Angular 2

# Basic Questions:

<http://www.code-sample.com/2016/06/angular-2-interview-questions-and.html>

<https://elhegazy.wordpress.com/2016/06/07/angular2-five-minutes-tutorialquestions-and-answers/>

What is angular?

***Angular:****is a JavaScript framework for building client-side applications, using HTML, CSS, and a programming language.*

The architecture diagram identifies the eight main building blocks of an Angular application:

1. Modules:
2. Components
3. Templates
4. Metadata
5. Data binding
6. Directives
7. Services
8. Dependency injection

# sModules

In angular system modules are called NgModules.

NgModule is a decorator function that takes a single metadata object whose properties describe the module. The most important properties are:

* declarations - the view classes that belong to this module. Angular has three kinds of view classes: **components**, **directives**, and **pipes**.
* exports - the subset of declarations that should be visible and usable in the component **templates** of other modules.
* imports - other modules whose exported classes are needed by component templates declared in this module.
* providers - creators of **services** that this module contributes to the global collection of services; they become accessible in all parts of the app.
* bootstrap - the main application view, called the root component, that hosts all other app views. Only the root module should set this bootstrap property.

What is Npm?

Npm, or node package manager: is a command line utility that interacts with a repository of open source projects, Become the package manager for JavaScript. Using npm we can install libraries, packages, and applications, along with their dependencies.

<https://angular.io/tutorial/toh-pt4>

The @Injectable() decorator tells TypeScript to emit metadata about the service. The metadata specifies that Angular may need to inject other dependencies into this service.

<https://angular.io/tutorial/toh-pt3>

In general, the declarations array contains a list of application components, pipes, and directives that belong to the module. A component must be declared in a module before other components can reference it.

What is the difference between annotation and decorator?

**The core differences and many more advantages on Angular2 vs. Angular 1 as following,**

1.      Angular2 has better performance.

2.      Angular2 has more powerful template system.

3.      Angular2 provide simpler APIs, lazy loading and easier to application debugging.

4.      Angular2 much more testable

5.      Angular2 provides to nested level components.

6.      Angular2 executes run more than two programs at the same time.

7.      Angular1 is controllers and $scope based but Angular2 is component based.

8.      The Angular2 structural directives syntax is changed like ng-repeat is replaced with \***ngFor** etc.

9.      In Angular2, local variables are defined using prefix (**#**) hash. You can see the below \*ngFor loop Example.

### ****3-We already use Angular 1, why do we need an Angular 2?****

* ***Angular 2****is*built for speed*(lets ask our self how?)*

*– It has faster initial loads.*

*– faster change detection.*

*–improved rendering times.*

*–Angular 2 is modern.*

*–It takes advantage of features provided in the latest JavaScript standards and beyond(Such as*classes, modules, and decorators*)*

*–It leverages web component technologies for building reusable user interface widgets.*

*–It supports both Greenfield and Legacy browsers, Edge, Chrome, Firefox and Internet Explorer back to IE9.*

*–It has fewer built-in directives to learn simpler binding.*

*–Enhances our productivity to improve our day-to-day workflow.*

# Q. Constructor VS ngOnInit

1. Construction is related to the class instantiation and has nothing to do with angularJs 2. On the other word a constructor can be used on any class and allow us put tp it for some initializing processing for the newly created instance.

**OnInit** is component’s life cycle hook which runs first after constructor when component is being initialized. OnInit allows us initialization processing relies on binding of the component.

2. Constructor is predefined default method of the typescript class there is nothing relation between angular2 and constructor. normally we use constructor to define/initialize some variables. but when we have some task related to angular's binding we move to angular's hook (life cycle hook) i.e ngOnInit. ngOnInit called just after the constructor call. we can also do same work in the constructor but its prefferable to use ngOnInit to start angular's binding.

<https://stackoverflow.com/questions/35845554/angular-2-component-constructor-vs-oninit>

<http://www.code-sample.com/2017/02/angular-2-constructor-vs-oninit.html>

# Components Questions:

What are the Components in angular 2?

The components are the main way to build or specify HTML elements and business logic on the page.

A component is a building block of Angular 2 application.Angular 2 application is created as a tree of components.A component is declared by using @Component() decorator function.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | @Component({  selector: 'first-component',  template: `<p>Hello from first component</p>`  })  export class FirstComponent {  } |

When we declare Component we define metadata for component.In this example we have defined selector and template metadata for the FirstComponent component.

