<https://github.com/WebPredict/angular-2-interview-questions>

<https://github.com/khan4019/angular-interview-questions>

<https://github.com/Yonet/Angular-Interview-Questions>

<https://github.com/sudheerj/angular-interview-questions>

<https://vsavkin.com/the-core-concepts-of-angular-2-c3d6cbe04d04> -> core concepts

**What is SPA ?**

in spa, it receives data in json; whereas in MPA it receives whole html file itself from server

<https://www.youtube.com/watch?v=F_BYg2QGsC0>

<https://www.quora.com/What-is-SPA-single-page-application>

<https://flaviocopes.com/single-page-application/>

<https://dzone.com/articles/what-is-a-single-page-application>

Difference between Angular 1.x and the higher version (2, 4 etc.) ?

Angular 2/4 is complete revamp of Angular 1. In fact, we can say that it was completely rewritten from the ground-up.

Angular 1 is controller and $scope based while Angular 2/4 is based on an architecture of component hierarchy, dependency injections and directives.

Angular 2/4 is much more geared towards Mobile development unlike Angular 1 as mobile development is much bug prone. If mobile development is handled first desktop development will not have much issues.

Angular 2/4 focuses much more on modularity as much of its core functionality has been transferred to its modules leading to a faster and lighter application in development.

Angular 2/4 is mainly written in Typescript Which introduces features like Class-based Object Oriented Programming, Generics and Static Typing to framework, which makes it more appealing to developers from non-JavaScript background and leading to shorter development time.

Unlike Angular 1, Angular 2/4 is mainly introduced to be compatible with modern browsers. Angular 2/4 not only introduces new features like lazy loading/Dynamic loading, Asynchronous template compilation, Simpler Routing and Reactive programming support-using RxJS but also improves on legacy features like dependency injection.

Difference between Angular 2 and 4 ?

The upgrade of the version from 2.0 to 4.0 has reduced it’s bundled file size by 60%. The code generated is reduced and has accelerated the application development. Here the developed code can be used for prod mode and debug.

Angular v4.0 is compatible with newer versions TypeScript 2.1 and TypeScript 2.2. This helps with better type checking and also enhanced IDE features for Visual Studio Code.

The view engine is introduced in Angular 4 where the produced code of components can be reduced up to 60%. The bundles are reduced to thousands of KBs.

Why skipped Angular 3?



The angular team already updated **@angular/router** with version 3.X before Angular 3 would have been released that's because of major development on router packages, like router-preload. So to avoid such confusion between Angular package with angular version. They decided to skip the Angular version 3 and directly released angular 4.

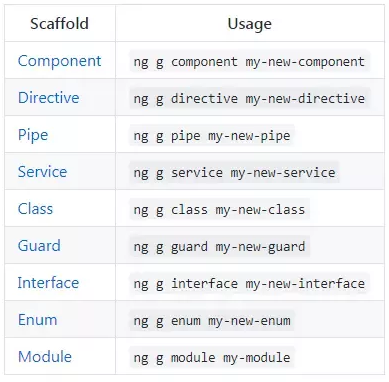
What is Angular cli?

Angular CLI stands for Angular Command Line Interface. As the name implies, it is a command line tool for creating angular apps.

It is recommended to use angular cli for creating angular apps as you don't need to spend time installing and configuring all the required dependencies and wiring everything together.

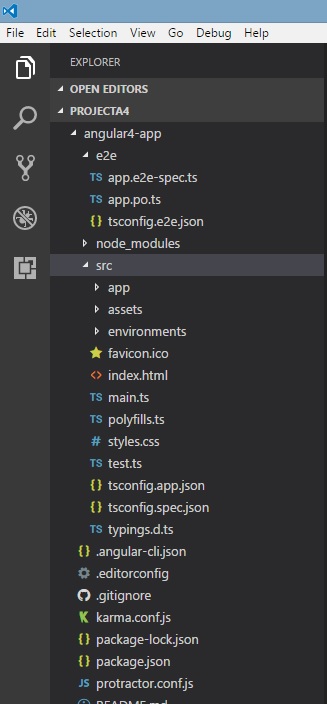
**Creating New Project: *ng new****<project-name> (for.eg., ng new Angular4-app)*

**Running the Project: *ng serve***



Explain Angular 4 Project Structure?

