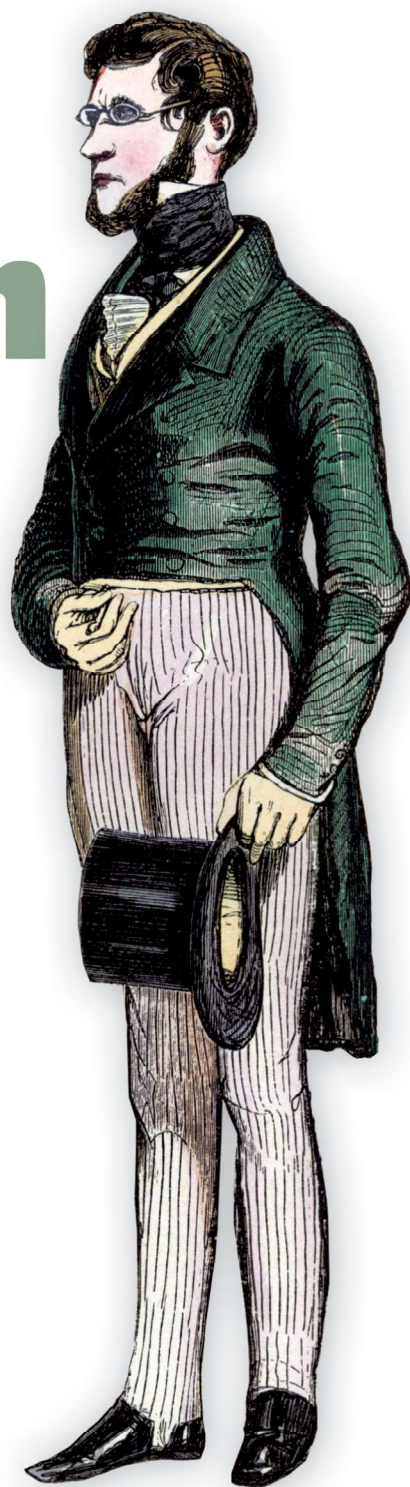


Angular и TypeScript

Сайтостроение
для профессионалов

Яков Файн
Антон Моисеев

 MANNING



Angular 2 Development with TypeScript

YAKOV FAIN
ANTON MOISEEV



MANNING
Shelter Island

Яков Файн, Антон Моисеев

Angular и TypeScript

Сайтостроение
для профессионалов



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

2018

Яков Файн, Антон Моисеев

Angular и TypeScript. Сайтостроение для профессионалов

Серия «Библиотека программиста»

Перевели с английского *Н. Вильчинский, Е. Зазноба*

Заведующая редакцией
Руководитель проекта
Ведущий редактор
Литературный редактор
Художественный редактор
Корректоры
Верстка

*Ю. Сергиенко
О. Сивченко
Н. Гринчик
Н. Хлебина
С. Заматевская
Е. Павлович, Е. Рафалюк-Бузовская
Г. Блинов*

ББК 32.988.02

УДК 004.738.5

Файн Я., Моисеев А.

Ф17 Angular и TypeScript. Сайтостроение для профессионалов. — СПб.: Питер, 2018. — 464 с.: ил. — (Серия «Библиотека программиста»).

ISBN 978-5-4461-0496-3

Если вы занимаетесь веб-разработкой — от веб-клиентов до полнофункциональных одностраничных приложений, — то фреймворк Angular будет для вас просто спасением. Этот ультрасовременный инструмент полностью интегрирован со статически типизированным языком TypeScript, который отлично вписывается в экосистему JavaScript.

Вы научитесь:

- Проектировать и строить модульные приложения.
- Правильно транспилировать TypeScript в JavaScript.
- Пользоваться новейшими инструментами JavaScript — в частности npm, Karma и Webpack.

Если вам знаком язык JavaScript — берите и читайте! Знаний TypeScript или AngularJS для изучения книги не требуется.

16+ (В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ISBN 978-1617293122 англ.
ISBN 978-5-4461-0496-3

© 2017 by Manning Publications Co. All rights reserved
© Перевод на русский язык ООО Издательство «Питер», 2018
© Издание на русском языке, оформление ООО Издательство «Питер», 2018
© Серия «Библиотека программиста», 2018

Права на издание получены по соглашению с Manning. 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.

Изготовлено в России. Изготовитель: ООО «Прогресс книга».

Место нахождения и фактический адрес: 191123, Россия, город Санкт-Петербург,
улица Радищева, дом 39, корпус Д, офис 415. Тел.: +78127037373.

Дата изготовления: 12.2017. Наименование: книжная продукция. Срок годности: не ограничен.

Налоговая льгота — общероссийский классификатор продукции ОК 034-2014, 58.11.12 —

Книги печатные профессиональные, технические и научные.

