

*Посвящаю эту книгу моим родителям Паулу и Пепе
и памяти моего любимого брата Хосе Рауля.*

Ты будешь вечно жить в наших сердцах.

Содержание

Об авторе	13
Благодарности	14
О технических рецензентах	15
Предисловие	16
Глава 1. Создание самого первого компонента в Angular 2	22
Обновленный подход.....	23
Веб-компоненты.....	24
Почему именно TypeScript?	25
Настройка рабочего окружения.....	26
Установка зависимостей.....	26
Установка TypeScript	30
Установка утилиты typings.....	31
Hello, Angular 2!	33
Классы TypeScript.....	34
Введение в декораторы	34
Компиляция TypeScript в понятный браузерам код на JavaScript.....	35
HTML-контейнер	36
Сопроводительные примеры книги	39
Соединяем все вместе.....	40
Усовершенствование интегрированной среды разработки	43
Sublime Text 3.....	43
Atom.....	44
Visual Studio Code	45
WebStorm	46
Использование Gulp с другими интегрированными средами разработки.....	46
Погружение в компоненты Angular 2	48
Улучшение производительности.....	48
Методы компонента и обновление данных	48
Добавление интерактивности в компонент.....	51

Улучшения для вывода данных в представлении и доработка пользовательского интерфейса.....	53
Итоги.....	57

Глава 2. Введение в TypeScript 58

Причины создания TypeScript.....	58
Преимущества TypeScript.....	59
Знакомство с ресурсами, посвященными TypeScript.....	61
Типы в TypeScript.....	62
Тип string	63
Тип number	64
Динамическая типизация с помощью типа any	64
Тип void	66
Автоматическое определение типов	66
Функции, лямбда-функции и поток выполнения.....	67
Аннотации типов в функциях	67
Параметры функций в TypeScript.....	69
Улучшенный синтаксис функций и область обработки с лямбда-функциями	71
Классы, интерфейсы и наследование классов	73
Анатомия классов: конструкторы, свойства и методы	73
Интерфейсы в TypeScript.....	76
Расширение классов с помощью наследования	78
Декораторы TypeScript.....	79
Декораторы классов.....	80
Декораторы свойств.....	81
Декораторы методов	83
Декораторы параметров	86
Работа с модулями в приложении	87
Внутренние модули.....	87
Внешние модули	89
Предстоящий путь.....	90
Итоги.....	90

Глава 3. Реализация свойств и событий

компонентов	92
Улучшенный синтаксис шаблонов.....	92
Привязка данных с помощью входных свойств.....	93
Дополнительные синтаксические конструкции для привязки выражений.....	94

Привязка событий к выходным свойствам.....	94
Входные и выходные свойства в действии	95
Передача данных в пользовательских событиях	100
Локальные ссылки в шаблонах.....	102
Альтернативный синтаксис для входных и выходных свойств.....	103
Настройка шаблона класса компонента.....	104
Внутренние и внешние шаблоны.....	104
Инкапсуляции CSS-стилей.....	105
Управление инкапсуляцией представлений	107
Итоги.....	108

Глава 4. Расширение компонентов с помощью директив и фильтров	110
Директивы в Angular 2.....	111
Встроенные директивы.....	111
Управление привязками шаблонов с помощью фильтров.....	114
Фильтры uppercase/lowercase.....	115
Фильтры number, percent и currency.....	115
Фильтр slice.....	117
Фильтр date.....	117
Фильтр JSON.....	118
Фильтр replace	118
Фильтры интернационализации	119
Фильтр async	120
Реализация списка заданий в проекте Pomodoro.....	120
Создание основного HTML-контейнера.....	121
Построение таблицы со списком заданий с помощью директив Angular.....	122
Переключение заданий в списке	128
Отображение изменений состояния в шаблонах.....	130
Встраивание дочерних компонентов	132
Создание собственных фильтров	136
Анатомия пользовательских фильтров.....	136
Пользовательские фильтры для форматирования времени.....	137
Фильтрация данных с помощью пользовательских фильтров	139
Создание собственных директив.....	140
Анатомия пользовательских директив	141
Создание подсказки для заданий с помощью пользовательской директивы.....	143

