Angular
для профессионалов

Адам Фримен

ПИТЕР®

Apress®

Адам Фримен

Angular для профессионалов

Серия «Для профессионалов»

Перевел с английского *Е. Матвеев*

Заведующая редакцией	*Ю. Сергиенко*
Ведущий редактор	*Н. Римицан*
Художественный редактор	*С. Заматевская*
Корректоры	*С. Беляева, Н. Викторова*
Верстка	*Н. Лукьянова*

ББК 32.988.02-018.1
УДК 004.43

Фримен А.

Ф88 Angular для профессионалов. — СПб.: Питер, 2018. — 800 с.: ил. — (Серия «Для профессионалов»).

ISBN 978-5-4461-0451-2

Выжмите из Angular — ведущего фреймворка для динамических приложений JavaScript — всё. Адам Фримен начинает с описания MVC и его преимуществ, затем показывает, как эффективно использовать Angular, охватывая все этапы, начиная с основ и до самых передовых возможностей, которые кроются в глубинах этого фреймворка.

Каждая тема изложена четко и лаконично, снабжена большим количеством подробностей, которые позволят вам стать действительно эффективными. Наиболее важные фичи даны без излишних подробностей, но содержат всю необходимую информацию, чтобы вы смогли обойти все подводные камни.

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

ISBN 978-1484223062 англ.
ISBN 978-5-4461-0451-2

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: 09.2017. 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 07.09.17. Формат 70×100/16. Бумага офсетная. Усл. п. л. 64,500. Тираж 1200. 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

Оглавление

1 Подготовка

Angular заимствует некоторые лучшие аспекты разработки на стороне сервера и использует их для расширения возможностей разметки HTML в браузере. Таким образом закладывается основа, которая упрощает и облегчает создание приложений с расширенной функциональностью. Приложения Angular строятся на базе паттерна проектирования MVC («модель — представление — контроллер»), который ориентирован на создание приложений, обладающих следующими характеристиками:

- *Простота расширения*: если вы понимаете основы, вам будет легко разобраться даже в самом сложном приложении Angular, а это означает, что вы сможете легко расширять приложения для поддержки новой полезной функциональности.
- *Удобство сопровождения*: приложения Angular просты в отладке, в них легко исправляются ошибки, а это означает простоту сопровождения кода в долгосрочной перспективе.
- *Удобство тестирования*: в Angular реализована хорошая поддержка модульного и сквозного тестирования. Следовательно, вы сможете находить и устранять дефекты до того, как ваши пользователи столкнутся с ними.
- *Стандартизация*: Angular работает на основе внутренней функциональности браузера, не создавая никаких препятствий для вашей работы. Это позволяет вам создавать веб-приложения, соответствующие стандартам, в которых задействована новейшая функциональность (например, различные API HTML5), популярные инструменты и фреймворки.

Angular — библиотека JavaScript с открытым кодом, финансированием разработки и сопровождения которой занимается компания Google. Она использовалась в ряде крупнейших и сложнейших веб-приложений. В этой книге вы узнаете все, что необходимо знать для использования Angular в ваших собственных проектах.

Что вам необходимо знать?

Чтобы книга принесла пользу, читатель должен быть знаком с основами веб-разработки, понимать, как работает HTML и CSS, а в идеале обладать практическими знаниями JavaScript. Для тех, кто забыл какие-то подробности, я кратко

напомню основные возможности HTML, CSS и JavaScript, используемые в книге, в главах 4, 5 и 6. Впрочем, вы не найдете здесь подробного справочника по элементам HTML и свойствам CSS. В книге, посвященной Angular, попросту не хватит места для полного описания HTML. Если вам требуется полный справочник по HTML и CSS, я рекомендую вам другую свою книгу, «The Definitive Guide to HTML5».

Много ли в книге примеров?

В книге *очень* много примеров. Angular лучше всего изучать на реальных примерах, поэтому я постарался включить в книгу как можно больше кода. Чтобы довести количество примеров до максимума, я воспользовался простой схемой, чтобы не приводить содержимое файлов снова и снова. Когда файл впервые встречается в главе, я привожу его полное содержимое, как в листинге 1.1. В заголовке листинга указывается имя файла и папка, в которой этот файл создается. При внесении изменений в код измененные команды выделяются жирным шрифтом.