Подписано в печать 08.12.17. Формат 70×100/16. Бумага офсетная. Усл. п. л. 37,410. Тираж 1000. Заказ 0000.

Отпечатано в ОАО «Первая Образцовая типография». Филиал «Чеховский Печатный Двор».

142300, Московская область, г. Чехов, ул. Полиграфистов, 1.

Сайт: www.chpk.ru. E-mail: marketing@chpk.ru

Факс: 8(496) 726-54-10, телефон: (495) 988-63-87

Краткое содержание

Предисловие.....	13
Предисловие к русскому изданию.....	16
Благодарности.....	17
Об этой книге.....	18
Об авторах.....	21
Глава 1. Знакомство с Angular.....	22
Глава 2. Приступаем к работе с Angular.....	47
Глава 3. Навигация с помощью маршрутизатора Angular.....	91
Глава 4. Внедрение зависимостей.....	137
Глава 5. Привязки, наблюдаемые объекты и каналы.....	169
Глава 6. Реализация коммуникации между компонентами.....	202
Глава 7. Работа с формами.....	239
Глава 8. Взаимодействие с серверами с помощью HTTP и WebSockets.....	272
Глава 9. Модульное тестирование Angular-приложений.....	315
Глава 10. Упаковка и развертывание приложений с помощью Webpack.....	348
Приложение А. Общее представление о ECMAScript 6.....	388
Приложение Б. TypeScript в качестве языка для приложений Angular.....	429
Об обложке.....	464

Оглавление

Предисловие.....	13
Предисловие к русскому изданию.....	16
Благодарности.....	17
Об этой книге.....	18
Как читать книгу.....	18
Структура издания.....	18
Соглашения, принятые в этой книге, и материалы для скачивания.....	20
Об авторах.....	21
Глава 1. Знакомство с Angular.....	22
1.1. Примеры фреймворков и библиотек JavaScript.....	23
1.1.1. Полноценные фреймворки.....	23
1.1.2. Легковесные фреймворки.....	23
1.1.3. Библиотеки.....	24
1.1.4. Что такое Node.js.....	25
1.2. Общий обзор AngularJS.....	26
1.3. Общий обзор Angular.....	30
1.3.1. Упрощение кода.....	30
1.3.2. Улучшение производительности.....	36
1.4. Инструментарий Angular-разработчика.....	38
1.5. Как все делается в Angular.....	42
1.6. Знакомство с приложением-примером.....	43
1.7. Резюме.....	46
Глава 2. Приступаем к работе с Angular.....	47
2.1. Первое приложение для Angular.....	47
2.1.1. Hello World с использованием TypeScript.....	48
2.1.2. Hello World с помощью ES5.....	52
2.1.3. Hello World с помощью ES6.....	54
2.1.4. Запуск приложений.....	55
2.2. Элементы Angular-приложения.....	56
2.2.1. Модули.....	56
2.2.2. Компоненты.....	57
2.2.3. Директивы.....	59
2.2.4. Краткое введение в привязку данных.....	60

2.3. Универсальный загрузчик модулей SystemJS	61
2.3.1. Обзор загрузчиков модулей	61
2.3.2. Загрузчики модулей против тегов <script>	62
2.3.3. Приступаем к работе с SystemJS	63
2.4. Выбираем менеджер пакетов	68
2.4.1. Сравниваем npm и jspm	70
2.4.2. Создаем проект Angular с помощью npm	71
2.5. Практикум: приступаем к работе над онлайн-аукционом	77
2.5.1. Первичная настройка проекта	79
2.5.2. Разработка главной страницы	81
2.5.3. Запуск онлайн-аукциона	89
2.6. Резюме	90
Глава 3. Навигация с помощью маршрутизатора Angular	91
3.1. Основы маршрутизации	92
3.1.1. Стратегии расположения	93
3.1.2. Составные части механизма навигации на стороне клиента	95
3.1.3. Навигация по маршрутам с помощью метода navigate()	103
3.2. Передача данных маршрутам	105
3.2.1. Извлечение параметров из объекта ActivatedRoute	105
3.2.2. Передача статических данных маршруту	108
3.3. Маршруты-потомки	108
3.4. Граничные маршруты	115
3.5. Создание одностраничного приложения, с несколькими областями отображения	120
3.6. Разбиение приложения на модули	124
3.7. «Ленивая» загрузка модулей	127
3.8. Практикум: добавление навигации в онлайн-аукцион	129
3.8.1. Создание ProductDetailComponent	130
3.8.2. Создание HomeComponent и рефакторинг кода	131
3.8.3. Упрощаем компонент ApplicationComponent	132
3.8.4. Добавление RouterLink в компонент ProductItemComponent	133
3.8.5. Изменение корневого модуля с целью добавления маршрутизации	134
3.8.6. Запуск аукциона	135
3.9. Резюме	136
Глава 4. Внедрение зависимостей	137
4.1. Шаблоны «Внедрение зависимостей» и «Инверсия управления»	138
4.1.1. Шаблон «Внедрение зависимостей»	138
4.1.2. Шаблон «Инверсия управления»	139
4.1.3. Преимущества внедрения зависимости	139

