

Саймон Холмс

# Стек MEAN

**Mongo  
Express  
Angular  
Node**



Саймон Холмс

# Стек MEAN

**Mongo,  
Express,  
Angular,  
Node**



Санкт-Петербург • Москва • Екатеринбург • Воронеж  
Нижний Новгород • Ростов-на-Дону • Самара • Минск

2017

ББК 32.988.02-018  
УДК 004.738.5  
Х72

**Холмс С.**

**X72** Стек MEAN. Mongo, Express, Angular, Node. — СПб.: Питер, 2017. — 496 с.: ил. — (Серия «Библиотека программиста»).  
ISBN 978-5-496-02459-4

Обычно при веб-разработке на всех уровнях стека используются разные языки программирования. База данных MongoDB, фреймворки Express и AngularJS и технология Node.js вместе образуют стек MEAN — мощную платформу, на всех уровнях которой применяется всего один язык: JavaScript. Стек MEAN привлекателен для разработчиков и бизнеса благодаря простоте и экономичности, а конечные пользователи любят MEAN-приложения за их скорость и отзывчивость.

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

ББК 32.988.02-018  
УДК 004.738.5

Права на издание получены по соглашению с Manning. Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

Информация, содержащаяся в данной книге, получена из источников, рассматриваемых издательством как надежные. Тем не менее, имея в виду возможные человеческие или технические ошибки, издательство не может гарантировать абсолютную точность и полноту приводимых сведений и не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 978-1617292033 англ.  
978-5-496-02459-4

© 2016 by Manning Publications Co.  
© Перевод на русский язык ООО Издательство «Питер», 2017  
© Издание на русском языке, оформление ООО Издательство «Питер», 2017  
© Серия «Библиотека программиста», 2017

# Краткое содержание

<b>Предисловие</b> .....	16
<b>Благодарности</b> .....	19
<b>Об этой книге</b> .....	21
<b>Часть I. Задаем отправную точку</b> .....	25
<b>Глава 1.</b> Знакомство с разработкой full-stack.....	27
<b>Глава 2.</b> Проектируем архитектуру на основе стека MEAN .....	53
<b>Часть II. Создание веб-приложения на платформе Node</b> .....	83
<b>Глава 3.</b> Создание и настройка проекта на стеке MEAN .....	85
<b>Глава 4.</b> Создание статического сайта с помощью Node и Express.....	117
<b>Глава 5.</b> Создание модели данных с помощью MongoDB и Mongoose.....	162
<b>Глава 6.</b> Создание API REST: делаем базу данных MongoDB доступной приложению .....	208
<b>Глава 7.</b> Потребление API REST: использование API из Express.....	256
<b>Часть III. Добавление динамической клиентской части с помощью Angular</b> .....	301
<b>Глава 8.</b> Добавление компонентов Angular в приложение Express.....	303
<b>Глава 9.</b> Создание одностраничного приложения с помощью Angular: фундамент .....	343
<b>Глава 10.</b> Создание одностраничного приложения с помощью Angular: следующий уровень .....	377

<b>Часть IV. Управление аутентификацией и пользовательскими сеансами .....</b>	<b>425</b>
<b>Глава 11. Аутентификация пользователей, управление сеансами и обеспечение безопасности API.....</b>	<b>427</b>
<b>Приложения.....</b>	<b>477</b>
<b>Приложение А. Установка стека.....</b>	<b>478</b>
<b>Приложение Б. Установка и подготовка вспомогательного программного обеспечения .....</b>	<b>482</b>
<b>Приложение В. Разбираемся со всеми представлениями .....</b>	<b>486</b>

# Оглавление

<b>Предисловие</b> .....	16
<b>Благодарности</b> .....	19
<b>Об этой книге</b> .....	21
Дорожная карта .....	22
Стандарты оформления кода .....	22
Скачивание кода .....	23
Автор в Интернете .....	23
Об иллюстрации на обложке .....	24
<b>Часть I. Задаем отправную точку</b> .....	25
<b>Глава 1. Знакомство с разработкой full-stack</b> .....	27
1.1. Зачем изучать full-stack .....	28
1.1.1. Краткая история веб-разработки .....	28
1.1.2. Тенденция к разработке full-stack .....	30
1.1.3. Преимущества разработки full-stack .....	31
1.1.4. Почему именно стек MEAN .....	31
1.2. Знакомимся с Node.js: веб-сервер/веб-платформа .....	32
1.2.1. JavaScript: единый язык программирования для всего стека .....	32
1.2.2. Быстрая, производительная и масштабируемая платформа .....	33
1.2.3. Использование предварительно собранных пакетов с помощью npm .....	37
1.3. Знакомимся с Express: фреймворк .....	38
1.3.1. Упрощаем настройку сервера .....	38
1.3.2. Маршрутизация URL для ответов .....	38
1.3.3. Представления: ответы HTML .....	39
1.3.4. Запоминаем посетителей с помощью поддержки сессий .....	39

