**What Is Angular?**

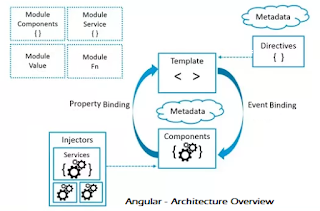
Angular is a most popular web development framework for developing mobile apps as well as desktop applications.

Angular framework is also utilized in the cross platform mobile development called IONIC and so it is not limited to web apps only.

Angular is an open source framework written and maintained by angular team at Google and the Father of Angular is [Misko Hevery](http://misko.hevery.com/about/" \t "_blank).

**Misko Hevery** - Agile Coach at Google, Attended Santa Clara University and Lives in Saratoga, CA.

Angular is written in TypeScript and so it comes with all the capabilities that typescript offers.

[](https://1.bp.blogspot.com/-dEK3X5tnvgA/WuLlDS2S_0I/AAAAAAAASZU/_xCvA4_tvEIHT3n3EujoVYphlHKx5QwXgCLcBGAs/s320/Angular-5-4-6-Architecture-Overview.png)

You don’t worry about the TypeScript versions. The compiler manages to the versioning related problems and Angular team working with Traceur compiler team to provide the support to build some extensions.

**What's New in Angular 8?**

Angular 8 being smaller, faster and easier to use and it will making Angular developers life easier. Angular version numbers have three parts**: major.minor.patch**. This release contains the following added and Improvements over the entire Angular platform including:-

**=> Added Support for *[TypeScript 3.2](https://www.code-sample.com/2017/06/angular-2-typescript-tutorial-for.html" \t "_blank)***

**=> Added a Navigation Type Available during Navigation in the Router**

**=> Added *pathParamsOrQueryParamsChange* mode for *runGuardsAndResolvers* in the Router**

**=> Allow passing state to *routerLink* Directives in the**[**Router**](https://www.code-sample.com/2018/05/angular-5-6-7-routing-and-navigation.html)

**=> Allow passing state to *NavigationExtras* in the Router**

**=> Restore whole object when navigating back to a page managed by Angular Router**

**=> Added support for *SASS***

**=> Resolve generated *Sass/Less* files to .css inputs**

**=> Added Predicate function mode for *runGuardsAndResolvers*:-**

*This option means guards and resolvers will ignore changes when a provided predicate function returns `false`. This supports use cases where an application needs to ignore some param updates but not others.*

*For example, changing a sort param in the URL might need to be ignored, whereas changing the `project` param might require re-run of guards and resolvers.*

**=> Added functionality to mark a control and its descendant controls as touched: -** add*markAllAsTouched () to AbstractControl*

**=> Added ng-new command that builds the project with Bazel**

**=> Use image based cache for windows BuildKite**

**=> Export *NumberValueAccessor* & *RangeValueAccessor* directives**

**=> Use shared *DomElementSchemaRegistry* instance for improve performance of platform-server*(@angular/platform-server*)**:-

*Right now the ServerRendererFactory2` creates a new instance of the `DomElementSchemaRegistry` for each and every request, which is quite costly (for the Tour of Heroes SSR example this takes around \*\*15%\*\* of the overall execution time)*

**=> Now the Performance Improvements on core, more consistent about “*typeof checks*”: -**

*When testing whether `value` is an object, use the ideal sequence of strictly not equal to `null` followed by `typeof value === 'object'` consistently. Specifically there's no point in using double equal with `null` since `undefined` is ruled out by the `typeof` check.*

*\*

*Also avoid the unnecessary ToBoolean check on `value.ngOnDestroy` in `hasOnDestroy()`, since the `typeof value.ngOnDestroy === 'function'` will only let closures pass and all closures are truish (with the notable exception of `document.all`, but that shouldn't be relevant for the `ngOnDestroy` hook)*

**=> In the Compiler-CLI, expose ngtsc as a TscPlugin**

**=> Restore whole object when navigating back to a page managed by**[**Angular Router**](https://www.code-sample.com/2018/05/angular-5-6-7-routing-and-navigation.html)**:-**

*This feature adds a few capabilities. First, when a `popstate` event fires the value of `history.state` will be read and passed into `NavigationStart`. In the past, only the `navigationId` would be passed here.*

*Additionally, `NavigationExtras` has a new public API called `state` which is any object that will be stored as a value in `history.state` on navigation.*

*For example, the object `{foo: 'bar'}` will be written to `history.state` here: -`router.navigateByUrl('/simple', {state: {foo: 'bar'}});`*

**What Are Components in Angular?**

Components are the most basic building block of a UI in Angular applications and it controls views (HTML/CSS). They also communicate with other components and services to bring functionality to your applications.

