Certified Angular 4 Developer
Sample Material
VS-1430

1. ANGULAR CORE CONCEPTS

1.1. Angular Architecture and Building Blocks

Angular is a platform and framework for building client applications in HTML and TypeScript. Angular is itself written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your apps.

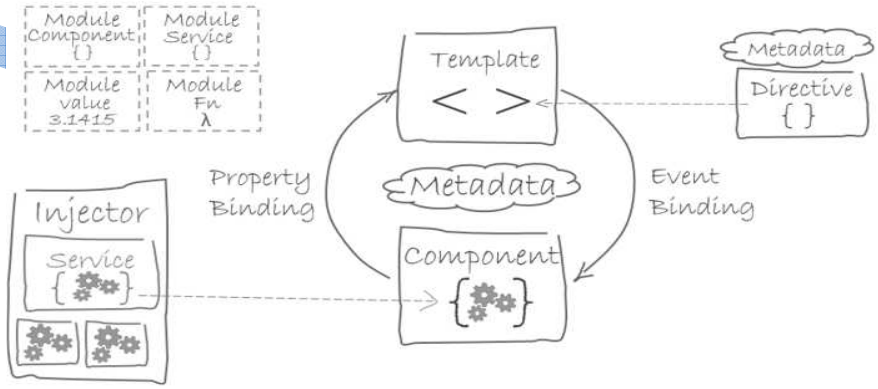
The basic building blocks of an Angular application are NgModules, which provide a compilation context for components. NgModules collect related code into functional sets; an Angular app is defined by a set of NgModules. An app always has at least a root module that enables bootstrapping, and typically has many more feature modules.

- ✓ Components define views, which are sets of screen elements that Angular can choose among and modify according to your program logic and data. Every app has at least a root component.
- ✓ Components use services, which provide specific functionality not directly related to views. Service providers can be injected into components as dependencies, making your code modular, reusable, and efficient.

Both components and services are simply classes, with decorators that mark their type and provide metadata that tells Angular how to use them.

- ✓ The metadata for a component class associates it with a template that defines a view. A template combines ordinary HTML with Angular directives and binding markup that allow Angular to modify the HTML before rendering it for display.
- ✓ The metadata for a service class provides the information Angular needs to make it available to components through Dependency Injection (DI).

An app's components typically define many views, arranged hierarchically. Angular provides the Router service to help you define navigation paths among views. The router provides sophisticated in-browser navigational capabilities.



1.2. Modules

Angular defines the NgModule, which differs from and complements the JavaScript (ES2015) module. An NgModule declares a compilation context for a set of components that is dedicated to an application domain, a workflow, or a closely related set of capabilities. An NgModule can associate its components with related code, such as services, to form functional units.

Every Angular app has a root module, conventionally named AppModule, which provides the bootstrap mechanism that launches the application. An app typically contains many functional modules.

Like JavaScript modules, NgModules can import functionality from other NgModules, and allow their own functionality to be exported and used by other NgModules. For example, to use the router service in your app, you import the Router NgModule.

Organizing your code into distinct functional modules helps in managing development of complex applications, and in designing for reusability. In addition, this technique lets you take advantage of lazy-loading—that is, loading modules on demand—in order to minimize the amount of code that needs to be loaded at startup.

1.3. Components

Every Angular application has at least one component, the root component that connects a component hierarchy with the page DOM. Each component defines a class that contains application data and logic, and is associated with an HTML template that defines a view to be displayed in a target environment.

The @Component decorator identifies the class immediately below it as a component, and provides the template and related component-specific metadata.

Decorators are functions that modify JavaScript classes. Angular defines a number of such decorators that attach specific kinds of metadata to classes, so that it knows what those classes mean and how they should work.

1.4. Templates, Directives, and Data Binding

A template combines HTML with Angular markup that can modify the HTML elements before they are displayed. Template directives provide program logic, and binding markup connects your application data and the document object model (DOM).

- ✓ Event binding lets your app respond to user input in the target environment by updating your application data.
- ✓ Property binding lets you interpolate values that are computed from your application data into the HTML.

Before a view is displayed, Angular evaluates the directives and resolves the binding syntax in the template to modify the HTML elements and the DOM, according to your program data and logic. Angular supports two-way data binding, meaning that changes in the DOM, such as user choices, can also be reflected back into your program data.

Your templates can also use pipes to improve the user experience by transforming values for display. Use pipes to display, for example, dates and currency values in a way appropriate to the user's locale. Angular provides predefined pipes for common transformations, and you can also define your own.

