

**Advanced Angular**

**Course Number:** ANG-172
**Duration:** 2 days

**Overview**

Take your skills to the next level with Accelebrate's Advanced Angular training. Students gain an understanding of application architecture and design best practices in Angular, as well as learn how to authenticate, unit test, and manage application state in an Angular application.

**Note:** This course is taught in the current version of Angular at the time of teaching.

**Prerequisites**

Students should have taken Accelebrate's [Introduction to Angular](file:////training/angular-introduction) class or have the equivalent experience.

**Materials**

All Angular training attendees receive comprehensive courseware.

**Software Needed on Each Student PC**

* Google Chrome
* Other modern browsers as desired
* IDE/development environment of your choice
* Other free software and lab files that Accelebrate would specify

**Objectives**

* Unit test all parts of an application including Components, Services, and Pipes
* Implement Authentication and Authorization in an Angular Application
* Optimize Angular Performance by changing Change Detection Strategies
* Setup new projects from scratch using the Angular CLI
* Scaffold modules, components, services, models, routes, and unit tests in accordance with best practices using the Angular CLI
* Write End-to-End Tests (optional; taught only if this applies to your group)
* Upgrade an existing application to the current Angular version (optional; taught only if this applies to your group)

**Outline**

* Introduction
* RxJS and Observables
	+ What is an Observable?
	+ Creating Observables
	+ What is an Observer?
	+ Observer Example
	+ Operators: map, switchMap, debounceTime, distinctUntilChanged
	+ Practical Application of using RxJS
	+ Subscriptions
	+ Unsubscribing from Observables in Angular (unsubscribe, Async Pipe, takeUntil)
	+ Subject
	+ Subject Example
	+ Subject Variants (AsyncSubject,BehaviorSubject, ReplaySubject)
	+ EventEmitter or Observable
	+ RxJS Operators and HTTP
* Unit Testing
	+ Tools: Jasmine, Karma
	+ Jasmine Syntax: describe, it, beforeEach, afterEach,matchers
	+ Setup and your First Test
	+ Testing Terminology: Mock, Stub, Spy, Fakes
	+ Angular Testing Terminology: TestBed, ComponentFixture, debugElement, async, fakeAsync, tick, inject
	+ Simple Component Test
	+ Detecting Component Changes
	+ Testing a Component with properties (inputs) and events (outputs)
	+ Testing a Component that uses the Router
	+ Testing a Component that depends on a Service
	+ Testing a Service and Mocking its HTTP requests
* Security
	+ Best Practices
	+ Preventing Cross-site Scripting (XSS)
	+ Trusting values with the DOMSanitizer
	+ HTTP Attacks (CSRF and CSSI)
	+ Authentication using JSON Web Tokens (JWT)
	+ Authorization: Router Guards
* Change Detection
	+ Understanding Zone.js and Change Detection
	+ Change Detection Strategies Default and OnPush
* Advanced Routing
	+ Lazy-loading Angular Modules (using Dynamic Imports)
	+ Nested or Child Routes
* Advanced Dependency Injection
	+ Providers
	+ Hierarchical Injection
	+ providedIn options: root, module, platform, any
* Pipes
	+ Creating a custom Pipe using PipeTransform
	+ Understanding Pure and Impure Pipes
* Conclusion
* Choose any two additional topics. If desired, the course can be customized to include more than two of these topics if other topics are scaled back or removed.
	+ npm QuickStart
		- Installing Dependencies
		- Understanding package.json and package-lock.json
		- Using npm as a Build Tool
	+ Managing Shared Application State using ngrx and Redux
		- Benefits Overview
		- Three Principles of Redux: Single Source of Truth, State is Read-Only, Pure Functions
		- Examples of Pure Functions
		- Reducers
		- Simple ngrx Example
		- Time-traveling with Redux Devtools
		- Full ngrx Example Application
	+ Upgrade Strategies from AngularJS
		- High-level Approaches
		- Concept Mapping AngularJS to Angular
		- UpgradeAdapter
		- What can be Upgraded or Downgraded
		- What cannot be Upgraded or Downgraded
		- UpgradeAdapter and Dependency Injection
	+ End-to-End Testing
		- What is Protractor?
		- Why Protractor?
		- Using Locators
		- Page Objects
		- Debugging E2E Tests