Angular 2/4 :

1. **What is Angular When was Angular 4 released?**

Angular is a TypeScript-based open-source front-end web application platformled .

Angular is a framework for building client applications in HTML and either JavaScript or a language like TypeScript that compiles to JavaScript.

Angular is a JavaScript framework for writing Single Page Applications (SPA). March 2017

Angular is a platform and framework to build client application in HTML and TypeScript. The basic building blocks of an angular application are NG- Module, which provide a compilation context for components. Below are the core concepts 1) Module 2) Components 3) Service 4) Route 5) Template 6) Forms Template driven form Reactive form 7) HTTPClient 8) Dependency Injection 9) Providers 10) Injectors

1. **What is new in Angular 4?**

* Better View Engine
* Animation Module
* TypeScript Improvements eg: Strict Null Checks, NgIf with NgElse, More Pipes
* Angular Universal allowing Server-side rendering

1. **What is NPM?**

Node Package Manager which is Equivalent of Nuget Package Manager in .Net. NPM helps in installing packages for an Angular application.

1. **What is CLI?**

* CLI stands for Command Line Interface
* CLI provides ng new, ng serve, ng build, ng test Commands
* CLI compiles TypeScript to JavaScript
* CLI uses Nodejs in the background for Angular Compilation

1. **What is a module?**

Every application has a root module conventionally named as appModule. which provides bootstrap mechanism that launches the application. Typically It contains many functional module. Ngmodule is a container. An ngmodule defined as class decorated with @NgModule.

Below are the import properties of NgModule. declarions- componets, directives and pipe are belong to this export- The subset of decalation that should be visiable and usable in import -othere modules which are exported providers : where services are be defined. bootstrap- The main application view , called the root components .

Example

import { NgModule } from '@angular/core'; import { BrowserModule } from '@angular/platform-browser'; @NgModule({ imports: [ BrowserModule ], providers: [ Logger ], declarations: [ AppComponent ], exports: [ AppComponent ], bootstrap: [ AppComponent ] }) export class AppModule { }

1. **What is a component?**

The component is a class. Component decorator allows to mark as an Angular component . Components are the most basic building block of an UI in an Angular application. It provides additional metadata that determines how the component should be processed, instantiated and used at runtime. A component needs to configured in NgModule in order to be usable by another component or application.

Example - @Component({ selector: 'app-customer-list', templateUrl: './customer-list.component.html', providers: [ CustomerService ] }) export class CustomerListComponent implements OnInit { /\* . . . \*/ }

1. **What is the dependency Injection ?**

Dependency Injection is often called DI. Dependency Injection is wired into an application with the help of providers and used in a component .

**8. What are the phases in an angular life cycle?**

1) ngOnChanges()

2) ngOnInit

3) ngDoCheck()

4) ngAfterContentInit()

5) ngAfterContentChecked

6) ngAfterViewInit()

7) ngAfterViewChecked()

8) ngDestory()

* 1. **What is a service?**

The service is a class. An injector decorator allows to mark as an angular service. Service serves data to components.

* 1. **What is ngZone?**

The ngZone is a service. It executes task inside or outside of angular. For example. Sometime UI doesn't want to execution, but excepting directly result

* 1. **What are the router events ?**

• NavigationStart

• NativationEnd

• NavigationCancel

• NavigationError

• RoutesRecongized

• RouteConfigLoadStart

• RouteConfigLoadEnd

* 1. **What is Injector bubbling?**

When a component request to dependency. Angular tries to satisfy that dependency with a provider registered in that component of own injector. If component's injectors not able to resolve the dependency,It passes the request up to its parent component's injector. If that injector can't satisfy the request , It passes to root component until it's satisfy. If It runs out of ancestors, angular throws an error.

* 1. **WHAT ARE THE DIFFERENCES BETWEEN ANGULAR AND ANGULAR JS?**
* Angular does not have a concept of “scope” or controllers, instead it uses a hierarchy of components as its main architectural concept
* Angular has a different expression syntax, focusing on “[ ]” for property binding, and “( )” for event binding
* Mobile development – desktop development is much easier when mobile performance issues are handled first
* Modularity – much core functionality has moved to modules, producing a lighter, faster core
* Modern browsers only – reducing the need for browser compatibility workarounds
* Angular recommends the use of Microsoft’s TypeScriptlanguage, which introduces the following features:
  + Class-based Object Oriented Programming
  + Static Typing
  + Generics
* TypeScriptis a superset of ECMAScript 6 (ES6), and is backwards compatible with ECMAScript 5 (i.e.: JavaScript). Angular also includes the benefits of ES6:
  + Lambdas
  + Iterators
  + For/Of loops
  + Python-style generators
  + Reflection
* Improved dependency injection– bindings make it possible for dependencies to be named
* Dynamic loading
* Asynchronous template compilation
* Simpler Routing
* Replacing controllers and $scope with components and directives – a component is a directive with a template
* Reactive programmingsupport using RxJS

#### WHAT’S NEW IN ANGULAR 4? AND WHAT ARE THE IMPROVEMENTS IN ANGULAR 4

Angular 4 contains some additional Enhancement and Improvement. Consider the following enhancements.