1.5. Services and Dependency Injection

For data or logic that is not associated with a specific view, and that you want to share across components, you create a service class. A service class definition is immediately preceded by the @Injectable decorator. The decorator provides the metadata that allows your service to be injected into client components as a dependency.

Dependency injection (or DI) lets you keep your component classes lean and efficient. They don't fetch data from the server, validate user input, or log directly to the console; they delegate such tasks to services.

1.6. Routing

The Angular Router NgModule provides a service that lets you define a navigation path among the different application states and view hierarchies in your app. It is modeled on the familiar browser navigation conventions:

- ✓ Enter a URL in the address bar and the browser navigates to a corresponding page.
- ✓ Click links on the page and the browser navigates to a new page.
- ✓ Click the browser's back and forward buttons and the browser navigates backward and forward through the history of pages you've seen.

The router maps URL-like paths to views instead of pages. When a user performs an action, such as clicking a link, that would load a new page in the browser, the router intercepts the browser's behavior, and shows or hides view hierarchies.

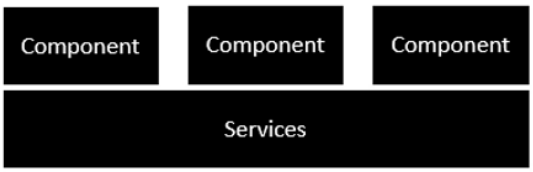
If the router determines that the current application state requires particular functionality, and the module that defines it has not been loaded, the router can lazy-load the module on demand.

The router interprets a link URL according to your app's view navigation rules and data state. You can navigate to new views when the user clicks a button, selects from a drop box, or in response to some other stimulus from any source. The Router logs activity in the browser's history journal, so the back and forward buttons work as well.

To define navigation rules, you associate navigation paths with your components. A path uses a URL-like syntax that integrates your program data, in much the same way that template syntax integrates your views with your program data. You can then apply program logic to choose which views to show or to hide, in response to user input and your own access rules.

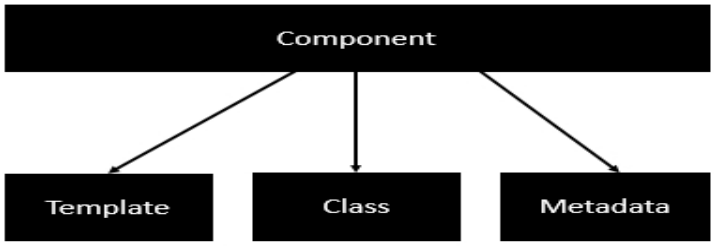
1.7. Angular Application Working

The below image shows the anatomy of an Angular 2 application. Each application consists of Components. Each component is a logical boundary of functionality for the application. You need to have layered services, which are used to share the functionality across components.



Following is the anatomy of a Component. A component consists of −

- ✓ Class − This is like a C++ or Java class which consists of properties and methods.
- ✓ Metadata − This is used to decorate the class and extend the functionality of the class.
- ✓ Template − This is used to define the HTML view which is displayed in the application.

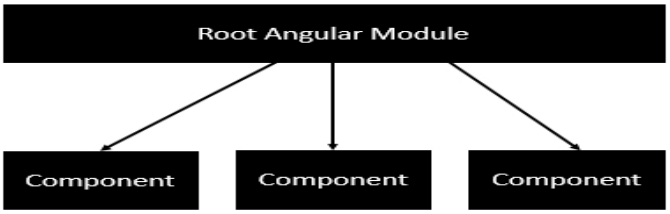


Following is an example of a component.

```
import { Component } from '@angular/core';
@Component ({
   selector: 'my-app',
   templateUrl: 'app/app.component.html'
})

export class AppComponent {
   appTitle: string = 'Welcome';
}
```

Each application is made up of modules. Each Angular 2 application needs to have one Angular Root Module. Each Angular Root module can then have multiple components to separate the functionality.



Following is an example of a root module.

```
import { NgModule }      from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
```

```
import { AppComponent }  from './app.component';

@NgModule ({
  imports:      [ BrowserModule ],
  declarations: [ AppComponent ],
  bootstrap:    [ AppComponent ]
})
export class AppModule { }
```

Each application is made up of feature modules where each module has a separate feature of the application. Each Angular feature module can then have multiple components to separate the functionality.

