

Рецепты TypeScript

Программирование на уровне типов для реальных задач

Стефан Баумгартнер

SPRINT
book

2025

Оглавление

Предисловие	11
Введение	13
Кому стоит прочесть эту книгу	14
Цели написания книги	14
Структура издания	15
Условные обозначения	17
Использование примеров кода	18
От издательства	19
Благодарности	20
Глава 1. Настройка проекта	22
1.1. Проверка типов JavaScript	23
1.2. Установка TypeScript	27
1.3. Поддержка типов «в стороне»	30
1.4. Миграция проекта на TypeScript	32
1.5. Загрузка типов из репозитория Definitely Typed	35
1.6. Настройка полнофункционального проекта	38
1.7. Настройка тестов	43
1.8. Типизация модулей ECMAScript из URL	46
1.9. Загрузка различных типов модулей в Node	50
1.10. Работа с Deno и зависимостями	53
1.11. Использование предопределенных конфигураций	56
Глава 2. Основные типы	58
2.1. Эффективное аннотирование	58
2.2. Работа с any и unknown	62
2.3. Выбор правильного типа объекта	65
2.4. Работа с типами кортежей	68
2.5. Различия между интерфейсами и псевдонимами типов	71
2.6. Определение перегрузок функций	74

2.7. Определение типов параметра <code>this</code>	77
2.8. Работа с <code>Symbol</code>	80
2.9. Понимание пространств имен значений и типов	83
Глава 3. Система типов	87
3.1. Моделирование данных с помощью типов объединений и пересечений	87
3.2. Явное определение моделей с размеченными типами объединения	92
3.3. Проверка полноты с помощью приема <code>assertNever</code>	95
3.4. Закрепление типов с помощью <code>Const Context</code>	99
3.5. Сужение типов с помощью предикатов типа	102
3.6. Понимание <code>void</code>	105
3.7. Работа с типами ошибок в предложениях <code>catch</code>	108
3.8. Создание моделей исключаящего ИЛИ с помощью приема <code>optional never</code>	111
3.9. Эффективное использование утверждений типа	114
3.10. Использование сигнатур индекса	117
3.11. Различение отсутствующих свойств и значений <code>undefined</code>	122
3.12. Работа с перечислениями	124
3.13. Определение номинальных типов в системе структурных типов	129
3.14. Включение свободного автозаполнения для подмножеств строк	133
Глава 4. Обобщенные типы	136
4.1. Обобщение сигнатур функций	136
4.2. Создание связанных аргументов функций	139
4.3. Избавление от <code>any</code> и <code>unknown</code>	143
4.4. Создание экземпляров обобщенного типа	146
4.5. Создание новых типов объектов	151
4.6. Изменение объектов с помощью сигнатур утверждений	156
4.7. Сопоставление типов с помощью карт типов	159
4.8. Использование <code>ThisType</code> для определения <code>this</code> в объектах	164
4.9. Добавление контекста <code>const</code> к параметрам обобщенного типа	168
Глава 5. Условные типы	173
5.1. Управление сигнатурами сложных функций	173
5.2. Фильтрация с помощью типа <code>never</code>	178
5.3. Группировка элементов по типу	182
5.4. Удаление определенных свойств объекта	188
5.5. Выведение типов в условных выражениях	191

Глава 6. Типы литералов шаблонов строк	197
6.1. Определение пользовательской системы событий.....	198
6.2. Создание обратных вызовов событий с помощью типов манипуляций со строками и переназначения ключей	200
6.3. Написание функции форматирования	204
6.4. Извлечение типов параметров формата.....	208
6.5. Работа с ограничениями рекурсии	211
6.6. Использование литералов шаблонов строк в качестве дискриминантов	215
Глава 7. Вариативные типы кортежей.....	219
7.1. Типизация функции concat.....	219
7.2. Типизация функции promisify	223
7.3. Типизация функции curry	227
7.4. Типизация гибкой функции curry	230
7.5. Типизация простейшей функции curry.....	234
7.6. Создание перечисления из кортежа	237
7.7. Разделение всех элементов сигнатуры функции.....	241
Глава 8. Вспомогательные типы	243
8.1. Установка определенных необязательных свойств.....	244
8.2. Изменение вложенных объектов	247
8.3. Переназначение типов	251
8.4. Получение всех необходимых ключей.....	254
8.5. Разрешение хотя бы одного свойства	257
8.6. Разрешение ровно одного свойства, всех свойств или ни одного	262
8.7. Преобразование типов объединения в типы пересечения	267
8.8. Использование библиотеки type-fest	274
Глава 9. Стандартная библиотека и определения внешних типов.....	279
9.1. Перебор объектов с помощью Object.keys	280
9.2. Явное выделение небезопасных операций с помощью утверждений типов и unknown	284
9.3. Работа с defineProperty	286
9.4. Расширение типов для Array.prototype.includes	292
9.5. Фильтрация нулевых значений.....	296
9.6. Расширение модулей.....	298
9.7. Расширение глобальных переменных	300
9.8. Добавление модулей, которые отличаются от JS, в граф модулей.....	304

Глава 10. TypeScript и React	308
10.1. Написание прокси-компонентов.....	309
10.2. Написание контролируемых компонентов	312
10.3. Типизация пользовательских хуков	315
10.4. Типизация обобщенных компонентов forwardRef.....	317
10.5. Предоставление типов для контекстного API.....	322
10.6. Типизация компонентов высшего порядка	327
10.7. Типизация обратных вызовов в синтетической системе событий React	330
10.8. Типизация полиморфных компонентов	333
Глава 11. Классы	338
11.1. Выбор правильного модификатора видимости.....	339
11.2. Явное переопределение методов.....	344
11.3. Описание конструкторов и прототипов	348
11.4. Использование обобщенных типов в классах	352
11.5. Решение об использовании классов или пространства имен.....	355
11.6. Написание статических классов	360
11.7. Работа со строгой инициализацией свойств	364
11.8. Работа с типами this в классах.....	369
11.9. Декораторы	373
Глава 12. Стратегии разработки типов	380
12.1. Написание простых в сопровождении типов.....	381
12.2. Поэтапное уточнение типов	384
12.3. Проверка контрактов с помощью satisfies	395
12.4. Тестирование сложных типов	399
12.5. Проверка типов данных во время выполнения с помощью Zod	402
12.6. Обход ограничений доступа к индексу	407
12.7. Принятие решения о том, использовать ли перегрузку функций или условные типы	410
12.8. Именованное обобщенных типов	417
12.9. Прототипирование в TypeScript Playground.....	419
12.10. Предоставление нескольких версий библиотеки	424
12.11. Знать, когда остановиться.....	428
Об авторе	430
Иллюстрация на обложке.....	431