В заключение | 381 |
| Предметный указатель | 383 |



ОБ АВТОРАХ

Павел Козловский (Pawel Kozlowski) обладает более чем 15-летним профессиональным опытом веб-разработки и использования самых разных веб-технологий, языков и платформ. Он одинаково хорошо разбирается с особенностями разработки как клиентских, так и серверных компонентов веб-приложений и всегда старается использовать самые производительные инструменты и приемы.

Павел является убежденным сторонником свободного, открытого программного обеспечения. Он с большим энтузиазмом участвует в работе над проектом AngularJS и ведет активную деятельность в сообществе пользователей AngularJS. Он также является одним из разработчиков Angular UI – комплекта компонентов, сопутствующего фреймворку AngularJS, где занимается разработкой директив Twitter Bootstrap для AngularJS.

В свободное от программирования время Павел занимается популяризацией AngularJS на различных конференциях и встречах.

Питер Бэкон Дарвин (Peter Bacon Darwin) занимается программированием уже больше двух десятилетий. Ему довелось работать с фреймворком .NET еще до того, как он был выпущен; Питер принимал участие в разработке IronRuby¹ и работал консультантом по информационным технологиям в Avanade и IMGROUP, пока не оставил их, чтобы присматривать за своими детьми и заниматься независимой разработкой.

Питер является заметной фигурой в сообществе AngularJS. Он недавно присоединился к команде разработки AngularJS в Google, как внештатный разработчик, и является одним из основателей проекта AngularUI. Он часто выступает с информацией о AngularJS на конференциях Devoxx UK и многочисленных встречах в Лондоне. Он так же ведет учебные курсы в AngularJS. Как консультант, он в первую очередь стремится помочь компаниям наиболее оптимально использовать фреймворк AngularJS.

1 Реализация языка Ruby для .NET. – Прим. перев.



БЛАГОДАРНОСТИ

От Павла Козловского

Даже не верится, что на протяжении последних месяцев, пока я работал над книгой, мне повезло сотрудничать с такими замечательными людьми. Эта книга едва ли появилась бы на свет без помощи и напряженного труда всех вас. Спасибо вам!

В первую очередь я хотел бы сказать «Спасибо!» всем участникам проекта AngularJS в компании Google. Вы – команда мечты, работающая над развитием удивительным фреймворка. Не останавливайтесь на достигнутом! Отдельную благодарность я хочу выразить Бреду Гину (Brad Green), Мишко Хеври (Miško Hevery), Игорю Минару (Igor Minar) и Войту Джину (Vojta Jina). Спасибо Бреду, что свел меня и Питера с издателем и побудил нас написать эту книгу. Спасибо Мишко за рецензирование нашей книги и за проявленное долготерпение, когда мы приставали с глупыми расспросами об AngularJS. Спасибо Игорю за неустанную поддержку и бесконечный поток советов, которые помогли сделать эту книгу лучше. Мы получили массу удовольствия от работы с вами!

Я хотел бы также выразить свою благодарность всему сообществу пользователей AngularJS, особенно тем, кто активно помогает в списках рассылки и на других форумах. Я не могу перечислить всех вас по именам, но ваши глубокомысленные вопросы служили для нас источником вдохновения при работе над этой книгой. Энергичное и доброжелательное сообщество, стоящее за фреймворком AngularJS, – это одна из причин, почему фреймворк получился таким удачным.

Спасибо всем сотрудникам издательства Packt Publishing: Роксане Хамбатта (Rukhsana Khambatta), Даяну Хаймсу (Dayan Hyames) и Аршаду Сопаривала (Arshad Sopariwala). Вы сделали весь процесс создания и публикации книги чрезвычайно легким и гладким. Спасибо вам!

Хочу также сказать спасибо моим коллегам из Amadeus, где я вплотную познакомился с разработкой клиентских компонентов веб-приложений. Прежде всего моим руководителям, Бертрану Ла-

порте (Bertrand Laporte) и Бруно Шабри (Bruno Chabrier). Спасибо Бертрану, что ввел меня в мир разработки клиентских компонентов и за поддержку в решении написать эту книгу. Спасибо Бруно, что позволил мне работать неполный рабочий день и сфокусироваться на этом проекте. Спасибо вам обоим за ваше великодушие. Спасибо Юлиану Дескоттеу (Julian Descottes) и Коринн Крич (Corinne Krich) за рецензирование первых рукописей книги и за весьма ценные отзывы.