1.8. Angular Installation

The Angular CLI, Angular applications, and Angular itself depend upon features and functionality provided by libraries that are available as npm packages.

You can download and install these npm packages with the npm client, which runs as a node.js application.

The yarn client is a popular alternative for downloading and installing npm packages. The Angular CLI uses yarn by default to install npm packages when you create a new project.

package.json

Both npm and yarn install packages identified in a package.json file. The CLI ng new command creates a default package.json file for your project. This package.json specifies a starter set of packages that work well together and jointly support many common application scenarios. You will add packages to package.json as your application evolves. You may even remove some.

dependencies and devDependencies

The package.json includes two sets of packages, dependencies and devDependencies. The dependencies are essential to running the application. The devDependencies are only necessary to develop the application.

The dependencies section of package.json contains:

- ✓ Angular packages: Angular core and optional modules; their package names begin @angular/.
- ✓ Support packages: 3rd party libraries that must be present for Angular apps to run.
- ✓ Polyfill packages: Polyfills plug gaps in a browser's JavaScript implementation.

Angular Packages

- ✓ @angular/animations: Angular's animations library makes it easy to define and apply animation effects such as page and list transitions.
- ✓ @angular/common: The commonly needed services, pipes, and directives provided by the Angular team. The HttpClientModule is also here, in the '@angular/common/http' subfolder.

- ✓ @angular/core: Critical runtime parts of the framework needed by every application. Includes all metadata decorators, Component, Directive, dependency injection, and the component lifecycle hooks.
- ✓ @angular/compiler: Angular's Template Compiler. It understands templates and can convert them to code that makes the application run and render. Typically you don't interact with the compiler directly; rather, you use it indirectly via platform-browser-dynamic when JIT compiling in the browser.
- ✓ @angular/forms: support for both template-driven and reactive forms.
- ✓ @angular/http: Angular's old, soon-to-be-deprecated, HTTP client.
- ✓ @angular/platform-browser: Everything DOM and browser related, especially the pieces that help render into the DOM. This package also includes the bootstrapStatic() method for bootstrapping applications for production builds that pre-compile with AOT.
- ✓ @angular/platform-browser-dynamic: Includes Providers and methods to compile and run the app on the client using the JIT compiler.
- ✓ @angular/router: The router module navigates among your app pages when the browser URL changes.
- ✓ @angular/upgrade: Set of utilities for upgrading AngularJS applications to Angular.

Polyfill packages

Many browsers lack native support for some features in the latest HTML standards, features that Angular requires. "Polyfills" can emulate the missing features. The default package.json installs the core-js package which polyfills missing features for several popular browser.

Support packages

- ✓ rxjs: Many Angular APIs return observables. RxJS is an implementation of the proposed Observables specification currently before the TC39 committee that determines standards for the JavaScript language.
- ✓ zone.js: Angular relies on zone.js to run Angular's change detection processes when native JavaScript operations raise events. Zone.js is an implementation of a specification currently before the TC39 committee that determines standards for the JavaScript language.

DevDependencies

The packages listed in the devDependencies section of the package.json help you develop the application on your local machine. You don't deploy them with the production application although there is no harm in doing so.

- ✓ @angular/cli: The Angular CLI tools.
- ✓ @angular/compiler-cli: The Angular compiler, which is invoked by the Angular CLI's build and serve commands.
- ✓ @angular/language-service: The Angular language service analyzes component templates and provides type and error information that TypeScript-aware editors can use to improve the developer's experience. For example, see the Angular language service extension for VS Code
- ✓ @types/... : TypeScript definition files for 3rd party libraries such as Jasmine and node.
- ✓ codelyzer: A linter for Angular apps whose rules conform to the Angular style guide.
- ✓ jasmine/... : packages to support the Jasmine test library.
- ✓ karma/... : packages to support the karma test runner.

- ✓ protractor: an end-to-end (e2e) framework for Angular apps. Built on top of WebDriverJS.
- ✓ ts-node: TypeScript execution environment and REPL for node.
- ✓ tslint: a static analysis tool that checks TypeScript code for readability, maintainability, and functionality errors.
- ✓ typescript: the TypeScript language server, including the tsc TypeScript compiler.

NPM, Git and Visual Studio Code Installation

To start working with Angular 2, you need to get the following key components installed.