1. Smaller & Faster Apps
2. View Engine Size Reduce
3. Animation Package
4. NgIf and ngFor Improvement
5. Template
6. NgIf with Else
7. Use of AS keyword
8. Pipes
9. HTTP Request Simplified
10. Apps Testing Simplified
11. Introduce Meta Tags
12. Added some Forms Validators Attributes
13. Added Compare Select Options
14. Enhancement in Router
15. Added Optional Parameter
16. Improvement Internationalization

#### HOW TO SET HTTP REQUEST HEADER IN ANGULAR 4 AND ANGULAR 2?

The HTTP Interceptors are used to Set Http Headers Request in Angular 4 using the import from “@angular/common/http”. The HTTP Interceptors are available in Angular 4.x versions.

The HTTP Interceptors are not supported in Angular 2. We are creating the HttpClient Injectable class to achieve this. You can see the below examples for set http headers request with and without HTTP interceptors.

#### WHAT IS THE USE OF INTERCEPTORS IN ANGULAR 4?

The Interceptors is a common used to set default headers for all responses.

#### WHAT IS THE FORROOT METHOD?

The forRoot is a static method and it’s very easy for developers to configure the modules and the best example is – **RouterModule.forRoot**.

The RouterModule also offers a **forChild**. It’s also a static method and use to configure the routes of lazy-loaded modules. The forRoot and forChild are the traditional names for methods that configure services in root.

#### WHAT IS THE DIFFERENCE BETWEEN `{`NGFOR`}` AND `{`NGFOROF`}` IN ANGULAR 2?

**Angular 2 – ngFor vs. ngForOf**

1. The [ngFor] is not a type safe
2. The [NgForOf] is a type Safe
3. The [NgFor] directive instantiates a template once per item from iterate
4. The [ngFor] and [ngForOf] are actually the selectors of the [NgForOf] directive and it is not two distinct things
5. The [ngFor] will be works like as collections
6. The [ngForOf] will be works like as generics

#### WHAT CLASSES SHOULD I ADD TO MODULE'S DECLARATIONS?

We can add the declarable classes like components, directives and pipes in the module’s declarations list and we can add only – components, directives and pipes classes in the @NgModule.

#### WHAT CLASSES SHOULD I NOT ADD TO MODULE'S DECLARATIONS?

We do not declare – Module, Service, objects, strings, numbers, functions, entity models, configurations, business logic, and helper classes in the module’s declarations.

#### WHAT HAPPEN WHEN I IMPORT THE SAME MODULE TWICE IN ANGULAR 4?

No problem! We can import the same module twice but Angular does not like modules with circular references.

So do not let Module “X” import Module “Y” which already imports Module “X”.

When four modules all import Module “X”, Angular estimate Module “X” once, the first time face it and does not do again. Actually, the modules help us to organize an application into associative blocks of functionality.

#### HOW TO GET AND LOG AN ERROR IN ANGULAR 4?

Two types of errors –

1. If the backend returns an unsuccessful response like – 404, 500 and so on
2. If something goes wrong in the client side like -network error etc.

In the both cases – We are using HttpErrorResponse and return the useful information on what went wrong in this call!

#### HOW ARE JWTS USED TO AUTHENTICATE ANGULAR 4 APPLICATIONS?

In Annular, the following Steps are used to building authentication and authorization for RESTful APIs and applications. It might help you –

1. The users send their credentials to the server which is verified by the database credentials. If everything is verified successfully, the JWT is sent back to them.
2. The JWT is saved in the user’s browser in local storage or in a cookie and so on.
3. The presence of a JWT saved in the browser is used as an indicator that a user is currently logged in.
4. The expiry time of JWT is continually checked to maintain an authenticated state in the Angular applications.
5. The client side routes are protected and access by authenticated users only.
6. When user sends the XHR requests for APIs, the JWT gets sent an Authorization header using your cookies or Bearer.
7. When XHR requests coming on the server, before send back the responses it’s validated first with configured app’s secret keys. If everything is looking good then returns successfully responses other send the back to the bad request.

There are several open source libraries are available for angular which are helps with JWTs and has the ability to Decode the JWT, Authorization header to XHR requests and so on.

#### WHAT IS JSON WEB TOKEN?

JSON Web Token (JWT) is an open standard which used for securely transmitting information between parties as a JSON object.

The JWTs can be signed with –

1. HMAC algorithm
2. RSA algorithm

#### WHEN SHOULD YOU USE JSON WEB TOKENS?

There are some scenarios where we can used JSON Web Tokens –

1. Authentication
2. Information Exchange

#### WHAT IS THE JSON WEB TOKEN STRUCTURE?

The JSON Web Tokens consist of three parts separated by dots (.), which are:

1. Header
2. Payload
3. Signature

**1) What is Angular 2 ?**  
    Angular 2 is released in the year 2016,Angular 2 is completely rewritten, so it has lot of improvements when compared with Angular 1.  
    From a performance standpoint, Angular 2 has faster and the other hand is designed from the ground up with mobile support.  
    In Angular 2, "everything is a component". Components are the building blocks of an Angular application.  
  
