

O'REILLY®

Профессиональный TypeScript

Разработка масштабируемых
JavaScript-приложений



Борис Черный

Programming TypeScript

Making Your JavaScript Applications Scale

Boris Cherny

Beijing • Boston • Farnham • Sebastopol • Tokyo

O'REILLY®

Профессиональный TypeScript

Разработка масштабируемых
JavaScript-приложений

Борис Черный



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

2021

ББК 32.988.02-018
УДК 004.738.5
Ч-49

Борис Черный

Ч-49 Профессиональный TypeScript. Разработка масштабируемых JavaScript-приложений. — СПб.: Питер, 2021. — 352 с.: ил. — (Серия «Бестселлеры O'Reilly»).

ISBN 978-5-4461-1651-5

Любой программист, работающий с языком с динамической типизацией, подтвердит, что задача масштабирования кода невероятно сложна и требует большой команды инженеров. Вот почему Facebook, Google и Microsoft придумали статическую типизацию для динамически типизированного кода.

Работая с любым языком программирования, мы отслеживаем исключения и вычитываем код строку за строкой в поиске неисправности и способа ее устранения. TypeScript позволяет автоматизировать эту неприятную часть процесса разработки.

TypeScript, в отличие от множества других типизированных языков, ориентирован на прикладные задачи. Он вводит новые концепции, позволяющие выражать идеи более кратко и точно, и легко создавать масштабируемые и безопасные современные приложения.

Борис Черный помогает разобраться со всеми нюансами и возможностями TypeScript, учит устранять ошибки и масштабировать код.

16+ (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 O'Reilly. 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.

Издательство не несет ответственности за доступность материалов, ссылки на которые вы можете найти в этой книге. На момент подготовки книги к изданию все ссылки на интернет-ресурсы были действующими.

ISBN 978-1492037651 англ. Authorized Russian translation of the English edition of Programming TypeScript ISBN 9781492037651 © 2019 Boris Cherny. This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

ISBN 978-5-4461-1651-5 © Перевод на русский язык
ООО Издательство «Питер», 2021
© Издание на русском языке,
оформление ООО Издательство «Питер», 2021
© Серия «Бестселлеры O'Reilly», 2021

Краткое содержание

Отзывы	10
Пролог	12
Глава 1. Вступление.....	18
Глава 2. TypeScript с высоты птичьего полета.....	21
Глава 3. Подробно о типах	33
Глава 4. Функции	67
Глава 5. Классы и интерфейсы.....	111
Глава 6. Продвинутое типы	145
Глава 7. Обработка ошибок	198
Глава 8. Асинхронное программирование, конкурентность и параллельная обработка	215
Глава 9. Фронтенд- и бэкенд-фреймворки	247
Глава 10. Пространства имен и модули	265
Глава 11. Взаимодействие с JavaScript	283
Глава 12. Создание и запуск TypeScript.....	306
Глава 13. Итоги	327
Приложение А. Операторы типов	329
Приложение Б. Утилиты типов.....	330
Приложение В. Область действия деклараций.....	331
Приложение Г. Правила написания файлов деклараций для сторонних модулей JavaScript	333
Приложение Д. Директивы с тремя слешами	342
Приложение Е. Флаги безопасности компилятора TSC	344
Приложение Ж. TSX.....	346
Об авторе.....	349
Об обложке	350

Оглавление

Отзывы	10
Пролог	12
Структура книги	12
Стиль	13
Использование примеров кода	15
Благодарности.....	16
От издательства	17
Глава 1. Вступление.....	18
Глава 2. TypeScript с высоты птичьего полета	21
Компилятор.....	21
Система типов	23
Настройка редактора кода.....	27
index.ts	30
Упражнения к главе 2.....	32
Глава 3. Подробно о типах	33
О типах	34
Типы от а до я.....	35
Итоги.....	65
Упражнения к главе 3.....	66
Глава 4. Функции	67
Объявление и вызов функций.....	67
Полиморфизм.....	90
Разработка на основе типов	108
Итоги.....	109
Упражнения к главе 4.....	110