Technically components are basically TypeScript classes that interact with the HTML files of the components, which get displayed on the browsers.

The component is the core functionality of Angular applications but you need to know to pass the data into the components to configure them.

Angular applications must have a root component that contains all other components.

Components are created using @Component decorator that is part of @angular/core module.

You can create your own project using Angular CLI, this command allows you to quickly create an Angular application like - generate components, services, pipes, directive, classes, and modules, and so on as per your requirements.

Explore in detail about [Angular Components click…](https://www.code-sample.com/2018/04/what-are-components-in-angular-6-54-and.html)

**What Is Modules?**

The NgModule is a TypeScript class marked by the @NgModule decorator.

The NgModule is a class and work with the @NgModule decorator function and also takes a metadata object that tells Angular how to compile and run module code.

The Angular module helps you to organize an application into associative blocks of functionality.

An angular module represents a core concept and plays a fundamental role in structuring Angular applications.

The NgModule is used to simplify the ways you define and manage the dependencies in your applications and also you can consolidate different components and services into associative blocks of functionality.

Every Angular application should have at least one module and it contains the components, service providers, pipes and other code files whose scope is defined by the containing NgModule.

The purpose of the module is to declare everything you create in Angular and group them together.

Explore in detail about [Angular module click…](https://www.blogger.com/www.code-sample.com/2018/04/modules-ngmodule-angular-4-5-6.html)

**What Are Angular Directives?**

Angular Directive is a TypeScript class which is declared as a @directive decorator.

The directives allow you to attach behavior to DOM elements and the @directive decorator provide you an additional metadata that determines how directives should be processed, instantiated, and used at run-time.

Explore in detail about [Angular Directives click…](https://www.code-sample.com/2018/05/angular-5-6-7-directives-decorator.html)

**What Is Dependency Injection (DI)?**

Dependency Injection is a powerful pattern for managing code dependencies. DI is a way to create objects that depend upon other objects.

Angular has its own DI framework pattern, and you really can't build an Angular application without Dependency injection (DI).

A DI system supplies the dependent objects when it creates an instance of an object.

Explore in detail about [Angular Dependency Injection (DI) click…](https://www.code-sample.com/2018/05/dependency-injection-di-angular-5-6.html)

**What Is Angular Pipe?**

Pipes transform displayed values within a template.

Use the @Pipe annotation to declare that a given class is a pipe. A pipe class must also implement a PipeTransform interface.

The @Pipe decorator allows you to define the pipe name that is globally available for use in any template in the across Angular apps.

Pipe class implements the “PipeTransform” interfaces transform method that accepts an input value and returns the transformed result.

There will be one additional argument to the transform method for each parameter passed to the pipe.

Explore in detail about [Angular Pipes Decorator click…](https://www.code-sample.com/2018/05/angular-5-6-7-pipes-decorator.html)

**What Is runGuardsAndResolvers function?**

This option means **guards** and **resolvers** will ignore changes when a provided predicate function returns `**false**`. This supports use cases where an application needs to ignore some param updates but not others.

For example, changing a sort param in the URL might need to be ignored, whereas changing the `**project**` param might require re-run of guards and resolvers.

**What Is “typeof checks” in Angular 8?**

**How To Performance Improvements on core in Angular 8?**

When testing whether `**value**` is an object, use the ideal sequence of strictly not equal to `**null**` followed by `**typeof value === 'object'**` consistently. Specifically there's no point in using double equal with `**null**` since `**undefined**` is ruled out by the `**typeof**` check.

Also avoid the unnecessary ToBoolean check on `**value.ngOnDestroy**` in `**hasOnDestroy**()`, since the `**typeof value.ngOnDestroy === 'function'**` will only let closures pass and all closures are truish (with the notable exception of `**document.all**`, but that shouldn't be relevant for the `**ngOnDestroy**` hook)

**How to restore whole object when navigating back to a page managed by Angular Router in Angular 8?**

This feature adds a few capabilities.

