

Оглавление

Часть I. Основы проектирования API	28
1 ■ Что такое проектирование API	29
2 ■ Проектирование API для пользователей	45
3 ■ Проектирование программного интерфейса	75
4 ■ Описание API с помощью формата описания	111
Часть II. Проектирование практического API ...	149
5 ■ Проектирование простого API	151
6 ■ Проектирование предсказуемого API	183
7 ■ Проектирование лаконичного и хорошо организованного API	213
Часть III. Контекстное проектирование API ...	233
8 ■ Проектирование безопасного API	235
9 ■ Изменение дизайна API	269
10 ■ Проектирование эффективного API для сети	311
11 ■ Проектирование API в контексте	345
12 ■ Документирование API	381
13 ■ Развитие API	413

Содержание

Предисловие	13
От автора	16
Благодарности	18
Об этой книге	20
Об авторе	24
Об иллюстрации на обложке	25

Часть I. Основы проектирование API..... 28

1	Что такое проектирование API..... 29
1.1.	Что такое API?..... 30
1.1.1.	API – это веб-интерфейс для программного обеспечения ... 30
1.1.2.	API превращают программное обеспечение в детали конструктора LEGO® 32
1.2.	Чем важна разработка API 35
1.2.1.	Открытый или закрытый API – это интерфейс для других разработчиков 36
1.2.2.	API создается, для того чтобы скрыть реализацию 37
1.2.3.	Страшные последствия плохо спроектированных API..... 38
1.3.	Элементы проектирования API 42
1.3.1.	Изучение принципов, выходящих за рамки проектирования программного интерфейса 42
1.3.2.	Изучение всех аспектов проектирования API 43
2	Проектирование API для пользователей 45
2.1	Правильная точка зрения для проектирования повседневных пользовательских интерфейсов 46
2.1.1	Когда вы фокусируетесь на том, как все работает, это приводит к возникновению сложных интерфейсов 46

2.1.2	<i>Когда вы фокусируетесь на том, что могут делать пользователи, это приводит к появлению простых интерфейсов</i>	48
2.2	Проектирование интерфейсов программного обеспечения	50
2.2.1	<i>API как панель управления программным обеспечением</i>	50
2.2.2	<i>Ориентация на точку зрения потребителя для создания простых API</i>	51
2.3	Определение целей API	54
2.3.1	<i>Отвечая на вопросы «что?» и «как?»</i>	55
2.3.2	<i>Определение входных и выходных данных</i>	56
2.3.3	<i>Выявляем недостающие цели</i>	58
2.3.4	<i>Идентификация всех пользователей</i>	61
2.3.5	<i>Использование таблицы целей API</i>	62
2.4	Избегаем точки зрения поставщика при проектировании API	64
2.4.1	<i>Как избежать влияния данных</i>	65
2.4.2	<i>Как избежать влияния кода и бизнес-логики</i>	67
2.4.3	<i>Как избежать влияния архитектуры программного обеспечения</i>	69
2.4.4	<i>Как избежать влияния организации, где работают люди</i>	70
2.4.5	<i>Определение точки зрения поставщика в таблице целей API</i>	72

3

	Проектирование программного интерфейса	75
3.1	Знакомство с REST API	77
3.1.1	<i>Анализ вызова REST API</i>	77
3.1.2	<i>Базовые принципы HTTP</i>	79
3.1.3	<i>Базовые принципы REST API</i>	80
3.2	Перенос целей в REST API	81
3.2.1	<i>Идентификация ресурсов и их связей с таблицей целей API</i>	82
3.2.2	<i>Идентификация действий, их параметров и результатов с помощью таблицы целей API</i>	84
3.2.3	<i>Представление ресурсов с помощью путей</i>	86
3.2.4	<i>Представление действий с помощью протокола HTTP</i>	88
3.2.5	<i>REST API и шпаргалка по HTTP</i>	92
3.3	Проектирование данных API	94
3.3.1	<i>Проектирование концепций</i>	94
3.3.2	<i>Проектирование ответов от концепций</i>	97
3.3.3	<i>Проектирование параметров из концепций или ответов</i>	98
3.3.4	<i>Проверка параметров источника данных</i>	99
3.3.5	<i>Проектирование других параметров</i>	101
3.4	Достижение баланса при решении проблем проектирования	101
3.4.1	<i>Примеры компромисса</i>	101