Очень, очень большое спасибо Питеру, согласившемуся стать моим соавтором. Питер, я наслаждался каждой минутой работы с тобой! Я и мечтать не мог о лучшем соавторе.

Наконец, и что особенно важно для меня, я хочу поблагодарить мою невесту Аню. Без твоей поддержки и терпения я не смог бы даже приступить к работе над этой книгой.

От Питера Бэкона Дарвина

Я хочу поблагодарить разработчиков из компании Google, давших нам AngularJS, и особенно тех, с кого все это началось: Мишко Хеври (Miško Hevery), Игоря Минара (Igor Minar), Бреда Грина (Brad Green) и Войта Джина (Vojta Jina). Они стали неиссякаемым источником энтузиазма. Спасибо моему соавтору, Павлу, ставшему основной движущей силой. Он придумал структуру этой книги и написал большую ее часть, а еще он – отличный парень, с которым было приятно работать. Спасибо удивительно активному сообществу, сплотившемуся вокруг AngularJS за столь короткое время, и особое спасибо участникам проекта AngularUI. Наконец, я должен сказать, что не смог бы закончить эту книгу без любви и поддержки моей супруги Келин (Kelyn) и моих детей, Лили (Lily) и Захария (Zachary).



О РЕЦЕНЗЕНТАХ

Стефан Биссон (Stephane Bisson) – работал программистом в консалтинговой компании ThoughtWorks. В настоящее время живет в Торонто (Канада). Участвовал в разработке нескольких полнофункциональных веб-приложений для медицинских и финансовых учреждений, и предприятий обрабатывающей промышленности.

Мишко Хеври (Miško Hevery) – работал консультантом по внедрению методов гибкой разработки в Google, где отвечал за обучение сотрудников Google методам автоматизации тестирования. Его старания позволили Google чаще выпускать новые версии веб-приложений, с неизменно высоким качеством. Прежде ему приходилось работать в Adobe, Sun Microsystems, Intel и Xerox, где он стал экспертом в разработке веб-приложений с применением таких технологий, как Java, JavaScript, Flex и ActionScript. Мишко активно участвует в жизни сообщества open source и является автором нескольких проектов с открытым исходным кодом, наиболее примечательным из которых является AngularJS (<http://angularjs.org>).

Ли Ховард (Lee Howard) закончил Аппалачский государственный университет (США) по направлению информационных технологий и в настоящее время занимает должность ведущего программиста-аналитика в северо-западном отделении просвещения (Northwest Area Health Education Center) Баптистского медицинского центра Уэйк Форест (Wake Forest Baptist Health Medical Center) в городе Уинстон-Сейлем (Северная Каролина, США). Им было разработано множество различных веб-приложений, упрощающих создание, регистрацию и проведение очных и электронных образовательных курсов для Northwest АНЕС. Им также было создано мобильное приложение CreditTrakr для устройств на базе iOS, позволяющее врачам и другим медицинским работникам следить за своим учебным расписанием с помощью мобильных устройств.



ПРЕДИСЛОВИЕ

AngularJS – относительно новый фреймворк MVC на JavaScript, но уже нашедший практическое применение. В нем используются новейшие подходы к обработке шаблонов и применяется прием двунаправленного связывания данных, обеспечивающие широчайшие возможности и простоту использования фреймворка. Разработчики постоянно сообщают о существенном сокращении объема кода, необходимого для реализации приложений на основе AngularJS, в сравнении с другими подходами.

Фреймворк AngularJS – выдающийся образец инженерной мысли. С сильным упором на тестирование и высокое качество кода, он способствует использованию передового опыта, накопленного всей экосистемой JavaScript. Не удивительно, что многие разработчики, привлеченные высоким качеством кода и новизной технологий, образовали весьма активное и доброжелательное сообщество вокруг AngularJS, способствующее росту популярности фреймворка.

С ростом популярности AngularJS, все больше и больше разработчиков будут использовать его в сложных проектах. Но, как это часто бывает, вы вскоре столкнетесь с проблемами, которые не освещены в стандартной документации или в простых примерах, которые можно найти в Интернете. Как и многие другие технологии, фреймворк AngularJS имеет свой комплекс идиом, шаблонов и приемов, раскрытых сообществом на основе накопленного коллективного опыта.