---

1.4. Знакомимся с MongoDB: база данных.....	39
1.4.1. Реляционные и документоориентированные базы данных.....	40
1.4.2. Документы MongoDB: хранилище данных JavaScript.....	40
1.4.3. Больше чем документоориентированная база данных.....	41
1.4.4. Для чего MongoDB не подходит.....	41
1.4.5. Использование Mongoose для моделирования данных и не только.....	42
1.5. Знакомимся с AngularJS: фреймворк клиентской части.....	43
1.5.1. jQuery и AngularJS.....	43
1.5.2. Двусторонняя привязка данных: работа с данными на странице.....	44
1.5.3. Использование AngularJS для загрузки новых страниц.....	45
1.5.4. Какова обратная сторона?.....	46
1.6. Вспомогательные утилиты.....	46
1.6.1. Применение Twitter Bootstrap для пользовательского интерфейса.....	47
1.6.2. Управление исходным кодом с помощью Git.....	48
1.6.3. Хостинг с помощью Heroku.....	49
1.7. Соединяем все вместе в реальном примере.....	49
1.7.1. Знакомство с примером приложения.....	49
1.7.2. Как компоненты стека MEAN работают вместе.....	51
1.8. Резюме.....	51
<b>Глава 2. Проектируем архитектуру на основе стека MEAN.....</b>	<b>53</b>
2.1. Традиционная архитектура стека MEAN.....	53
2.2. Выходим за рамки одностраничных приложений.....	55
2.2.1. Сложности со сканированием сайтов.....	55
2.2.2. Веб-аналитика и история браузера.....	56
2.2.3. Скорость начальной загрузки.....	57
2.2.4. SPA или не SPA, вот в чем вопрос.....	58
2.3. Разработка гибкой архитектуры MEAN.....	59
2.3.1. Требования к движку блога.....	59
2.3.2. Архитектура движка блога.....	61
2.3.3. Рекомендуемое решение: создание внутреннего API для слоя данных.....	64
2.4. Планирование реального приложения.....	66
2.4.1. Планирование приложения на высоком уровне.....	66
2.4.2. Проектирование архитектуры приложения.....	68
2.4.3. Конечный продукт.....	71
2.5. Разбиваем разработку на этапы.....	72
2.5.1. Этапы ускоренной разработки прототипа.....	72
2.5.2. Шаги создания Loc8r.....	74

---

2.6. Аппаратная архитектура .....	79
2.6.1. Аппаратные средства для разработки приложений.....	79
2.6.2. Аппаратное обеспечение для промышленной эксплуатации .....	80
2.7. Резюме.....	82
<b>Часть II. Создание веб-приложения на платформе Node .....</b>	<b>83</b>
<b>Глава 3. Создание и настройка проекта на стеке MEAN .....</b>	<b>85</b>
3.1. Краткий обзор Express, Node и npm .....	87
3.1.1. Описание пакетов с помощью package.json .....	87
3.1.2. Установка зависимостей Node с помощью npm .....	89
3.2. Создание проекта Express .....	90
3.2.1. Установка отдельных частей .....	91
3.2.2. Проверка установленных пакетов .....	91
3.2.3. Создание каталога проекта.....	92
3.2.4. Конфигурирование установки Express.....	92
3.2.5. Создание проекта Express и его проверка.....	94
3.2.6. Перезапуск приложения .....	98
3.2.7. Автоматический перезапуск приложения с помощью NODEMON .....	98
3.3. Модифицируем Express для MVC .....	99
3.3.1. Обзор MVC с высоты птичьего полета.....	99
3.3.2. Изменение структуры каталогов .....	100
3.3.3. Новые каталоги views и routes .....	101
3.3.4. Отделение контроллеров от маршрутов.....	102
3.4. Импорт Bootstrap для быстроты и адаптивности макетов.....	105
3.4.1. Скачивание Bootstrap и добавление его в приложение.....	106
3.4.2. Использование Bootstrap в приложении.....	107
3.5. Выводим приложение в Интернет с помощью Heroku.....	110
3.5.1. Установка и настройка Heroku.....	110
3.5.2. Запускаем сайт в Интернет с помощью Git .....	112
3.6. Резюме.....	116
<b>Глава 4. Создание статического сайта с помощью Node и Express.....</b>	<b>117</b>
4.1. Описание маршрутов в Express.....	119
4.1.1. Различные файлы контроллеров для разных логических наборов .....	120
4.1.2. Запрос файлов контроллера.....	121
4.1.3. Настройка маршрутов.....	121
4.2. Создание основных контроллеров .....	122
4.2.1. Настройка контроллеров .....	122
4.2.2. Тестирование контроллеров и маршрутов .....	123