**What are the difference between components and directives?**

|  |  |
| --- | --- |
| [**@Components**](http://www.code-sample.com/2016/04/angular-2-components-vs-directives.html) | [**@Directive**](http://www.code-sample.com/2016/04/angular-2-components-vs-directives.html) |
| 1.       @Component meta-data annotation is used  to register the components. | @Directive meta-data annotation is used  to register the directives. |
| 2.       The components are used to create UI widgets. | The directives are used to add behavior to existing DOM elements. |
| 3.       The components are used to split to application into smaller parts. | The directives are use to design a reusable components. |
| 4.       Only one component is used per DOM element. | More than one directive are used per DOM element. |
| 5.       In the components, @View, template and templateUrl are mandatory in the components. | The directive do not have @View etc. |

What are Components Life-Cycles?

**What is @Inputs in Angular 2?**

@Input allows you to pass data into your controller and templates through html and defining custom properties. This allows you to easily reuse components and have them display different values for each instance of the renderer.

<https://egghead.io/lessons/angular-2-pass-values-into-angular-2-components-with-input>

**What is Outputs in Angular 2?**

**@Output** decorator is used to binds a property of a component to send the data from child component to parent component and this is a one-way communication.

**@Output** decorates output properties and its binds a property of the type of angular EventEmitter.

<http://www.code-sample.com/2016/06/angular-2-outputs.html>

What is hidden property in Angular 2?

**How do components communicate with each other?**

By creating service component communicate with each other

**How would you create a component to display error messages throughout your application?**

**In Angular 2**, the ngModel provides error objects for each of the built-in input validators. You can access these errors from a reference to the ngModel itself then build useful messaging around them to display to your users.

And also, we can use the properties “**pristine**” and “**touched**” to display error messages.

1. If we want to display errors after the user fills something in a field, use the pristine property.
2. If we want to display errors after the user put the focus on a field, use the touched property.

<div \*ngIf="(!loginForm.controls.email.valid && !loginForm.controls.email.pristine)">

\*\*Email is required.

</div>

<http://www.code-sample.com/2017/05/angular-2-error-message-on-form-submit.html>

# Services Questions:

What is an Angular 2 Service?

Service is a class that encapsulates some methods (GET/POST/PUT) and provides it result as a service for across your application.

What are the features of Angular 2 Service?

The Angular 2 is using services concept and it provide the multiple features to us that are,

1. Services are singleton objects.
2. Services are capable of returning the data in the form promises or observables.
3. Service class is decorated with Injectable decorator.
4. The Injectable decorator is required only if our service class is making use of some Angular injectable like Http, Response and HttpModule service within it.

What are the differences between Observables & Promises?

1. **Promise**: - Promises are only called once and it can return only a single value at a time and the Promises are not cancellable.
2. **Observables**: - Observables handle multiple values over time and it can return multiple values and the Observables are cancellable.
3. The Observables are more advanced than Promises.

<http://www.code-sample.com/2017/05/angular-2-observables-vs-promises.html>

How To Create & Call Angular 2 Services in Components?

**How do we create a singleton service in Angular 2?**

<http://www.code-sample.com/2017/05/create-angular-2-services-example.html>

How HTTP Client is Interact with Angular 2 Servers?

What the best way to inject one service into another in angular 2?

# Pipe Questions:

# Component Questions:

**What is Directives in Angular 2?**

 There are 3 types of directives in Angular 2.

1.     **Components** **Directives** - directives with a template

2.     **Structural Directives** - change the DOM layout by adding and removing DOM elements.

3.     **Attribute Directives** - change the appearance or behavior of an element, component, or other directive.

**What are components directives?**

A component is a directive with a template and the @Component decorator is actually a @Directive decorator extended with template oriented features.

1.     To register a component, we use @Component meta-data annotation.

2.     The directives are used to add behavior to existing DOM elements.

3.     The directives are used to design a reusable component.

4.     Only one component can be present per DOM element.

5.     Multiple directives are used per DOM element.

6.     The directive does not have @View etc.

**What are structural directives?**

The “Structural directives” are responsible for HTML layout. They shape or reshape the DOM structure; it is using for adding, removing and manipulating the elements.

The “Structural directives” is used to enable an element as a template for creating additional elements. If you want to create structural directive that time you should have knowledge of **<template>** elements and structural directives are easy to recognize.