Несколько слов об именах пользовательских директив и фильтров...	145
Итоги	146

Глава 5. Создание приложений с помощью компонентов Angular 2

Введение в деревья компонентов	148
Общие соглашения для масштабируемых приложений	149
Соглашение об именовании файлов и модулей	151
Обеспечение непрерывной масштабируемости с помощью фасадов или баррелей.....	152
Внедрение зависимостей в Angular 2	153
Внедрение зависимостей в дереве компонентов.....	157
Переопределение поставщиков в иерархии инструментов внедрения зависимостей	161
Расширенная поддержка инструментов внедрения зависимостей для пользовательских сущностей	163
Инициализация приложения с помощью функции bootstrap()	165
Подготовка структуры каталогов приложения Pomodoro	167
Приведение приложения в соответствие с рекомендациями Angular 2.....	170
Общий контекст	170
Создание фасадного модуля, включающего пользовательскую баррель поставщиков.....	175
Создание компонентов	177
Начальная загрузка приложения	186
Итоги	187

Глава 6. Асинхронная обработка данных в Angular 2

Способы асинхронной обработки информации.....	189
Наблюдаемые объекты.....	191
Реактивное функциональное программирование в Angular 2	193
Библиотека RxJS.....	196
Введение в HTTP API	199
В каких случаях используются классы Request и RequestOptionsArgs	200
Объект Response.....	201
Обработка ошибок при выполнении Http-запросов.....	202
Внедрение класса Http и псевдоним HTTP_PROVIDERS.....	202

Пример: отслеживание данных по HTTP.....	204
Итоги.....	209
Глава 7. Маршрутизация в Angular 2.....	210
Добавление поддержки маршрутизатора Angular 2	211
Настройка службы маршрутизации.....	212
Создание нового компонента для демонстрационных целей.....	214
Настройка декоратора RouteConfig с помощью экземпляров RouteDefinition	216
Директивы маршрутизатора RouterOutlet и RouterLink	218
Принудительный переход по маршрутам	220
CSS-ловушки для активных маршрутов	222
Параметры обработки маршрута.....	222
Передача динамических параметров в маршрутах	223
Разбор параметров маршрута с помощью службы RouteParams	225
Определение дочерних маршрутизаторов	227
События жизненного цикла маршрутизатора	230
Событие CanActivate.....	231
Событие OnActivate.....	233
События CanDeactivate и OnDeactivate.....	234
События CanReuse и OnReuse	235
Переадресация на другие маршруты	237
Точная настройка базового пути	238
Настройка генерируемых URL-адресов с учетом стратегии размещения	239
Асинхронная загрузка компонентов с помощью AsyncRoutes	240
Итоги.....	241
Глава 8. Формы и аутентификация в Angular 2.....	243
Двухсторонняя привязка данных в Angular 2.....	244
Директива NgModel.....	245
Типы привязки к форме с помощью директивы NgModel	247
Отслеживание изменений и проверка допустимости данных	251
Элементы управления, группы элементов управления и класс FormBuilder	255
Объекты Control и Validator	256
Элементы управления в модели DOM и директива ngControl.....	257
Группировка элементов в модели DOM с помощью директивы NgControlGroup.....	258

Императивное определение групп элементов управления с помощью ControlGroup	260
Соединение модели DOM и контроллера с помощью ngFormModel	262
Пример компонента входа пользователя.....	263
Контекст функции входа.....	263
Шаблон формы входа.....	266
Компонент входа.....	266
Применение пользовательских проверок допустимости к элементам управления.....	268
Отображение изменений состояния в элементах управления	270
Фиктивная служба проверки подлинности клиента.....	271
Блокирование несанкционированного доступа.....	277
Реакция пользовательского интерфейса на состояние аутентификации пользователя	279
Улучшение управления доступом	281
Разработка собственной директивы безопасности RouterOutlet	282
Итоги	288