Листинг 1.1. Пример документа

```
import { NgModule } from "@angular/core";
import { BrowserModule } from "@angular/platform-browser";
import { ProductComponent } from "./component";
import { FormsModule, ReactiveFormsModule  } from "@angular/forms";
import { PaAttrDirective } from "./attr.directive";

@NgModule({
    imports: [BrowserModule, FormsModule, ReactiveFormsModule],
    declarations: [ProductComponent, PaAttrDirective],
    bootstrap: [ProductComponent]
})
export class AppModule { }
```

Этот листинг позаимствован из главы 15. Неважно, что он делает; просто запомните, что при первом использовании каждого файла в главе будет приводиться полный листинг, как в листинге 1.1. Во втором и всех последующих примерах я привожу только измененные элементы, то есть *неполный листинг*. Частичные листинги легко узнать, потому что они начинаются и заканчиваются многоточием (. . .), как показано в листинге 1.2.

Листинг 1.2. Неполный листинг

```
...
<table class="table table-sm table-bordered table-striped">
    <tr><th></th><th>Name</th><th>Category</th><th>Price</th></tr>
    <tr *ngFor="let item of getProducts(); let i = index"
        [pa-attr]="getProducts().length < 6 ? 'bg-success' : 'bg-warning'">
        <td>{{i + 1}}</td>
        <td>{{item.name}}</td>
        <td [pa-attr]="item.category == 'Soccer' ? 'bg-info' : null">
```

```
            {{item.category}}
        </td>
        <td [pa-attr]="'bg-info'">{{item.price}}</td>
    </tr>
</table>
...
```

Листинг 1.2 также взят из главы 15. В нем приводится только контент элемента body, и часть команд выделена жирным шрифтом. Так я привлекаю ваше внимание к части примера, которая демонстрирует описываемый прием или возможность. В неполных листингах приводятся только части, изменившиеся по сравнению с полным листингом, приведенным где-то ранее. В некоторых случаях изменения приходится вносить в разных частях одного файла; тогда я просто опускаю некоторые элементы или команды для краткости, как показано в листинге 1.3.

Листинг 1.3. Часть команд опущена для краткости

```
import { ApplicationRef, Component } from "@angular/core";
import { Model } from "./repository.model";
import { Product } from "./product.model";
import { ProductFormGroup } from "./form.model";

@Component({
    selector: "app",
    templateUrl: "app/template.html"
})
export class ProductComponent {
    model: Model = new Model();
    form: ProductFormGroup = new ProductFormGroup();

    // ...Другие свойства и методы опущены для краткости...

    showTable: boolean = true;
}
```

Эта схема позволила мне включить в книгу больше примеров, однако она усложняет поиск описания конкретных возможностей. По этой причине все главы, в которых описываются возможности Angular, начинаются со сводной таблицы с описанием различных возможностей, представленных в этой главе, и листингов, демонстрирующих их использование.

Где взять примеры кода?

Проекты примеров всех глав этой книги можно загрузить на сайте *www.apress.com*. Архив загружается бесплатно и включает все поддерживающие ресурсы, необходимые для воссоздания примеров без необходимости вводить весь код заново. Загружать код не обязательно, но возможность простого копирования кода в проекты позволит вам легко экспериментировать с примерами.

Как подготовить среду разработки?

В главе 2 вы познакомитесь с Angular на примере создания простого приложения. В ходе рассмотрения примера я объясню, как подготовить среду разработки для работы с Angular.

Как связаться с автором

Если у вас возникнут проблемы с запуском примеров или вы обнаружите ошибки в книге, свяжитесь со мной по адресу *adam@adam-freeman.com*; я постараюсь вам помочь.

Итоги

В этой главе я кратко обрисовал содержимое и структуру книги. Разработку приложений Angular лучше всего изучать на примерах, поэтому в следующей главе мы с ходу возьмемся за дело. Вы узнаете, как подготовить среду разработки и как использовать ее для создания вашего первого приложения Angular.

2 Первое приложение

Лучший способ начать работу с Angular — просто взяться за дело и создать веб-приложение. В этой главе я покажу, как настроить среду разработки, и проведу вас по основным этапам процесса создания простейшего приложения, начиная с построения статического макета функциональности и применения возможностей Angular для построения динамического веб-приложения (пусть и несложного). В главах 7–10 будет рассмотрено создание более сложных и реалистичных приложений Angular, но пока будет достаточно простого примера, который представит основные компоненты приложений Angular и заложит основу для других глав в этой части книги.