4.2. Инъекторы и поставщики	142
4.2.1. Как объявлять поставщики	144
4.3. Пример приложения, задействующего Angular DI	145
4.3.1. Внедрение сервиса продукта	145
4.3.2. Внедрение Http-сервиса	148
4.4. Переключение внедряемых объектов — это просто	150
4.4.1. Объявление поставщиков с помощью useFactory и useValue	153
4.4.2. Использование OpaqueToken	156
4.5. Иерархия инъекторов	157
4.5.1. viewProviders	159
4.6. Практикум: использование DI для приложения онлайн-аукциона	160
4.6.1. Изменение кода для передачи идентификатора продукта в качестве параметра	163
4.6.2. Изменение компонента ProductDetailComponent	163
4.7. Резюме	167
Глава 5. Привязки, наблюдаемые объекты и каналы	169
5.1. Привязка данных	169
5.1.1. Привязки к событиям	171
5.1.2. Привязка к свойствам и атрибутам	171
5.1.3. Привязка в шаблонах	176
5.1.4. Двухсторонняя привязка данных	179
5.2. Реактивное программирование и наблюдаемые потоки	182
5.2.1. Что такое «наблюдаемые потоки» и «наблюдатели»	182
5.2.2. Наблюдаемые потоки событий	185
5.2.3. Отменяем наблюдаемые потоки	190
5.3. Каналы	194
5.3.1. Пользовательские каналы	195
5.4. Практикум: фильтрация продуктов онлайн-аукциона	197
5.5. Резюме	201
Глава 6. Реализация коммуникации между компонентами	202
6.1. Коммуникация между компонентами	202
6.1.1. Входные и выходные свойства	203
6.1.2. Шаблон «Посредник»	210
6.1.3. Изменяем шаблоны во время работы программы с помощью ngContent	215
6.2. Жизненный цикл компонента	219
6.2.1. Использование ngOnChanges	222
6.3. Краткий обзор определения изменений	226

6.4. Открываем доступ к API компонента-потомка	229
6.5. Практикум: добавление в онлайн-аукцион функциональности для оценивания товаров	231
6.6. Резюме	238
Глава 7. Работа с формами	239
7.1. Обзор форм HTML	240
7.1.1. Стандартная функциональность браузера	240
7.1.2. Forms API в Angular	242
7.2. Шаблон-ориентированные формы	243
7.2.1. Обзор директив	244
7.2.2. Доработка формы HTML	246
7.3. Реактивные формы	248
7.3.1. Модель формы	248
7.3.2. Директивы форм	249
7.3.3. Переработка формы-примера	254
7.3.4. Использование FormBuilder	256
7.4. Валидация данных формы	256
7.4.1. Валидация реактивных форм	257
7.4.2. Выполнение валидации для примера формы регистрации	265
7.5. Практикум: добавление валидации в форму поиска	267
7.5.1. Изменение корневого модуля для добавления поддержки Forms API	267
7.5.2. Добавление списка категорий в компонент SearchComponent	268
7.5.3. Создание модели формы	269
7.5.4. Переработка шаблона	269
7.5.5. Реализация метода onSearch()	271
7.5.6. Запуск онлайн-аукциона	271
7.6. Резюме	271
Глава 8. Взаимодействие с серверами с помощью HTTP и WebSockets	272
8.1. Краткий обзор API Http-объектов	273
8.2. Создание веб-сервера с применением Node и TypeScript	276
8.2.1. Создание простого веб-сервера	276
8.2.2. Обслуживание формата JSON	278
8.2.3. Живая перекомпиляция TypeScript и перезагрузка кода	280
8.2.4. Добавление RESTful API для обслуживания товаров	281
8.3. Совместное использование Angular и Node	283
8.3.1. Статические ресурсы, имеющиеся на сервере	283
8.3.2. Выполнение GET-запросов с помощью Http-объекта	286