First, when a `**popstate**` event fires the value of `**history.state**` will be read and passed into `**NavigationStart**`. In the past, only the `**navigationId**` would be passed here.

Additionally, `**NavigationExtras**` has a new public API called `**state**` which is any object that will be stored as a value in `**history.state**` on navigation.

For example, the object **`{name: 'anil'}**` will be written to `**history.state**` here: -`**router.navigateByUrl('/simple', {state: {name: 'anil'}});**`

**What Are the Navigation Type Available during Navigation in the Angular 8 Router?**

**What Is Bazel?**

Google open sourced the software responsible for building most of our projects under the name Bazel. Bazel is a powerful tool which can keep track of the dependencies between different packages and build targets.

**What Are the features of Bazel?**

Bazel is independent of the tech stack.

It has a smart algorithm for determining the build dependencies

**What Are the Angular 8 Best practices?**

Don't modify your copy of Angular

Avoid Angular APIs marked in the documentation as “Security Risk.”

Preventing cross-site scripting (XSS)

Keep current with the latest Angular library releases.

Check the Angular log for security-related updates in the regularly.

Remember, whether a value is safe depends on context, so choose the right context for your intended use of the value.

Normally, Angular automatically sanitizes the URL, disables the dangerous code, and in development mode, logs this action to the console.

**What Are the Angular Security Principles?**

Security Principles of Angular Applications:

1.          You should avoid direct use of the DOM APIs.

2.          You should enable Content Security Policy (CSP) and configure your web server to return appropriate CSP HTTP headers.

3.          You should Use the offline template compiler.

4.          You should Use Server Side XSS protection.

5.          You should Use DOM Sanitizer.

6.          You should Preventing CSRF or XSRF attacks.

**What Is Cross Site Scripting (XSS) Attack?**

The Cross Site Scripting (XSS) attack is a type of injection and attackers inject your web applications using the client side scripts and malicious code into web pages.

An attacker can insert vulnerability scripts and malicious code in your web applications.

The Cross Site Scripting (XSS) attacks are common on web browsers and it carried out on websites around 84% (approximately).

**How To Preventing Cross Site Scripting (XSS) in Angular?**

**How Angular Protects Us From XSS Attacks?**

The Cross Site Scripting (XSS) attack is a type of injection and attackers inject your web applications using the client side scripts and malicious code into web pages.

An attacker can insert vulnerability scripts and malicious code in your web applications.

The Angular treats all values as untrusted by default. This is the great advantages of Angular.

When a value is Inserted Vulnerability into the DOM from –

1.          A Template

2.          Property

3.          Attribute

4.          Style

5.          Class Binding

6.          Interpolation

Angular recognizes the value as unsafe and automatically sanitizes and removes the script tag and other security vulnerabilities.

For more detail [about Angular security explore in detail….](https://www.code-sample.com/2017/11/angular-security-xss-csrf.html)

**Stayed Informed** – [*Angular Book - All in One (Including Version 2, 4, 5, 6 and 7)*](https://www.code-sample.com/2018/11/angular-7-6-5-4-interview-questions-book.html)

**Stayed Informed** – [*Angular Interview Question (Including Version 2, 4, 5, 6 and 7)*](https://www.code-sample.com/2018/05/angular-2-4-5-6-7-8-interview-questions.html)

**Angular 8 beta** will be release on March/April 2019.

The following table contains our current target release dates for the next two major versions of Angular:

| **Date** | **Stable Release** | **Compatibility** |
| --- | --- | --- |
| March/April 2019 | 8.0.0 | ^7.0.0 |
| September/October 2019 | 9.0.0 | ^8.0.0 |

Are you curious to know about the [**Recently Stable Released Angular 7.2.3**](https://www.code-sample.com/2018/04/what-new-in-angular-7-features.html), click to explore in details?

Are you interested to know about, [**What’s New in Angular 7.2.3?**](https://www.code-sample.com/2018/04/what-new-in-angular-7-features.html)

You can [**download Angular 7.2.3 stableproduction release**](https://github.com/angular/angular/releases/tag/7.0.0) form GitHub.

For more detail on Release schedule, go to [**https://angular.io/guide/releases**](https://angular.io/guide/releases)