The two familiar examples of structural directive as,

1. \*ngIf
2. \*ngfor

**What are attribute directives?**

Attribute directives are used to change the behavior, appearance or look of an element on a user input or via data from the service.

What is router-outlet directive in Angular 2?

**Router-outlet directive: -**Router-outlet directive is used to render the components for specific location of your applications. Both the template and **templateUrl** render the components where you use this directive.

# Styling Questions:

# Routing Questions:

**What is routing in angular 2?**

“The Router is use to map applications URLs to application components. There are three main components that you are using to configure routing.”

1.           **Routes**: - It uses to describe our application's Routes.

2.           **Router Imports: -** It uses to import our application's Routes.

3.           **RouterOutlet**: - It is a placeholder component and use to get expanded to each route's content.

4.           **RouterLink**: - It is use to link to application's routes.

**Routes: -** The Routes is uses to describe our application's Routes. The “RouterModule.forRoot” method in the module imports to configure the router.

Five concepts that need Routes Representation

1.           Path (a part of the URL)

2.           Route Parameters

3.           Query/Matrix Parameters

4.           Name outlets

5.           A tree of route segments targeting outlets

# Dependency Injection Questions

What is Dependency Injection (DI) in Angular 2?

<http://www.code-sample.com/2016/04/dependency-injection-in-angular-2.html>

**What are the difference between @Inject and @Injectable?**

@Inject() is a special technique for letting Angular know that a parameter must be injected.

**@Injectable()** marks a class as available to an injector for instantiation. An injector reports an error when trying to instantiate a class that is not marked as **@Injectable()**.

**How to use Dependency Injection (DI) correctly in Angular 2?**

The basics Steps of Dependency injection,

1.      A class with @Injectable() to tell angular 2 that it’s to be injected “UserService”.

2.      A class with a constructor that accepts a type to be injected.

# Ng-Modules Questions:

What is the purpose of NgModule?

How do you decide to create a new NgModule?

What would you have in a shared module?

What is the purpose of exports in a NgModule?

Angular 2 Modules vs. JavaScript Modules

What is the difference between a module's forRoot() and forChild() methods and why do you need it?

# Traceur Compiler Questions:

What is Traceur compiler? Why in Angular 2?

The [Traceur](http://www.code-sample.com/2015/03/what-is-traceur-compiler.html) is a JavaScript compiler. The Traceur compiler used to allow us to use the features from the future. The Traceur compiler is fully supported to [ECMAScript](http://www.code-sample.com/2015/03/what-is-ecmascript-es6.html)(ES6) and ES.vNext also.

# <http://www.codecompiled.com/angular-2-interview-questions-answers/>

#### What is Angular 2

Angular 2 is a JavaScript framework for developing web applications.  
It is a complete rewrite of Angular 1.0 ,so learning AngularJS 1.0 is not required for creating Angular 2 applications.  
It is well suited for developing mobile applications unlike AngularJS 1.0 which was suitable for building desktop applications.It is also much faster than Angular 1.0.

It supports the modern browsers as well as the older browsers.Applications are better structured than previous versions of Angular.It supports server side rendering for faster rendering of views even on slow devices such as mobile.The size of Angular 2 library is smaller compared to previous versions.Also Angular 2 applications use ahead of time compilation which makes them faster.

#### Which languages are used to write Angular 2 applications

Angular 2 applications can be written in any of the following languages:

* Typescript   Prefered Language for developing Angular 2 applications.
* Javascript
* Dart

We don’t have to worry about the JavaScript or ECMAScript version as its the compiler’s responsibility to manage the version issues.

As Angular 2 is written in TypeScript so it is preferable to write Angular 2 applications in TypeScript or ECMA6. [*Typescript*](http://www.codecompiled.com/overview-of-typescript/)is the prefered language to use for developing Angular 2 applications.

#### Components

A component is a building block of Angular 2 application.Angular 2 application is created as a tree of components.A component is declared by using @Component() decorator function.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | @Component({  selector: 'first-component',  template: `<p>Hello from first component</p>`  })  export class FirstComponent {  } |

When we declare Component we define metadata for component.In this example we have defined selector and template metadata for the FirstComponent component.

#### Modules

Angular apps consists of different modules.Modules consists of collection of components,directives and services.