3.4.2	Баланс между удобством для пользователя и соответствием	103
3.5	Почему REST важен при проектировании любого API	104
3.5.1	Знакомство с архитектурным стилем REST	104
3.5.2	Влияние ограничений REST на проектирование API	106
4	Описание API с помощью формата описания	111
4.1	Что такое формат описания API?	112
4.1.1	Спецификация OpenAPI (OAS)	113
4.1.2	Зачем использовать формат описания API?	115
4.1.3	Когда использовать формат описания API	118
4.2	Описание ресурсов и действий API с помощью OAS	119
4.2.1	Создание документа OAS	120
4.2.2	Описание ресурса	121
4.2.3	Описание операций в ресурсе	122
4.3	Описание данных API с помощью OpenAPI и JSON Schema	126
4.3.1	Описание параметров запроса	127
4.3.2	Описание данных с помощью JSON Schema	130
4.3.3	Описание ответов	134
4.3.4	Описание параметров тела	137
4.4	Эффективное описание API с помощью OAS	140
4.4.1	Повторное использование компонентов	140
4.4.2	Описание параметров пути	143

Часть II. Проектирование практического API ..149

5	Разработка простого API	151
5.1	Разработка простых представлений	152
5.1.1	Выбор кристально ясных имен	153
5.1.2	Выбор простых в использовании типов данных и форматов	155
5.1.3	Выбор готовых к использованию данных	157
5.2	Проектирование простых взаимодействий	159
5.2.1	Запрос простых входных данных	160
5.2.2	Выявление всех возможных ошибок	162
5.2.3	Возвращение информативного сообщения об ошибке	163
5.2.4	Возвращение исчерпывающего сообщения об ошибке	168
5.2.5	Возвращение информативного сообщения об успешном результате	170
5.3	Проектирование простых потоков	172
5.3.1	Построение простой цепочки целей	174
5.3.2	Предотвращение ошибок	176
5.3.3	Объединение целей	178
5.3.4	Проектирование потоков без сохранения состояния	180

6	Проектирование предсказуемого API	183
6.1	Согласованность	184
6.1.1	Проектирование согласованных данных	185
6.1.2	Проектирование согласованных целей	188
6.1.3	Четыре уровня согласованности	189
6.1.4	Копируя других: следование общепринятым практикам и соблюдение стандартов	190
6.1.5	Согласованность – это сложно, и все нужно делать по-умному	194
6.2	Адаптируемость	195
6.2.1	Предоставление и принятие разных форматов	195
6.2.2	Интернационализация и локализация	199
6.2.3	Фильтрация, разбиение на страницы и сортировка	202
6.3	Быть видимым	204
6.3.1	Предоставление метаданных	205
6.3.2	Создание гипермедиа-API	206
6.3.3	Использование преимуществ протокола HTTP	210
7	Проектирование лаконичного и хорошо организованного API	213
7.1	Организация API	213
7.1.1	Организация данных	215
7.1.2	Организация ответных сообщений	217
7.1.3	Организация целей	219
7.2	Определение размера API	225
7.2.1	Выбор детализации данных	226
7.2.2	Выбор детализации целей	228
7.2.3	Выбор детализации API	229
Часть III. Контекстное проектирование API		233
8	Проектирование безопасного API	235
8.1	Обзор безопасности API	237
8.1.1	Регистрация потребителя	237
8.1.2	Получение учетных данных для использования API	238
8.1.3	Выполнение API-вызова	240
8.1.4	Проектирование API с точки зрения безопасности	241
8.2	Разделение API на части для облегчения управления доступом	243
8.2.1	Определение гибких, но точных групп	245
8.2.2	Определение простых, но менее детализированных групп ..	247
8.2.3	Выбор стратегии	250
8.2.4	Определение групп с помощью формата описания API	251
8.3	Проектирование с учетом управления доступом	254