<https://angular.io/guide/file-structure>



The Angular 4 app folder has the following **folder structure** −

* **e2e** − end to end test folder. Mainly e2e is used for integration testing and helps ensure the application works fine.
* **node\_modules** − The npm package installed is node\_modules. You can open the folder and see the packages available.
* **src** − This folder is where we will work on the project using Angular 4.

The Angular 4 app folder has the following **file structure** −

* **.angular-cli.json** − It basically holds the project name, version of cli, etc.
* **.editorconfig** − This is the config file for the editor.
* **.gitignore** − A .gitignore file should be committed into the repository, in order to share the ignore rules with any other users that clone the repository.
* **karma.conf.js** − This is used for unit testing via the protractor. All the information required for the project is provided in karma.conf.js file.
* **package.json** − The package.json file tells which libraries will be installed into node\_modules when you run npm install.
* **protractor.conf.js** − This is the testing configuration required for the application.
* **tsconfig.json** − This basically contains the compiler options required during compilation.
* **tslint.json** − This is the config file with rules to be considered while compiling.

### **Assets -** You can save your images, js files in this folder.

### **Environment**

This folder has the details for the production or the dev environment. The folder contains two files.

* environment.prod.ts
* environment.ts

Both the files have details of whether the final file should be compiled in the production environment or the dev environment.

The additional file structure of Angular 4 app folder includes the following −

### **favicon.icon -** This is a file that is usually found in the root directory of a website.

### **index.html -** This is the file which is displayed in the browser.

### **main.ts** - main.ts is the file from where we start our project development. It starts with importing the basic module which we need.

platformBrowserDynamic().bootstrapModule(AppModule);

### **polyfill.ts -** This is mainly used for backward compatibility.

### **styles.css -** This is the style file required for the project.

### **test.ts -** Here, the unit test cases for testing the project will be handled.

### **tsconfig.app.json -** This is used during compilation, it has the config details that need to be used to run the application.

### **tsconfig.spec.json -** This helps maintain the details for testing.

### **typings.d.ts -** It is used to manage the TypeScript definition.

What are building blocks of Angular 4?

Modules, Component, Template, Directives, Data Binding, Services, Dependency Injection, Routing.

**Angular architecture ?**

<https://www.code-sample.com/2018/01/angular-4-and-5-architecture-overview.html>

<https://angular.io/guide/architecture>

**What is component in Angular 4?**

<https://angular.io/guide/architecture-components>

Components are like the basic building block in an Angular application. Major part of the development with Angular 4 is done in the components. Components are basically classes that interact with the .html file of the component, which gets displayed on the browser.

Components are defined using the @component decorator. A component has a selector, template, style and other properties, using which it specifies the metadata required to process the component.

The file structure has the app component and it consists of the following files

* app.component.css
* app.component.html
* app.component.spec.ts
* app.component.ts

**Directive VS Component ?**

<https://stackoverflow.com/questions/32680244/directive-v-s-component-in-angular>

|  |  |
| --- | --- |
| **Components** | **Directives** |
| A @Component requires a view. | @Directive does not require a view. |
| The components are used to split the application into smaller parts. | The directives are used to design reusable components. |
| A component, rather than adding/modifying behavior, actually creates its own view. | Directives add behavior to an existing DOM element or an existing component instance. |
| Only one component is used per DOM elements. | More than one directive can be used per DOM elements. |
| @Component meta-data annotation is used to register the components. | @Directive is used to register the directive. |

What are Modules?

<https://angular-2-training-book.rangle.io/modules/introduction>

In Angular, a module is a mechanism to group components, directives, pipes and services that are related to an application.