- ✓ Npm − This is known as the node package manager that is used to work with the open source repositories. Angular JS as a framework has dependencies on other components. And npm can be used to download these dependencies and attach them to your project.
- ✓ Git − This is the source code software that can be used to get the sample application from the github angular site.
- ✓ Editor − There are many editors that can be used for Angular JS development such as Visual Studio code and WebStorm.

npm Installation

Let's now look at the steps to get npm installed. The official site for npm is https://www.npmjs.com/

Step 1 − Go to the "get stated with npm" section in the site.
Step 2 − In the next screen, choose the installer to download, depending on the operating system. For the purpose of this exercise, download the Windows 64 bit version.
Step 3 − Launch the installer. In the initial screen, click the Next button.
Step 4 − In the next screen, Accept the license agreement and click the next button.
Step 5 − In the next screen, choose the destination folder for the installation and click the Next button.
Step 6 − Choose the components in the next screen and click the Next button. You can accept all the components for the default installation.
Step 7 − Click the Install button.
Step 8 − Once the installation is complete, click the Finish button.
Step 9 − To confirm the installation, in the command prompt you can issue the command npm version. You will get the version number of npm.

Installation of Visual Studio Code

Following are the features of Visual Studio Code −

- ✓ Light editor when compared to the actual version of Visual Studio.
- ✓ Can be used for coding languages such as Clojure, Java, Objective-C and many other languages.
- ✓ Built-in Git extension.
- ✓ Built-in IntelliSense feature.
- ✓ Many more extensions for development.

The official site for Visual Studio code is https://code.visualstudio.com/ and installation steps are

Step 1 − After the download is complete, please follow the installation steps. In the initial screen, click the Next button.
Step 2 − Accept the license agreement and click the Next button.
Step 3 − Choose the destination location for the installation and click the next button.
Step 4 − Choose the name of the program shortcut and click the Next button.
Step 5 − Accept the default settings and click the Next button.
Step 6 − Click the Install button.
Step 7 − In the final screen, click the Finish button to launch Visual Studio Code.

Certifications

Accounting, Banking & Finance
- Certified GST Professional
- Certified AML-KYC Compliance Officer
- Certified Business Accountant
- Certified BASEL III Professional
- Certified GAAP Accounting Standards Professional
- Certified Treasury Markets Professional

Big Data
- Certified Hadoop and Mapreduce Professional

Cloud Computing
- Certified Cloud Computing Professional

Design
- Certified Interior Designer

Digital Media
- Certified Social Media Marketing Professional
- Certified Inbound Marketing Professional
- Certified Digital Marketing Professional

Foreign Trade
- Certified Export Import (Foreign Trade) Professional

Health, Nutrition and Well Being
- Certified Fitness Instructor

Hospitality
- Certified Restaurant Team Member (Hospitality)

Human Resources
- Certified HR Compensation Manager
- Certified HR Staffing Manager
- Certified Human Resources Manager
- Certified Performance Appraisal Manager

Office Skills
- Certified Data Entry Operator
- Certified Office Administrator

Project Management
- Certified Master in Project Management
- Certified Scrum Specialist

Real Estate
- Certified Real Estate Consultant

Marketing
- Certified Marketing Manager

Quality
- Certified Six Sigma Green Belt Professional
- Certified Six Sigma Black Belt Professional
- Certified TQM Professional

Logistics & Supply Chain Management
- Certified International Logistics Professional
- Certified Logistics & SCM Professional
- Certified Supply Chain Management Professional

Legal
- Certified IPR & Legal Manager
- Certified Labour Law Analyst
- Certified Business Law Analyst
- Certified Corporate Law Analyst

Information Technology
- Certified Angular JS Professional
- Certified Basic Network Support Professional
- Certified Business Intelligence Professional
- Certified Core Java Developer
- Certified E-commerce Professional
- Certified IT Support Professional
- Certified PHP Professional
- Certified Selenium Professional

Mobile Application Development
- Certified Android Apps Developer
- Certified iPhone Apps Developer

Security
- Certified Ethical Hacking and Security Professional
- Certified Network Security Professional

Management
- Certified Corporate Governance Professional
- Certified Corporate Social Responsibility Professional
- Certified Leadership Skills Professional

Life Skills
- Certified Business Communication Specialist
- Certified Public Relations Officer

Media
- Certified Advertising Manager
- Certified Advertising Sales Professional

Sales, BPO
- Certified Sales Manager
- Certified Telesales Executive

& many more job related certifications

Contact us at:
V-Skills
011-473 44 723 or info@vskills.in
www.vskills.in