Глава 9. Анимация компонентов в Angular 2

Анимация с помощью стандартной CSS-разметки.....	290
Анимация с помощью классов-событий CSS	293
Анимация компонентов с помощью класса AnimationBuilder.....	295
Отслеживание состояния анимации с помощью класса Animation	300
Разработка пользовательских директив анимации	301
Взгляд в будущее: ngAnimate 2.0	307
Итоги.....	308

Глава 10. Модульное тестирование в Angular 2

Почему тестирование необходимо?.....	310
Средства модульного тестирования в Angular 2.....	311
Внедрение зависимостей в модульных тестах.....	312
Настройка среды тестирования	315
Реализация средства выполнения тестов.....	315
Настройка команд NPM.....	318
Пользовательские функции сопоставления Angular 2	319
Тестирование фильтров	320
Тестирование компонентов	322

Переопределение зависимостей компонентов для улучшения тестирования	330
Тестирование маршрутов	332
Тестирование переадресации.....	334
Тестирование служб.....	334
Реализация фиктивных Http-ответов с помощью MockBackend.....	338
Тестирование директив.....	342
Предстоящий путь.....	343
Использование Jasmine в сочетании с Karma	344
Получение отчетов об охвате кода тестами.....	344
Реализация E2E-тестов	345
Итоги.....	345
Предметный указатель.....	347

Об авторе

Пабло Дилеман (Pablo Deeleman) в 90-х годах работал дизайнером пользовательских интерфейсов, когда открыл для себя Интернет. В то время модем со скоростью 14 400 бит/с был ключом в волшебный мир, а самой популярной была игра с названием «Создай свой сайт».

После получения степени бакалавра (с отличием) в области маркетинга и поработав в рекламной индустрии, он остался верен себе и погрузился в самостоятельное изучение, превратившись в увлеченного дизайнера пользовательских интерфейсов и разработчика веб-приложений с блестяще подготовленными CSS-макетами и толстыми клиентами на JavaScript. Пабло создал бесчисленное множество интерактивных проектов и веб-приложений для настольных и мобильных устройств.

В течение нескольких лет он сделал карьеру дизайнера и разработчика, успешно руководившего интернет-проектами для широкого круга клиентов. Работал с европейскими туроператорами, начинающими компаниями из Кремниевой долины, международными веб-сайтами с тяжелым интернет-трафиком, порталами глобальных банковских систем, компаниями, занимающимися созданием игровых программ для мобильных устройств, и это далеко не полный список. Появление Node.js и фреймворков одностраничных приложений стало поворотным пунктом в его карьере, заставив сосредоточиться на создании взаимодействий с пользователем, управляемых с помощью JavaScript.

Пожив и поработав в нескольких странах, Пабло Дилеман обосновался в Барселоне (Испания) и теперь возглавляет разработку интерфейсов в Барселонской студии компании Gameloft, мирового лидера в области разработки игровых приложений для мобильных устройств, известного такими играми, как «Despicable Me: Minions Rush» и «Asphalt 8».

Большую часть свободного времени, когда он не пишет книги и не принимает участия в корпоративных мероприятиях и переговорах, Пабло посвящает своему увлечению – игре на фортепиано и гитаре.

Благодарности

Книга, которую вы сейчас держите в руках, является результатом больших усилий и массы затраченного на нее времени. Однажды кто-то мудро заметил, что писать книгу о фреймворке, находящемся в стадии альфа-версии, – это то же самое, что стрелять навскидку по движущейся мишени, и это действительно так. В процессе написания книги автор и команда проекта сбились со счета, сколько раз пришлось ее переписывать, чтобы привести в соответствие с самой последней реализацией фреймворка. Немногие способны выдержать накал такого сражения, не разочароваться и не согласиться с мыслью о прекращении проекта. Поэтому я хочу выразить благодарность команде издательства Packt и прежде всего Саманте Гонсалвес (Samantha Gonsalves). Теплые слова их поддержки подпитывали меня энергией, так необходимой для продвижения проекта.

