

TYPESCRIPT

БЫСТРО

ЯКОВ ФАЙН
АНТОН МОИСЕЕВ



 MANNING

ЯКОВ ФАЙН
АНТОН МОИСЕЕВ

TYPESCRIPT БЫСТРО



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

2021

ББК 32.988.02-018
УДК 004.738.5
Ф17

Файн Яков, Моисеев Антон

Ф17 TypeScript быстро. — СПб.: Питер, 2021. — 528 с.: ил. — (Серия «Для профессионалов»).

ISBN 978-5-4461-1725-3

«TypeScript быстро» научит вас секретам продуктивной разработки веб- или самостоятельных приложений. Она написана практиками для практиков.

В книге разбираются актуальные для каждого программиста задачи, объясняется синтаксис языка и описывается разработка нескольких приложений, в том числе нетривиальных — так вы сможете понять, как использовать TypeScript с популярными библиотеками и фреймворками. Вы разберетесь с превосходным инструментарием TypeScript и узнаете, как объединить в одном проекте TypeScript и JavaScript. Среди продвинутых тем, рассмотренных авторами, – декораторы, асинхронная обработка и динамические импорты. Прочитав эту книгу, вы поймете, что именно делает TypeScript особенным.

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 Manning Publications. 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.

№ 978-1617295942 англ.
| 978-5-4461-1725-3

© 2020 by Manning Publications Co. All rights reserved.
© Перевод на русский язык ООО Издательство «Питер», 2021
© Издание на русском языке, оформление ООО Издательство «Питер», 2021
© Серия «Для профессионалов», 2021

Краткое содержание

| | |
|--------------------|----|
| Введение..... | 15 |
| Благодарности..... | 17 |
| О книге..... | 18 |

ЧАСТЬ 1

ОСНОВЫ СИНТАКСИСА TYPESCRIPT

| | |
|---|-----|
| Глава 1. Знакомство с TypeScript..... | 24 |
| Глава 2. Базовые и пользовательские типы..... | 39 |
| Глава 3. Объектно-ориентированное программирование с классами и интерфейсами..... | 67 |
| Глава 4. Перечисления и обобщенные типы..... | 98 |
| Глава 5. Декораторы и продвинутые типы..... | 124 |
| Глава 6. Инструменты..... | 152 |
| Глава 7. Использование TypeScript и JavaScript в одном проекте..... | 188 |

ЧАСТЬ 2

ИСПОЛЬЗОВАНИЕ TYPESCRIPT В БЛОКЧЕЙН-ПРИЛОЖЕНИИ

| | |
|---|-----|
| Глава 8. Разработка собственного блокчейн-приложения..... | 214 |
| Глава 9. Разработка узла блокчейна на основе браузера | 238 |
| Глава 10. Клиент-серверное взаимодействие посредством Node.js, TypeScript и WebSocket | 266 |
| Глава 11. Разработка приложений Angular с помощью TypeScript..... | 317 |
| Глава 12. Разработка клиента блокчейна на Angular | 352 |
| Глава 13. Разработка приложений React.js с помощью TypeScript..... | 367 |
| Глава 14. Разработка блокчейн-клиента в React.js | 405 |
| Глава 15. Разработка приложений Vue.js с помощью TypeScript | 433 |
| Глава 16. Разработка блокчейн-клиента на Vue.js | 464 |
| Приложение А. Современный JavaScript | 486 |

Оглавление

| | |
|----------------------------|-----------|
| Введение | 15 |
| Благодарности | 17 |
| О книге | 18 |
| Для кого эта книга | 18 |
| Структура книги | 18 |
| О коде..... | 20 |
| Об авторах | 21 |
| Об обложке..... | 22 |

Часть 1

Основы синтаксиса TypeScript

| | |
|--|-----------|
| Глава 1. Знакомство с TypeScript | 24 |
| 1.1. Зачем программировать в TypeScript..... | 24 |
| 1.2. Типичные рабочие процессы TypeScript..... | 29 |
| 1.3. Использование компилятора TypeScript | 31 |
| 1.4. Знакомство с Visual Studio Code | 35 |
| Итоги..... | 38 |
| Глава 2. Базовые и пользовательские типы..... | 39 |
| 2.1. Объявление переменных с типами | 40 |
| 2.1.1. Базовые аннотации типов | 41 |
| 2.1.2. Типы в объявлениях функций..... | 45 |
| 2.1.3. Объединенный тип | 47 |