What is NgModule?

**The purpose of a NgModule is to declare each thing you create in Angular, and group them together.**

To be able to define modules we have to use the decorator NgModule.

import {NgModule } from '@angular/core';

@NgModule ({

imports: [ ... ],

declarations: [ ... ],

bootstrap: [ ... ]

})

export class AppModule { }

* **Declarations -** It is an array of components created. If any new component gets created, it will be imported first and the reference will be included in declarations.
* **Imports -** It is an array of modules required to be used in the application
* **Providers is for services** (~ models: the classes getting and handling data).
* **Bootstrap -** This includes the main app component for starting the execution.

**How Bootstrapping in angular works ?**

<https://angular.io/guide/bootstrapping>

<https://stackoverflow.com/questions/44821879/can-angular-have-more-than-one-bootstrap-component>

<https://blog.angularindepth.com/how-to-manually-bootstrap-an-angular-application-9a36ccf86429>

What are Directives and its types?

<https://www.c-sharpcorner.com/article/sort-of-directives-in-angular-6-and-how-to-use-them/>

ngStyle and ngClass

<https://codecraft.tv/courses/angular/built-in-directives/ngstyle-and-ngclass/>

[style] vs [ngStyle]

<https://blog.angulartraining.com/whats-the-difference-between-style-and-ngstyle-in-angular-68a3301c2ae6>

when to use ngStyle vs ngClass

<https://stackoverflow.com/questions/26919963/what-is-the-difference-between-ng-class-and-ng-style>

How does communication happen between components?

<https://angularfirebase.com/lessons/sharing-data-between-angular-components-four-methods/>

<https://www.youtube.com/watch?v=eqz35AQoVcs&list=PLC3y8-rFHvwgKhaLU8GTyF-5Bb8qT-wzV> -> codeevolution series on component interaction

**@ViewChild('') @ViewChildren('') @ContentChild('') @ContentChildren('')**

<https://www.youtube.com/watch?v=lreYTcPxPRk&index=6&list=PLC3y8-rFHvwgKhaLU8GTyF-5Bb8qT-wzV>

<https://www.youtube.com/watch?v=v7N2J88SP-s>

Explain Component Lifecycle?

<https://angular.io/guide/lifecycle-hooks>

<https://codecraft.tv/courses/angular/components/lifecycle-hooks/>

<https://www.ryadel.com/en/angular-ngonchanges-onchanges-lifecycle-hook-changes-async/>

<https://www.youtube.com/watch?v=ZZYo-YkixTA&index=10&list=PLC3y8-rFHvwgKhaLU8GTyF-5Bb8qT-wzV>

<https://www.youtube.com/watch?v=BYwfrSlJFfY>

<https://dev6.com/angular/when-to-use-ngoninit-and-constructor-in-angular-2-components/>

What is Templates?

A template combines HTML with Angular markup that can modify HTML elements before they are displayed.

**Angular 4** uses the **<ng-template>** as the tag instead of **<template>** which is used in Angular2. The reason Angular 4 changed **<template>** to **<ng-template>** is because there is a name conflict between the **<template>** tag and the html **<template>** standard tag. It will deprecate completely going ahead. This is one of the major changes in Angular 4.

Use of <ng-template>?

<https://www.youtube.com/watch?v=JZq3GBnOe4c>

<https://medium.freecodecamp.org/everything-you-need-to-know-about-ng-template-ng-content-ng-container-and-ngtemplateoutlet-4b7b51223691>

What is Data-binding in Angular 4 and its types?

<https://www.c-sharpcorner.com/article/data-binding-in-angular-6/>

Services in Angular 4?

<https://medium.com/@tomastrajan/total-guide-to-angular-6-dependency-injection-providedin-vs-providers-85b7a347b59f>

<https://www.techiediaries.com/angular-httpclient/>

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

<https://www.youtube.com/watch?v=PPnm_G_rDtc>

<https://coryrylan.com/blog/tree-shakeable-providers-and-services-in-angular>

**@Injectabls vs @Inject**

<https://stackoverflow.com/questions/37315317/what-is-the-difference-between-inject-and-injectable-in-angular-2-typescript>

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

## <https://stackoverflow.com/questions/36198785/how-do-i-create-a-singleton-service-in-angular-2>

What is Interface?