Я также хотел бы особо поблагодарить своего друга Хорхе Феррандо (Jorge Ferrando) за помощь и подсказки в процессе работы над книгой. Его опыт работы с Angular 2 был бесценен при выборе между разными видами подачи обучающего материала. Кроме того, следует упомянуть моих коллег-разработчиков: Хавьера Гомеса (Javier Gómez), Альфонсо Фернандеса (Alfonso Fernández), Франа Ириэла (Fran Iruela) и Педро Нарцисо (Pedro Narciso).

Еще я хотел бы поблагодарить всех моих наставников, направивших мою профессиональную деятельность на протяжении нескольких лет, и особенно сотрудников компаний Casumo и Gameloft: Расмуса Свеннингсона (Rasmus Svenningson), Кима Ларсена (Kim Larsen), Джозефа Галеа (Josef Galea), Стива Аттарда (Steve Attard), Идена Аззопарди (Iden Azzopardi), Ренальда Далли (Renald Dalli), Мэтью Борга (Matthew Borg), Марка Бусуттила (Mark Busuttil), Жерара Гинэ (Gerard Giné), Антонио Гонсалеса (Antonio González), Альберта Пуэртоласа (Albert Puértolas), Рафаэля Марфил (Rafael Marfil) и всегда вдохновляющего меня Стюарда Лангриджа (Stuart Langridge).

О технических рецензентах

Йоханнес Вебер (Johannes Weber) – увлеченный разработчик и консультант в области применения веб-технологий в промышленных JS-приложениях. Работает в компании Mayflower GmbH (Мюнхен, Германия), где специализируется на миграциях SPA и MPA. В свободное время (co)организует Мюнхенские конференции AngularCamp и JS-Kongress.de по AngularJS. Йоханнес сотрудничает с сайтом ESnextNews.com, который готов еженедельно присылать вам по электронной почте пять отличных ссылок по теме ECMAScript.next.

Предисловие

За последние годы Angular 1.x стал одним из наиболее популярных фреймворков на JavaScript, предназначенных для создания современных веб-приложений, больших и малых. Но по мере увеличения размеров и сложности приложений все явственнее проявлялись его недостатки в отношении производительности и масштабируемости. Angular 2 был полностью переписан для удовлетворения требований разработчиков современных веб-приложений, которым необходимы высокая производительность и гибкость.

Фреймворк Angular 2 разработан в соответствии с современными веб-стандартами и предоставляет определенную свободу в выборе языка, обеспечивая полную поддержку не только ES6 и TypeScript, но и широко распространенных сегодня ES5, Dart и CoffeeScript. Его встроенный механизм внедрения зависимостей позволяет создавать масштабируемые модульные приложения с ясным и выразительным кодом, значительно упрощая сопровождение и разработку, основанную на тестировании. Везде, где бы не применялся Angular 2, он демонстрирует беспрецедентный уровень быстродействия и производительности благодаря новой системе обнаружения изменений, которая работает в пять раз быстрее, чем ее предыдущая реализация. Более прозрачные представления и непревзойденный синтаксис стандартных шаблонов предоставляют бесконечный список мощных функций для создания мобильных и настольных веб-приложений нового поколения.

Фреймворк Angular 2 сохранит и упрочит свою роль в разработке современных веб-приложений на годы вперед. Но именно из-за его архитектуры, разрушающей стереотипы, освоение Angular 2 может стать непосильной задачей для новичков.

И здесь на помощь придет эта книга, цель которой – помочь читателю избежать желудочных коликов при изучении документации с описанием фреймворка и его программного интерфейса, предложив практически подход к освоению фреймворка, позволяющий сразу перейти к его использованию. И это будет сделано с самого ее начала.

Книга описывает процесс построения веб-проекта на новой платформе, основанной на синтаксисе TypeScript, начиная с основных понятий, примеров компонентов и их использования для создания

функционала, усложняющегося с каждой главой, пока в конце книги не будет получено завершенное, протестированное и готовое к эксплуатации веб-приложение.

Какие темы охватывает книга

Глава 1 «Создание самого первого компонента в Angular 2» знакомит с веб-компонентами – строительными блоками всех приложений, реализованных с помощью Angular 2.

Глава 2 «Введение в TypeScript» знакомит с синтаксисом и особенностями типизированного надмножества ECMAScript 6, выбранного командой Angular для создания Angular 2.

Глава 3 «Реализация свойств и событий компонентов» описывает поведение компонентов, которые изменяют свое состояние в соответствии с данными, полученными через входные свойства, и передают данные в виде событий через выходные свойства.

Глава 4 «Расширение компонентов с помощью директив и фильтров» содержит полное описание фильтров, используемых для обработки данных в шаблонах, а также встроенных директив, расширяющих возможности компонентов. В этой главе читатель также узнает, как создавать собственные фильтры и директивы

Глава 5 «Создание приложений с помощью компонентов Angular 2» включает повторение изученных ранее элементов и показывает, как обеспечить произвольное масштабирование проектов на Angular 2 и выполнение соглашений об именовании и оформлении кода.

Глава 6 «Асинхронная обработка данных в Angular 2» рассматривает внедрение и развертывание HTTP-соединений со сторонними службами с помощью модуля Http для создания собственных клиентов.

Глава 7 «Маршрутизация в Angular 2» знакомит с механизмом маршрутизации в Angular 2 и его встроенными директивами, обеспечивающими различные способы загрузки компонентов посредством маршрутов и обработку состояний через программный интерфейс History API.

Глава 8 «Формы и аутентификация в Angular 2» демонстрирует различные способы создания веб-форм с двухсторонней привязкой данных к элементам ввода, а также создание сложных форм и проверку их допустимости.

Глава 9 «Анимация компонентов в Angular 2» рассматривает инструменты и классы для реализации анимационных эффектов, начиная с использования только стилей CSS, управляемых директивами

Angular 2, и заканчивая сложными переходами, управляемыми из JavaScript через компоновщики анимации Angular 2.

Глава 10 «Модульное тестирование в Angular 2» содержит пошаговое описание приемов тестирования приложений и типичных шаблонов развертывания модульных тестов для компонентов, директив, фильтров, маршрутов и служб.

Что потребуется для работы с книгой

Для работы с примерами в этой книге прежде всего понадобится веб-браузер, обновленный до последней версии. Мы рекомендуем Google Chrome или Mozilla Firefox, хотя Angular 2 поддерживает все современные браузеры.

Также понадобится программа-терминал, так как многие операции будут выполняться через консольные команды `npm`, для запуска которых также понадобятся установленные в системе Node.js и `npm`. Прочие необходимые модули и процедуры и порядок их установки будут описаны по ходу работы.

И наконец, для программирования вам потребуется текстовый редактор. В *главе 1 «Создание самого первого компонента в Angular 2»* приводится подробное описание лучших на настоящий момент интегрированных сред разработки приложений, использующих фреймворк Angular 2.

Кому адресована эта книга

Эта книга адресована веб-разработчикам, которые хотят освоить разработку современных мобильных и настольных веб-приложений следующего поколения с помощью Angular 2. От читателя не требуется опыт работы с Angular 1.x или 2, но предполагается хорошее владение JavaScript. Она отлично подойдет новичкам в Angular, нуждающимся в пояснении и определении его концепций.

Соглашения

В этой книге используется несколько разных стилей оформления текста для выделения разных видов информации. Ниже приведены примеры этих стилей с объяснением их назначения.

Программный код в тексте, имена таблиц баз данных, имена папок, имена файлов, расширения файлов, фиктивные адреса URL, пользовательский ввод и ссылки в Twitter будут выглядеть так: «В

результате этого действия в корневой папке проекта появится новый файл `tsconfig.json`, содержащий параметры, необходимые компилятору TypeScript для преобразования кода компонента в обычный код на JavaScript спецификации ECMAScript 5, который способны обработать современные браузеры».