Не беспокойтесь, если что-то из материала этой главы покажется непонятным. Поначалу изучение Angular создает немало трудностей, поэтому эта глава всего лишь дает общее представление об основной последовательности разработки приложений Angular и о том, как стыкуются различные компоненты. На первых порах что-то может остаться непонятным, но когда вы перевернете последнюю страницу, вы будете понимать все, что происходит в этой главе, а также многое другое.

Подготовка среды разработки

Разработка Angular-приложений требует определенной подготовки. Далее я объясню, как выполнить необходимые действия для создания вашего первого проекта. Angular широко поддерживается популярными инструментами разработки, так что вы можете выбрать тот вариант, который лучше подходит лично вам.

Установка Node.js

Многие инструменты, используемые для разработки приложений Angular, зависят от Node.js (также известной как Node) — простой и эффективной исполнительной среды для серверных приложений, написанных на JavaScript, которая была создана в 2009 году. Node.js использует ядро JavaScript, задействованное в браузере Chrome, и предоставляет API для выполнения кода JavaScript за пределами окружения браузера.

Среда Node.js пользовалась большим успехом как сервер приложений, но для этой книги она интересна прежде всего тем, что заложила основу для нового поколения кроссплатформенной разработки и средств построения. Некоторые умные решения, принятые командой разработки Node.js, а также кроссплатформенная поддержка со стороны исполнительной среды JavaScript для Chrome открыли замечательные возможности, которые были моментально замечены увлеченными разработчиками инструментальных средств. Короче говоря, среда Node.js стала важнейшим инструментом разработки веб-приложений.

Очень важно, чтобы вы загрузили ту же версию Node.js, которая используется в этой книге. Хотя среда Node.js относительно стабильна, время от времени в API происходят критические изменения, которые могут нарушить работоспособность приведенных в книге примеров.

Я использую версию 6.10.1, самую актуальную версию с долгосрочной поддержкой на момент написания книги. Возможно, вы выберете для своих проектов более свежий выпуск, но для примеров книги следует придерживаться версии 6.10.1. Полный набор всех выпусков 6.10.1 с программами установки для Windows и Mac OS и двоичными пакетами для других платформ доступен по адресу *https://nodejs.org/dist/v6.10.1*.

В ходе установки Node.js не забудьте указать ключ для добавления пути к исполняемым файлам Node.js в переменную окружения. После завершения установки выполните следующую команду:

```
node -v
```

Если установка прошла так, как положено, команда выводит следующий номер версии:

```
V6.10.1
```

Установка Node.js включает менеджер пакетов проекта NPM (Node Package Manager). Выполните следующую команду, чтобы убедиться в том, что NPM работает:

```
npm -v
```

Если все работает так, как должно, выводится следующий номер версии:

```
3.10.10
```

Установка пакета angular-cli

Пакет `angular-cli` стал стандартным инструментом создания и управления пакетами Angular в ходе разработки. В исходной версии книги я показывал, как создавать пакеты Angular «с нуля»; это довольно долгий и ненадежный процесс, который упрощается благодаря `angular-cli`. Чтобы установить `angular-cli`, откройте новую командную строку и введите следующую команду:

```
npm install --global @angular/cli@1.0.0
```

В системе Linux или macOS вам, возможно, придется использовать команду `sudo`.

Установка Git

Система управления версиями Git нужна для управления некоторыми пакетами, необходимыми для разработки Angular. Если вы работаете в Windows или macOS, загрузите и запустите программу установки по адресу *https://git-scm.com/downloads*. (Возможно, в macOS вам придется изменить настройки безопасности для открытия программы установки, которую разработчики не снабдили цифровой подписью.)

Система Git заранее устанавливается в большинстве дистрибутивов Linux. Если вы захотите установить самую свежую версию, обратитесь к инструкциям для своего дистрибутива по адресу *https://git-scm.com/download/linux*. Например, для Ubuntu — моего дистрибутива Linux — используется следующая команда:

```
sudo apt-get install git
```

Завершив установку, откройте новую командную строку и выполните следующую команду, чтобы проверить правильность установки Git:

```
git --version
```

Команда выводит версию установленного пакета Git. На момент написания книги новейшей версией Git для Windows была версия 2.12.0, для macOS — 2.10.1, а для Linux – 2.7.4.