8.3.3. Распаковка наблюдаемых объектов в шаблонах с помощью AsyncPipe.....	289
8.3.4. Внедрение HTTP в сервис	290
8.4. Обмен данными между клиентом и сервером через веб-сокеты	294
8.4.1. Выдача данных с Node-сервера.....	295
8.4.2. Превращение WebSocket в наблюдаемый объект	298
8.5. Практикум: реализация поиска товара и уведомлений о ценовых предложениях	304
8.5.1. Реализация поиска товара с использованием HTTP	306
8.5.2. Распространение по сети ценовых предложений аукциона с использованием веб-сокетов	310
8.6. Резюме	314
Глава 9. Модульное тестирование Angular-приложений	315
9.1. Знакомство с Jasmine	316
9.1.1. Что именно тестировать	319
9.1.2. Порядок установки Jasmine.....	319
9.2. Средства, предоставляемые библиотекой тестирования Angular	321
9.2.1. Тестирование сервисов.....	322
9.2.2. Тестирование навигации с помощью маршрутизатора	323
9.2.3. Тестирование компонентов.....	324
9.3. Пример: тестирование приложения для составления прогнозов погоды	325
9.3.1. Настройка SystemJS.....	327
9.3.2. Тестирование маршрутизатора приложения для составления прогнозов погоды.....	327
9.3.3. Тестирование сервиса погоды.....	330
9.3.4. Тестирование компонента Weather	333
9.4. Запуск тестов с помощью Karma	337
9.5. Практикум: модульное тестирование онлайн-аукциона.....	340
9.5.1. Тестирование ApplicationComponent.....	342
9.5.2. Тестирование ProductService	342
9.5.3. Тестирование StarsComponent	343
9.5.4. Запуск тестов.....	346
9.6. Резюме	347
Глава 10. Упаковка и развертывание приложений с помощью Webpack	348
10.1. Знакомство с Webpack.....	350
10.1.1. Hello World с Webpack	352
10.1.2. Как использовать загрузчики	356

10.1.3. Как использовать дополнительные модули	361
10.2. Создание базовой конфигурации Webpack для Angular.....	361
10.2.1. Команда npm run build	365
10.2.2. Команда npm start.....	366
10.3. Создание конфигураций для разработки и для коммерческого применения.....	367
10.3.1. Конфигурация для разработки	367
10.3.2. Конфигурация для коммерческого применения.....	368
10.3.3. Специальный файл определения типов.....	371
10.4. Что такое Angular CLI.....	374
10.4.1. Запуск нового проекта с Angular CLI.....	375
10.4.2. Команды CLI	375
10.5. Практикум: развертывание онлайн-аукциона с помощью Webpack.....	377
10.5.1. Запуск Node-сервера.....	378
10.5.2. Запуск клиента аукциона	379
10.5.3. Запуск тестов с помощью Karma.....	383
10.6. Резюме	386
Приложение А. Общее представление о ECMAScript 6.....	388
А.1. Порядок запуска примеров кода	389
А.2. Литералы шаблонов	390
А.2.1. Строки с переносами	390
А.2.2. Тегированные шаблонные строки.....	391
А.3. Необязательные параметры и значения по умолчанию.....	392
А.4. Область видимости переменных	393
А.4.1. Поднятие переменных	393
А.4.2. Блочная область видимости с использованием let и const.....	395
А.4.3. Блочная область видимости для функций	397
А.5. Стрелочные функции, this и that.....	397
А.5.1. Операторы rest и spread	400
А.5.2. Генераторы	403
А.5.3. Деструктурирование.....	405
А.6. Итерация с помощью forEach(), for-in и for-of.....	408
А.6.1. Использование метода forEach().....	408
А.6.2. Использование цикла for-in.....	409
А.6.3. Использование for-of	410
А.7. Классы и наследование.....	410
А.7.1. Конструкторы.....	412
А.7.2. Статические переменные	413

A.7.3. Геттеры, сеттеры и определение методов.....	414
A.7.4. Ключевое слово <code>super</code> и функция <code>super</code>	414
A.8. Асинхронная обработка с помощью промисов.....	416
A.8.1. Ад обратного вызова	417
A.8.2. Промисы ES6	418
A.8.3. Одновременное разрешение сразу нескольких промисов	421
A.9. Модули	422
A.9.1. Импорт и экспорт данных	423
A.9.2. Загрузка модулей в ES6	424
Приложение Б. TypeScript в качестве языка для приложений Angular	429
Б.1. Зачем создавать Angular-приложения на TypeScript	430
Б.2. Роль транспиляторов	431
Б.3. Начало работы с TypeScript.....	432
Б.3.1. Установка и применение компилятора TypeScript.....	433
Б.4. TypeScript как расширенная версия JavaScript	436
Б.5. Необязательные типы.....	436
Б.5.1. Функции	438
Б.5.2. Параметры по умолчанию.....	439
Б.5.3. Необязательные параметры.....	439
Б.5.4. Выражения стрелочных функций.....	440
Б.6. Классы.....	443
Б.6.1. Модификаторы доступа	445
Б.6.2. Методы	446
Б.6.3. Наследование.....	448
Б.7. Обобщения	450
Б.8. Интерфейсы	452
Б.8.1. Объявление пользовательских типов с помощью интерфейсов.....	453
Б.8.2. Использование ключевого слова <code>implements</code>	454
Б.8.3. Использование <code>callable</code> -интерфейсов	456
Б.9. Добавление метаданных класса с помощью аннотаций.....	459
Б.10. Файлы определения типов	460
Б.10.1. Установка файлов определения типов	460
Б.10.2. Управление стилем кода с помощью TSLINT	462
Б.11. Обзор процесса разработки TypeScript и Angular	463
Об обложке	464

Предисловие

Мы принялись подбирать хороший фреймворк для JavaScript примерно четыре года назад. Тогда мы работали над платформой для страховой компании и большая часть интерфейса системы была написана с помощью фреймворка Apache Flex (ранее известного как Adobe Flex). Это отличный инструмент для разработки веб-интерфейсов, но он требует наличия Flash Player, с некоторых пор впавшего в немилость.