**2) What is Type Script ?**  
    TypeScript is a free and open-source programming language developed by Microsoft. It is a superset of JavaScript and compiles to JavaScript through a process     called transpilation  
  
**3) What is the default port number to run(in local) angular 2 application in browser ?**  
     http://localhost:4200/  
  
**4) What are the softwares need to install for angular 2 ?**  
    install Node.js and npm. It is recommended that you have node version 4.6.x or greater and npm 3.x.x or greater  
  
**5) What are the commands to check Node and Npm versions using command prompt ?**  
    C:\Users\user>npm -v  
      Output : 5.5.1  
  
    C:\Users\user>node -v  
      Output : v8.9.3  
 **6) How to Install the Angular CLI using command prompt ?**  
   > npm install -g @angular/cli  
  
**7) How to create a new Angular 2 application using command prompt ?**  
    > ng new angular My-Sample-App  
  
**8) How to launch the Angular 2 application using command prompt ?**  
   > cd angular My-Sample-App  
   > ng serve --open  
  
**9) How to create a new component in angular 2 application using command prompt ?**  
   > ng generate component My-First-Component  
  
**10) How to create a new service in angular 2 application using command prompt ?**  
   > ng generate service My-First-Service  
  
**11) How to create a Module in angular 2 application using command prompt ?**  
    > ng g module My-First-Module  
 **12) How to create a Directive in angular 2 application using command prompt ?**  
   > ng g directive my-new-directive  
  
**13) How to update Angular CLI old version to a new version ?**  
   you must update both the global package and your project's local package.  
   Global package:  
   > npm uninstall -g @angular/cli  
   > npm cache verify  
   > npm install -g @angular/cli@latest  
  
    Local project package:  
   > rm -rf node\_modules dist # use rmdir /S/Q node\_modules dist in Windows Command Prompt; use rm -r -fo node\_modules,dist in Windows PowerShell  
   > npm install --save-dev @angular/cli@latest  
   > npm install  
  
**14) What are the supported Editors in TypeScript ?**  
    1. Visual Studio Code  
    2. Eclipse  
    3. WebStorm  
    4. Atom  
    5. Sublime Text  
 **15) Can we learn AngularJS 1 before learning Angular 2 ?**  
     No. You can think of AngularJS 1 and Angular 2 as 2 different frameworks. The concepts, the API's and patterns that we use to build applications are very      different between these 2 versions. So there is no need to learn AngularJS 1 before you learn Angular 2  
  
**16) Can we learn Angular 2 before learning Angular 4 ?**  
     Yes,Building an application using Angular 2 and Angular 4 is not very different. We still use the same concepts, APIs and patterns.  
  
**17) What is the difference between AngularJS and Angular 2 ?**  
    AngularJS and was released in the year 2010  
    AngularJS applications around the concept of controllers,View and $scope in AngularJS  
    AngularJs is 5 times  slow compared to Angular 2.  
    AngularJS was not built for mobile devices  
    AngularJS no more language choices,no TypeScript used  
  
    Angular 2 is released in the year 2016  
    Angular 2 both controllers and $scope are gone,Angular 2 is entirely component based, which means we create a set of independent or loosely coupled components  
    Angular 2 is 5 times faster compared to AngularJS.  
    AngularJS was built for mobile devices  
    Angular 2 we have more language choices. In additon to nativa JavaScript we can use TypeScript, Dart, PureScript, Elm, etc

**18) What is a component in Angular 2 ?**  
    A component in Angular is a class with a template and a decorator. So in simple terms a component in Angular is composed of these 3 things  
    **Template**- Defines the user interface. Contains the HTML, directives and bindings  
    **Class**- Contains the code required for template. Just like a class in any object oriented programming language like C# or Java, a class in angular can contain     methods and properties. Properties contain the data that we want to display in the view template and methods contain the logic for the view. We use TypeScript to     create the class.  
   **Decorator**- We use the Component decorator provided by Angular to add metadata to the class. A class becomes an Angular component, when it is decorated with the     Component decorator.  
  
 **19) What are the differences between template and templateUrl properties and when to use one over the other ?**  
    Angular2 recommends to extract templates into a separate file, if the view template is longer than 3 lines. Let's understand why is it better to extract a view     template into a seprate file, if it is longer than 3 lines.  
  
     With an inline template   
    We loose Visual Studio editor intellisense, code-completion and formatting features.  
    TypeScript code is not easier to read and understand when it is mixed with the inline template HTML.  
  
