

Modern **Angular**

Architecture, Concepts, Implementation



Manfred Steyer

Modern Angular

Architecture, Concepts, Implementation

Manfred Steyer

March 29, 2026

Contents

1	Getting Started with Angular	2
	Tooling	2
	Development Environment	2
	Node.js	3
	Angular CLI	3
	Example Project Repository	3
	Getting Started with the Angular CLI	4
	Generating a New Project	4
	Starting an Angular Application	4
	Project Structure of CLI Projects	6
	Inspecting the Generated Source Code	7
	App Component with Signals and Data Bindings	7
	Bootstrapping the App Component	9
	Connecting the Root Component to the Start Page	10
	Installing Additional Packages	11
	Adding Components and Styles	12
	Configuring the Angular CLI	12
	Initializing the Linter	14
	Building the Application	14
	Summary	14
2	Signal-Based Components	15
	Creating a Signal-based Component	15
	Scaffolding Components with the Angular CLI	16
	Suffixes and the Updated Angular Style Guide	17
	Data Model	17
	Component Logic	18
	Importing Dependencies for the Template	22
	Template and Data Binding	23
	Event Bindings	25
	Property Bindings	25
	Built-in Control Flow	25
	Interpolation and Pipes	26
	Conditional Styling	27
	Calling the Component	28
	Running and Debugging the Application	29
	Starting the Application	29
	Discovering Errors in the Developer Console	29

Debugging the Application in the Browser	30
Debugging with Visual Studio Code	31
Data Access	31
Data Access Using the HttpClient	31
Data Access Using the httpResource	36
Sub Components with Properties and Events	40
Preparations	40
Planning Properties and Events	42
Components with InputSignals and OutputEmitterRefs	43
Component Template	44
Binding to a Sub-Component	45
Content Projection	47
Writable Inputs with ModelSignals	49
Two-way Bindings	51
Summary	54
3 Reactive Design with Signals	55
Building Blocks of Reactive Design	55
Computed Signals	55
Resources	57
httpResource	57
rxResource	59
resource (Promise-based)	61
Effects	63
Signal Semantics in Angular	66
Signals in the Component Lifecycle	66
Auto-Tracking and the Reactive Context	67
Discussing Explicit Effects	69
Untracking	70
Glitch-Free Property	70
Equality and Immutability	71
Establishing a Reactive Flow	73
Thinking in Terms of the Signal Graph	73
Implementing the Reactive Flow	74
Summary	77
4 Navigation & Lazy Loading with the Router	78
Overview	78
Getting Started with the Router	79
Setting up Routing Configuration	80
Adding a Placeholder in the App	82
Setting up Hyperlinks to Activate Routes	83
Trying out Your Setup	85
Programmatically Changing Routes	85
Parameterized Routes	86
Types of Routing Parameters	86

Reading Parameters with <code>ActivatedRoute</code>	86
Using <code>withComponentInputBinding()</code> for Reading Parameters	87
Configuring Parameterized Routes	89
Linking to Parameterized Routes	90
Hierarchical Routing with Child Routes	92
Overview of Child Routes	92
Implementing Child Components	93
Configuring Child Routes	96
Hyperlinks to Child Routes	97
Lazy Loading of Routes	98
Setting up Routes for Lazy Loading	98
Lazy Loading Individual Components with the Router	101
Verify Lazy Loading in the Browser	102
Preloading	102
Working with Query Strings and Hash Fragments	104
Path Routing vs. Hash Routing	105
<code>PathLocationStrategy</code>	106
<code>HashLocationStrategy</code>	107
Summary	109
5 State Management with Services & Signals	110
Services	110
Generating a Service	110
Implementing a Service	111
Injecting a Service	112
Injection Context	113
Services with Dependencies	115
Exchanging Services with Providers	116
Short-Hand Syntax for Providers	118
Provider Functions	119
Component-local Services	120
Route-local Services and Auto Cleanup	123
State Management	124
Implementing a Store with Signals	125
Consuming the Store	127
Delegated Signals	129
Lifetime and Scopes	131
Outlook: NgRx Signal Store	132
Summary	132
6 Signal Forms	133
A First Signal Form	133
Setting up the Component	133
Understanding the <code>FieldTree</code> Type	135
Binding to the Template	136