Глава 5. Классы и интерфейсы	111
Классы и наследование	111
super	117
Использование this в качестве возвращаемого типа	117
Интерфейсы	119
Классы структурно типизированы.....	126
Классы объявляют и значения, и типы	127
Полиморфизм	131
Примеси	132
Декораторы.....	135
Имитация финальных классов	138
Паттерны проектирования.....	139
Итоги.....	142
Упражнения к главе 5.....	144
Глава 6. Продвинутое типы.....	145
Связи между типами.....	145
Тотальность	165
Продвинутое типы объектов	167
Продвинутое функциональные типы	177
Условные типы	180
Запасные решения	185
Имитация номинальных типов	191
Безопасное расширение прототипа	193
Итоги.....	196
Упражнения к главе 6.....	197
Глава 7. Обработка ошибок.....	198
Возврат null	199
Выбрасывание исключений	200
Возврат исключений.....	203

Тип Option	205
Итоги.....	213
Упражнение к главе 7.....	214
Глава 8. Асинхронное программирование, конкурентность и параллельная обработка.....	215
Цикл событий	216
Работа с обратными вызовами.....	218
Промисы как здоровая альтернатива.....	221
asunc и await	227
Async-поток	228
Типобезопасная многопоточность	231
Итоги.....	245
Упражнения к главе 8.....	246
Глава 9. Фронтенд- и бэкенд-фреймворки.....	247
Фронтенд-фреймворки	247
Типобезопасные API	260
Бэкенд-фреймворки.....	262
Итоги.....	264
Глава 10. Пространства имен и модули.....	265
Краткая история модулей JavaScript	266
import, export.....	269
Пространства имен	274
Слияние деклараций	279
Итоги.....	281
Упражнение к главе 10.....	282
Глава 11. Взаимодействие с JavaScript	283
Декларации типов	283
Поэтапная миграция из JavaScript в TypeScript	292
Поиск типов для JavaScript	298

Использование стороннего кода JavaScript	301
Итоги.....	305
Глава 12. Создание и запуск TypeScript	306
Создание проекта в TypeScript	306
Запуск TypeScript на сервере	317
Запуск TypeScript в браузере	318
Публикация TypeScript-кода на NPM.....	321
Директивы с тремя слешами.....	322
Итоги.....	326
Глава 13. Итоги.....	327
Приложение А. Операторы типов	329
Приложение Б. Утилиты типов.....	330
Приложение В. Область действия деклараций	331
Генерирует ли декларация тип	331
Допускает ли декларация слияние.....	331
Приложение Г. Правила написания файлов деклараций для сторонних модулей JavaScript.....	333
Типы экспорта	334
Расширение модуля.....	337
Приложение Д. Директивы с тремя слешами	342
Внутренние директивы	343
Нежелательные директивы.....	343
Приложение Е. Флаги безопасности компилятора TSC	344
Приложение Ж. TSX	346
Об авторе	349
Об обложке	350

Отзывы

Отличная книга для углубленного изучения TypeScript. Она демонстрирует все преимущества использования системы типов и помогает обрести уверенность при работе с JavaScript.

*Минко Гечев,
инженер команды Angular в Google*

Книга «Профессиональный TypeScript. Разработка масштабируемых JavaScript-приложений» помогла мне быстро освоить инструменты и внутреннее устройство этого языка. Она дала ответы на все мои вопросы с помощью реальных примеров. Глава «Продвинутые типы» сломала терминологические барьеры и показала, как TypeScript позволяет создать безопасный и удобный код.

*Шон Гров,
сооснователь OneGraph*

Борис создал обширное руководство по TypeScript. Прочтите его вдоль и поперек. А затем еще разочек.