8 Оглавление

| | |
|---|-----------|
| 2.2. Определение пользовательских типов | 50 |
| 2.2.1. Использование <code>type</code> | 50 |
| 2.2.2. Использование классов в качестве пользовательских типов | 52 |
| 2.2.3. Интерфейсы в качестве пользовательских типов..... | 54 |
| 2.2.4. Структурная система типов против номинальной | 57 |
| 2.2.5. Пользовательские объединения типов | 60 |
| 2.3. Типы <code>any</code> и <code>unknown</code> , а также пользовательские защиты типов..... | 62 |
| 2.4. Мини-проект..... | 64 |
| Итоги..... | 65 |
| Глава 3. Объектно-ориентированное программирование с классами и интерфейсами | 67 |
| 3.1. Работа с классами..... | 68 |
| 3.1.1. Знакомство с наследованием классов..... | 68 |
| 3.1.2. Модификаторы доступа <code>public</code> , <code>private</code> , <code>protected</code> | 70 |
| 3.1.3. Статические переменные и пример Одиночки..... | 73 |
| 3.1.4. Метод <code>super()</code> и ключевое слово <code>super</code> | 76 |
| 3.1.5. Абстрактные классы..... | 78 |
| 3.1.6. Перегрузка метода | 81 |
| 3.2. Работа с интерфейсами | 87 |
| 3.2.1. Обеспечение выполнения контракта | 88 |
| 3.2.2. Расширение интерфейсов | 90 |
| 3.2.3. Программирование через интерфейсы..... | 92 |
| Итоги | 96 |
| Глава 4. Перечисления и обобщенные типы | 98 |
| 4.1. Использование <code>enums</code> | 98 |
| 4.1.1. Численные значения <code>enums</code> | 99 |
| 4.1.2. Строковые перечисления | 102 |
| 4.1.3. Использование перечислений <code>const</code> | 105 |
| 4.2. Использование обобщений..... | 106 |
| 4.2.1. Разъяснение обобщений | 106 |
| 4.2.2. Создание собственных обобщенных типов..... | 112 |

| | |
|---|------------|
| 4.2.3. Создание обобщенных функций..... | 116 |
| 4.2.4. Обеспечение возвращаемого типа функции высшего порядка ... | 121 |
| Итоги..... | 123 |
| Глава 5. Декораторы и продвинутые типы | 124 |
| 5.1. Декораторы..... | 125 |
| 5.1.1. Создание декораторов классов..... | 127 |
| 5.1.2. Создание декораторов методов | 133 |
| 5.2. Отображенные типы | 135 |
| 5.2.1. Отображенный тип Readonly..... | 135 |
| 5.2.2. Объявление собственных отображенных типов | 140 |
| 5.2.3. Другие встроенные отображенные типы | 141 |
| 5.3. Условные типы | 144 |
| 5.3.1. Ключевое слово infer..... | 148 |
| Итоги..... | 151 |
| Глава 6. Инструменты..... | 152 |
| 6.1. Карты кода..... | 153 |
| 6.2. Линтер TSLint..... | 156 |
| 6.3. Связывание кода с помощью Webpack | 159 |
| 6.3.1. Связывание JavaScript с помощью Webpack | 161 |
| 6.3.2. Связывание TypeScript с помощью Webpack | 166 |
| 6.4. Использование компилятора Babel | 171 |
| 6.4.1. Использование Babel с JavaScript | 175 |
| 6.4.2. Использование Babel с TypeScript | 177 |
| 6.4.3. Использование Babel с TypeScript и Webpack | 180 |
| 6.5. Инструменты для рассмотрения..... | 182 |
| 6.5.1. Знакомство с Deno..... | 182 |
| 6.5.2. Знакомство с псс..... | 184 |
| Итоги..... | 187 |
| Глава 7. Использование TypeScript и JavaScript в одном проекте | 188 |
| 7.1. Файлы определений типов | 188 |
| 7.1.1. Знакомство | 189 |

| | |
|---|-----|
| Глава 10. Клиент-серверное взаимодействие посредством Node.js, TypeScript и WebSocket..... | 266 |
| 10.1. Разрешение конфликтов с помощью правила длиннейшей цепочки..... | 267 |
| 10.2. Добавление сервера в блокчейн | 270 |
| 10.3. Структура проекта..... | 271 |
| 10.4. Файлы конфигураций проекта | 273 |
| 10.4.1. Настройка компиляции TypeScript..... | 273 |
| 10.4.2. Что находится в package.json | 275 |
| 10.4.3. Настройка nodemon | 276 |
| 10.4.4. Выполнение блокчейн-приложения | 277 |
| 10.5. Краткое знакомство с WebSockets | 284 |
| 10.5.1. Сравнение протоколов HTTP и WebSocket..... | 285 |
| 10.5.2. Передача данных от сервера Node к простому клиенту | 286 |
| 10.6. Рассмотрение процессов уведомления..... | 291 |
| 10.6.1. Рассмотрение кода сервера..... | 294 |
| 10.6.2. Рассмотрение кода клиента..... | 304 |
| Итоги..... | 315 |
| Глава 11. Разработка приложений Angular с помощью TypeScript..... | 317 |
| 11.1. Генерация и запуск нового приложения с помощью Angular CLI..... | 319 |
| 11.2. Рассмотрение сгенерированного приложения | 322 |
| 11.3. Сервисы Angular и внедрение зависимостей | 328 |
| 11.4. Приложение с внедрением ProductService..... | 332 |
| 11.5. Программирование через абстракции в TypeScript | 336 |
| 11.6. Начало работы с HTTP-запросами | 337 |
| 11.7. Начало работы с формами..... | 342 |
| 11.8. Основы маршрутизации..... | 346 |
| Итоги..... | 351 |
| Глава 12. Разработка клиента блокчейна на Angular | 352 |
| 12.1. Запуск блокчейн-приложения Angular | 352 |
| 12.2. Обзор AppComponent..... | 355 |
| 12.3. Рассмотрение Transaction Form Component..... | 359 |