    With an external view template  
   We have Visual Studio editor intellisense, code-completion and formatting features and  
   Not only the code in "app.component.ts" is clean, it is also easier to read and understand  
  
  
**20) What is AppModule ?**  
    AppModule is the root module which bootstraps and launches the angular application. You can name it anything you want, but by convention it is named AppModule.  
     It imports 2 system modules - BrowserModule and NgModule  
    BrowserModule - Every application that runs in a browser needs this module. In a later video in this course we will discuss NgIf and NgFor directives which are     also provided by this module.  
    NgModule - @component decorator adds metadata to an angular component class, similarly @NgModule decorator adds metadata to the angular module class  
  
**21) How to Error Handle in angular2.0 ?**  
  
    private handleError(error: Response) {  
        console.error(error);  
        return Observable.throw(error.json().error());  
     }

**22) What is Pipes in Angular 2 ? What are those ?**  
 Transform data before display we called it as Pipes  
 Built in pipes include lowercase, uppercase, decimal, date, percent, currency etc  
 To apply a pipe on a bound property use the pipe character " | "  
   <td>{{employee.code | uppercase}}</td>  
   We can also chain pipes<td>{{employee.dateOfBirth | date:'fullDate' | uppercase }}</td>  
  
**23) What is ngFor in angular 2 ? How to define ngFor in angular 2 with example ?**  
   1) ngFor is usually used to display an array of items

   2) Since ngFor is a structural directive it is prefixed with \*  
   Example: \*ngFor='let employee of employees'  
  
            <tr \*ngFor='let employee of employees'>  
            <td>{{employee.code}}</td>  
            <td>{{employee.name}}</td>  
            <td>{{employee.gender}}</td>  
            <td>{{employee.annualSalary}}</td>  
            <td>{{employee.dateOfBirth}}</td>  
           </tr>  
  
**24) What is ngIf in angular 2 ? How to define ngIf in angular 2 with example ?**  
    ngIf structural directive displays the row "No employees to display" when employees property does not exist or when there are ZERO employees in the array.  
     Example : <tr \*ngIf="!employees || employees.length==0">  
               <td colspan="5">  
                No employees to display  
            </td>  
           </tr>  
  
**25) What is Data Binding in angular 2?**  
  Data binding help us coordinate communication between a component and its view template. Data binding consist of One-Way Data-Binding and Two-Way Data-Binding  
  
**26) How many ways can data bind in angular2 ?**  
  In Angular data-binding can be broadly classified into 3 categories  
  Data Binding          Description  
  One way data-binding From Component to View Template  
  One way data-binding From View Template to Component  
  Two way data-binding From Component to View Template & From View template to Component  
  
**27) What is One way data-binding (From Component to View Template) ?**  
   One way data-binding - From Component to View Template : To display read-only data on a view template we use one-way data binding technique  
   In the following example, Angular pulls the value of the firstName property from the component and inserts it between the opening and closing <h1> element.

    import { Component } from '@angular/core';  
  
    @Component({  
       selector: 'my-app',  
       template: `  
                   <h1>{{ firstName }}</h1>  
                 `  
     })  
    export class AppComponent {  
          firstName: string = 'Srini';  
      }  
  
   Output :  Srini

**28) What is the difference between Interpolation and Property binding ?**  
   Interpolation is a special syntax that Angular converts into a property binding.  
   Interpolation is just a convenient alternative to property binding.  
   In some cases like when we need to concatenate strings we have to use interpolation instead of property binding as shown in the example below.  
        <img src='http://www.pragimtech.com/{{imagePath}}' />  
    When setting an element property to a non-string data value, you must use property binding. In the following example, we are disabling a button by binding to the      boolean property isDisabled.  
       <button [disabled]='isDisabled'>Click me</button>  
   If we use interpolation instead of property binding, the button is always disabled irrespective of isDisabled class property value  
  <button disabled='{{isDisabled}}'>Click me</button>  
  
**29) How to resolve this below error ?**  
    No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'http://localhost:12345' is therefore not allowed access.  
    We get this error because our Angular application and Web API service are in different projects. Because they are present in different projects the port numbers     are different. Since the port numbers are different, the request from the angular project to the web api project is a cross domain request which violates the same      origin policy and as a result the browser blocks the request for security reasons  
  
    To fix this error include the following setting in web.config file of the Web API project  
     <system.webServer>  
       <httpProtocol>  
         <customHeaders>  
           <add name="Access-Control-Allow-Origin" value="\*" />  
           <add name="Access-Control-Allow-Headers" value="Content-Type" />  
           <add name="Access-Control-Allow-Methods"  
            value="GET, POST, PUT, DELETE, OPTIONS" />  
       </customHeaders>  
      </httpProtocol>  
     </system.webServer>  
  
**30) What is an Observable ?**  
   Observable is an asynchronous pattern. In the Observable pattern we have an Observable and an Observer. Observer observes the Observable. In many implementations an   Observer is also called as a Subscriber.  
   An Observable can have many Observers (also called Subscribers).  
   Observable emits items or notifications over time to which an Observer (also called Subscriber) can subscribe.  
   When a subscriber subscribes to an Observable, the subscriber also specifies a callback function.  
   This subscriber callback function is notified as and when the Observable emits items or notifications.  
   Within this callback function we write code to handle data itmes or notifications received from the Observable.