4.3. Создание представлений.....	125
4.3.1. Краткий обзор Bootstrap .....	125
4.3.2. Настройка каркаса HTML с помощью шаблонов Jade и Bootstrap .....	127
4.3.3. Создание шаблона.....	132
4.4. Добавление остальных представлений .....	137
4.4.1. Страница Details (Подробности).....	138
4.4.2. Создаем страницу Add Review (Добавление отзыва) .....	141
4.4.3. Страница About (О нас) .....	143
4.5. Извлечение данных из представлений и их интеллектуализация.....	146
4.5.1. Как переместить данные из представления в контроллер .....	147
4.5.2. Работаем со сложными повторяющимися данными .....	150
4.5.3. Манипуляции данными и представлениями с помощью кода .....	154
4.5.4. Инструкции include и mixin для создания пригодных для повторного использования компонентов макета.....	155
4.5.5. Законченная домашняя страница .....	157
4.5.6. Модификация остальных представлений и контроллеров .....	159
4.6. Резюме .....	161
<b>Глава 5. Создание модели данных с помощью MongoDB и Mongoose.....</b>	<b>162</b>
5.1. Подключение приложения Express к MongoDB с помощью Mongoose .....	165
5.1.1. Добавление Mongoose в приложение .....	166
5.1.2. Добавление в приложение соединения с Mongoose.....	166
5.2. Зачем нужно моделировать данные .....	174
5.2.1. Что такое Mongoose и как он работает .....	176
5.2.2. Как Mongoose моделирует данные.....	177
5.2.3. Анализ пути схемы .....	177
5.3. Описание простых схем Mongoose .....	179
5.3.1. Основы настройки схемы .....	179
5.3.2. Использование географических данных в MongoDB и Mongoose .....	182
5.3.3. Создание более сложных схем с поддокументами .....	183
5.3.4. Окончательная схема .....	189
5.3.5. Компиляция схем MongoDB в модели .....	191
5.4. Использование командной оболочки MongoDB для создания базы данных MongoDB и добавления данных.....	192
5.4.1. Основы командной оболочки MongoDB .....	193
5.4.2. Создание базы данных MongoDB .....	194
5.5. Введение базы данных в промышленную эксплуатацию .....	198
5.5.1. Настройка MongoLab и получение URI базы данных .....	198
5.5.2. Помещение данных в базу данных .....	201
5.5.3. Заставляем приложение использовать правильную базу данных .....	203
5.6. Резюме.....	207

<b>Глава 6. Создание API REST: делаем базу данных MongoDB доступной приложению</b> .....	208
6.1. Правила API REST .....	209
6.1.1. URL запросов .....	210
6.1.2. Методы запроса .....	211
6.1.3. Ответы и коды состояния .....	214
6.2. Настройка API в Express .....	216
6.2.1. Создание маршрутов .....	217
6.2.2. Создание заглушек для контроллеров .....	220
6.2.3. Возврат JSON из запроса Express .....	220
6.2.4. Включение модели .....	221
6.2.5. Тестирование API .....	222
6.3. Методы GET: чтение данных из MongoDB .....	224
6.3.1. Поиск отдельного документа в MongoDB с помощью Mongoose .....	224
6.3.2. Поиск отдельного поддокумента на основе идентификаторов .....	229
6.3.3. Поиск нескольких документов с помощью геопространственных запросов .....	232
6.4. Методы POST: добавление данных в MongoDB .....	240
6.4.1. Создание новых документов в MongoDB .....	241
6.4.2. Проверка данных с помощью Mongoose .....	242
6.4.3. Создание новых поддокументов в MongoDB .....	243
6.5. Методы PUT: обновление данных в MongoDB .....	247
6.5.1. Использование Mongoose для обновления документа в MongoDB .....	247
6.5.2. Метод Mongoose save .....	248
6.5.3. Обновление существующего поддокумента в MongoDB .....	250
6.6. Метод DELETE: удаление данных из MongoDB .....	252
6.6.1. Удаление документов из MongoDB .....	252
6.6.2. Удаление поддокумента из MongoDB .....	253
6.7. Резюме .....	255
<b>Глава 7. Потребление API REST: использование API из Express</b> .....	256
7.1. Обращение к API из Express .....	257
7.1.1. Добавление в наш проект модуля request .....	258
7.1.2. Настройка параметров по умолчанию .....	258
7.1.3. Модуль request .....	259
7.2. Использование списков данных из API: домашняя страница Loc8r .....	261
7.2.1. Разделение обязанностей: перенос визуализации в поименованную функцию .....	261
7.2.2. Создание запроса API .....	262
7.2.3. Использование данных ответа API .....	263