8.3.1	Какие данные необходимы для управления доступом	254
8.3.2	Адаптация дизайна при необходимости	255
8.4	Обработка конфиденциальных данных и важных вещей	257
8.4.1	Обработка конфиденциальных данных	258
8.4.2	Обработка конфиденциальных целей	261
8.4.3	Проектирование безопасных сообщений об ошибках	264
8.4.4	Выявление проблем, связанных с архитектурой и протоколом	266
9	Изменение дизайна API	269
9.1	Проектирование изменений API	271
9.1.1	Избегайте критических изменений в выходных данных	272
9.1.2	Как избежать критических изменений во входных данных и параметрах	277
9.1.3	Как избежать критических изменений в сообщениях об успехе или ошибках	281
9.1.4	Как избежать критических изменений в целях и потоках	284
9.1.5	Предотвращение нарушений в системе безопасности и критических изменений	286
9.1.6	Невидимый контракт интерфейса	287
9.1.7	Критическое изменение – не всегда проблема	288
9.2	Управление версиями API	289
9.2.1	Управление версиями API и реализации	290
9.2.2	Выбор представления управления версиями API с точки зрения потребителя	292
9.2.3	Выбор детализации	295
9.2.4	Влияние управления версиями API за пределами проектирования	301
9.3	Проектирование API с учетом расширяемости	302
9.3.1	Проектирование расширяемых данных	302
9.3.2	Проектирование расширяемых взаимодействий	306
9.3.3	Проектирование расширяемых потоков	307
9.3.4	Проектирование расширяемых API	308
10	Проектирование эффективного API для сети	311
10.1	Обзор проблем передачи данных по сети	312
10.1.1	Подготовка сцены	313
10.1.2	Анализ проблем	315
10.2	Обеспечение эффективности передачи данных по сети на уровне протокола	320
10.2.1	Активация сжатия и постоянных соединений	320
10.2.2	Активация кеширования и условных запросов	321
10.2.3	Выбор политики кеширования	325

10.3 Обеспечение эффективности передачи данных по сети на уровне дизайна	326
10.3.1 Активация фильтрации	327
10.3.2 Выбор соответствующих данных для представлений списка	330
10.3.3 Агрегирование данных	332
10.3.4 Предложение разных представлений	334
10.3.5 Активация расширения	336
10.3.6 Активация запросов	338
10.3.7 Предоставление более релевантных данных и целей	340
10.3.8 Создание разных слоев API	343

11 Проектирование API в контексте

11.1 Адаптация передачи данных к целям и характеру данных	347
11.1.1 Управление длительными процессами	347
11.1.2 Уведомление потребителей о событиях	349
11.1.3 Потокковая передача событий	352
11.1.4 Обработка нескольких элементов	357
11.2 Соблюдение полного контекста	365
11.2.1 Знание существующих практик и ограничений потребителей	365
11.2.2 Тщательно учитываем ограничения поставщика	370
11.3 Выбор стиля API в соответствии с контекстом	373
11.3.1 Сравнение API на базе ресурсов, данных и функций	374
11.3.2 За пределами API на базе HTTP	379

12 Документирование API

12.1 Создание справочной документации	384
12.1.1 Документирование моделей данных	385
12.1.2 Документирование целей	389
12.1.3 Документирование безопасности	396
12.1.4 Обзор API	398
12.1.5 Генерирование документации из кода реализации: плюсы и минусы	399
12.2 Создание руководства пользователя	400
12.2.1 Документирование вариантов использования	401
12.2.2 Документирование безопасности	404
12.2.3 Предоставление обзора общепринятого поведения и принципов	404
12.2.4 Мышление вне статической документации	404
12.3 Предоставление адекватной информации разработчикам	405
12.4 Документирование изменений API и устаревшие функции	409

13	Развитие API	413
13.1	Жизненный цикл API	414
13.2	Создание руководства по проектированию API	415
13.2.1	Что включить в руководство по проектированию API	416
13.2.2	Постоянное создание руководств	420
13.3	Проверка API	422
13.3.1	Оспаривание потребностей и их анализ	424
13.3.2	Линтирование	427
13.3.3	Проверка дизайна с точки зрения поставщика	430
13.3.4	Проверка дизайна с точки зрения потребителя	432
13.3.5	Проверка реализации	433
13.4	Общайтесь и делитесь информацией	435

Предисловие

На протяжении более десяти лет для большинства разработчиков проектирование API всегда подразумевало архитектуру REST. Такая ситуация сложилась в связи с регулярным выходом книг и блогов об API, которые продвигают мировоззрение RESTful – и это порождает некую заикленность на данном направлении, если не сказать догматизм. Эта книга знаменует появление руководств по проектированию API следующего поколения, которые помогут нам преодолеть ограничения, тормозившие более десяти лет. Практический подход к разработке API по-прежнему основывается на REST, но автор очень старался сформировать представление о разработке API в практических ситуациях за вычетом имеющихся догм.

Арно Лоре знакомит нас с основами проектирования API, которые легко можно найти в других отраслевых изданиях, и раскрывает тему просто и доступно, рассматривая широкий круг вопросов в удобной для читателя форме. Я знаком с Арно уже несколько лет и считаю его одним из немногих высокопрофессиональных специалистов, которые не только знают, как создавать API технически, но и понимают проблемы, связанные с их доставкой потребителю, и знают, в каких случаях API могут оказывать положительное или отрицательное влияние на опыт разработчика. Арно фокусируется не только на проектировании API, но и на предоставлении правильно спроектированного API целевой аудитории продуманным образом.

Я лично наблюдал, как Арно принимал самое активное участие в дискуссиях, касающихся API, по всему миру, извлекая из них все самое полезное для себя. Достаточно просмотреть его страницу в Twitter или перейти по хештегу, посвященному какому-нибудь популярному мероприятию, связанному с API, чтобы понять, о чем я говорю. Арно использует уникальный подход, слушая доклады представителей из индустрии API и обрабатывая информацию, которой они делятся, и затем освеща-

ет в Twitter важные моменты беседы, представляя сведения об API в удобном для восприятия формате. Я рад, что Арно свел накопленные им знания воедино и изложил в книге, продолжая не только оттачивать собственные навыки, но и делиться с другими людьми своими знаниями и уникальным подходом к проектированию API. Арно принадлежит к редкой породе API-аналитиков, которые заботятся о развитии темы: впитывают знания об API, усваивают их и распространяют в доступной форме, так что их становится действительно легко применить в мире бизнеса.

После того как, начиная с 2012 года, проектирование API стало набирать обороты и стала очевидной доминирующая роль спецификации OpenAPI (ранее известной как Swagger), Арно был одним из немногих экспертов в области API, которые усердно старались раскрыть потенциал этой спецификации, а также разрабатывали инновационные инструменты и визуализации, связанные со стандартом спецификации API с открытым исходным кодом – чтобы понять не только спецификацию, но и то, как она может реализовываться, представлять и даже организовывать многие принципы проектирования API, необходимые для достижения успеха. Понадобится много работать, прежде чем вы поймете, что OpenAPI – не только документация; большинство разработчиков API так и не проходят этот путь до конца. Арно понимает, что OpenAPI – это к тому же и основа проектирования API для любой платформы, что помогает определить каждую остановку на протяжении всего жизненного цикла API. Эта книга – первое руководство по проектированию API из тех, что я видел, в котором OpenAPI и проектирование API объединены так вдумчиво и эффективно, что многие разработчики почувствуют несомненную пользу чтения.

Потратьте время, чтобы понять, чем Арно делится с вами. Это книга не предназначена для беглого пролистывания, и она уж точно не из серии «прочитал и забыл». Это справочник. Руководство по переходу проектирования ваших API на новый уровень. Оно дает вам набор концепций API, с которыми вы знакомы, и предоставляет вам чертежи для создания «Тысячелетнего сокола» или даже «Звезды Смерти» (если, конечно, пожелаете) из набора строительных блоков API.

Рекомендую вам прочитать эту книгу, а затем отложить ее на месяц. Затем приступайте к созданию API и переходу от проектирования к фактическому развертыванию и общедоступности – и поделитесь своими знаниями с разработчиками. А пока ждете отзывов, сядьте и снова возьмитесь за книгу. Вы начнете понимать глубину и ценность знаний, которыми Арно делится с вами. Возвращайтесь к чтению до тех пор, пока вы не научитесь в полной мере создавать API – не идеальные образцы,

а именно тот API, который вам нужен, чтобы достучаться до потребителей, на которых вы рассчитываете повлиять.

— *Кин Лейн, апологет API*

От автора

На протяжении своей карьеры я большей частью соединял программные блоки, используя различные технологии программных интерфейсов – начиная с простых файлов и баз данных и заканчивая удаленными программными интерфейсами на базе RPC, Corba, Java RMI, веб-сервисов SOAP и веб-API. В эти годы мне посчастливилось работать над разнородными распределенными системами, смешивая очень старую технологию мейнфреймов с современными облачными системами и всем, что находится посередине между ними. Мне также посчастливилось работать с обеими сторонами программных интерфейсов в различных контекстах. Я работал над IVR (Interactive Voice Response), веб-сайтами и мобильными приложениями, созданными поверх огромных систем сервис-ориентированной архитектуры. Я создавал закрытые и общедоступные веб-сервисы и веб-API для веб-приложений и приложений на стороне сервера. Все эти годы я много жаловался на ужасные программные интерфейсы – и попадал в ловушки, создавая точно такие же.

Шли годы, и технология развивалась, переходя от RPC к веб-сервисам SOAP и веб-API. Объединять программное обеспечение становилось все проще и проще с технической точки зрения. Но независимо от используемой технологии, я понял, что программный интерфейс – это гораздо больше, чем просто связующая система или побочный продукт программного проекта. После посещения первых конференций по API в 2014 году на API Days в Париже я понял, что есть множество других людей, которые, как и я, сражаются с API. Вот почему в 2015 году я начал вести блог API Handyman, а также стал участвовать в конференциях, посвященных API. Я хотел поделиться своим опытом с другими и помочь им избежать попадания в те же ловушки, в которые попал я. Когда я писал о веб-API и обсуждал их, это не только позволяло мне помогать другим, но и давало мне возможность узнать о них еще больше.

После двух лет ведения блогов и выступлений на конференциях я пришел к решению написать книгу. Я хотел адресовать ее прежде всего себе – тому, кто испытывал столько затруднений. К счастью, издательство Manning Publications искало человека, готового написать книгу о спецификации OpenAPI, формате описания API (об этом мы поговорим в главе 4). Я воспользовался шансом, предложив свою книгу *Design of Everyday APIs*, и ее приняли. На это название меня вдохновила книга «Дизайн привычных вещей» Дона Нормана (MIT Press, 1998) – вам обязательно стоит прочитать ее. Позже первоначальный заголовок заменили более простым, *The Design of Web APIs*. Должен признать, что он мне нравится больше; теперь у меня нет ощущения, что я покушаюсь на лавры Дона Нормана.

Поначалу в книге освещалось проектирование повседневных вещей + API + REST в сравнении с gRPC в сравнении с GraphQL. Усвоить все это было бы довольно трудно, но я хотел, чтобы изложенные в книге принципы можно было использовать для любого типа API. Месяц за месяцем содержание совершенствовалось, в результате чего получилась книга, которая сейчас называется *The Design of Web APIs* («Проектирование веб-API»). Я решил сосредоточиться на REST API и использовать их в качестве примера для изучения принципов проектирования веб- и удаленных API, что выходит за рамки простого проектирования API. Если бы такая книга попала в прежние времена, я был бы очень рад ее изучить; надеюсь, вам тоже понравится!

Благодарности

Два года. Мне потребовалось два года, чтобы написать книгу. Да, это немало, за-то в итоге получилась замечательная книга, которая, надеюсь, вам понравится. Я работал над ней не в одиночку. Мне многих нужно поблагодарить за непосредственную помощь в работе, но не меньше я благодарен и тем, кто эту работу сделал возможной. Прежде всего я благодарен своей жене Синзии и дочери Элизабетте. Большое вам спасибо за вашу поддержку и терпение, пока я проводил вечера и выходные за написанием книги. Я очень вас люблю.

Далее я хотел бы поблагодарить всех сотрудников издательства Manning Publications. Многие даже не представляют, сколько людей работает над книгой с самого начала ее создания! Каждый из этих людей проделал замечательную работу. Главным образом я бы хотел поблагодарить моего редактора Майка Стивенса, который поверил в этот проект. Особая благодарность двум редакторам – консультантам по аудитории, Кевину Харрелду и Дженнифер Стаут, и техническому редактору Майклу Ланду. Вы действительно очень помогли мне! Без вашего участия эта книга не стала бы такой, какой ее видит читатель. И отдельное merci beaucoup моему редактору текста из ESL Рэйчел Хэд, которая проделала поистине потрясающую работу, корректируя мой «французский английский». Спасибо производственному редактору Дейрдре Хайам, редактору Фрэнсис Буран, корректору Мелоди Долаб и техническому редактору Полу Гребенцу. Также я хотел бы поблагодарить своих рецензентов: Эндрю Гвоздзевич, Энди Кирша, Бриджера Хауэлла, Эвина Квока, Энрико Маццареллу, Мохаммада Али Бацци, Нараянан Джаяратчаган, Питера Пола Селларса, Равиша Шарму, Санджива Кумара Джайсвала, Серхио Пачеко, Шона Хиксона, Шона Смита, Винсента Терона и Уильяма Руденмальма.

Особая благодарность Ивану Гончарову, переславшему мне 15 марта 2017 года электронное письмо от издательства Manning Publications. Оно искало человека, который написал бы книгу об API, в результате чего

и появилась на свет «Проектирование веб-API». Я рад, что по счастливой случайности мы встретились на REST Fest 2015.

Спасибо всем, кто нашел время, чтобы прочитать рукопись на разных этапах ее подготовки и предоставил неоценимую поддержку и отзывы. Отдельная благодарность Изабель Реуса и Мехди Меджауи за «полевое тестирование» содержания книги и за обратную связь. И спасибо всем тем, кто занимается проектированием API, – людям, с которыми я встречался и работал на протяжении многих лет. Я вложил в эту книгу все, чему научился у вас!

И еще один раз поблагодарю Майка Амундсена, Кина Лэйна и Мехди Меджауи, теперь уже за поддержку и помощь в то время, когда я начал вести блог API Handyman (в 2015 году). Без вас этой книги бы не было.

Об этой книге

Эта книга была написана, для того чтобы помочь вам проектировать веб-API, которые не просто удовлетворяют выраженные потребности. Книга поможет вам проектировать выдающиеся веб-API, которые могут использоваться любым человеком в самых разных контекстах и которые также являются безопасными, долговечными, расширяемыми, эффективными и реализуемыми. В ней раскрываются все аспекты проектирования веб-API и дается полный обзор экосистемы веб-API и того, как проектировщики API могут внести в нее свой вклад.

Кому адресована эта книга?

Данная книга, очевидно, предназначена для всех тех, кому необходимо проектировать веб-API. Это могут быть разработчики, работающие над серверной частью для мобильных приложений или веб-сайтов, или те, кому нужно соединять микросервисы воедино, или же это могут быть владельцы продуктов, работающие над API в качестве продукта и всем, что находится посередине. На самом деле эту книгу могут прочитать все те, кто работает над проектом, связанным с созданием API.

Как устроена эта книга: дорожная карта

Книга состоит из трех частей, которые охватывают 13 глав.

В первой части представлены основные понятия и навыки, необходимые для проектирования API.

В главе 1 обсуждается, что такое API, чем важно его проектирование, и что из себя представляют элементы проектирования API.

В главе 2 объясняется, как точно определить назначение API – его реальные цели, – сосредоточив внимание на точке зрения пользователей API и программного обеспечения, использующего API, и избегая точки зрения организации и программного обеспечения, предоставляющего API.

Глава 3 знакомит вас с протоколом HTTP, REST API и архитектурным стилем REST. Она учит, как проектировать интерфейс веб-программирования (содержащий ресурсы, действия над ресурсами, данные, параметры и ответы) на основе определенных целей.

Глава 4 знакомит вас со спецификацией OpenAPI и демонстрирует, как описывать API структурированным и стандартным способом, используя такой формат описания API.

Вторая часть посвящена тому, как проектировать API типа «не заставляйте меня думать», которые будут просты для понимания и использования.

В главе 5 объясняется, как проектировать простые представления данных, сообщения об ошибках и успешных результатах, а также потоки вызовов API, которые сразу же будут понятны, и их легко будет использовать.

В главе 6 рассказывается, как создавать еще более простые для понимания и удобные в использовании API-интерфейсы, пользователи которых (люди или машины) смогут угадать, как работают API, делая их согласованными, адаптируемыми и обнаруживаемыми.

В главе 7 показано, как организовать и оценить все аспекты API, чтобы их было легко понять и использовать.

В третьей части показано, что проектировщики API должны учитывать контекст, окружающий API, и весь контекст, окружающий сам процесс проектирования API.

В главе 8 описывается безопасность API и то, как проектировать безопасные API.

В главе 9 рассказывается, как изменить API, не оказывая чрезмерного влияния на его пользователей, и как и когда использовать управление версиями. В ней также демонстрируется, как проектировать API, которые будут легко расширять с нуля.

Глава 10 посвящена тому, как создавать веб-API для эффективной работы в сети.

Глава 11 раскрывает весь контекст, который должны учитывать проектировщики при проектировании API. Она включает в себя адаптацию механизмов обмена данными (запрос/ответы, асинхронность, события и пакетную обработку), оценку и адаптацию к ограничениям потребителей или поставщиков и выбор адекватного стиля API (на базе ресурсов, функций или данных).

В главе 12 объясняется, как проектировщики API участвуют в создании различных типов документации API, используя преимущества такого формата описания API как спецификацию OpenAPI.

В главе 13 показано, как проектировщики могут участвовать в развитии множества API, принимая участие в жизненном цикле API. Особое внимание уделяется руководству по проектированию API и обзору.

Эту книгу следует читать от корки до корки, каждую главу по порядку. Каждая новая глава расширяет то, что было изучено в предыдущих. При этом, как только вы закончите главы 1, 2 и 3, вы можете перейти к любой главе, посвященной теме, которую вам необходимо срочно исследовать.

О коде

В данной книге содержится множество примеров исходного кода как в пронумерованных списках, так и в обычном тексте. В обоих случаях исходный код форматируется шрифтом фиксированной ширины наподобие этого, чтобы отделить его от обычного текста. Иногда код также печатается **жирным шрифтом**, чтобы выделить код, который изменился по сравнению с предыдущим вариантом, например когда в существующую строку кода добавляется новая функция.

Во многих случаях оригинальный исходный код был переформатирован; мы добавили разрывы строк и переработали отступы, чтобы разместить доступное пространство страницы в книге. В редких случаях даже этого было недостаточно, и листинги содержат маркеры продолжения строки (↵). Кроме того, комментарии в исходном коде часто удаляются из листингов, когда описание кода приводится в тексте. Многие листинги сопровождаются кодовыми аннотациями для выделения важных понятий.

Исходный код примеров из этой книги доступен для скачивания с веб-сайта издателя по адресу <https://www.manning.com/books/the-design-of-web-apis>.

Дискуссионный форум liveBook

Покупка книги *The Design of Web APIs* дает право свободного доступа к закрытому веб-форуму издательства Manning Publication, где можно высказать свои замечания о книге, задать технические вопросы и получить помощь от авторов и других пользователей. Чтобы получить доступ к форуму, перейдите по ссылке <https://livebook.manning.com/book/the-design-of-everyday-apis/welcome/v-11/discussion>. Вы также можете узнать больше о форумах издательства и правилах поведения на них по адресу <https://livebook.manning.com/discussion>.

Издательство Manning обязуется предоставить читателям площадку для содержательного диалога не только между читателями, но и между читателями и авторами. Но авторы не обязаны выделять определенное количество времени для участия, т. к. их вклад в работу форума является добровольным (и неоплачиваемым). Мы предлагаем читателям задавать авторам действительно непростые вопросы, чтобы их интерес не угасал! Форум и архив предыдущих обсуждений будут доступны на веб-сайте издательства, пока книга находится в печати.

Другие интернет-ресурсы

Существует множество онлайн-ресурсов об API. Вот два моих любимых:

- *API Developer Weekly Newsletter* (<https://apideveloperweekly.com/>) – лучший способ узнать, что происходит в мире API и открыть для себя новые источники информации об API;
- сайт Web API Events (<https://webapi.events/>), он будет информировать вас о предстоящих конференциях, посвященных API.

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге, – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв прямо на нашем сайте www.dmkpress.com, зайдя на страницу книги, и оставить комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com, при этом напишите название книги в теме письма.

Если есть тема, в которой вы квалифицированы, и вы заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Скачивание исходного кода примеров

Скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте www.dmkpress.com или www.дмк.рф на странице с описанием соответствующей книги.

Список опечаток

Хотя мы приняли все возможные меры, для того чтобы удостовериться в качестве наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг – возможно, ошибку в тексте или в коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от расстройств и поможете нам улучшить последующие версии этой книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу dmkpress@gmail.com, и мы исправим это в следующих тиражах.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Manning очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконно выполненной копией любой нашей книги, пожалуйста, сообщите нам адрес копии или веб-сайта, чтобы мы могли применить санкции.

Пожалуйста, свяжитесь с нами по адресу dmkpress@gmail.com со ссылкой на подозрительные материалы.

Мы высоко ценим любую помощь по защите наших авторов, помогающую нам предоставлять вам качественные материалы.

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru