<https://www.wisdomjobs.com/e-university/angular-2-interview-questions.html>

**What Are The New Features Of Angular2?**

**Angular 2 is written entirely in Typescript and meets the ECMAScript 6 specification :**

* **Component-Based-** Angular 2 is entirely component based. Controllers and $scope are no longer used. They have been replaced by components and directives.
* **Directives-** The specification for directives is considerably simplified, although they are still subject to change. With the @Directive annotation, a directive can be declared.
* **Dependency Injection**- Because of the improved dependency injection model in Angular2 there are more opportunities for component / object-based work.
* **Use of TypeScript**-TypeScript is a typed super set of JavaScript which has been built and maintained by Microsoft and chosen by the AngularJS team for development. The presence of types makes the code written in TypeScript less prone to run-time errors. In recent times, the support for ES6 has been greatly improved and a few features from ES7 have been added as well.
* **Generics-** TypeScript has generics which can be used in the frontend.
* **Lambdas with TypeScript-** In TypeScript, lambdas are available.
* **Forms and Validations-** Forms and validations are an important aspect of frontend development. Within Angular 2 the Form Builder and Control Group are defined.

**What Is The Need Of Angular2?**

Angular 2 is not just a typical upgrade but a totally new development. The whole framework is rewritten from the ground. Angular 2 got rid of many things like $scope, controllers, DDO, jqLite, angular.module etc.

It uses components for almost everything. Imagine that even the whole app is now a component. Also it takes advantage of ES6 / TypeScript syntax. Developing Angular 2 apps in TypeScript has made it even more powerful.

Apart from that, many things have evolved and re-designed like the template engine and many more.

**What Is Typescript ?**

TypeScript is a typed super set of JavaScript which has been built and maintained by Microsoft and chosen by the AngularJS team for development.

**What Is The Need For Typescript In Angular2?**

**Understanding the need for TypeScript file in Angular2 applications :**JavaScript rules in Web development. Its the most popular language for developing web application UI. For may application developers having exposure in languages like Java and C#, creating the front end of a Web application in JavaScript is a very cumbersome process. For example if the user wants to create a class Employee in JavaScript. There is no class keyword in JavaScript so the code will be as follows-

<html>  
<head>  
</head>  
<body>  
<script>  
function Employee()  
{  
this.name="";  
this.id="";  
this.Validate=function()  
{  
alert("Validate");  
}  
}  
</script>  
</body>  
</html>  
**Same can be written using TypeScript as follows-**

class Employee  
{  
public name : string = "";  
public id : string = "";  
Validate()  
{  
alert("validate");  
}  
}

This Customer.ts will compile to the above JavaScript code.

So TypeScript provides the following advantages over JavaScript-

* Structure the code- There were many different coding styles for JavaScript. This leads to unstructured code. With TypeScript we create structured code.
* Use object-oriented programming paradigms and techniques-  There is lack of object-oriented design paradigms and techniques in JavaScript. This is not the case in TypeScript. It makes use of Objected Oriented features like Polymorphism, Inheritance etc.
* Standard Coding guidelines- There is no Type checking in JavaScript. The code style needs to be defined. Hard to enforce style guide. TypeScript overcomes this issue with features like Code Analysis and Navigation, Documentation, Intellisense etc.

**What Is Ecmascript ?**

ECMAScript is a subset of JavaScript. JavaScript is basically ECMAScript at its core but builds upon it. Languages such as ActionScript, JavaScript, JScript all use ECMAScript as its core. As a comparison, AS/JS/JScript are 3 different cars, but they all use the same engine... each of their exteriors is different though, and there have been several modifications done to each to make it unique. Angular2 supports ES6 and higher versions.

**What Is @ngmodule?**

@NgModule is a decorator function. A decorator function allows users to mark something as Angular 2 thing (could be a module or component or something else) and it enables you to provide additional data that determines how this Angular 2 thing will be processed, instantiated and used at the runtime. So, whenever user writes @NgModule, it tells the Angular 2 module, what’s going to be included and used in and using this module.

**What Is Traceur Compiler ?**

Traceur is a JavaScript.next-to-JavaScript-of-today compiler that allows you to use features from the future today. Traceur supports ES6 as well as some experimental ES.next features. Traceur's goal is to inform the design of new JavaScript features which are only valuable if they allow you to write better code.

**What Is Component In Angularjs 2 ?**

In Angular, a Component is a special kind of directive that uses a simpler configuration which is suitable for a component-based application structure.

**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.

**What Is @outputs In Angular?**

Components push out events using a combination of an @Output and an EventEmitter. This allows a clean separation between reusable Components and application logic.