<https://www.youtube.com/watch?v=Be3-ZWtC1Lo>

<https://vegibit.com/how-to-use-an-interface-in-angular/>

What is Routing in Angular 4 and how to achieve it?

Routing basically means navigating between pages.

<https://www.techiediaries.com/angular-router/>

<https://www.youtube.com/watch?v=nH2S2LV8k0I&list=PLFhf0VXftAyDRmmGIAYe014xUgSHU06S4>

**Is it possible to have a multiple router-outlet in the same template?**

<https://www.techiediaries.com/angular-router-multiple-outlets/>

<https://stackoverflow.com/questions/34628848/angular2-multiple-router-outlet-in-the-same-template>

**preloading**

<https://www.youtube.com/watch?v=ceMsvxyuc6Q&list=PL6n9fhu94yhWNJaDgh0mfae_9xoQ4E_Zj&index=37>

<https://www.youtube.com/watch?v=oiqrYF0cWCw&list=PL6n9fhu94yhWNJaDgh0mfae_9xoQ4E_Zj&index=38>

**Angular lazy Loading**

<https://www.youtube.com/watch?v=75XFBIKLPQY&t=353s>

**Angular images lazy loading**

<http://www.talkingdotnet.com/how-to-lazy-load-images-angular-6/>

What is Pipes?

A pipe takes in data as input and transforms it to a desired output.

<https://www.youtube.com/watch?v=2ZBEwsXyk3M>

<https://www.youtube.com/watch?v=y8lwG8IM82k>

**custom pipes**

<https://toddmotto.com/angular-pipes-custom-pipes>

**pure vs Impure pipes and asunc pipe ?**

<https://angular.io/guide/pipes?source=post_page--------------------------->

<https://stackoverflow.com/questions/39285550/what-is-impure-pipe-in-angular>

What is View Encapsulation?

<https://codecraft.tv/courses/angular/components/templates-styles-view-encapsulation/>

View Encapsulation(encapsulation:ViewEncapsulation.none) --> enables css to apply all components

<https://www.youtube.com/watch?v=awayPS4Fa28&index=12&list=PLqq-6Pq4lTTbh7bUrKwyCWr6ABBFEd4bD>

What is Change Detection Mechanism?

* <https://dzone.com/articles/how-to-use-change-detection-in-angular>
* <https://www.toptal.com/angular/angular-change-detection>
* <https://www.youtube.com/watch?v=Ol6dFGs0Wxo>

What is Observables?

<https://stackoverflow.com/questions/42342781/why-should-we-use-subscribe-over-map-in-angular2>

<https://stackoverflow.com/questions/37364973/angular-promise-vs-observable>

<https://www.youtube.com/watch?v=mSV0AihITxo>

<https://www.youtube.com/watch?v=JFx3amVu1Yg>

Difference between Observable and Promises ?

Difference between Observable and Promises in single line?

What is the use of subscripe() and map() methods?

<https://www.youtube.com/watch?v=T9wOu11uU6U&list=PL55RiY5tL51pHpagYcrN9ubNLVXF8rGVi&index=1>

you subscribe to the observable. .subscribe is the function that actually executes the observable. It takes three callback parameters as follow:

.subscribe(success, failure, complete);

**for example:**

.subscribe(

function(response) { console.log("Success Response" + response)},

function(error) { console.log("Error happened" + error)},

function() { console.log("the subscription is completed")}

);

**.map()** is a transformer that will transform the result to whatever you return before it's passed to the success callback you should called it once on either one of them.