Блоки программного кода оформляются так:

```
<body>
  <nav class="navbar navbar-default navbar-static-top">
    <div class="container">
      <div class="navbar-header">
        <strong class="navbar-brand">My Pomodoro Timer</strong>
      </div>
    </div>
  </nav>
  <pomodoro-timer></pomodoro-timer>
</body>
```

Когда потребуется привлечь ваше внимание к определенному фрагменту в блоке программного кода, он будет выделяться жирным:

```
<body>
  <nav class="navbar navbar-default navbar-static-top">
    <div class="container">
      <div class="navbar-header">
        <strong class="navbar-brand">My Pomodoro Timer</strong>
      </div>
    </div>
  </nav>
  <pomodoro-timer></pomodoro-timer>
</body>
```

Ввод и вывод в командной строке будут оформляться так:

```
$ npm install angular2 es6-shim es6-promise reflect-metadata rxjs zone.js
--save
```

Новые термины и важные слова будут выделены жирным. Текст, отображаемый на экране, например в меню или в диалогах, будет оформляться так: «Раздел **learn** предоставляет доступ к обучающим материалам, позволяющим ускорить освоение языка».



Предупреждения и важные сообщения будут выделены подобным образом.



Подсказки и советы будут выглядеть так.

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв прямо на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com, при этом напишите название книги в теме письма.

Если есть тема, в которой вы квалифицированы, и вы заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Поддержка пользователей

Если вы гордый владелец книги, изданной в Packt, мы готовы предоставить вам дополнительные услуги, чтобы ваша покупка принесла вам максимальную пользу.

Загрузка исходного кода примеров

Вы сможете загрузить файлы с примерами кода на странице:

<https://github.com/deeleman/learning-angular2>.

Кроме того, скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте www.dmkpress.com или www.дмк.рф в разделе «Читателям – Файлы к книгам».

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы удостовериться в качестве наших текстов, ошибки всё равно случаются. Если вы найдёте ошибку в одной из наших книг – возможно, ошибку в тексте или в коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от расстройств и можете нам улучшить последующие версии этой книги.

Если вы найдёте какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу dmkpress@gmail.com, и мы исправим это в следующих тиражах.

Нарушение авторских прав

Пиратство в Интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Packt очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в Интернете с незаконно выполненной копией любой нашей книги, пожалуйста, сообщите нам адрес копии или веб-сайта, чтобы мы могли принять меры.

Пожалуйста, свяжитесь с нами по адресу электронной почты dmkpress@gmail.com со ссылкой на подозрительные материалы.

Мы высоко ценим любую помощь по защите наших авторов, помогающую нам предоставлять вам качественные материалы.

Вопросы

Вы можете присылать любые вопросы, касающиеся данной книги, по адресу dm@dmk-press.ru или questions@packtpub.com. Мы постараемся разрешить возникшие проблемы.

Глава 1

Создание самого первого компонента в Angular 2

Если проследить тенденции за последнюю пару лет, можно отметить всплеск появления современных веб-фреймворков и библиотек на JavaScript для создания интерфейсов. Уже достигнута стадия, когда каждый день рождаются новые фреймворки, что заставляет разработчиков тщательно оценивать их, чтобы определить, стоит ли тратить время и силы на их освоение, а также их пригодность для применения в дальнейших проектах.

В конце концов, некоторые из них признаются наиболее актуальными. Вполне возможно, что следующие названия вам уже известны: Backbone, Ember, Knockout, Angular 1 и т. д.

В битву за господство в мире JavaScript вступили такие новые фреймворки, как React или Aurelia, отдающие предпочтение веб-компонентам и теневой модели DOM, которые являются краеугольным камнем их архитектуры. Приложения, построенные на их основе, получаются более модульными, масштабируемыми и простыми в сопровождении, не говоря уже об их непревзойденном уровне производительности.

С момента создания фреймворка Angular 1 прошло достаточно времени, чтобы его недостатки стали очевидными и с ними нельзя уже было мириться. Пришло время, когда отдельных улучшений в коде оказалось недостаточно. Потребовалось полностью сменить сам подход. В результате появился совершенно новый фреймворк, Angular 2, разработанный с нуля и полностью отвечающий новейшим тенден-

циям в отрасли. Центральное место в нем занимают веб-компоненты, а мощь теневой модели DOM используется для уменьшения времени реакции веб-сущностей на изменение состояния. Кроме того, Angular 2 включает систему обнаружения изменений состояния, встроенную в каждый компонент и отвечающую за организацию связей во всем дереве компонентов, составляющих приложение.