Working with Schemas	139
Using Separate Schemas	139
Controlling Behavior	140
Debouncing	141
Validating Against Zod and Standard Schema	143
Submitting Forms	144
Logic for Submitting Forms	144
Template for Submitting	146
Further Submit Actions	147
Custom Validators	147
Implementing a First Custom Validators	147
Refactoring Validators into Functions	148
Showing Validation Errors	149
Conditional Validation	151
Multi-Field Validators	152
Accessing Sibling Fields	154
Tree Validators	155
Asynchronous Validators	156
HTTP Validators	158
Large and Nested Forms	159
Form Groups	159
Form Arrays	161
Validating Form Arrays	163
Subforms	164
Working with Form Metadata	166
Reading Metadata	167
Displaying Metadata	168
Defining Custom Metadata	169
Reading Custom Metadata	170
Null and Undefined Values	171
Custom Fields	174
Summary	176
7 Modern Testing with Vitest	177
Vitest	177
Structure of a Vitest Test	177
Skipping Tests	179
Turning on Browser Mode	179
Running Tests	180
Angular and Vitest	181
Preparing the TestBed	181
A First Component Test	183
Working with Locators	184
Locating Elements via DebugElement	186
Defining Default Timeouts	186

Mocking and Spies	187
Mocking Services	187
Mocking Services on the Component Level	189
Mocking Child Components and Shallow Testing	189
Mocking HTTP Calls	190
Gray-Box Testing with Spies	194
Testing Routed Components	196
Testing Timers and Debouncing	196
Mocking Asynchronicity and Delays	197
Fake Timers	198
Testing Services	200
Determining Test Coverage	201
Summary	203
8 Sustainable Architectures for Modern Angular	204
Vertical Slicing	204
Reasons for Vertical Slicing	204
Finding Boundaries	205
Event Storming	207
Different Models	209
Different Slicing in the Frontend	209
Structuring Verticals	211
The Architecture Matrix	211
Feature-Local Source Code	212
Implementation Options	213
Implementing a Modulith	214
Project Structure for a Modulith	214
Information Hiding	215
Enforcing your Architecture with Sheriff	216
Visualizing Dependencies with Detective	218
Lightweight Path Mappings	220
Lightweight Stores and Your Architecture	221
Unidirectional Data Flow	221
Where to Put a Lightweight Store?	222
Granularity of a Store	223
Communication Between Stores	224
Preventing Cycles, Redundancies, and Inconsistencies	225
Summary	225
9 State Management with NgRx Signal Store	226
A First SignalStore	226
Creating a Signal Store	226
Defining State	228
Providing Resources	230
Providing Computed Signals	231
Providing Methods	232

Setting up Hooks	233
Consuming the Store	233
Inspecting the Store with the Redux DevTools	235
Connecting to the Redux DevTools	235
Disabling DevTools in Production	235
Mutations	238
Creating Mutations	238
Consuming Mutations	241
Using rxMutation	243
Reactive Methods	244
rxMethod	244
signalMethod	249
Entity Management and Normalization	250
Entities	250
Entity Maps	253
Identifying Entities with IDs	254
Managing Several Entities in a Store	255
Normalization	256
Event API: Flux and Redux	259
Mental Model	260
Events	261
Reducer	262
Event Handlers	263
Dispatching Events	264
Custom Features	265
Defining Custom Features	265
Using Custom Features	267
Summary	269
10 Signal Queries & Component Communication	270
Content Projection	270
Referencing Parent Components	272
View and Content	274
Interacting with Content	275
Interacting with the View	278
Questioning the Use of viewChild and viewChildren	279
Static Child Components	279
Communication via Template Variables	280
Communication via Services	281
Summary	283
11 Directives, Templates, and Containers	285
Attribute Directives	285
Defining Directives	285
Communicating with the Environment	288
Directives and Template Variables	289

Controlled DOM-Manipulations	289
Code-based Content Projection	293
Templates and Containers	293
Passing Parameters to Templates	295
Structural Directives	298
Desugaring Structural Directives	299
Implementing a Simple DataTable	300
Using ViewContainerRef Directly to Display Templates	304
Accessing the ViewContainerRef via Signal Queries	306
Dynamic Components	306
Modal Dialogs	306
Instantiating Component via Code	313
Summary	315
12 Initialization & Route Changes	316
Initializers	316
Application Initializers	316
Environment Initializers	318
Platform Initializers	319
Guards	319
Preventing Route Activation	320
Preventing Route Deactivation	321
Router Events	324
Resolver	326
HttpInterceptors	328
Summary	330
13 Agentic UI & AI Assistants with Hashbrown	331
Implementing an Assistant with Tool Calling	331
Setting up Hashbrown	331
Using the chatResource	335
Providing Tools	337
Under the Hood	338
Generative UI and Component Control	340
UI chat with uiChatResource	341
Components for chat: Dumb Components with Smart Wrappers	344
Describing Components	345
Under the Hood: Structured Output	346
Supporting Different Models	347
Applying Few-Shot Prompting	347
Natural Language Queries with Code Generation	350
Approach	350
Implementation with Hashbrown	352
Runtime Functions	354
System Prompt with One-Shot Prompting	356
Summary	357

14 Monorepos & Reusable Libraries	358
Angular CLI-based Repos	358
Creating a Monorepo	359
Structure of Libraries	360
Trying Out the Library in the Monorepo	363
Building and Publishing the npm Package	364
Consuming the npm Package	366
Faster Builds and More Convenience with Nx	367
Creating an Nx Workspace	367
Module Boundaries	369
Nx with Sheriff and Detective	371
Incremental Builds with Nx	371
Distributed Cache with Nx Cloud	371
Even Faster: Parallelization with Nx Cloud	372
Summary	372
15 Internationalization	374
Overview	374
Installing @angular/localize	375
Marking Texts	375
Marking Strings in the Component Class	376
Extracting Texts	377
Integrating Translated Texts into Builds	379
Setting the Language for Development	380
Specifying Translation Texts at Runtime	382
Taking Grammatical Forms into Account	382
Supporting Different Formats	384
Outlook: Community Solutions	385
Summary	385
16 Modern Patterns for Authentication & Authorization	386
Cookie-based Authentication	386
Security-Attributes for Cookies	386
Cookies and XSRF	387
Token-based Security	388
OAuth 2	389
Authenticating Users with OpenID Connect	390
JSON Web Token	390
OAuth 2 and OIDC Flows	392
Client-side OAuth 2	393
Current Recommendation: Server-side OAuth 2	393
Summary	395
17 Defer, SSR & Hydration	396
Deferrable Views	396
Using Defer Blocks	396

Triggers	397
SSR & Hydration	398
Adding Server-Side Rendering	398
Incremental Hydration	399
Event Replay	401
Different Implementations for Server and Client	401
Different Implementations via DI	401
Checking the Platform at Runtime	403
Hybrid Routing	403
Prerendering Routes With Parameters	405
Working with the HTTP request and response	406
Summary	407
18 Micro Frontends: Scaling Across Multiple Teams	408
What are Micro Frontends?	408
Motivation Behind Micro Frontends	408
Challenges to Keep in Mind	409
Micro Frontends and Self-Contained Systems	410
Native Federation	411
Using Native Federation	414
Native Federation: Setting up a Micro Frontend	414
Native Federation: Setting up a Shell	415
Exposing a Router Config	418
Communication between Micro Frontends	420
Multi-Version and Multi-Framework Solutions	421
Abstracting Micro Frontends with Web Components	422
Loading Web Components in a Shell	423
Sharing Zone.js	426
Web Components with own Routes	427
Workaround for Routers in Web Component	427
The Cost of Micro Frontends	428
Summary	428
19 Analyzing Your Architecture with Forensic Techniques	429
The Example Application Examined	429
Analyzing Layering	430
Forensic Analysis for Architects: A Brief Overview	432
Using Detective	432
Change Coupling	433
Hotspots as an Indicator of Architectural Problems	434
Team Alignment and Conway’s Law	435
From Detective to Code Scene	437
Critical Review	437
Summary	438