7.2.4. Изменение данных перед их отображением: приводим в порядок расстояния.....	265
7.2.5. Перехват возвращаемых API ошибок .....	267
7.3. Получение от API отдельных документов: страница Details приложения Loc8r .....	273
7.3.1. Настройка URL и маршрутов для обращения к конкретным документам MongoDB .....	274
7.3.2. Разделение обязанностей: перемещаем визуализацию в поименованную функцию .....	275
7.3.3. Выполнение запросов к API с использованием уникального идентификатора из параметра URL .....	276
7.3.4. Передаем данные из API в представление .....	277
7.3.5. Отладка и исправление ошибок с представлением .....	278
7.3.6. Создание страниц ошибок в зависимости от статуса .....	281
7.4. Добавление данных в БД через API: добавляем отзвыы Loc8r .....	284
7.4.1. Настройка маршрутизации и представлений .....	285
7.4.2. Отправка данных отзвыов API методом POST .....	290
7.5. Защита целостности данных с помощью их проверок .....	292
7.5.1. Проверка на уровне схемы с помощью Mongoose .....	293
7.5.2. Проверка на уровне приложения с помощью Node и Express .....	297
7.5.3. Проверка в браузере с помощью jQuery .....	298
7.6. Резюме .....	300
<b>Часть III. Добавление динамической клиентской части с помощью Angular .....</b>	<b>301</b>
<b>Глава 8. Добавление компонентов Angular в приложение Express .....</b>	<b>303</b>
8.1. Настройка и запуск Angular .....	304
8.1.1. Открываем для себя двустороннюю привязку данных .....	305
8.1.2. Готовим все для настоящих достижений (и кода JavaScript) .....	308
8.2. Отображение и фильтрация списка для домашней страницы .....	312
8.2.1. Добавление Angular в приложение Express .....	312
8.2.2. Перемещение выдачи данных из Express в Angular .....	313
8.2.3. Фильтры Angular для форматирования данных .....	318
8.2.4. Директивы Angular для создания HTML-сниппетов .....	322
8.3. Получение данных из API .....	328
8.3.1. Сервисы для данных .....	328
8.3.2. Выполнение HTTP-запросов к API из Angular .....	330
8.3.3. Добавляем HTML-геолокацию для поиска местоположений, находящихся рядом с вами .....	333
8.4. Обеспечиваем надлежащую работу форм .....	340
8.5. Резюме .....	341

<b>Глава 9. Создание одностраничного приложения с помощью</b>	
Angular: фундамент .....	343
9.1. Подготовительные работы для Angular SPA .....	344
9.1.1. Создание каталога app_client для приложения на стороне клиента .....	345
9.1.2. Создание основного файла приложения SPA .....	345
9.1.3. Добавление основного файла приложения SPA в макет Jade .....	346
9.2. Переключение с Express-маршрутизации на Angular-маршрутизацию .....	346
9.2.1. Отключение маршрутизации Express .....	348
9.2.2. Добавляем в приложение ngRoute (angular-route) .....	350
9.3. Добавление первых представлений, контроллеров и сервисов .....	352
9.3.1. Создаем представление Angular .....	353
9.3.2. Добавляем контроллер к маршруту .....	355
9.3.3. Рекомендуемое решение для контроллера: использование синтаксиса controllerAs .....	358
9.3.4. Сервисы .....	361
9.3.5. Использование фильтров и директив .....	365
9.4. Улучшаем производительность браузера .....	369
9.4.1. Обертываем каждый файл в IIFE .....	369
9.4.2. Внедряем зависимости вручную для защиты от сокращения .....	371
9.4.3. Используем UglifyJS для сокращения и конкатенации сценариев .....	372
9.5. Резюме .....	376
<b>Глава 10. Создание одностраничного приложения с помощью</b>	
Angular: следующий уровень .....	377
10.1. Полнофункциональное SPA: перестаем основываться на серверном приложении .....	378
10.1.1. Создание изолированной HTML-страницы хоста .....	379
10.1.2. Создаем допускающие многократное использование директивы для каркаса страницы .....	381
10.1.3. Удаление # из URL .....	387
10.2. Добавление дополнительных страниц и динамическое внедрение HTML .....	389
10.2.1. Добавление в SPA нового маршрута и страницы .....	389
10.2.2. Создание фильтра для преобразования разрывов строк .....	393
10.2.3. Передача HTML в привязку Angular .....	395
10.3. Более сложные представления и параметры маршрутизации .....	397
10.3.1. Развертываем фреймворк страницы .....	398
10.3.2. Параметры URL в контроллерах и сервисах .....	400
10.3.3. Создаем представление для страницы Details (Подробности) .....	403
10.4. Использование компонентов AngularUI для создания модального всплывающего окна .....	407
10.4.1. Подготовка AngularUI .....	408
10.4.2. Добавление и использование обработчика нажатий .....	409

