

**Angular Architecture and Best Practices**

**Course Number:** ANG-176
**Duration:** 4 days

**Overview**

This Angular Architecture and Best Practices training course teaches attendees how to build a robust and scalable Angular architecture that is easy to refactor and maintain. Students also learn component communication techniques, state management, code organization, general best practices, performance considerations, and more.

**Prerequisites**

All Angular training students must have existing knowledge of Angular and TypeScript.

**Materials**

All 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**

* Organize features and modules
* Properly structure components
* Facilitate component communication
* Manage state in Angular applications

**Outline**

* Introduction
* Planning the Application Architecture
	+ Architecture Considerations
	+ Architecture Planning Template
	+ The Angular Style Guide
* Organizing Features and Modules
	+ Organizing Features and Modules
	+ Core and Shared Modules
	+ Preventing Reimport of Core
	+ Reviewing Module Organization
	+ Custom Libraries
* Structuring Components
	+ Container and Presentation Components
	+ Passing State with Input and Output Properties
	+ Change Detection Strategies
	+ ngOnChanges: Reference vs. Value
	+ Cloning Techniques
	+ Component Inheritance
* Component Communication
	+ Component Communication Techniques
	+ RxJS Subjects
	+ Creating an Event Bus Service
	+ Creating an Observable Service
	+ Unsubscribing from Observables
* State Management
	+ The Need for State Management
	+ State Management Options
	+ Using Services
	+ Using NgRx
	+ Using ngrx-data
	+ Using Observable Store
	+ Reviewing State Management Options
* Additional Considerations
	+ Interfaces, Classes, and Enums
	+ Functions versus Pipes
	+ Adding a Memo Decorator
	+ HttpClient and RxJS Operators
	+ Using Interceptors
* Conclusion