И здесь вам на помощь придет эта книга – она постарается показать вам, как писать нетривиальные приложения на основе фреймворка AngularJS. В место описания особенностей работы фреймворка, эта книга концентрируется на описании особенностей использования AngularJS в сложных веб-приложениях. Здесь вы найдете ответы на многие вопросы, которые часто задаются в сообществе AngularJS.

Проще говоря, эта книга написана разработчиками приложений, для разработчиков приложений, на основе вопросов, часто задаваемых разработчиками приложений. Из этой книги вы узнаете:

- как создавать полноценные, надежные приложения с использованием имеющихся служб и директив AngularJS;
- как расширять AngularJS (создавать собственные директивы, службы, фильтры), если стандартных его возможностей оказывается недостаточно;
- как настраивать проекты приложений на основе AngularJS (организация кода, сборка, тестирование, профилирование).

О чем рассказывается в этой книге

Глава 1, «Дзен Angular», служит введением в проект и фреймворк AngularJS. В этой главе вы познакомитесь с философией проекта, основными понятиями и базовыми составляющими.

Глава 2, «Сборка и тестирование», закладывает фундамент примера приложения, используемого в качестве иллюстрации на протяжении всей книги. Она знакомит с предметной областью и охватывает такие темы, как сборка и тестирование систем.

Глава 3, «Взаимодействие с серверными компонентами», рассказывает, как организовать получение данных со стороны сервера и как эффективно передавать их пользовательскому интерфейсу, действующему под управлением AngularJS. В этой главе детально будет расматриваться доступный прикладной интерфейс API.

Глава 4, «Отображение и форматирование данных», предполагает, что отображаемые данные уже приняты от серверной части веб-приложения, и демонстрирует, как можно организовать отображение этих данных в пользовательском интерфейсе. Здесь обсуждается применение директив AngularJS для отображения пользовательского интерфейса и фильтров для форматирования данных.

Глава 5, «Создание сложных форм», иллюстрирует, как дать пользователям возможность манипулировать данными в формах и знакомит с различными типами полей ввода. Она охватывает различные способы ввода, поддерживаемые фреймворком AngularJS, и подробно рассказывает о приемах проверки данных в формах.

Глава 6, «Навигация», покажет, как организовать отдельные экраны в приложениях с поддержкой навигации. Глава начинается с описания роли адресов URL в одностраничных веб-приложениях и знакомит читателя с ключевыми службами AngularJS, обеспечивающими управление адресами URL и навигацией.

Глава 7, «Безопасность приложений», погрузит вас в детали обеспечения безопасности одностраничных веб-приложений, написанных с использованием AngularJS. В ней описываются понятия и приемы аутентификации и авторизации пользователей.

Глава 8, «Создание собственных директив», служит введением в разработку одних из самых интереснейших компонентов AngularJS: директив. Она познакомит читателя со структурой типовых директив и продемонстрирует приемы их тестирования.

Глава 9, «Создание сложных директив», основана на главе 8, «Создание собственных директив» и охватывает некоторые более сложные темы. Она наполнена практическими примерами действующих директив, ясно иллюстрирующими приемы их создания.

Глава 10, «Создание интернационализированных веб-приложений на основе AngularJS», описывает приемы интернационализации приложений на базе AngularJS. Охватывает темы, включающие приемы перевода шаблонов а также управления настройками, зависящими от национальных установок.

Глава 11, «Создание надежных веб-приложений на основе AngularJS» концентрируется на нефункциональных требованиях, предъявляемых к веб-приложениям. Она раскрывает закулисные механизмы AngularJS и знакомит читателя с характеристиками производительности. Хорошее знание внутреннего устройства AngularJS поможет вам избежать ловушек, связанных со снижением производительности приложений.

Глава 12, «Упаковка и развертывание веб-приложений на основе AngularJS» проведет вас через процедуру подготовки законченного веб-приложения к развертыванию. Она покажет, как оптимизировать загрузку приложений, особо остановившись на организации начальной страницы.

Что потребуется при чтении этой книги

Для опробования любых примеров использования AngularJS, что приводятся в этой книге, вам потребуются только веб-браузер и текстовый редактор (или интегрированная среда разработки по вашему выбору). Но, чтобы получить максимум от этой книги, мы рекомендуем также установить платформу `node.js` (<http://nodejs.org/>) и ее диспетчер пакетов `npm` со следующими модулями: