

УДК 004.432

ББК 32.972.1

Л78

Л78 Арно Лоре

Проектирование веб-API / Пер. с англ. Д. А. Беликова. – М.: ДМК Пресс, 2020. – 440 с.

ISBN 978-5-97060-861-6

Книга, написанная с учетом многолетнего опыта автора в разработке API, научит вас, как собирать требования, как найти баланс между техническими и бизнес-целями и как принимать во внимание запросы потребителя. Рассматриваются основные характеристики API, принципы его изменения, документирования и проверки. Эффективные методы разработки проиллюстрированы множеством интересных примеров.

Рассматриваются основные характеристики API, принципы его изменения, документирования и проверки. Эффективные методы разработки проиллюстрированы множеством интересных примеров.

Издание предназначено для разработчиков, обладающих минимальным опытом в создании и использовании API-интерфейсов.

УДК 004.432

ББК 32.972.1

Original English language edition published by Manning Publications USA, USA. Copyright © 2018
Russian-language edition copyright © 2020 by DMK Press. All rights reserved.

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.

ISBN 978-5-97060-861-6 (рус.)

ISBN 978-1-61729-510-2 (анг.)

© 2019 by Manning Publications Co.

© Оформление, издание, ДМК Пресс, 2020

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	Потоковая 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
12.2.1	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