

УДК 004.738.5:004.4Node.js
ББК 32.973.202-018.2
К28

Каскиаро М., Маммино Л.

К28 Шаблоны проектирования Node.js / пер. с англ. А. Н. Киселева. – М.: ДМК Пресс, 2017. – 396 с.: ил.

ISBN 978-5-97060-485-4

Node.js – программная платформа, позволяющая легко и просто создавать масштабируемые серверные приложения на языке JavaScript.

В книге описаны асинхронная, однопоточная архитектура платформы, а также шаблоны асинхронного управления потоком выполнения и потоками данных. Рассмотрен подробный список реализаций распространенных, а также некоторых уникальных шаблонов проектирования в Node.js.

Издание адресовано разработчикам и архитекторам программного обеспечения, обладающим основными навыками владения JavaScript и желающим получить глубокое понимание, как проектируются и разрабатываются приложения уровня предприятия на основе Node.js.

УДК 004.738.5:004.4Node.js
ББК 32.973.202-018.2

Copyright ©Packt Publishing 2016. First published in the English language under the title 'Node.js Design Patterns - Second Edition – (9781785885587)'

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

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

ISBN 978-1-78588-558-7 (анг.)
ISBN 978-5-97060-485-4 (рус.)

© 2016 Packt Publishing
© Перевод, оформление, издание, ДМК Пресс, 2017

Содержание

Об авторах.....	11
О технических рецензентах.....	14
Предисловие.....	15
Глава 1. Добро пожаловать в платформу Node.js.....	21
Философия Node.js.....	21
Небольшое ядро.....	22
Небольшие модули.....	22
Небольшая общедоступная область.....	23
Простота и прагматизм.....	23
Введение в Node.js 6 и ES2015.....	24
Ключевые слова let и const.....	24
Стрелочные функции.....	26
Синтаксис классов.....	28
Расширенные литералы объектов.....	29
Коллекции Map и Set.....	30
Коллекции WeakMap и WeakSet.....	31
Литералы шаблонов.....	32
Другие особенности ES2015.....	33
Шаблон Reactor.....	33
Медленный ввод/вывод.....	33
Блокирующий ввод/вывод.....	34
Неблокирующий ввод/вывод.....	35
Демультимплексирование событий.....	36
Введение в шаблон Reactor.....	37
Неблокирующий движок libuv платформы Node.js.....	38
Рецепт платформы Node.js.....	39
Итоги.....	40
Глава 2. Основные шаблоны Node.js.....	41
Шаблон Callback.....	41
Стиль передачи продолжений.....	42
Синхронный или асинхронный?.....	44
Соглашения Node.js об обратных вызовах.....	48
Система модулей и ее шаблоны.....	51
Шаблон Revealing Module.....	51
Пояснения относительно модулей Node.js.....	52
Шаблоны определения модулей.....	58
Шаблон Observer.....	63
Класс EventEmitter.....	63
Создание и использование класса EventEmitter.....	64

Распространение ошибок	65
Создание произвольного наблюдаемого объекта	66
Синхронные и асинхронные события	67
Класс EventEmitter и обратные вызовы	68
Комбинирование EventEmitter и обратных вызовов	68
Итоги	69

Глава 3. Шаблоны асинхронного выполнения

с обратными вызовами	70
Сложности асинхронного программирования	70
Создание простого поискового робота	71
Ад обратных вызовов	72
Использование обычного JavaScript	73
Дисциплина обратных вызовов	74
Применение дисциплины обратных вызовов	74
Последовательное выполнение	76
Параллельное выполнение	80
Ограниченное параллельное выполнение	85
Библиотека async	88
Последовательное выполнение	89
Параллельное выполнение	91
Ограниченное параллельное выполнение	92
Итоги	93

Глава 4. Шаблоны асинхронного выполнения

с использованием спецификации ES2015, и не только	94
Promise	94
Что представляет собой объект Promise?	95
Реализации Promises/A+	97
Перевод функций в стиле Node.js на использование объектов Promise	98
Последовательное выполнение	99
Параллельное выполнение	101
Ограниченное параллельное выполнение	102
Обратные вызовы и объекты Promise в общедоступных программных интерфейсах	103
Генераторы	105
Введение в генераторы	105
Асинхронное выполнение с генераторами	108
Последовательное выполнение	110
Параллельное выполнение	112
Ограниченное параллельное выполнение	114
Async/await с использованием Babel	117
Установка и запуск Babel	118
Сравнение	119
Итоги	119

Глава 5. Программирование с применением потоков данных	121
Исследование важности потоков данных	121
Буферизация и потоковая передача данных	121
Эффективность с точки зрения памяти	122
Эффективность с точки зрения времени	124
Способность к объединению	126
Начало работы с потоками данных	127
Анатомия потоков данных	128
Потоки данных для чтения	128
Потоки данных для записи	132
Дуплексные потоки данных	135
Преобразующие потоки данных	136
Соединение потоков с помощью конвейеров	138
Управление асинхронным выполнением с помощью потоков данных	140
Последовательное выполнение	140
Неупорядоченное параллельное выполнение	142
Неупорядоченное ограниченное параллельное выполнение	145
Шаблоны конвейерной обработки	147
Объединение потоков данных	147
Ветвление потоков данных	150
Слияние потоков данных	151
Мультиплексирование и демультиплексирование	153
Итоги	158
Глава 6. Шаблоны проектирования	159
Фабрика	160
Универсальный интерфейс для создания объектов	160
Механизм принудительной инкапсуляции	161
Создание простого профилировщика кода	162
Составные фабричные функции	164
Реальное применение	167
Открытый конструктор	168
Генератор событий, доступный только для чтения	168
Реальное применение	169
Прокси	170
Приемы реализации прокси	171
Сравнение различных методов	172
Журналирование обращений к потоку для записи	173
Место прокси в экосистеме – ловушки для функций и АОП	174
Прокси в стандарте ES2015	174
Реальное применение	176
Декоратор	176
Приемы реализации декораторов	176
Декорирование базы данных LevelUP	177
Реальное применение	179

Адаптер	180
Использование LevelUP через интерфейс файловой системы	180
Реальное применение	183
Стратегия	183
Объекты для хранения конфигураций в нескольких форматах	184
Реальное применение	186
Состояние	187
Реализация простого сокета, защищенного от сбоя	188
Макет	191
Макет диспетчера конфигурации	192
Реальное применение	193
Промежуточное программное обеспечение	194
Промежуточное программное обеспечение в Express	194
Промежуточное программное обеспечение как шаблон	195
Создание фреймворка промежуточного программного обеспечения для OMQ	196
Промежуточное программное обеспечение, использующее генераторы Koa	201
Команда	204
Гибкость шаблона	205
Итоги	208
Глава 7. Связывание модулей	210
Модули и зависимости	211
Наиболее типичные зависимости в Node.js	211
Сцепленность и связанность	212
Модули с поддержкой состояния	212
Шаблоны связывания модулей	214
Жесткие зависимости	214
Внедрение зависимостей	218
Локатор служб	222
Контейнер внедрения зависимостей	227
Связывание плагинов	230
Плагины как пакеты	230
Точки расширения	232
Расширение, управляемое плагинами и приложением	232
Реализация плагина выхода из системы	235
Итоги	242
Глава 8. Универсальный JavaScript для веб-приложений	243
Использование кода совместно с браузером	244
Совместное использование модулей	244
Введение в Webpack	248
Знакомство с волшебством Webpack	248
Преимущества использования Webpack	250
Использование ES2015 с помощью Webpack	250

Основы кросс-платформенной разработки	252
Ветвление кода во время выполнения.....	252
Ветвление кода в процессе сборки.....	253
Замена модулей	255
Шаблоны проектирования для кросс-платформенной разработки.....	257
Введение в React.....	258
Первый компонент React	259
Что такое JSX?!.....	260
Настройка Webpack для транскомпиляции JSX.....	262
Отображение в браузере.....	263
Библиотека React Router.....	264
Создание приложений на универсальном JavaScript.....	268
Создание многократно используемых компонентов.....	268
Отображение на стороне сервера	271
Универсальное отображение и маршрутизация.....	274
Универсальное извлечение данных	275
Итоги	282

Глава 9. Дополнительные рецепты асинхронной обработки.....284

Подключение модулей, инициализируемых асинхронно.....	284
Канонические решения.....	285
Очереди на инициализацию.....	286
Реальное применение	289
Группировка асинхронных операций и кэширование.....	290
Реализация сервера без кэширования и группировки операций.....	290
Группировка асинхронных операций.....	292
Кэширование асинхронных запросов	294
Группировка и кэширование с использованием объектов Promise	297
Выполнение вычислительных заданий.....	299
Решение задачи выделения подмножеств с заданной суммой.....	299
Чередование с помощью функции setImmediate	302
Чередование этапов алгоритма извлечения подмножеств с заданной суммой.....	302
Использование нескольких процессов.....	304
Итоги	310

Глава 10. Шаблоны масштабирования и организации архитектуры.....311

Введение в масштабирование приложений	312
Масштабирование приложений на платформе Node.js	312
Три измерения масштабируемости.....	312
Клонирование и распределение нагрузки.....	314
Модуль cluster	315
Взаимодействия с сохранением состояния.....	322
Масштабирование с помощью обратного проксирования	325
Использование реестра служб	328

Одноранговое распределение нагрузки	333
Декомпозиция сложных приложений	336
Монолитная архитектура	336
Архитектура на микрослужбах	338
Шаблоны интеграции в архитектуре на микрослужбах	341
Итоги	346
Глава 11. Шаблоны обмена сообщениями и интеграции	348
Введение в системы обмена сообщениями	349
Шаблоны однонаправленного обмена и вида «Запрос/ответ»	349
Типы сообщений	350
Асинхронный обмен сообщениями и очереди	351
Обмен сообщениями, прямой и через брокера	352
Шаблон «Публикация/подписка»	353
Минимальное приложение для общения в режиме реального времени	354
Использование Redis в качестве брокера сообщений	357
Прямая публикация/подписка с помощью библиотеки ØMQ	359
Надежная подписка	362
Шаблоны конвейеров и распределения заданий	369
Шаблон распределения/слияния в ØMQ	370
Конвейеры и конкурирующие потребители в AMQP	374
Шаблоны вида «Запрос/ответ»	378
Идентификатор корреляции	378
Обратный адрес	382
Итоги	386
Предметный указатель	387