

## Example applications

The following is a list of the example applications in the Angular documentation.

### Fundamentals

These examples demonstrate minimal, fundamental concepts.

#### Getting started application

Introductory application demonstrating Angular features. For more information, see [Getting started](#).

#### Launching your app

Demonstrates the Angular bootstrapping process. For more information, see [Launching your app with a root module](#).

#### Structure of Angular applications

Demonstrates the fundamental architecture of Angular applications. For more information, see [Introduction to Angular concepts](#).

#### Setting the document title

Demonstrates customizing the document title in the browser. For more information, see [Set the document title](#).

### Tour of Heroes tutorial application

The Tour of Heroes is a comprehensive tutorial that guides you through the process of building an application with many of Angular's most popular features.

#### Tour of Heroes: completed application

Completed Tour of Heroes example application. For more information, see [Tour of Heroes app and tutorial](#).

#### Tour of Heroes: Creating an application

Initial Tour of Heroes example application for beginning the tutorial. For more information, see [Create a new project](#).

#### Tour of Heroes: The hero editor

First step of the Tour of Heroes example application. For more information, see [The hero editor](#).

### **Tour of Heroes: Display a selection list**

Second step of the Tour of Heroes example application. For more information, see [Display a selection list](#).

### **Tour of Heroes: Create a feature component**

Third step of the Tour of Heroes example application. For more information, see [Create a feature component](#).

### **Tour of Heroes: Add services**

Fourth step of the Tour of Heroes example application. For more information, see [Add services](#).

### **Tour of Heroes: Add in-app navigation with routing**

Fifth step of the Tour of Heroes example application. For more information, see [Add in-app navigation with routing](#).

### **Tour of Heroes: Get data from a server**

Sixth and final step of the Tour of Heroes example application. For more information, see [Get data from a server](#).

## **Working with templates**

These examples demonstrate features of Angular templates.

### **Accessibility**

Demonstrates building Angular applications in a more accessible way. For more information, see [Accessibility](#).

### **Animations**

Demonstrates Angular's animation features. For more information, see [Introduction to Angular animations](#).

### **Attribute, class, and style bindings**

Demonstrates Angular attribute, class, and style bindings. For more information, see [Attribute, class, and style bindings](#).

### **Attribute directives**

Demonstrates Angular attribute directives. For more information, see [Attribute directives](#).

### **Binding syntax**

Demonstrates Angular's binding syntax. For more information, see [Binding syntax: an overview](#).

### **Built-in directives**

Demonstrates Angular built-in directives. For more information, see [Built-in directives](#).

### **Built-in template functions**

Demonstrates Angular built-in template functions. For more information, see the [\\$any\(\)](#) type cast function section of [Template expression operators](#).

### **Content projection**

Demonstrates how to use Angular's content projection feature when creating reusable components.

### **Interpolation**

Demonstrates Angular interpolation. For more information, see [Interpolation and template expressions](#).

### **Template expression operators**

Demonstrates expression operators in Angular templates. For more information, see [Template expression operators](#).

### **Template reference variables**

Demonstrates Angular's template reference variables. For more information, see [Template reference variables](#).

### **<ngcontainer>**

Demonstrates **<ngcontainer>**. For more information, see the [ng-container](#) section of [Built-in directives](#).

### **Pipes**

Demonstrates Angular pipes. For more information, see [Transforming Data Using Pipes](#).

### **Property binding**

Demonstrates property binding in Angular. For more information, see [Property binding](#).

### **Structural directives**

Demonstrates Angular structural directives. For more information, see Structural directives.

### **Two-way binding**

Demonstrates two-way data binding in Angular applications. For more information, see Two-way binding.

### **Template syntax**

Comprehensive demonstration of Angular's template syntax. For more information, see Template reference variables.

### **User input**

Demonstrates responding to user actions. For more information, see User input.

## **Working with components**

These examples demonstrate features of Angular components.

### **Component interaction**

Demonstrates how Angular shares data between components. For more information, see Component interaction.

### **Component styles**

Demonstrates styling in Angular applications. For more information, see Component styles.

### **Dynamic component loader**

Demonstrates how to dynamically load components. For more information, see Dynamic component loader.

### **Elements**

Demonstrates using Angular custom elements. For more information, see Angular elements overview.

### **Event binding**

Demonstrates binding to events in Angular. For more information, see Event binding.

## **@Input() and @Output()**

Demonstrates @Input() and @Output() in components and directives. For more information, see @Input() and @Output() properties.

## **Lifecycle hooks**

Demonstrates Angular lifecycle hooks such as ngOnInit() and ngOnChanges(). For more information, see Hooking into the component lifecycle.

# **Dependency injection**

## **Dependency injection fundamentals**

Demonstrates fundamentals of Angular dependency injection. For more information, see Dependency injection.

## **Dependency injection features**

Demonstrates many of the features of Angular dependency injection. For more information, see Dependency injection in action.

## **Providing dependencies in NgModules**

Demonstrates providing services in NgModules. For more information, see Providing dependencies in modules.

## **Hierarchical dependency injection**

Demonstrates Angular injector trees and resolution modifiers. For more information, see Hierarchical injectors.

## **Dependency injection with providers and viewProviders**

Demonstrates how providers and viewproviders affect dependency injection. For more information, see the Providing services in @Component() section of Hierarchical injectors.

## **Resolution modifiers and dependency injection**

Demonstrates Angular's resolution modifiers, such as @Self(). For more information, see the Modifying service visibility section of Hierarchical injectors.

# **Forms**

## **Forms overview**

Demonstrates foundational concepts of Angular forms. For more information, see Introduction to forms in Angular.

### **Reactive forms**

Demonstrates Angular’s reactive forms. For more information, see [Reactive forms](#).

### **Template-driven forms**

Demonstrates Angular template-driven forms. For more information, see [Building a template-driven form](#).

### **Form validation**

Demonstrates validating forms in Angular. For more information, see [Validating form input](#).

### **Dynamic forms**

Demonstrates creating dynamic forms. For more information, see [Building dynamic forms](#).

## **NgModules**

### **NgModules**

Demonstrates fundamentals of NgModules. For more information, see [NgModules](#).

### **Feature modules**

Demonstrates using feature modules in Angular. For more information, see [Feature modules](#).

### **Lazy loading NgModules**

Demonstrates lazy loading NgModules. For more information, see [Lazy-loading feature modules](#).

## **Routing**

### **Router**

Demonstrates Angular’s routing features. For more information, see [Router](#).

### **Router tutorial**

Demonstrates Angular’s fundamental routing techniques. For more information, see [Using Angular routes in a single-page application](#).

## **Documentation**

### **Style guide for Documentation contributions**

Demonstrates Angular documentation style guidelines. For more information, see Angular documentation style guide.

## **Server communication**

### **HttpClient**

Demonstrates server interaction using HTTP. For more information, see Communicating with backend services using HTTP.

## **Workflow**

### **Security**

Demonstrates security concepts in Angular applications. For more information, see Security.

### **Testing**

For the sample application that the testing guides describe, see the sample app.

Demonstrates techniques for testing Angular. For more information, see Testing.

## **Hybrid Angular applications**

### **AngularJS to Angular concepts: Quick reference**

Demonstrates Angular for those with an AngularJS background. For more information, see AngularJS to Angular concepts: Quick reference.