10.4.3. Создаем модальное окно Bootstrap с помощью AngularUI .....	411
10.4.4. Передача данных в модальное окно .....	414
10.4.5. Форма для отправки отзыва .....	417
10.5. Резюме .....	424

## **Часть IV. Управление аутентификацией**

<b>и пользовательскими сеансами .....</b>	<b>425</b>
---	------------

<b>Глава 11. Аутентификация пользователей, управление сеансами и обеспечение безопасности API .....</b>	<b>427</b>
11.1. Подход к аутентификации в стеке MEAN .....	428
11.1.1. Традиционный серверный подход .....	429
11.1.2. Подход с использованием всего стека MEAN .....	430
11.2. Создание схемы для пользователей в MongoDB .....	433
11.2.1. Одностороннее шифрование паролей: хеши и соль .....	433
11.2.2. Создание схемы Mongoose .....	434
11.2.3. Задание зашифрованных путей с помощью методов Mongoose .....	435
11.2.4. Проверяем введенный пароль .....	436
11.2.5. Генерация веб-маркера JSON .....	437
11.3. Создание API аутентификации с помощью Passport .....	440
11.3.1. Установка и конфигурация Passport .....	441
11.3.2. Создание конечных точек API для возврата веб-маркеров JSON .....	444
11.4. Защита конечных точек API .....	449
11.4.1. Добавление промежуточного ПО аутентификации в маршруты Express .....	450
11.4.2. Использование информации из JWT в контроллере .....	452
11.5. Создание сервиса аутентификации Angular .....	455
11.5.1. Управление сеансами пользователей в Angular .....	456
11.5.2. Предоставляем пользователям возможность регистрироваться, входить в приложение и выходить из него .....	457
11.5.3. Использование данных из JWT в сервисе Angular .....	459
11.6. Создание страниц регистрации и входа в приложение .....	461
11.6.1. Создаем страницу регистрации .....	461
11.6.2. Создаем страницу входа .....	465
11.7. Работаем с аутентификацией в приложении Angular .....	468
11.7.1. Меняем навигацию .....	468
11.7.2. Добавляем данные пользователя в отзыв .....	471
11.8. Резюме .....	475

---

<b>Приложения</b> .....	477
<b>Приложение А. Установка стека</b> .....	478
А.1. Установка Node и npm.....	478
А.1.1. Установка Node в Windows.....	478
А.1.2. Установка Node в Mac OS X.....	479
А.1.3. Установка Node в Linux.....	479
А.1.4. Проверяем установку путем сверки версий.....	479
А.2. Глобальная установка Express.....	479
А.3. Установка MongoDB.....	480
А.3.1. Установка MongoDB в Windows.....	480
А.3.2. Установка MongoDB в Mac OS X.....	480
А.3.3. Установка MongoDB в Linux.....	480
А.3.4. Запуск MongoDB в качестве сервиса.....	481
А.3.5. Проверка номера версии MongoDB.....	481
А.4. Установка Angular.....	481
<b>Приложение Б. Установка и подготовка вспомогательного программного обеспечения</b> .....	482
Б.1. Twitter Bootstrap.....	482
Б.1.1. Получение темы Amelia.....	483
Б.1.2. Чистка каталогов.....	484
Б.2. Установка Git.....	484
Б.3. Установка подходящего интерфейса командной строки.....	484
Б.4. Настройка Heroku.....	485
Б.4.1. Подписка на Heroku.....	485
Б.4.2. Установка набора инструментов Heroku.....	485
Б.4.3. Вход в Heroku с помощью терминала.....	485
<b>Приложение В. Разбираемся со всеми представлениями</b> .....	486
В.1. Перемещение данных из представлений в контроллеры.....	486
В.1.1. Страница Details (Подробности).....	486
В.1.2. Страница Add Review (Добавление отзыва).....	490
В.1.3. Страница About (О нас).....	491