## **1. Angular 4 vs. Angular 2**

There is no path breaking difference between angular 2 and angular 4. Angular 4 is simply the next version of Angular 2. The underlying concepts are same. If you know Angular 2, you can easily switch to Angular 4.  
Angular 4 is **backward compatible** with Angular 2 for most Applications.   
  
There are some under the hood changes to reduce the **size of the AOT(Ahead-of-Time)** compiler generated code. Migrating to Angular 4 may reduce production bundle.  
Angular 4 is **faster** than Angular 2. The apps developed in Angular 4 are five times faster and smaller as compared to Angular 2.  
**TypeScript 2.1 and 2.2 compatiblity**. Before Angular 4, only TypeScript 1.8 was supported. So with Angular 4, we have all new features of Typescript available.  
Unlike Angular 2, the **animations** have been pulled out of @angular/core and are moved into their package in the Angular 4. If you don't use animations, this extra code will not end up in the production bundle.  
We can now use new if/else style syntax with **\*ngif structural directive**. In Angular 2, it was not possible to use "else" statement with nglf, but now it has been made possible in Angular 4.   
  
Angular 4 has introduced a new **titlecase pipe**. It changes the first letter of each word into uppercase  
  
"**As**" keyword is the new addition to the template syntax to simplify the "let" syntax  
Integration of **Angular Universal** - The integration of Angular Universal permits developers to run Angular on a server.  
The **template tag is now deprecated**: you should use the "ng-template" tag instead as Angular has its own template tag: ng-template now.

## **Why Angular 4? What’s New in Angular 4?**

**Router ParamMap**  
Starting from version 4, it is possible to use **paramMap** to get the route- and query-parameter from a related route. The use of **Map** brings advantages in terms of type security.   
The old had an unsecure type (type params = {[key: string]: any}) and the value could have all possible types.   
The new way provides either a string, or an array of strings related to the used method (paramMap.get(): string and paramMap.getAll(): string[])

**Animations**  
Earlier all the functions of animations were the part of @angular/core module, which means the code were always included in the bundle even if animations were not used.  
  
In Angular 4, Animations are to be provided in the module BrowserAnimationsModule from @angular/platform-browser/animations. This avoids creating bundles with unnecessary large sizes.     
  
**ngif**  
We can now use new if/else style syntax with \*ngif structural directive.  
  
**NgComponentOutlet**  
To build and produce components dynamically at runtime involved relatively much programming work. With the introduction of \*ngComponentOutlet-Directive in Angular 4, it is possible to build dynamic components in a declarative way.    
  
**TypeScript 2.1/2.2**  
We have the support of most latest TypeScript versions in Angular 4 which helps in improving the speed of ngc-Compiler.

**Angular Universal**  
With Angular Universal, it is possible to render Angular applications on the web server. With that, websites can be optimized better for search engines as JavaScript is no longer necessary for initially rendering the page content.

## **3. What is the use of Interceptors?**

Interceptors are used to intercept and/or mutate outgoing requests or incoming responses. It can be really useful for features like caching and logging.  
Interceptors can be used on multiple scenarios, i.e. setting the Origin for each outgoing request, adding authentication token to every request etc.

## **4. What is Angular?**

Angular is a framework for building client applications in HTML, CSS and Javascript(or language like Typescript which compiles into Javascript).

## **5. Why do we need Angular?**

We can develop application using Javascript and Jquery. But as the application become complex, code in Javascript and Jquery become difficult to maintain. We then require to structure the application code properly by incorporating object oriented features. That is why a framework like Angular has been evolved to make web application development and maintenance faster and easier. The application in Javascript is hard to test. Applications build in Angular are easily testable.

* Angular gives our application a clean object oriented structure that is easy to understand and easy to maintain.
* Angular come with a lot of utility code that can be reused in various applications.
* Applications build in Angular are easily testable.

## **6. What is Node js?**

It is basically a runtime environment for executing Javascript code outside the browser.

## **7. Angular CLI**

Angular CLI stands for Command-line Interface.

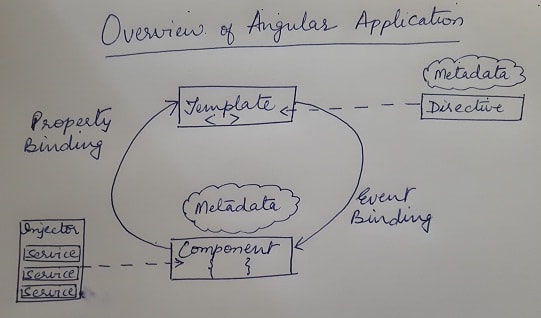
## **7. What is TypeScript?**

TypeScript is a superset of Javascript. So any valid Javascript code is also a TypeScript code. TypeScript has many additional features that Javascript doesn't offer:

* **Strong Type** - You can specify variable type at the time of declaring a variable which makes code easier to maintain and catching errors become easier. Although this feature is optional.
* **Object Oriented features** - TypeScript brings many object oriented features which have been missing in Javascript like classes, interfaces, constructor, access modifier, properties etc.
* In TypeScript, We can catch **error** at compile time instead of runtime
* With TypeScript, We also get **intellisense** in the code editor.