После попыток создать несколько вводных проектов, написанных на JavaScript, мы обнаружили заметное снижение производительности наших разработчиков. Задача, выполнить которую с использованием Flex можно было за один день, требовала три дня работы в других фреймворках JavaScript, включая AngularJS. Основная причина спада — нехватка типов в JavaScript, недостаточная поддержка IDE и отсутствие поддержки компилятора.

Когда мы узнали, что компания Google начала разработку фреймворка Angular с TypeScript в качестве рекомендуемого языка, мы взяли данный фреймворк на вооружение летом 2015 г. В то время о нем было известно немного. В нашем распоряжении была лишь информация из некоторых блогов и видеороликов, записанных на конференции, где команда разработчиков Angular представляла новый фреймворк. Нашим основным источником информации был его исходный код. Но вскоре мы обнаружили у Angular отличный потенциал, поэтому решили начать создавать с его помощью обучающее ПО для отдела разработки нашей компании. В то же время Майк Стивенс (Mike Stephens), сотрудник издательства Manning, искал авторов, заинтересованных в написании книги об Angular. Именно так и появилось это издание.

После года работы с Angular и TypeScript мы можем подтвердить: этот дуэт предоставляет наиболее продуктивный способ создания средних и крупных веб-приложений, которые могут работать в любом современном браузере, а также на мобильных платформах. Перечислим основные причины, позволившие нам прийти к такому выводу.

- ❑ *Четкое разделение интерфейса и логики.* Код, который отвечает за пользовательский интерфейс, и код, реализующий логику приложения, четко разделены. Интерфейс не должен создаваться в HTML, существуют другие продукты с поддержкой его нативной отрисовки для iOS и Android.
- ❑ *Модульность.* Существует простой механизм разбиения приложения на модули с возможностью их ленивой загрузки.
- ❑ *Поддержка навигации.* Маршрутизатор поддерживает сложные сценарии навигации в одностраничных приложениях.
- ❑ *Слабое связывание.* Внедрение зависимостей (Dependency Injection, DI) позволяет установить связь между компонентами и сервисами. С помощью связывания

и событий есть возможность создавать слабо связанные компоненты, которые можно использовать повторно.

- ❑ *Жизненный цикл компонентов.* Каждый компонент проходит через тщательно определенный жизненный цикл, разработчикам доступны процедуры, позволяющие перехватывать важные события компонентов.
- ❑ *Определение изменений.* Автоматический (и быстрый) механизм определения изменений дает возможность отказаться от обновления интерфейса вручную, также предоставляя способ выполнить тонкую настройку этого процесса.
- ❑ *Отсутствие ада обратных вызовов.* Angular поставляется вместе с библиотекой RxJS, которая позволяет построить процесс обработки асинхронных данных, основанный на подписке, что помогает избежать появления ада обратных вызовов.
- ❑ *Формы и валидация.* Поддержка форм и пользовательская валидация хорошо спроектированы. Вы можете создавать формы, добавляя директивы к элементам формы как в шаблонах, так и программно.
- ❑ *Тестирование.* Фреймворк поддерживает модульное (блочное) и сквозное тестирование, вы можете интегрировать тесты в свой автоматизированный процесс сборки.
- ❑ *Упаковка и оптимизация с помощью Webpack.* Упаковка и оптимизация кода с помощью Webpack (и различных плагинов для него) позволяет поддерживать небольшой размер разворачиваемых приложений.
- ❑ *Инструментальные средства.* Инструментальные средства поддерживаются так же хорошо, как и для платформ Java и .NET. Анализатор кода TypeScript указывает на ошибки по мере набора текста, а инструмент для создания временных платформ и развертывания (Angular CLI) позволяет избегать написания стереотипного кода и сценариев конфигурирования.
- ❑ *Лаконичный код.* Использование классов и интерфейсов TypeScript делает код более лаконичным, а также упрощает его чтение и написание.
- ❑ *Компиляторы.* Компилятор TypeScript генерирует код JavaScript, который человек может прочитать. Код TypeScript может быть скомпилирован для версий JavaScript ES3, ES5 или ES6. Компиляция перед выполнением (Ahead-of-time, AoT) кода Angular (не путать с компилированием TypeScript) помогает избавиться от необходимости поставлять с вашим приложением компилятор для Angular, что еще больше снижает накладные расходы, требуемые для фреймворка.
- ❑ *Отрисовка на стороне сервера.* Angular Universal превращает ваше приложение в HTML на этапе сборки кода офлайн. Данный код может быть использован для отрисовки на стороне сервера, что, в свою очередь, значительно улучшает индексирование поисковыми системами и SEO.

Таким образом, Angular 2 готов к работе сразу после установки.

С точки зрения управления этот фреймворк может быть привлекательным, поскольку в мире более миллиона разработчиков предпочитают AngularJS и большинство из них переключатся на Angular. Эта версия была выпущена в сентябре

2016 г., и каждые полгода будут выпускаться новые крупные релизы. Команда разработчиков Angular потратила два года на разработку Angular 2.0, и к моменту выхода фреймворка более полумиллиона разработчиков уже использовали его. Наличие большого количества работников с определенными навыками — важный фактор, который следует учитывать при выборе технологий для новых проектов. Кроме того, платформами Java или .NET пользуются более 15 миллионов разработчиков, и многие из них сочтут синтаксис TypeScript гораздо более привлекательным, чем синтаксис JavaScript, благодаря поддержке классов, интерфейсов, обобщенных типов, аннотаций, переменных членов класса, а также закрытых и открытых переменных, не говоря о полезном компиляторе и поддержке известных IDE.

Писать книгу об Angular было трудно, поскольку мы начали работу с ранних альфа-версий фреймворка, которые изменились по мере перехода к бета- и релиз-версиям. Но нам нравится результат, и мы уже начали разрабатывать с помощью Angular проекты для решения задач реальной сложности.

Предисловие к русскому изданию

В эпоху бурного развития Интернета требования к сложности и качеству сайтов постоянно растут. Пользователям и организациям уже недостаточно простых статических страниц. В связи с этим давно наметился спрос на полноценные веб-приложения, однако до недавнего времени он удовлетворялся в основном за счет таких закрытых технологий, как Adobe Flex и JavaFX.

Около десяти лет назад ситуация начала меняться. Последовательная стандартизация языка JavaScript привела к появлению множества инструментов для веб-разработки, которые не требуют дополнений и поддерживаются во всех современных браузерах. Пожалуй, самым известным из них стал фреймворк Angular.

Angular разрабатывается компанией Google и при этом является полностью бесплатным и открытым. Это устоявшийся программный продукт, который пользуется большой популярностью как на корпоративном рынке, так и в небольших студиях. Количество программистов, работающих с ним, давно перевалило за миллион.

Эта книга посвящена Angular 2 — современной версии данного фреймворка. Он вобрал в себя все последние веяния в мире веб-разработки, сохранив традиционные черты, характерные для первой версии. Главной особенностью Angular 2 является его изначальная ориентированность на TypeScript. И хотя поддержка JavaScript (ES5 и ES6) по-прежнему присутствует, TypeScript остается основным языком разработки. Это, в сущности, надстройка над ECMAScript 6 со строгой типизацией. Развитая система типов и программных интерфейсов позволяет писать более безопасный и выразительный код, который легче сопровождать. Кроме того, TypeScript имеет отличную поддержку в современных средах разработки.

Еще одной ключевой особенностью Angular 2 является четкое разделение интерфейса и бизнес-логики. Это позволяет создавать компоненты, которые могут работать в том числе на таких мобильных операционных системах, как iOS и Android. Angular 2 позволяет разбивать код на модули, которые загружаются по мере необходимости и содержат описание своих зависимостей. Это значительно упрощает процедуру тестирования (как модульного, так и сквозного) и облегчает интеграцию тестов в процесс сборки.

Разработчикам доступен богатый инструментарий для написания и развертывания приложений. Автодополнение кода, упаковка и оптимизация, отрисовка на стороне сервера, компиляция (статическая и на лету) — все это становится возможным благодаря современным технологиям, на которых основаны Angular 2 и TypeScript. В частности, для управления проектами предусмотрена утилита командной строки Angular CLI, а для создания гибких пользовательских интерфейсов компания Google предоставляет библиотеку компонентов Angular Material 2.

На протяжении двух лет разработкой Angular 2 занимались сотни программистов. Все старания были направлены на то, чтобы вы получили современный инструмент, полностью готовый к работе сразу после установки. Этот фреймворк покрывает собой все аспекты создания сложных веб-приложений, начиная с обработки событий и навигации и заканчивая развертыванием в промышленных масштабах. Все это и многое другое подробно рассматривается на страницах книги.