| | |
|---|-----|
| 12.4. Обзор Block Component..... | 361 |
| 12.5. Обзор сервисов | 363 |
| Итоги..... | 366 |
| Глава 13. Разработка приложений React.js с помощью TypeScript..... | 367 |
| 13.1. Разработка простейшей веб-страницы при помощи React | 368 |
| 13.2. Генерация и запуск нового приложения с помощью Create React App..... | 371 |
| 13.3. Управление состоянием компонента | 377 |
| 13.3.1. Добавление состояния в компоненты, основанные на классе | 377 |
| 13.3.2. Использование хуков для управления состоянием..... | 379 |
| 13.4. Разработка метеоприложения..... | 382 |
| 13.4.1. Добавление хука состояния в компонент App..... | 383 |
| 13.4.2. Получение данных при помощи хука useEffect в компоненте App..... | 387 |
| 13.4.3. Использование свойств | 393 |
| 13.4.4. Как дочерний компонент может передавать данные родителю? | 399 |
| 13.5. Что такое виртуальная DOM? | 402 |
| Итоги..... | 404 |
| Глава 14. Разработка блокчейн-клиента в React.js | 405 |
| 14.1. Запуск клиента и сервера обмена сообщениями | 406 |
| 14.2. Что изменилось в директории lib..... | 409 |
| 14.3. Умный компонент App | 411 |
| 14.3.1. Добавление транзакции..... | 413 |
| 14.3.2. Генерация нового блока | 417 |
| 14.3.3. Объяснение хуков useEffect() | 417 |
| 14.3.4. Мемоизация с помощью хука useCallback() | 419 |
| 14.4. Компонент представления TransactionForm | 423 |
| 14.5. Компонент представления PendingTransactionsPanel | 427 |
| 14.6. Компоненты представления BlocksPanel и BlockComponent | 429 |
| Итоги | 431 |

| | |
|--|-----|
| Глава 15. Разработка приложений Vue.js с помощью TypeScript | 433 |
| 15.1. Разработка простейшей веб-страницы с помощью Vue | 434 |
| 15.2. Генерация и запуск приложения с помощью Vue CLI | 438 |
| 15.3. Разработка одностраничных приложений с маршрутизацией | 446 |
| 15.3.1. Генерация нового приложения с Vue Router | 447 |
| 15.3.2. Отображение списка товаров в представлении Home | 451 |
| 15.3.3. Передача данных с помощью Vue Router | 458 |
| Итоги | 463 |
| Глава 16. Разработка блокчейн-клиента на Vue.js | 464 |
| 16.1. Запуск клиента и сервера обмена сообщениями | 465 |
| 16.2. Компонент App | 468 |
| 16.3. Компонент представления TransactionForm | 473 |
| 16.4. Компонент представления PendingTransactionsPanel | 478 |
| 16.5. Компоненты представления BlocksPanel и Block | 480 |
| Итоги | 484 |
| Эпилог | 485 |
| Приложение А. Современный JavaScript | 486 |
| А.1. Как запускать образцы кода | 486 |
| А.2. Ключевые слова let и const | 487 |
| А.2.1. Ключевое слово var и поднятие | 487 |
| А.2.2. let и const для работы в области блока | 489 |
| А.3. Шаблонные литералы | 490 |
| А.3.1. Размеченные шаблонные строки | 491 |
| А.4. Опциональные параметры и значения по умолчанию | 493 |
| А.5. Выражения стрелочных функций | 494 |
| А.6. Оператор остатка (rest) | 496 |
| А.7. Оператор распространения | 498 |
| А.8. Деструктуризация | 500 |
| А.8.1. Деструктуризация объектов | 500 |
| А.8.2. Деструктуризация массивов | 503 |

| | |
|---|-----|
| A.9. Классы и наследование | 504 |
| A.9.1. Конструкторы | 506 |
| A.9.2. Ключевое слово <code>super</code> и функция <code>super()</code> | 507 |
| A.9.3. Статические члены класса | 509 |
| A.10. Асинхронная обработка | 511 |
| A.10.1. Ад обратных вызовов | 511 |
| A.10.2. Промисы | 512 |
| A.10.3. Разрешение нескольких промисов одновременно | 515 |
| A.10.4. <code>async-await</code> | 516 |
| A.11. Модули | 518 |
| A.11.1. Импорты и экспорты | 521 |
| A.12. Транспиляторы | 523 |