**What Is Primeng? How Can It Be Used With Angular2?**

PrimeNG is a collection of rich UI components for Angular 2. PrimeNG is a sibling of the popular JavaServer Faces Component Suite, PrimeFaces. All widgets are open source and free to use under Apache License 2.0, a commercial friendly license. PrimeNG is developed by PrimeTek Informatics, a company with years of expertise in developing open source UI components. AngularJS makes it possible to use predefined components for development like tables etc. This helps developers save time and efforts. Using PrimeNG developers can create awesome applications in no time

**What Are Differences Between Components And Directives?**

**Components :**

* For register component we use @Component meta-data annotation.
* Component is a directive which use shadow DOM to create encapsulate visual behavior called components.Components are typically used to create UI widgets.
* Component is used to break up the application into smaller components.
* Only one component can be present per DOM element.
* @View decorator or templateurl template are mandatory in the component.

**Directives :**

* For register directives we use @Directive meta-data annotation.
* Directives is used to add behavior to an existing DOM element.
* Directive is use to design re-usable components.
* Many directive can be used in a per DOM element.
* Directive don’t have View.

**We Already Use Angular 1, Why Do We Need An Angular 2?**

**Angular 2 is built for speed :**

* 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.

**What Is An Angular 2 Component?**

Each component is comprised of a template, which is the HTML for the user interface. Add to that a class for the code associated with a view. The class contains the properties and methods, which perform actions for the view,A component also has metadata, which provides additional information about the component to Angular.

**What Is The Languages That You Can Use To Build Angular2 Application?**

**ECMAScript, or ES.**

* ES 3 is supported by older browsers.
* ES 5 is the version currently supported by most modern browsers.
* The ES 6 specification was recently approved and renamed ES 2015(Most browsers don't yet support ES 2015).

**How Can We Setting Up Our Development Environment For Angular 2?**

Setting up our development environment for Angular 2 requires two basic steps:

1. Install npm, or node package manager.
2. Set up the Angular 2 application.

**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.

**How Can We Setting Up Angular 2 Application?**

* Create an application folder.
* Create the tsconfig file(To configure the TypeScript compiler).
* Create the package.json file(To define the libraries and scripts we need).
* Create the typings.json file(That specifies a missing TypeScript type definition file).
* Install the libraries and typing files.
* Create the host Web page.(Normally index.html).
* Create the main.ts file(To bootstrap the Angular application with the root component).

**What Are The Security Threats Should We Be Aware Of In Angular 2 Application?**

Just like any other client side or web application, angular 2 application should also follow some of the basic guidelines to mitigate the security risks. Some of them are:

* Avoid using/injecting dynamic Html content to your component.
* If using external Html, that is coming from database or somewhere outside the application, sanitize it.
* Try not to put external urls in the application unless it is trusted. Avoid url re-direction unless it is trusted.
* Consider using AOT compilation or offline compilation.
* Try to prevent XSRF attack by restricting the api and use of the app for known or secure environment/browsers.

**What Are The Advantages Of Using Angular 2 Over Angular 1?**

**Angular 2 is a platform not only a language:**

* Better Speed and Performance: No $Scope in Angular 2, AOT
* Simpler Dependency Injection
* Modular, cross platform
* Benefits of ES6 and Typescript.
* Flexible Routing with Lazy Loading Features
* Easier to Learn

**How Routing Works In Angular 2.?**

Routing is a mechanism which enables user to navigate between views/components. Angular 2 simplifies the routing and provide flexibility to configure and define at module level (Lazy loading).

The angular application has single instance of the Router service and whenever URL changes, corresponding Route is matched from the routing configuration array. On successful match, it applies redirects and the router builds a tree of ActivatedRoute objects and contains the current state of the router. Before redirection, the router will check whether new state is permitted by running guards (CanActivate). Route Guards is simply an interface method that router runs to check the route authorization. After guard runs, it will resolve the route data and activate the router state by instantiation the required components into <router-outlet> </router-outlet>.

**What Are Event Emitters And How It Works In Angular 2?**

Angular 2 doesn’t have bi-directional digest cycle, unlike angular 1. In angular 2, any change occurred in the component always gets propagated from the current component to all its children in hierarchy. If the change from one component needs to be reflected to any of its parent component in hierarchy, we can emit the event by using Event Emitter api.

In short, EventEmitter is class defined in @angular/core module which can be used by components and directives to emit custom events.

@output() somethingChanged = new EventEmitter();

We use somethingChanged.emit(value) method to emit the event. This is usually done in setter when the value is being changed in the class.

This event emit can be subscribed by any component of the module by using subscribe method.

myObj.somethingChanged.subscribe(val) => this.myLocalMethod(val));

**What Is The Use Of Codelyzer In Angular 2 Application.?**

All enterprise applications follows a set of coding conventions and guidelines to maintain code in better way. Codelyzer is an open source tool to run and check whether the pre-defined coding guidelines has been followed or not. Codelyzer does only static code analysis for angular and typescript project.

Codelyzer runs on top of tslint and its coding conventions are usually defined in tslint.json file. Codelyzer can be run via angular cli or npm directly. Editors like Visual Studio Code and Atom also supports codelyzer just by doing a basic settings.

To set up the codelyzer in Visual Studio code, we can go to File -> Preferences -> User Settings and add the path for tslint rules.

Hide   Copy Code

{  
  "tslint.rulesDirectory": "./node\_modules/codelyzer",  
  "typescript.tsdk": "node\_modules/typescript/lib"  
}  
To run from cli: ng lint.   
To run from npm: npm run lint

**How Would You Optimize The Angular 2 Application For Better Performance?**

ell, optimization depends on the type and size of application and many other factors. But in general, I would consider the following points while optimizing the angular 2 app:

* Consider AOT compilation.
* Make sure the application is bundled, uglified, and tree shaking is done.
* Make sure the application doesn’t have un-necessary import statements.
* Make sure that any 3rd party library, which is not used, is removed from the application.
* Have all dependencies and dev-dependencies are clearly separated.
* I would consider lazy loading instead of fully bundled app if the app size is more.

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

Well, in most of the cases, the 3rd party library comes with its own .d.ts file for its type definition. In some cases, we need to extend the existing type by providing some more properties to it or if we need to define additional types to avoid Typescript warning.

If we need to extend the type definition for external library, as a good practice, we should not touch the node\_modules or existing typings folder. We can create a new folder, say “custom-typings” and keep all customized type definition in that.

To define typings for application (JavaScript/Typescript) objects, we should define interfaces and entity classes in models folder in the respective module of the application.

For those cases, we can define or extend the types by creating our own “.d.ts” file.

**What Is Shadow Dom? How Is It Helping Angular 2 To Perform Better?**

Shadow DOM is a part of the HTML spec which allows developers to encapsulate their HTML markup, CSS styles and JavaScript. Shadow DOM, along with a few other technologies, gives developers the ability to build their own 1st class tags, web components and APIs just like the <audio> tag. Collectively, these new tags and APIs are referred to as Web Components. Shadow DOM provides better separation of concern along with lesser conflict of styles and scripts with other HTML DOM elements.

Since shadow DOM are static in nature, it’s a good candidate to be cached as it is not accessible to developer. The cached DOM would be rendered faster in the browser providing better performance. Moreover, shadow DOM can be managed comparatively well while detecting the change in angular 2 application and re-paint of view can be managed efficiently.

**What Is Aot Compilation?**

AOT compilation stands for Ahead Of Time compilation, in which the angular compiler compiles the angular components and templates to native JavaScript and HTML during the build time. The compiled Html and JavaScript is deployed to the web server so that the compilation and render time can be saved by the browser.

**What Are The Advantages And Disadvantages Of Aot Compilation?**

**Advantages :**

* Faster download: Since the app is already compiled, many of the angular compiler related libraries are not required to be bundled, the app bundle size get reduced. So, the app can be downloaded faster.
* Lesser No. of Http Requests: If the app is not bundled to support lazy loading (or whatever reasons), for each associated html and css, there is a separate request goes to the server. The pre-compiled application in-lines all templates and styles with components, so the number of Http requests to the server would be lesser.
* Faster Rendering: If the app is not AOT compiled, the compilation process happens in the browser once the application is fully loaded. This has a wait time for all necessary component to be downloaded, and then the time taken by the compiler to compile the app. With AOT compilation, this is optimized.
* Detect error at build time: Since compilation happens beforehand, many compile time error can be detected, providing a better degree of stability of application.

**Disadvantages**

* Works only with HTML and CSS, other file types need a previous build step.
* No watch mode yet, must be done manually (bin/ngc-watch.js) and compiles all the files.
* Need to maintain AOT version of bootstrap file (might not be required while using tools like cli).
* Needs cleanup step before compiling.

**What Are The Core Differences Between Observables And Promises?**

**Promises vs Observables :**

**Promises:**

* returns a single value.
* not cancellable.

**Observables:**

* works with multiple values over time.
* cancellable.
* supports map, filter, reduce and similar operators.
* proposed feature for ES 2016.
* use Reactive Extensions (RxJS).
* an array whose items arrive asynchronously over time.

**Difference Between Constructor And Ngoninit?**

**Differences - Constructor Vs. ngOnInit**

**Angular 2 Constructors:-**

1. The constructor is a default method runs when component is being constructed.
2. The constructor is a typescript feature and it is used only for a class instantiations and nothing to do with Angular 2.
3. The constructor called first time before the ngOnInit().

**Angular 2 ngOnInit:-**

1. The ngOnInit event is an Angular 2 life-cycle event method that is called after the first ngOnChanges and the ngOnInit method is use to parameters defined with @Input otherwise the constructor is OK.
2. The ngOnInit is called after the constructor and ngOnInit is called after the first ngOnChanges.
3. The ngOnChanges is called when an input or output binding value changes.