Благодарности

Оба автора хотели бы поблагодарить издательство Manning Publications и Алана М. Кауниота (Alain M. Couniot) за его предложения по улучшению технической стороны книги и Коди Сэнда (Cody Sand) — за техническую редактуру. Благодарим также следующих рецензентов, предоставивших ценную обратную связь: Криса Коппенбаргера (Chris Coppenbarger), Дэвида Баркола (David Barkol), Дэвида Ди Мария (David DiMaria), Фредрика Энгберга (Fredrik Engberg), Ирака Илиша Рамоса Эрнандеса (Irach Ilish Ramos Hernandez), Джереми Брайана (Jeremy Bryan), Камала Раджа (Kamal Raj), Лори Уилкинс (Lori Wilkins), Мауро Керциоли (Mauro Quercioli), Себастьяна Нишеля (Sébastien Nichèle), Полину Кесельман (Polina Keselman), Субира Растоги (Subir Rastogi), Свена Лесекана (Sven Lösekann) и Висама Цагала (Wisam Zaghal).

Яков хотел бы поблагодарить своего лучшего друга Сэмми (Sammy) за создание теплой и комфортной атмосферы во время работы над этой книгой. К сожалению, Сэмми не умеет разговаривать, но, безусловно, любит Якова. Порода Сэмми — золотистый ретривер.

Антон хотел бы поблагодарить Якова Файна (Yakov Fain) и издательство Manning Publications за полученную возможность стать соавтором книги и приобрести бесценный писательский опыт. Он также благодарен своей семье за терпение, которое она проявляли, пока он работал над книгой.

Об этой книге

Программы на Angular могут быть написаны с помощью двух версий JavaScript (ES5 и ES6), а также на языках Dart или TypeScript. Сам фреймворк разработан с использованием TypeScript, и именно его мы будем задействовать во всех примерах нашей книги. В приложении Б вы найдете раздел «Зачем создавать Angular-приложения»¹ на TypeScript», где мы объясняем причины, по которым выбрали этот язык.

Поскольку мы по своей природе практики, то и книгу писали для практиков. Мы не только объясним особенности фреймворка, приводя простые примеры кода, но и на протяжении нескольких глав покажем, как создать одностраничное приложение для онлайн-аукционов.

Во время написания и редактирования данной книги мы провели несколько семинаров с использованием примеров кода. Это позволило получить раннюю (и крайне положительную) обратную связь об изложенном материале. Мы очень надеемся на то, что вам понравится изучать Angular.

Как читать книгу

Ранние версии книги начинались с глав, посвященных ECMAScript 6 и TypeScript. Некоторые редакторы предложили переместить этот материал в приложения, чтобы читатели смогли быстрее приступить к изучению Angular. Мы согласились с таким изменением. Но если вы еще не знакомы с синтаксисом ECMAScript 6 и TypeScript, то информация, представленная в приложениях, поможет разобраться в примерах кода каждой главы.

Структура издания

Книга состоит из десяти глав и двух приложений.

В главе 1 приводится обзор архитектуры Angular, кратко рассматриваются популярные фреймворки и библиотеки JavaScript. Вы также познакомитесь с приложением для онлайн-аукционов, которое начнете разрабатывать в главе 2.

Вы будете разрабатывать эту программу с помощью TypeScript. Информация, изложенная в приложении Б, позволит приступить к работе с этим замечательным языком, представляющим собой расширенный набор функций языка JavaScript. Вы узнаете не только о том, как писать классы, интерфейсы и обобщенные классы, но и как скомпилировать код TypeScript для современных версий JavaScript, которые могут быть развернуты во всех браузерах. В TypeScript

¹ Здесь и далее: имеется в виду приложение, созданное с помощью Angular. — *Примеч. ред.*

реализована большая часть синтаксиса спецификации ECMAScript 6 (она рассматривается в приложении А), а также синтаксис, который будет включен в будущие релизы ECMAScript.

В главе 2 вы начнете разрабатывать простые приложения с помощью Angular и создадите свои первые Angular-компоненты. Вы узнаете, как работать с загрузчиком модулей SystemJS, и мы предложим вам свою версию стартового проекта Angular, который может быть использован в качестве отправной точки для всех примеров приложений этой книги. В конце главы вы создадите первую версию главной страницы для онлайн-аукциона.

Глава 3 посвящена маршрутизатору Angular, предлагающему гибкий способ настройки навигации в одностраничных приложениях. Вы узнаете, как сконфигурировать маршруты для компонентов-предков и компонентов-потомков, передавать данные из одного маршрута в другой и выполнять «ленивую» загрузку модулей. В конце главы вы проведете рефакторинг для онлайн-аукциона, разбив его на несколько элементов и добавив для них маршрутизацию.

В главе 4 представлена информация о шаблоне (паттерне) «Внедрение зависимостей» и его реализации в Angular. Вы познакомитесь с концепцией поставщиков, которая позволит указать, как создавать объекты внедряемых зависимостей. В новой версии онлайн-аукциона вы примените внедрение зависимостей для наполнения данными представления **Product Details** (Информация о продукте).