**For example to convert result into ‘json’ format:**

.map(res => res.json())

**What are the Securities Threats should we be Aware of in Angular 2 Applications?**

<http://pankajagarwal.in/?p=393>

<https://angular.io/guide/security>

## What is the use of codelyzer in angular 2 applications?

## <https://stackoverflow.com/questions/39943403/how-to-use-codelyzer-in-angular-2>

## <https://syndicode.com/2019/04/23/codelyzer-for-static-code-analysis-of-angular-typescript-projects/>

## AOT vs JIT in detail ?

## <https://www.youtube.com/watch?v=-h_zSpjL4TY>

## <https://stackoverflow.com/questions/41450226/angular-2-just-in-time-jit-vs-ahead-of-time-aot-compilation>

## PrimeNG,Angular Material and bootstrap ?

## What Is Traceur Compiler ?

## <https://www.code-sample.com/2015/03/what-is-traceur-compiler.html>

## How Would You Define Custom Typings To Avoid Editor Warnings?

## <https://stackoverflow.com/questions/38971984/how-to-add-custom-typings-in-typescript-2-0>

## <https://www.bennadel.com/blog/3169-adding-custom-typings-files-d-ts-in-an-angular-2-typescript-application.htm>

## What's new in angular latest release ?

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

## What are the difference between Renderer and ElementRef in angular 2?

## <https://stackoverflow.com/questions/39785118/difference-between-renderer-and-elementref-in-angular-2>

## What is hidden property in Angular 2?

## <https://www.talkingdotnet.com/dont-use-hidden-attribute-angularjs-2/>

## <https://stackoverflow.com/questions/39777990/angular2-conditional-display-bind-to-hidden-property-vs-ngif-directive>

## How can you add an active class to a selected element in a list component?

## <https://stackoverflow.com/questions/40986464/angular-2-select-clicked-list-item-add-active-class-and-remove-from-sibling>

## What would you have in a shared module?

## <https://www.youtube.com/watch?v=HFRAL5Msa_o>

## <https://medium.com/frontend-fun/angular-4-shared-modules-18ac50f24852>

**More CLIENT QUESTIONS:**

**Why should we use constructor?**

<https://css-tricks.com/understanding-javascript-constructors/>

**What is Transpiler?**

<https://stackoverflow.com/questions/47752815/transpilation-process-in-angular-2>

the default transpiler used in Angular -> Babel. you can check inside node\_modules folder in project directory. Also check tsconfig.json where its transpiled to ES2015.

**How compiler works in Angular 2?**

<https://stackoverflow.com/questions/48942691/how-angular-builds-and-runs>

**Shadow DOM?**

Angular follows regular/shadow DOM

<https://blog.thoughtram.io/angular/2015/06/29/shadow-dom-strategies-in-angular2.html>

<https://stackoverflow.com/questions/48985400/how-to-access-dom-elements-in-angular-4-service>

<https://blog.nrwl.io/understanding-angular-ivy-incremental-dom-and-virtual-dom-243be844bf36>

<https://www.youtube.com/watch?v=9cH1sYHxfR0>

**In ngFor template how to change the background color for even indexed elements?**

<https://stackoverflow.com/questions/53273162/angular-6-ngfor-display-different-styles-for-first-odd-even-and-last>

**I18n internalization**

<https://alligator.io/angular/internationalization/>

**attribute vs property**

<https://www.youtube.com/watch?v=N8FBmB2jme8&t=331s>

<https://www.youtube.com/watch?v=Ut-ATm23_Pg&pbjreload=10>

**best way to declare global ts file and access it**

<https://stackoverflow.com/questions/36158848/what-is-the-best-way-to-declare-a-global-variable-in-angular-2-typescript>

**Auth0 and AuthGuard**

[**https://www.youtube.com/watch?v=i\_dHFvi1BJc&list=WL&index=2&t=1471s**](https://www.youtube.com/watch?v=i_dHFvi1BJc&list=WL&index=2&t=1471s)