Angular modules are created using the NgModules() decorator function.  
Every Angular application consists of a root module apart from other feature modules.Every modules is created using NgModule decorator function.

#### Template

View of a component is declared by using the template.It is the template which is rendered.

We can define template in line in the component template metadata property:

|  |  |
| --- | --- |
| 1  2  3  4 | @Component({  selector: 'hello-component',  template: '{{Hello}}'  }) |

We can also define templates in separate html file and use the templateUrl property in component:

|  |  |
| --- | --- |
| 1  2  3  4 | @Component({  selector: 'hello-component',  templateUrl: './hello.html'  }) |

#### Data bindings which are supported in Angular 2

Interpolation   In interpolation binding we specify the binding using expressions

|  |  |
| --- | --- |
| 1 | <h2>{{employee.name}}</h2> |

Property Binding  In property binding we bind the custom property using square brackets

|  |  |
| --- | --- |
| 1 | <h2 [innerText]="employee.name"></h2> |

Event Binding   In event binding we enclose the event name in parenthesis and assign the event handler method to the event:

|  |  |
| --- | --- |
| 1 | <button (click)="gotoDetail()">View Details</button> |

Two-way Binding  In Angular 2 we define two way binding as:

|  |  |
| --- | --- |
| 1 | <input [(ngModel)]="employee.name"/> |

#### How Angular 2 application is launched

Angular 2 application is launched by bootstrapping the root module.

#### What is router-outlet

The route which is matched by the router is used display the component.The template defined by the component is displayed in an area defined by the router-outlet

#### NgModule

Its a decorator function which has one argument ,a metadata object with properties describing the module.  
Some of its important properties are:

* declarations views which belong to this module.
* exports declarations which are visible in the components of other modules.
* imports other modules whose classes are needed in this module.
* providers services which are provided by this module
* bootstrap This property is set by the root module

|  |  |
| --- | --- |
| 1  2  3  4  5 | @NgModule({  providers: list of providers,  exports:list of components,  imports:list of components  }) |

### What are the advantages of Angular 2 over Angular 1

Better performance because of these reasons

* Better change detection.
* Ahead of Time compilation (AOT) improves rendering speed.
* Lazy Loading.
* TypeScript can be used for developing Angular 2 applications.
* Better syntax and application structure.

### What is lazy loading in Angular2

Angular 2 application is a collection of modules and components.There are two ways we can load Modules:

* Eager Module loading  Loading module at application startup
* Lazy loading  Loading Module only when required

Module which is required can be loaded instead of loading all the modules at application initialization.This has the obvious advantage of improving the application startup time.

We enable lazy loading in Angular 2 by using the loadChildren property in route

|  |  |
| --- | --- |
| 1 | { path: 'URL', loadChildren: 'modulePath#ClassName' } |

### AOT compilation

AOT compilation stands for  Ahead Of Time compilation, in it angular compiles  components to native JavaScript and HTML during the build time instead of runtime.This drastically improves the performance of the Angular 2 application.With Just in time compilation ,the compilation happens on the users browser.In the case of Ahead of time compilation ,the compilation process happens at the build time.So this improves the rendering of the application UI.

TypeScript

What is TypeScript?

TypeScript is a typed superset of JavaScript that compiles to plain JavaScript.

What is operator?

An operator defines some function that will be performed on the data.

Object Oriented:

**Method:** Methods are functions that run in context of an object. To call a method on an object, we first have to have an instance of that object.

**Constructor:**

A *constructor* is a special method that is executed when a new instance of the class is being created.Usually, the constructor is where you perform any initial setup for new objects.Constructor methods must be named constructor. They can optionally take parameters but they can’t return any values, since they are called when the class is being instantiated.

**Inheritance** is a way to indicate that a class receives behavior from a parent class. Then we can override, modify or augment those behaviors on the new class.

Book

## Agular CLI:

<https://github.com/angular/angular-cli>

To create a new project $ ng new angular-hello-world

**Run application:**

$ ng serve

\*\* NG Live Development Server is running on http://localhost:4200. \*\*

Port 4200 is already in use. Use '--port' to specify a different port

$ ng serve --port 9001

http://localhost:9001

To generate hello-world component

$ ng generate component hello-world

A basic Component has two parts:

1. A Component annotation

2. A component definition class

to pass data into the child component. (page 25)

Angular provides a way to do this: the @Input annotation