# Angular Development with TypeScript, Second Edition

# 1. Table of Contents


## 1.1. Introducing Angular

### 1.1.1. A High-Level Overview of Angular

### 1.1.2. Using Node Package Manager

#### Specifying project dependencies in package.json

#### Semantic versioning

### 1.1.3. Introducing Angular CLI

#### Generating a new Angular project

#### Reviewing the generated code

#### Generating artifacts in your project

#### Serving apps in dev mode

#### JiT and AoT Compilations

#### Creating bundles with the -prod option

### 1.1.4. Introducing the sample ngAuction app

### 1.1.5. Summary

## 1.2. Chapter 2. The main building blocks of an Angular app

### 1.2.1. The ng generate command of Angular CLI

### 1.2.2. Components

### 1.2.3. Directives

#### Creating a custom directive

### 1.2.4. Services

### 1.2.5. Pipes

#### Creating a custom pipe

### 1.2.6. Modules

#### Feature modules

### 1.2.7. Configuring Angular CLI projects with .angular-cli.json

### 1.2.8. First steps with data binding

#### One- and two-way data binding in action

### 1.2.9. Hands-on: Getting Started with ngAuction

#### The initial project setup for ngAuction

#### Hands-on: Generating components for ngAuction

### 1.2.10. Summary

## 1.3. Chapter 3. Navigation with Router

### 1.3.1. Routing basics

### 1.3.2. Location Strategies

#### Hash-based navigation

#### History API-based navigation

### 1.3.3. The building blocks of client-side navigation

### 1.3.4. Passing Data to Routes

#### Extracting parameters from ActivatedRoute

#### Passing query parameters to a route

### 1.3.5. Child Routes

### 1.3.6. Guarding routes

#### Implementing the CanActivate guard

#### Implementing the CanDeactivate guard

#### Implementing the Resolve guard

### 1.3.7. Developing a SPA with multiple router outlets

### 1.3.8. Lazy Loading of modules

#### Preloaders

### 1.3.9. Hands-on: Adding navigation to ngAuction

### 1.3.10. Summary

## 1.4. Chapter 4. Dependency Injection in Angular

### 1.4.1. The Dependency Injection and Inversion of Control patterns

#### The Dependency Injection pattern

#### The Inversion of Control pattern

#### Benefits of dependency injection

### 1.4.2. Injectors and providers

#### How to declare a provider

### 1.4.3. A simple app with Angular DI

#### Injecting a product service

#### Injecting the HttpClient service

### 1.4.4. Switching injectables made easy

#### Declaring providers with useFactory and useValue

#### Using InjectionToken

### 1.4.5. Dependency injection in a modularized app

#### Providers in lazy-loaded modules

#### Providers in eagerly-loaded modules

### 1.4.6. Hands-on: using Angular Material in ngAuction

#### A brief overview of the Angular Material library

====Adding Angular Material library to the project ==== Adding a feature module with AM components ==== Modifying the appearance of the NavbarComponent ==== Modifying the UI of the SearchComponent ==== Replacing the carousel with an image ==== More fixes with spacing ==== Using md-card in ProductItemComponent ==== Adding styles to HomeComponent === Summary

## 1.5. Chapter 5. Reactive programming in Angular

### 1.5.1. Handling events without observables

### 1.5.2. Turning DOM events into observables

### 1.5.3. Handling observable events using Forms API

### 1.5.4. How to cancel HTTP requests with switchMap

### 1.5.5. Using AsyncPipe

### 1.5.6. Observables and the router

### 1.5.7. Flex Layout and ObservableMedia

#### Using Flex Layout directives

#### ObservableMedia service

### 1.5.8. Hands-on: re-writing ngAuction

#### Why re-writing the ngAuction app from scratch?

#### Generating a new ngAuction app

#### Creating a custom Angular Material theme with Saas

#### Adding a toolbar to the top level component

#### Creating the product service

#### Creating the home module

#### Configuring routes

#### Running ngAuction

### 1.5.9. Summary

### 1.5.10. Summary

## 1.6. Chapter 6. Implementing Components Communication

### 1.6.1. Inter-component communication

### 1.6.2. Input and output properties

#### Input properties

#### Output properties and custom events

### 1.6.3. Implementing the mediator design pattern

#### Using a common parent as a mediator

#### Using an injectable service as a mediator

### 1.6.4. Changing templates at runtime with ngContent

#### View encapsulation modes

#### Projecting onto multiple areas

### 1.6.5. Component lifecycle

### 1.6.6. A high-level overview of change detection

### 1.6.7. Exposing a child component’s API

### 1.6.8. Hands-on: adding the product detail view to ngAuction

### 1.6.9. Summary

## 1.7. Chapter 7. Working With Angular Forms API

### 1.7.1. Overview of HTML Forms

#### Introducing a User Registration Form

### 1.7.2. Template-Driven Form API

#### Form directives

#### Submitting form

### 1.7.3. Reactive Forms API

#### Creating form model with form controls

#### Binding form model to the template with form directives

#### Submitting form

### 1.7.4. Form Validation

#### Built-in validators

#### Creating synchronous custom validators

#### Creating asynchronous custom validators

### 1.7.5. Hands-on: Adding validation to the ngAuction search form

### 1.7.6. Summary

## 1.8. Chapter 8. Interacting with Servers Using HTTP and WebSockets

### 1.8.1. Brief Overview of the Http and HttpClient Objects Services

### 1.8.2. Creating a Web Server with Node and TypeScript

#### Creating a Web Server with Node.js and Express frameworks

#### Serving JSON

#### Auto-reloading of the Node.js app with ts-node and nodemon

#### Adding the RESTful API for Serving Products

### 1.8.3. Bringing Angular and Node Together

#### Static Resources on the Server

#### Making GET Requests with HttpClient Service

#### Unwrapping Observables Inside Templates with Async pipe

### 1.8.4. Configuring a proxy

### 1.8.5. Using HttpClient Service

#### Interceptors

#### Progress Events

#### Building an app for prod deployment on the server

### 1.8.6. Client-Server Communications via WebSocket

#### Pushing Data From Node Server

#### Turning WebSocket into Observable

### 1.8.7. Hands-on: Implementing Product Search and Bid Notifications in ngAuction

#### Implementing Product Search using HttpClient

#### Broadcasting ngAuction Bids using WebSocket

### 1.8.8. Summary

## 1.9. Chapter 9. Testing Angular Applications

### 1.9.1. Unit and End-to-End testing

### 1.9.2. Getting to know the Jasmine Framework

### 1.9.3. What Comes with Angular Testing Library

#### Testing Services

#### Testing Component Router

#### Testing Components

### 1.9.4. Running Tests with Karma

### 1.9.5. End-to-End Testing with Protractor

#### Intro to the Protractor framework

### 1.9.6. Hands-on: Unit Testing of ngAuction

### 1.9.7. Summary

## 1.10. Chapter 10. Maintaining App State with NgRX

### 1.10.1. Introducing unidirectional data flow

### 1.10.2. Intorducing ngrx

#### Managing application state with @ngrx/store package

#### Integrating ngrx store with Angular router

#### Managing side-effects with @ngrx/effects package

### 1.10.3. Debugging Angular application with ngrx devtools

### 1.10.4. Modularizing ngrx-based Angular application

### 1.10.5. Hands-on: Refactoring ngAuction to use ngrx

### 1.10.6. Summary

## 1.11. Appendix A. An Overview of ECMAScript

### 1.11.1. How to Run Code Samples

### 1.11.2. Template Literals

#### Multi-line Strings

#### Tagged Template Strings

### 1.11.3. Optional Parameters and Default Values

### 1.11.4. Scope of Variables

#### Hoisting of variables declarations

#### Block Scoping With let and const

#### Block Scope for Functions

### 1.11.5. Arrow Function Expressions, This, and That

### 1.11.6. The Rest operator

### 1.11.7. The Spread operator

### 1.11.8. Generators

### 1.11.9. Destructuring

#### Destructuring objects

#### Destructuring arrays

### 1.11.10. Iterating with forEach(), for-in, and for-of

#### Using forEach()

#### Using for-in

#### Using for-of

### 1.11.11. Classes and Inheritance

#### Constructors

#### Static Variables

#### Getters, Setters, and Method Definitions

#### The super Keyword and the super Function

### 1.11.12. Asynchronous Processing

#### A Callback Hell

#### ES6 Promises

#### Resolving Several Promises at Once

#### async-await

#### Modules

#### Imports and Exports

# 1.12. Appendix B. TypeScript Essentials

## 1.12.1. Why Writing Angular Apps in TypeScript

## 1.12.2. The role of transpilers

## 1.12.3. Getting Started With TypeScript

#### Installing and using the TypeScript compiler

#### TypeScript as a Superset of JavaScript

## 1.12.4. How to run the code samples

## 1.12.5. Optional Types

## 1.12.6. Functions

#### Default Parameters

#### Optional Parameters

#### Arrow Function Expressions

## 1.12.7. Classes

#### Access Modifiers

#### Methods

#### Inheritance

## 1.12.8. Interfaces

#### Declaring Custom Types with Interfaces

#### Enforcing API contracts with the implements keyword

## 1.12.9. Generics

## 1.12.10. The readonly modifier

## 1.12.11. Decorators

#### 1.12.12. Dynamic imports

#### 1.12.13. Using Type Definition Files

#### 1.12.14. Controlling code style with TSLint

## 1.13. Appendix C: Using Node Package Manager

#### 1.13.1. Specifying project dependencies in package.json

#### 1.13.2. Semantic versioning

## 1.14. Appendix D: RxJS essentials

#### 1.14.1. Getting familiar with RxJS terminology

#### 1.14.2. Observable, observer, and subscriber

#### 1.14.3. Creating observables

#### 1.14.4. Operators map, filter, and reduce

#### 1.14.5. Using Observer's API

#### 1.14.6. Using RxJS Subject

#### 1.14.7. The flatMap operator

#### 1.14.8. The switchMap operator

#### 1.14.9. Error handling with the catch operator