*Блейк Эмбри, инженер в Opendoor,
автор TypeScript Node and Typings
(«Типизации и Node в TypeScript»)*

*Посвящается Саше и Михаилу.
Возможно, и они однажды полюбят тины.*

Пролог

Эта книга предназначена для программистов всех направлений: JavaScript-инженеров, C#-разработчиков, сторонников Java, любителей Python, Ruby и Haskell. На каком бы языке вы ни писали, если вам известны функции, переменные, классы и связанные с ними ошибки, то эта книга для вас. Наличие некоторого опыта в JavaScript, включая базовые знания об объектной модели документа (DOM) и обмене информацией по сети, поможет вам освоить представленный материал. Хотя мы и не сильно углубляемся в эти понятия, на них основаны приведенные примеры.

Работая с любым языком программирования, мы отслеживаем исключения и вычитываем код строку за строкой в поиске неисправности и способа ее устранения. TypeScript позволяет автоматизировать эту неприятную часть процесса разработки.

Если раньше вы не встречались со статической типизацией, то узнаете о типах из этой книги. Я расскажу, как с их помощью снизить риск программного сбоя, улучшить документирование кода и масштабировать его для большего числа пользователей, инженеров и серверов. Без сложных терминов я объясню материал, подключая вашу интуицию и память, и в этом мне помогут примеры.

TypeScript, в отличие от множества других типизированных языков, преимущественно практический. Он вводит новые концепции, позволяющие выражать идеи более кратко и точно и играючи создавать современные приложения безопасным способом.

Структура книги

Я постарался передать вам теоретическое понимание работы TypeScript и достаточное количество практических советов по написанию кода.

TypeScript — практический язык, поэтому в книге теория и практика в основном дополняют друг друга, но в первых двух главах преимущественно освещается теория, а ближе к концу представлена только практика.

Мы рассмотрим такие основы, как компилятор, модуль проверки типов и сами типы. Далее обсудим их разновидности и операторы, после чего перейдем к углубленным темам, таким как особенности системы типов, обработка ошибок и асинхронное программирование. В завершение я расскажу, как использовать TypeScript с вашими любимыми фреймворками (фронтенд и бэкенд), производить миграцию существующего JavaScript-проекта в TypeScript и запускать TypeScript-приложение в продакшене.

Большинство глав завершаются набором упражнений. Попробуйте выполнить их самостоятельно, чтобы лучше усвоить материал. Ответы к ним доступны по адресу <https://github.com/bcherny/programming-typescript-answers>.

Стиль

В коде я старался придерживаться единого стиля, поэтому отмечу, какие мои личные предпочтения он содержит.

- ❑ Точка с запятой только при необходимости.
- ❑ Отступы двумя пробелами.
- ❑ Короткие имена переменных вроде `a`, `f` или `_` там, где программа является фрагментом кода или ее структура важнее деталей.

Я считаю, что некоторых аспектов этого стиля стоит придерживаться и вам. К ним относятся:

- ❑ Использование синтаксиса JavaScript и новейших возможностей этого языка (последняя версия JavaScript обычно называется `esnext`). Так ваш код будет соответствовать последним стандартам и иметь хорошую функциональную совместимость с поисковыми системами. Это поможет снизить временные затраты на набор новых сотрудников и позволит оценить все преимущества таких мощных инструментов, как стрелочные функции, промисы и генераторы.
- ❑ Использование оператора расширения `spread` для длительного сохранения неизменности структур данных¹.

¹ Если у вас нет опыта работы в JavaScript, то вот пример: чтобы объекту `o` добавить свойство `k` со значением `3`, можно либо изменить `o` напрямую — `o.k = 3`, либо применить это изменение к `o`, создав тем самым *новый* объект — `let p = {...o, k: 3}`.

- ❑ Привычка убеждаться, что у любого элемента есть тип, по возможности выведенный. Старайтесь не злоупотреблять явными типами. Так вы сделаете код лаконичным и чистым, а также повысите его безопасность, выявляя неверные типы, а не используя временные решения.
- ❑ Стремление сохранить код переиспользуемым и обобщенным (см. раздел «Полиморфизм» на с. 90).

Конечно, все это не ново, но для TypeScript имеет особую актуальность. Его встроенный низкоуровневый компилятор, поддержка типов `read-only` (только для чтения), мощный интерфейс типов, глубокая поддержка полиморфизма и полностью структурная система типов позволяют добиться хорошего стиля, в то время как сам язык остается выразительным и верным для лежащего в его основе JavaScript.