Определяющие черты Angular 2 выводят его за рамки простого фреймворка, основанного на веб-компонентах, поскольку его возможности охватывают практически все необходимое для построения современных веб-приложений: взаимодействие компонентов, универсальную поддержку различных платформ и устройств, превосходную технологию внедрения зависимостей, гибкий расширенный механизм маршрутизации для ослабления связей между компонентами, расширенную поддержку HTTP-сообщений, анимацию и интернационализацию. И это далеко не полный список.

В этой главе мы:

- узнаем причины уникальности Angular 2 в сравнении с предыдущими версиями;
- настроим среду для работы с Angular 2 и TypeScript;
- выберем интегрированную среду для удобной разработки приложений на основе Angular 2;
- создадим первый веб-компонент Angular 2 и поместим его в веб-страницу;
- добавим в веб-компонент простейшие интерактивные возможности;
- познакомимся с основными вспомогательными функциями форматирования выходных данных.

Обновленный подход

Как упоминалось выше, Angular 2 – это совершенно новый фреймворк Angular, основанный на архитектуре, изначально построенный на TypeScript, строгом надмножестве JavaScript, добавляющем статическую типизацию, поддержку интерфейсов и декораторов.

Другими словами, приложения, использующие Angular 2, основаны на архитектуре, включающей взаимосвязанные деревья веб-компонентов с их собственными интерфейсами ввода/вывода. Каждый компонент внутренне использует полностью переделанный механизм внедрения зависимостей. Справедливости ради следует отметить, что это лишь упрощенное описание того, чем Angular 2 яв-

ляется на самом деле. Однако даже простой проект, использующий Angular, наследует все его черты. В следующих главах мы познакомимся с организацией взаимодействий компонентов и внедрением зависимостей, после чего перейдем к маршрутизации, веб-формам и обмену информацией по протоколу HTTP. Это определение также объясняет, почему в книге не будет явных ссылок на Angular 1.x.

Очевидно, что нет смысла тратить время и место, ссылаясь на бесполезную для раскрытия описываемой темы информацию. Кроме того, поскольку предполагается, что читатель может быть незнаком с Angular 1.x, такие ссылки здесь явно излишни.

Веб-компоненты

Веб-компоненты представляют концепцию, которая охватывает четыре технологии, разработанные для создания элементов с высоким уровнем визуальной наглядности и возможностью повторного использования, что ведет к улучшению модульности, последовательности и облегчению обслуживания веб-приложений. К этим технологиям относятся:

- **шаблоны:** фрагменты разметки HTML, определяющие структуру отображаемого содержимого;
- **пользовательские элементы:** эти шаблоны содержат не только традиционные HTML-элементы, но и пользовательские оболочки, улучшающие визуальное представление элементов или расширяющие их функциональные возможности;
- **теневая модель DOM:** обеспечивает защищенную среду для инкапсуляции правил CSS-макета и JavaScript-модели поведения любого пользовательского элемента;
- **импорт HTML:** позволяет размещать в HTML-шаблоне не только HTML-элементы, но и другие HTML-документы.

Вообще говоря, компонент Angular 2 – это пользовательский элемент с шаблоном для размещения HTML-структуры макета, оформление которой определяется CSS-стилем, инкапсулированным в контейнер теневой модели DOM. Попробуем перевести это на обычный язык. Вспомним элемент управления HTML5, ограничивающий ввод числовых значений определенным диапазоном. Его удобно использовать, когда требуется дать возможность вводить значения, заключенные между двумя предопределенными границами. Если вы не использовали его раньше, вставьте следующий фрагмент разметки в пустой HTML-шаблон и загрузите его браузер:

```
<input id="mySlider" type="range" min="0" max="100" step="10">
```

Конец ознакомительного фрагмента.
Приобрести книгу можно
в интернет-магазине
«Электронный универс»
e-Univers.ru