[**https://www.codeproject.com/Articles/1203649/MEAN-Stack-with-Angular-Auth-Auth-JWT-Authoriza**](https://www.codeproject.com/Articles/1203649/MEAN-Stack-with-Angular-Auth-Auth-JWT-Authoriza)

**How do you ensure angular performance**

<https://netbasal.com/optimizing-the-performance-of-your-angular-application-f222f1c16354>

<https://medium.com/@spp020/44-quick-tips-to-fine-tune-angular-performance-9f5768f5d945>

**which file runs first in an angular 4 app when i run the app?**

<https://stackoverflow.com/questions/51691628/which-file-runs-first-in-an-angular-4-app-when-i-run-the-app>

**Angular Forms**

Kudvenkat,codeevultion

**Angular Testing**

**some stuff:**

<https://stackoverflow.com/questions/54424501/how-to-implement-tree-shaking-in-angular-7-application>

<https://programmingwithmosh.com/react/react-vs-angular/>

What is Rxjs?

**RxJS** (Reactive Extensions for JavaScript) is a library for reactive programming using observables that makes it easier to compose asynchronous or callback-based code (**RxJS** Docs). ... The library also provides utility functions for creating and working with observables.

What is Typescript?

Typescript is a **superset of JavaScript** which primarily provides optional static typing, classes and interfaces. Typescript is compiled to provide clean and simple JavaScript code which runs on any browser.

TypeScript is a typed superset of JavaScript that compiles to plain JavaScript which runs on any browser or JavaScript engine.

<https://medium.com/@vigowebs/frequently-asked-typescript-interview-questions-and-answers-8e09d8429c48>

<https://www.talkingdotnet.com/typescript-interview-questions/>

What are the benefits of Typescript?

* It helps in code structuring.
* Use class based object oriented programming.
* Impose coding guidelines.
* Offers type checking.
* Compile time error checking.
* Intellisense.

### Which are the different data types supported by TypeScript?

TypeScript supports following data types.

* Boolean var bValue: boolean = false;
* Number var age: number = 16;
* String var name: string = "jon";
* Array var list:number[] = [1, 2, 3];
* Enum

|  |  |
| --- | --- |
| 1  2 | enum Color {Red, Green, Blue};  var c: Color = Color.Green; |

* Any var unknownType: any = 4;
* Void

|  |  |
| --- | --- |
| 1  2 | function NoReturnType(): void {  } |

### How can you get Typescript and install it?

TypeScript can be installed and managed via npm, the Node.js package manager. To install TypeScript, first ensure the npm is installed properly. And then run following command to install TypeScript globally on your system.

|  |  |
| --- | --- |
|  | **npm install -g typescript** |

### How do you compile Typescript files?

To compile any .ts file into .js use following command.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **tsc <TypeScript File Name>**  For example, to compile “Helloworld.ts”   |  |  | | --- | --- | |  | **tsc helloworld.ts** | |

### Which object oriented terms are supported by TypeScript?

Typescript supports following object oriented terms.

* Modules
* Classes
* Interfaces
* Data Types
* Member functions

### How do you implement inheritance in TypeScript?

Using extends keyword, we can implement inheritance.

|  |  |
| --- | --- |
|  | class Animal {      public domestic:boolean;      constructor(public name: string) { }  }    class Cat extends Animal {      constructor(name: string, domestic: boolean)      {        super(name);        this.domestic = true;      }  }    class Tiger extends Animal {      constructor(name: string, domestic: boolean)      {        super(name);        this.domestic = false;      }  } |

### How to call base class constructor from child class in TypeScript?

Using super(), we can call base class constructor, as seen in above code.

### What is the default access modifier for members of a class in TypeScript?

### In TypeScript, each member of class is **public** by default.

**What are all the other access modifiers that Typescript supports?**

Typescript supports access modifiers public, private and protected which determine the accessibility of a class member as given below:

* public - All the members of the class, its child classes, and the instance of the class can access.
* protected - All the members of the class and its child classes can access them. But the instance of the class cannot access.
* private - Only the members of the class can access them.