Установка редактора

В разработке приложений Angular может использоваться любой редактор для программистов. Выбор возможных вариантов огромен. В некоторых редакторах предусмотрена расширенная поддержка работы с Angular, включая выделение ключевых терминов и хорошую интеграцию с инструментарием. Если у вас еще нет любимого редактора для разработки веб-приложений, в табл. 2.1 представлены некоторые популярные варианты. Материал книги не зависит от какого-либо конкретного редактора; используйте тот редактор, в котором вам удобнее работать.

Один из важнейших критериев при выборе редактора — возможность фильтрации содержимого проекта, чтобы вы могли сосредоточиться на работе с некоторым подмножеством файлов. Проект Angular может содержать кучу файлов, многие из которых имеют похожие имена, поэтому возможность быстро найти и отредактировать нужный файл чрезвычайно важна. В редакторах эта функция может быть реализована разными способами: с выводом списка файлов, открытых для редактирования, или возможностью исключения файлов с заданным расширением.

ПРИМЕЧАНИЕ

Если вы работаете в Visual Studio (полноценной среде Visual Studio, не в Visual Studio Code), процесс работы с проектами Angular становится еще сложнее, особенно если вы захотите добавить Angular в проект ASP.NET Core MVC. Я планирую выпустить отдельное дополнение по использованию Angular в Visual Studio, вы сможете бесплатно загрузить его из репозитория GitHub этой книги.

Таблица 2.1. Популярные редакторы с поддержкой Angular

Название	Описание
Sublime Text	Sublime Text — коммерческий кроссплатформенный редактор с пакетами для поддержки большинства языков программирования, фреймворков и платформ. За подробностями обращайтесь по адресу www.sublimetext.com
Atom	Atom — бесплатный кроссплатформенный редактор с открытым кодом, уделяющий особое внимание возможностям настройки и расширения. За подробностями обращайтесь по адресу atom.io
Brackets	Brackets — бесплатный редактор с открытым кодом, разработанный компанией Adobe. За подробностями обращайтесь по адресу brackets.io
WebStorm	WebStorm — платный кроссплатформенный редактор с множеством интегрированных инструментов, чтобы вам не пришлось пользоваться командной строкой во время разработки. За подробностями обращайтесь по адресу www.jetbrains.com/webstorm
Visual Studio Code	Visual Studio Code — бесплатный кроссплатформенный редактор с открытым кодом, разработанный компанией Microsoft, с хорошими возможностями расширения. За подробностями обращайтесь по адресу code.visualstudio.com

Установка браузера

Остается принять последнее решение: выбрать браузер, который будет использоваться для проверки результатов работы в ходе разработки. Все браузеры последнего поколения обладают хорошей поддержкой для разработчика и хорошо сочетаются с Angular. Я использовал в книге Google Chrome; этот же браузер я могу порекомендовать и вам.

Создание и подготовка проекта

После того как вы установите Node.js, `angular-cli`, редактор и браузер, в вашем распоряжении окажется все необходимое для запуска процесса разработки.

Создание проекта

Выберите подходящую папку и выполните следующую команду в режиме командной строки, чтобы создать новый проект с именем todo:

```
ng new todo
```

Команда `ng` предоставляется пакетом `angular-cli`, а подкоманда `ng new` создает новый проект. Процесс установки создает папку с именем todo, которая содержит все файлы конфигурации, необходимые для разработки Angular, некоторые временные файлы, упрощающие начальную стадию разработки, и пакеты NPM, необходимые для разработки, запуска и развертывания приложений Angular. (Пакетов NPM довольно много; это означает, что создание проекта может занять много времени.)

Создание файла пакета

NPM использует файл с именем package.json для чтения списка программных пакетов, необходимых для проекта. Файл package.json создается `angular-cli` как часть инфраструктуры проекта, но он содержит только базовые пакеты, необходимые для разработки Angular. Для приложения этой главы понадобится пакет Bootstrap, не входящий в базовый набор пакетов. Отредактируйте файл project.json в папке todo и добавьте пакет Bootstrap, как показано в листинге 2.1.

ВНИМАНИЕ

К тому моменту, когда вы будете читать книгу, будут выпущены новые версии по крайней мере части пакетов из листинга 2.1. Чтобы ваши результаты соответствовали результатам примеров в этой и других главах, очень важно использовать конкретные версии, указанные в листинге. Если у вас возникнут проблемы с примерами этой или какой-либо из последующих глав, попробуйте использовать исходный код из архива, прилагаемого к книге; его можно загрузить на сайте издательства apress.com. Если же положение окажется совсем безвыходным, отправьте мне сообщение по адресу adam@adam-freeman.com, я постараюсь помочь вам.

Листинг 2.1. Содержимое файла package.json в папке todo

```
{
  "name": "todo",
  "version": "0.0.0",
  "license": "MIT",
  "scripts": {
    "ng": "ng",
    "start": "ng serve",
    "build": "ng build",
    "test": "ng test",
    "lint": "ng lint",
    "e2e": "ng e2e"
  },
  "private": true,

  "dependencies": {
    "@angular/common": "^4.0.0",
    "@angular/compiler": "^4.0.0",
    "@angular/core": "^4.0.0",
    "@angular/forms": "^4.0.0",
    "@angular/http": "^4.0.0",
    "@angular/platform-browser": "^4.0.0",
    "@angular/platform-browser-dynamic": "^4.0.0",
    "@angular/router": "^4.0.0",
    "core-js": "^2.4.1",
    "rxjs": "^5.1.0",
    "zone.js": "^0.8.4",
    "bootstrap": "4.0.0-alpha.4"
  },

  "devDependencies": {
    "@angular/cli": "1.0.0",
```

```
    "@angular/compiler-cli": "^4.0.0",
    "@types/jasmine": "2.5.38",
    "@types/node": "~6.0.60",
    "codelyzer": "~2.0.0",
    "jasmine-core": "~2.5.2",
    "jasmine-spec-reporter": "~3.2.0",
    "karma": "~1.4.1",
    "karma-chrome-launcher": "~2.0.0",
    "karma-cli": "~1.0.1",
    "karma-jasmine": "~1.1.0",
    "karma-jasmine-html-reporter": "^0.2.2",
    "karma-coverage-istanbul-reporter": "^0.2.0",
    "protractor": "~5.1.0",
    "ts-node": "~2.0.0",
    "tslint": "~4.5.0",
    "typescript": "~2.2.0"
  }
}
```

В файле package.json перечислены пакеты, необходимые для начала разработки приложений Angular, и некоторые команды для их использования. Все параметры конфигурации будут описаны в главе 11, а пока достаточно понимать, для чего нужна каждая секция файла package.json (табл. 2.2).

Таблица 2.2. Секции файла package.json

Имя	Описание
scripts	Список сценариев, запускаемых в режиме командной строки. Секция scripts в листинге запускает команды, используемые для компиляции исходного кода, и сервер HTTP для разработки
dependencies	Список пакетов NPM, от которых зависит работа веб-приложения. Для каждого пакета указан номер версии. В секции dependencies в листинге перечислены базовые пакеты Angular; библиотеки, от которых зависит Angular; и CSS-библиотека Bootstrap, которая используется для стилевого оформления контента HTML в книге
devDependencies	Список пакетов NPM, которые используются в разработке, но не нужны для работы приложения после его развертывания. В этой секции перечислены пакеты, компилирующие файлы TypeScript, предоставляющие функциональность сервера HTTP в процессе разработки и обеспечивающие тестирование

Установка пакета NPM

Чтобы файл package.json был обработан NPM для загрузки и установки указанного в нем пакета Bootstrap, выполните следующую команду из папки todo:

```
npm install
```

NPM выдает несколько предупреждений относительно обрабатываемых пакетов, но сообщений об ошибках быть не должно.

Запуск сервера

Инструментарий и базовая структура находятся на своих местах; пришло время убедиться в том, что все работает нормально. Выполните следующие команды из папки todo:

```
npm serve --port 3000 --open
```

Команда запускает сервер HTTP для разработки, который был установлен `angular-cli` и настроен для работы с инструментарием разработчика Angular. Запуск занимает немного времени для подготовки проекта, а вывод выглядит примерно так:

```
** NG Live Development Server is running on http://localhost:3000 **
Hash: b8843310528d229c2540
Time: 11251ms
chunk    {0} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 158 kB {4}
[initial] [rendered]
chunk {1} main.bundle.js, main.bundle.js.map (main) 3.69 kB {3} [initial]
          [rendered]
chunk {2} styles.bundle.js, styles.bundle.js.map (styles) 9.77 kB {4} [initial]
          [rendered]
chunk {3} vendor.bundle.js, vendor.bundle.js.map (vendor) 2.37 MB [initial]
          [rendered]
chunk {4} inline.bundle.js, inline.bundle.js.map (inline) 0 bytes [entry]
          [rendered]
          webpack: Compiled successfully.
```

Не беспокойтесь, если в вашем случае вывод будет выглядеть немного иначе — главное, чтобы после завершения подготовки появилось сообщение `Compiled successfully`. Через несколько секунд откроется окно браузера (рис. 2.1); это означает, что запуск проекта прошел успешно и в нем используется временный контент, сгенерированный `angular-cli`.

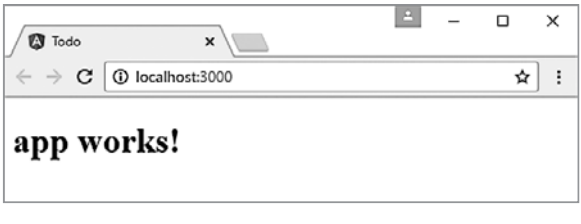

Рис. 2.1. Временный контент HTML

Редактирование файла HTML

Хотя пакет `angular-cli` добавил временный контент, мы сейчас удалим все лишнее и начнем с заготовки HTML, содержащей статический контент. Позже эта заготовка будет расширена для Angular. Отредактируйте файл index.html в папке todo/src и включите в него контент из листинга 2.2.

Листинг 2.2. Содержимое файла index.html в папке todo/src

```
<!DOCTYPE html>
<html>
<head>
    <title>ToDo</title>
    <meta charset="utf-8" />
    <link href="node_modules/bootstrap/dist/css/bootstrap.min.css"
          rel="stylesheet" />
</head>
<body class="m-a-1">

    <h3 class="bg-primary p-a-1">Adam's To Do List</h3>

    <div class="m-t-1 m-b-1">
        <input class="form-control" />
        <button class="btn btn-primary m-t-1">Add</button>
    </div>

    <table class="table table-striped table-bordered">
        <thead>
            <tr>
                <th>Description</th>
                <th>Done</th>
            </tr>
        </thead>
        <tbody>
            <tr><td>Buy Flowers</td><td>No</td></tr>
            <tr><td>Get Shoes</td><td>No</td></tr>
            <tr><td>Collect Tickets</td><td>Yes</td></tr>
            <tr><td>Call Joe</td><td>No</td></tr>
        </tbody>
    </table>
</body>
</html>
```

Сервер HTTP для разработки `angular-cli` добавляет фрагмент JavaScript в контент HTML, передаваемый браузеру. JavaScript открывает обратное подключение к серверу и ждет сигнала на перезагрузку страницы; сигнал отправляется при обнаружении сервером изменений в любых файлах из каталога todo. Как только вы сохраните файл index.html, сервер обнаружит изменения и отправит сигнал. Браузер перезагружается и выводит новый контент (рис. 2.2).

ПРИМЕЧАНИЕ

При внесении изменений в группу файлов может случиться так, что браузер не сможет загрузить и выполнить приложение, особенно в последующих главах, когда приложения станут более сложными. В большинстве случаев сервер HTTP для разработки инициирует перезагрузку в браузере, и все будет нормально, но если у него возникнут затруднения, просто нажмите кнопку обновления в браузере или перейдите по адресу http://localhost:3000, чтобы восстановить нормальную работу.

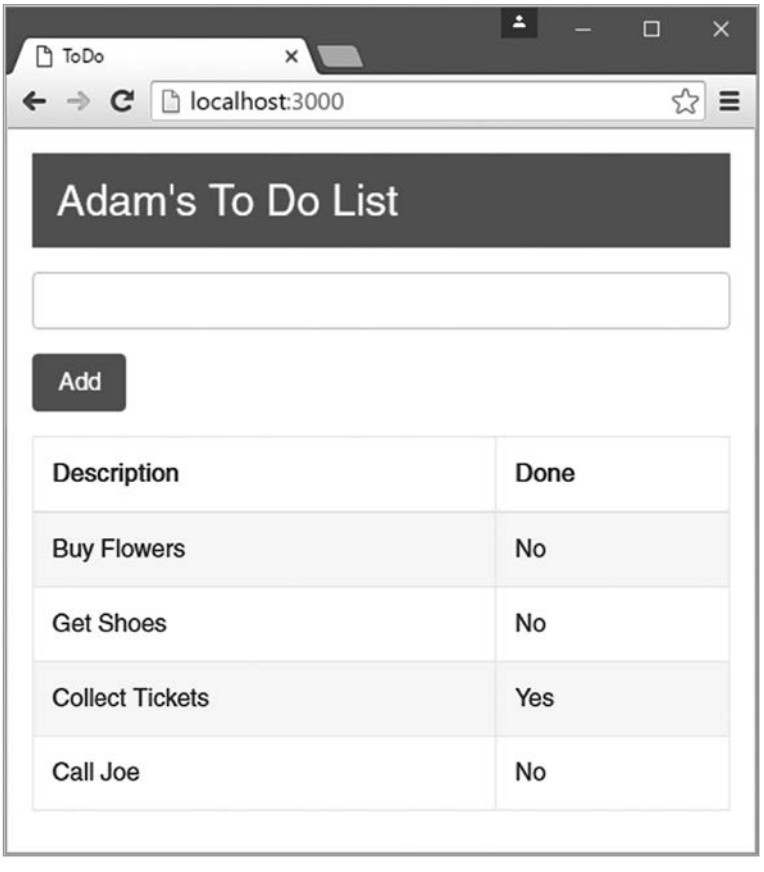

Рис. 2.2. Изменение содержимого файла HTML

Элементы HTML в файле index.html показывают, как будет выглядеть простое приложение Angular, которое мы создадим в этой главе. Его ключевые элементы — заголовок с именем пользователя, поле ввода, кнопка для добавления новой задачи в список и таблица со всеми задачами и признаками их завершения.

В этой книге я использую для оформления контента HTML превосходный CSS-фреймворк Bootstrap. Работа Bootstrap базируется на назначении элементам классов:

```
...
<h3 class="bg-primary p-a-1">Adam's To Do List</h3>
...
```

Элементу `h3` назначены два класса. Класс `bg-primary` назначает цветом фона элемента первичный цвет текущей темы Bootstrap. Я использую тему по умолчанию, в которой первичным цветом является темно-синий; также доступны другие цвета из темы, включая `bg-secondary`, `bg-info` и `bg-danger`. Класс `p-a-1` добавляет ко

всем сторонам элемента фиксированные отступы, чтобы текст был окружен небольшим свободным пространством.

В следующем разделе мы удалим из файла разметку HTML, разобьем ее на несколько меньших фрагментов и используем для создания простого приложения Angular.

> **ИСПОЛЬЗОВАНИЕ ПРЕДВАРИТЕЛЬНОЙ ВЕРСИИ BOOTSTRAP**
>
> В этой книге используется предварительная версия фреймворка Bootstrap. В то время, когда я пишу эти строки, команда Bootstrap занимается разработкой Bootstrap версии 4 и уже опубликовала несколько ранних выпусков. Эти выпуски считаются «альфа-версиями», но они содержат качественный код и работают достаточно стабильно для использования в примерах книги.
>
> Когда мне пришлось выбирать между версией Bootstrap 3, которая скоро станет устаревшей, и предварительной версией Bootstrap 4, я решил использовать новую версию, несмотря на то что некоторые имена классов, используемых для оформления элементов HTML, могут измениться в окончательной версии. Это означает, что для получения предполагаемых результатов в примерах вы должны использовать ту же версию Bootstrap — как и в остальных пакетах, перечисленных в файле package.json из листинга 2.1.

Добавление функциональности Angular в проект

Статическая разметка HTML в файле index.html заменяет простейшее приложение. Пользователь может просматривать список задач, помечать выполненные и создавать новые задачи. В дальнейших разделах мы добавим в проект Angular и воспользуемся некоторыми базовыми возможностями для того, чтобы вдохнуть жизнь в приложение. Для простоты предполагается, что у приложения всего один пользователь и нам не нужно беспокоиться о сохранении данных в приложении, а это означает, что изменения в списке будут потеряны при закрытии или перезагрузке окна браузера. (В последующих примерах, включая приложение SportsStore, создаваемое в главах 7–10, будет продемонстрирован механизм долгосрочного хранения данных.)

Подготовка файла HTML

Первым шагом на пути включения Angular в приложение станет подготовка файла index.html (листинг 2.3).

Листинг 2.3. Подготовка к использованию Angular в файле index.html

```
<!DOCTYPE html>
<html>
<head>
    <title>ToDo</title>
```