В главе 5 мы рассмотрим разные виды привязки данных, познакомим вас с реактивным программированием с помощью наблюдаемых потоков данных и покажем, как работать с каналами. Глава заканчивается новой версией онлайн-аукциона: в нее добавлен наблюдаемый поток событий, используемый для фильтрации популярных продуктов на главной странице.

Глава 6 посвящена разработке компонентов, которые могут общаться друг с другом в слабо связанной манере. Мы рассмотрим их входные и выходные параметры, шаблон проектирования «Посредник», а также жизненный цикл компонента. Кроме того, в этой главе приводится обзор механизма обнаружения изменений, представленного в Angular. Онлайн-аукцион приобретет функциональность оценки продуктов.

Глава 7 посвящена обработке форм в Angular. После рассмотрения основ Forms API мы обсудим валидацию форм и реализуем ее в поисковом компоненте для еще одной версии онлайн-аукциона.

В главе 8 мы объясним, как клиентское Angular-приложение может взаимодействовать с серверами, используя протоколы HTTP и WebSocket, и рассмотрим примеры. Вы создадите серверное приложение, применяя фреймворки Node.js и Express. Далее вы развернете фрагмент онлайн-аукциона, написанный на Angular, на сервере Node. С помощью клиентской части (front end) аукциона можно будет начать общаться с сервером Node.js с применением протоколов HTTP и WebSocket.

Глава 9 посвящена модульному тестированию. Мы рассмотрим основные принципы работы с Jasmine и библиотекой для проверки Angular. Вы узнаете, как тестировать сервисы, компоненты и маршрутизатор, сконфигурировать и использовать

Карта для запуска тестов, а также реализуете несколько модульных тестов для онлайн-аукциона.

Глава 10 посвящена автоматизации процессов сборки и развертывания. Вы увидите, как можно применять упаковщик Webpack для минимизации и упаковки вашего кода для развертывания.

Кроме того, вы узнаете, как использовать Angular CLI для генерации и развертывания проектов. Размер развернутой версии онлайн-аукциона снизится с 5,5 Мбайт (в разработке) до 350 Кбайт (на производстве).

В приложении А вы познакомитесь с новым синтаксисом, добавленным в ECMAScript 2015 (также известным как ES6). Приложение Б содержит вводную информацию о языке TypeScript.

Соглашения, принятые в этой книге, и материалы для скачивания

В данной книге содержится множество примеров исходного кода, который отформатирован с использованием моношириного шрифта, чтобы выделить его на фоне обычного текста. Нередко оригинальный исходный код переформатирован; добавлены разрывы строк, а также переработаны отступы с учетом свободного места на страницах. Иногда, когда этого оказалось недостаточно, листинги содержат метки продолжения строки (➡). Кроме того, комментарии к исходному коду были удалены там, где код описан в тексте. Во многих листингах приведены комментарии, указывающие на важные концепции.

Исходный код примеров книги доступен для загрузки с сайта издателя <https://www.manning.com/books/angular-2-development-with-typescript>.

Авторы также создали репозиторий на GitHub, в котором содержатся примеры исходного кода, на <https://github.com/Farata/angular2typescript>. Если в результате выхода новых релизов Angular какие-то примеры перестанут работать, то на указанной странице вы можете рассказать об этом и авторы постараются разобраться с проблемой.

Об авторах



Яков Файн (Yakov Fain) является сооснователем двух компаний: Farata Systems и SuranceBay. Farata Systems — фирма, занимающаяся IT-консалтингом. Яков руководит отделом обучения и проводит семинары по Angular и Java по всему миру. SuranceBay — производственная компания, которая автоматизирует различные рабочие потоки в отрасли страхования в Соединенных Штатах, предлагая приложения, использующие ПО как услугу (Software as a Service, SaaS). В этой компании Яков руководит различными проектами.

Яков — большой энтузиаст Java. Он написал множество книг, посвященных разработке ПО, а также более тысячи статей в блоге по адресу yakovfain.com. Кроме того, его книга *Java Programming for Kids, Parents and Grandparents* («Программирование на Java для детей, родителей, бабушек и дедушек») доступна для бесплатного скачивания на нескольких языках по адресу <http://myflex.org/books/java4kids/java4kids.htm>. Его никнейм в Twitter — @yfain.



Антон Моисеев (Anton Moiseev) — ведущий разработчик ПО в компании SuranceBay. Он более десяти лет занимается разработкой промышленных приложений с помощью технологий Java и .NET и имеет богатый опыт создания многофункциональных интернет-приложений для разных платформ. Сфера деятельности Антона — веб-технологии, причем он специализируется на приемах, позволяющих фронтенду и бэкенду без проблем взаимодействовать друг с другом. Он провел множество занятий, посвященных фреймворкам AngularJS и Angular 2.

Время от времени Антон пишет статьи в блоге по адресу antonmoiseev.com. Его никнейм в Twitter — @anton-moiseev.