Еще пара заметок до перехода к основному материалу.

JavaScript не предоставляет указатели и ссылки. Вместо них используются значения и ссылочные типы. Значения являются неизменными и включают такие элементы, как *strings* (строки), *numbers* (числа) и *booleans* (логические значения), в то время как ссылочные типы часто указывают на изменяемые структуры данных вроде массивов, объектов и функций. Когда я использую слово «значение», то обычно имею в виду либо значение JavaScript, либо ссылку.

И последнее. Написание идеального TypeScript-кода затрудняется взаимодействием с JavaScript, некорректно типизированными сторонними библиотеками и наследованным кодом. Также ему мешает спешка. В книге я буду часто советовать вам избегать компромиссов. В реальности достаточную корректность кода будете определять только вы и ваша команда.

Типографские соглашения

Ниже приведен список используемых обозначений.

Курсив

Используется для обозначения новых терминов.

Моноширинный

Применяется для оформления листингов программ и программных элементов в обычном тексте, таких как имена переменных и функций, базы данных, типы данных, переменные окружения, инструкции и ключевые слова.

Моноширинный жирный

Обозначает команды или другой текст, который должен вводиться пользователем. Иногда используется в листингах для привлечения внимания.



Так обозначаются примечания общего характера.



Так выделяются советы и предложения.



Так обозначаются предупреждения и предостережения.

Использование примеров кода

Вспомогательный материал (примеры кода, упражнения и пр.) доступен для загрузки по адресу: <https://github.com/bcherny/programming-typescript-answers>.

В общем случае все примеры кода из этой книги вы можете использовать в своих программах и в документации. Вам не нужно обращаться в издательство за разрешением, если вы не собираетесь воспроизводить существенные части программного кода. Например, если вы разрабаты-

ваете программу и используете в ней несколько отрывков программного кода из книги, вам не нужно обращаться за разрешением. Однако в случае продажи или распространения компакт-дисков с примерами из этой книги вам следует получить разрешение от издательства O'Reilly. Если вы отвечаете на вопросы, цитируя данную книгу или примеры из нее, получение разрешения не требуется. Но при включении существенных объемов программного кода примеров из этой книги в вашу документацию вам необходимо будет получить разрешение издательства.

За получением разрешения на использование значительных объемов программного кода примеров из этой книги обращайтесь по адресу permissions@oreilly.com.

Благодарности

Написание этой книги заняло год и потребовало множества бессонных ночей, ранних подъемов и потерянных выходных. В ее основе лежат многолетние наработки — сниппеты и заметки, собранные воедино, доработанные и обогащенные теорией.

Хочу поблагодарить O'Reilly за предоставленную возможность работы над книгой и лично редактора Анжелу Руфино за поддержку на протяжении всего процесса. Благодарю Ника Нэнса за помощь в создании раздела «Типобезопасные API» и Шьяма Шешадри за помощь с подразделом «Angular» (см. с. 256). Спасибо моим научным редакторам: Даниэлю Розенвассеру из команды TypeScript, который провел огромное количество времени, вычитывая рукопись и знакомя меня с нюансами системы типов, а также Джонатану Кримеру, Якову Файну, Полу Байингу и Рэйчел Хэд за технические правки и обратную связь. Спасибо моей семье — Лизе и Илье, Вадиму, Розе и Алику, Фаине и Иосифу — за вдохновение на реализацию этого проекта.

Самую большую благодарность выражаю Саре Гилфорд, которая поддерживала меня на протяжении всего процесса написания, даже когда отменялись планы на вечер и выходные или я внезапно начинал рассуждать на тему различных деталей системы типов. Я бесконечно благодарен ей за поддержку, без которой точно не справился бы.

От издательства

Ваши замечания, предложения, вопросы отправляйте по адресу comp@piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На веб-сайте издательства www.piter.com вы найдете подробную информацию о наших книгах.