Browser don't understand TypeScript code, TypeScript compiles into JavaScript at the time when we build the application.

## **8. Architecture of Angular Apps**

Angular applications are designed by composing HTML templates with Angularized markup, writing component classes to manage those templates, adding application logic in services, and boxing components and services in modules. 

## **9. Building blocks of Angular Apps**

**Components** - Angular embraces component based architecture which allows us to work with smaller and maintainable piece of code that can be reused at several places. Each Angular App has one or more components. A component controls a patch of screen called a view. It encapsulates Data, HTML template and Logic for a view (area of the screen that the users see).   
  
Every application has a root component that we call as App component.  
  
**Modules** - It is container for group of related components, i.e. in an employee module, we can have components for displaying employees details. Every Angular app has at least one NgModule class, the root module, conventionally named AppModule.  
  
**Templates** - You define a component's view with its companion template. A template is a form of HTML that tells Angular how to render the component.  
  
**Metadata** - Metadata tells Angular how to process a class.  
  
**Example**  
  
In fact, CourseListComponent really is just a class. It's not a component until you tell Angular about it.  
  
To tell Angular that CourseListComponent is a component, attach metadata to the class.  
  
In TypeScript, you attach metadata by using a decorator. Here's some metadata for CourseListComponent:

@Component({  
  selector:    'app-course-list',  
  templateUrl: './course-list.component.html',  
  providers:  [ courseService ]  
})  
  
export class CourseListComponent implements OnInit {  
/\* . . . \*/  
}

**Directives** - A directive is a class with a @Directive decorator. A component is a directive-with-a-template; a @Component decorator is actually a @Directive decorator extended with template-oriented features.  
  
Directives either alter the layout structure (for example, ngSwitch) or modify aspects of DOM elements and components (for example, ngStyle and ngClass).  
  
Of course, you can also write your own directives. Components such as HeroListComponent are one kind of custom directive.  
  
**Services** - A service is typically a class with a narrow, well-defined purpose. Components are big consumers of services. Component classes should be lean. They don't fetch data from the server, validate user input, or log directly to the console. They delegate such tasks to services. A component's job is to enable the user experience and nothing more. It mediates between the view (rendered by the template) and the application logic.   
  
Angular factor your application logic into services and make those services available to components through dependency injection.  
  
**Dependency injection** is a way to supply a new instance of a class with the fully-formed dependencies it requires. Most dependencies are services. Angular uses dependency injection to provide new components with the services they need.

## **10. Steps to follow in order to use component.**

* Create a Component
* Register it in a module
* Add an element in an HTML markup

1. **Question 1. What Are Angular 4?**

**Answer :**

On 13 December 2016 Angular 4 was announced, skipping 3 to avoid confusion due to the misalignment of the router package’s version which was already distributed as v3.3.0. The final version was released on March 23, 2017. Angular 4 is backward compatible with Angular 2.

Angular version 4.3 is a minor release, meaning that it contains no breaking changes and that it is a drop-in replacement for 4.x.x.

1. **Question 2. What Are The Features Of Angular 4.3?**

**Answer :**

**Features in Angular version 4.3 are:**

* + Introducing Http Client, a smaller, easier to use, and more powerful library for making HTTP Requests.
  + New router life cycle events for Guards and Resolvers. Four new events: GuardsCheckStart, GuardsCheckEnd, ResolveStart, ResolveEnd join the existing set of life cycle event such as NavigationStart.
  + Conditionally disable.

1. **Question 3. What Is Angular?**

**Answer :**

Angular (commonly referred to as “Angular 2+” or “Angular 2“):

Is a TypeScript-based open-source front-end web application platform bed by the Angular Team at Google and by a community of individuals and corporations to address all of the parts of the developer’s workflow while building complex web application. Angular is a complete rewrite from the same team that built AngularJS.

Angular is a framework for building client applications in HTML and either JavaScript or a language like TypeScript that compiles to JavaScript. Angular combines declarative templates, dependency injection, end to end tooling, and integrated best practices to solve development challenges. Angular empowers developers to build applications that live on the web, mobile, or the desktop.

1. **Question 4. What Is Angular Js?**

**Answer :**

AngularJS is a structural framework for dynamic web apps. It lets you use HTML as your template language and lets you extend HTML syntax to express your application’s components clearly and succinctly. AngularJS data binding and dependency injection eliminate much of the code you would otherwise have to write. And it all happens within the browser, making it an ideal partner with any server technology.

AngularJS is what HTML would have been, had it been designed for applications. HTML is a great declarative language for static documents. It does not contain much in the way of creating applications, and as a result building web applications is an exercise in.

1. **Question 5. What Do I Have To Do To Trick The Browser Into Doing What I Want?**

**Answer :**

**The impedance mismatch between dynamic applications and static documents is often solved with:**

**A library** – a collection of functions which are useful when writing web apps. Your code is in charge and it calls into the library when it sees fit. E.g., jQuery.

**Frameworks** – a particular implementation of a web application, where your code fills in the details. The framework is in charge and it calls into your code when it needs something app specific.

E.g., durandal, ember, etc.

AngularJS takes another approach. It attempts to minimize the impedance mismatch between document centric HTML and what an application needs by creating new HTML constructs. AngularJS teaches the browser new syntax through a construct we call directives.

**Examples include:**

* + Data binding, as in {{}}.
  + DOM control structures for repeating, showing and hiding DOM fragments.
  + Support for forms and form validation.
  + Attaching new behavior to DOM elements, such as DOM event handling.
  + Grouping of HTML into reusable components.

1. **Question 6. What Are The Differences Between Angular And Angular Js?**

**Answer :**

Angular was a ground-up rewrite of AngularJS and has many unique features.

* + Angular does not have a concept of “scope” or controllers; instead it uses a hierarchy of components as its main architectural concept
  + Angular has a different expression syntax, focusing on “[ ]” for property binding, and “( )” for event binding
  + Mobile development – desktop development is much easier when mobile performance issues are handled first
  + Modularity – much core functionality has moved to modules, producing a lighter, faster core
  + Modern browsers only – reducing the need for browser compatibility workarounds
  + Angular recommends the use of Microsoft’s Typescript language, which introduces the following features:
  + Class-based Object Oriented Programming
  + Static Typing
  + Generics

Typescript a superset of ECMAScript 6 (ES6), and is backwards compatible with ECMAScript 5 (i.e.: JavaScript).

**Angular also includes the benefits of ES6:**

* + Lambdas
  + Iterators
  + For/Of loops
  + Python-style generators
  + Reflection
  + Improved dependency injection– bindings make it possible for dependencies to be named
  + Dynamic loading
  + Asynchronous template compilation
  + Simpler Routing
  + Replacing controllers and $scope with components and directives – a component is a directive with a template.
  + Reactive programming support using RxJS.

1. **Question 7. What’s New In Angular 4? And What Are The Improvements In Angular 4?**

**Answer :**

Angular 4 contains some additional Enhancement and Improvement.

**Consider the following enhancements:**

* + Smaller & Faster Apps
  + View Engine Size Reduce
  + Animation Package
  + NgIf and ngFor Improvement
  + Template
  + Ng If with Else
  + Use of AS keyword
  + Pipes
  + HTTP Request Simplified
  + Apps Testing Simplified
  + Introduce Meta Tags
  + Added some Forms Validator Attributes
  + Added Compare Select Options
  + Enhancement in Router
  + Added Optional Parameter
  + Improvement Internationalization

1. **Question 8. Why Angular 4? What’s New In Angular 4?**

**Answer :**

* + It Makes work easier compared with angular 2 and other models.
  + Writing code is lots of cleaner and lesser.
  + It Improve the execution performance for Data binding and so on.
  + It has included Animations features.
  + In Angular 4, no need to apply observable methods because Angular analyses every page’s DOM and it is automatically modifies to page’s DOM.
  + It is also supported by Visual Studio, IntelliJ, And NET IDES and so on.
  + Migration is really very soft and cleaner.
  + And So On…

**Angular 2 apps will work using Angular 4 without changing anything. Angular 4 offers lots-of enhancements i.e.**

* + Smaller & Faster Apps
  + View Engine Size Reduce
  + Animation Package
  + NgIf and ngFor Improvement
  + Overriding Template
  + NgIf with Else
  + Use of AS keyword
  + Pipes
  + HTTP Request Simplified
  + Apps Testing Simplified
  + Introduce Meta Tags
  + Added some Forms Validator Attributes
  + Added Compare Select Options
  + Enhancement in Router
  + Added Optional Parameter
  + Improvement Internationalization
  + Meaningful errors handling methodology
  + Animations

1. **Question 9. How To Set Http Request Header In Angular 4 And Angular 2?**

**Answer :**

The HTTP Interceptors are used to Set Http Headers Request in Angular 4 using the import from “@angular/common/http”. The HTTP Interceptors are available in Angular 4.x versions.

The HTTP Interceptors are not supported in Angular 2. We are creating the Http Client Injectable class to achieve this. You can see the below examples for set http headers request with and without HTTP interceptors.

1. **Question 10. What Is The Use Of Interceptors In Angular 4?**

**Answer :**

The Interceptors is a common used to set default headers for all responses.

1. **Question 11. What Is The For Root Method?**

**Answer :**

The for Root is a static method and it’s very easy for developers to configure the modules and the best example is – RouterModule.for Root.

The Router Module also offers a for Child. It’s also a static method and use to configure the routes of lazy-loaded modules. The for Root and for Child are the traditional names for methods that configure services in root.

1. **Question 12. What Is The Difference Between `{'ngfor'}` And `{`ngforof`}` In Angular 2?**

**Answer :**

**Angular 2 – ngFor vs. ngFor:**

* + The [ngFor] is not a type safe.
  + The [NgForOf] is a type Safe.
  + The [NgFor] directive instantiates a template once per item from iterate.
  + The [ngFor] and [ngForOf] are actually the selectors of the [NgForOf] directive and it is not two distinct things.
  + The [ngFor] will be works like as collections.
  + The [ngForOf] will be works like as generics.

1. **Question 13. What Classes Should I Add To Module's Declarations?**

**Answer :**

We can add the declarable classes like components, directives and pipes in the module’s declarations list and we can add only – components, directives and pipes classes in the @NgModule.

1. **Question 14. What Classes Should I Not Add To Module's Declarations?**

**Answer :**

We do not declare – Module, Service, objects, strings, numbers, functions, entity models, configurations, business logic, and helper classes in the module’s declarations.

1. **Question 15. What Happen When I Import The Same Module Twice In Angular 4?**

**Answer :**

* + No problem! We can import the same module twice but Angular does not like modules with circular references.
  + So do not let Module “X” import Module “Y” which already imports Module “X”.
  + When four modules all import Module “X”, Angular estimate Module “X” once, the first time face it and does not do again. Actually, the modules help us to organize an application into associative blocks of functionality.

1. **Question 16. How To Get And Log An Error In Angular 4?**

**Answer :**

**Two types of error:**

* + If the backend returned an unsuccessful response like – 404, 500 and so on.
  + If something goes wrong in the client side like -network error etc.

In the both cases – We are using Http Error Response and return the useful information on what went wrong in this call!

1. **Question 17. How Are Jwts Used To Authenticate Angular 4 Applications?**

**Answer :**

**In Annular, the following Steps are used to building authentication and authorization for RESTful APIs and applications. It might help you** –

* + The users send their credentials to the server which is verified by the database credentials. If everything is verified successfully, the JWT is sent back to them.
  + The JWT is saved in the user’s browser in local storage or in a cookie and so on.
  + The presence of a JWT saved in the browser is used as an indicator that a user is currently logged in.
  + The expiry time of JWT is continually checked to maintain an authenticated state in the Angular applications.
  + The client side routes are protected and access by authenticated users only.
  + When user sends the XHR requests for APIs, the JWT gets sent an Authorization header using your cookies or Bearer.
  + When XHR requests coming on the server, before send back the responses it’s validated first with configured app’s secret keys. If everything is looking good then returns successfully responses other send the back to the bad request.

There are several open source libraries are available for angular which are helps with JWTs and has the ability to Decode the JWT, Authorization header to XHR requests and so on.

1. **Question 18. What Is Json Web Token?**

**Answer :**

JSON Web Token (JWT) is an open standard which used for securely transmitting information between parties as a JSON object.

**The JWTs can be signed with** –

* + HMAC algorithm
  + RSA algorithm

1. **Question 19. When Should You Use Json Web Tokens?**

**Answer :**

**There are some scenarios where we can used JSON Web Tokens –**

* + Authentication
  + Information Exchange

1. **Question 20. What Is The Json Web Token Structure?**

**Answer :**

**The JSON Web Tokens consist of three parts separated by dots (.), which are:**

* + Header
  + Payload
  + Signature

1. **Question 21. Explain Component Decorators In Angular 4?**

**Answer :**

A decorator is the core concept when developing an angular framework with version 2 and above. It may become a core language feature for JavaScript soon. In angular 4, decorators are used extensively and are also used to compile a code.

**There are 4 different types of decorators:**

* + Class decorators
  + Property decorators
  + Method decorators
  + Parameter decorators

A decorator is a function that is invoked with a prefix “@” symbol and is immediately followed by a class, parameter, method, or property. A decorator returns the same thing which was given as an input but in an augmented form.

1. **Question 22. Write The Cli Command To Generate A Component In Angular 4?**

**Answer :**

Components are just simple classes which are declared as components with the help of component decorators.

It becomes easy to create an application which already works, with the help of angular CLI commands. “Ng generate” is used to generate components, routes, services, and pipes. Simple test shells are also created with the help of this CLI command. For generating a component in angular4 with the help of CLI command.

**you need to follow the following syntax**-

* + ng generate component component name;

It generates the component and adds the component to module declarations.

1. **Question 23. Explain The Component Directory Structure Of Angular 4?**

**Answer :**

**Here are the elements which are present in the component directory structure anf modules: –**

**Module.ts-** in this, the angular module is declared. @NgModule decorator is used which initializes the different aspects of angular applications. AppComponent is also declared in it.

**Components.ts**- it simply defines the components in angular and this is the place where the app-root sector is also defined. A title attribute is also declared in the component.

**Component.html**- it is the template file of the application which represents the visual parts of our components.

1. **Question 24. Explain Ngfor Directive With An Example?**

**Answer :**

The ngFor directive instantiates a template for every element of the given iterator. The different local variables of the ngFor directive can be used in iterations. The ngFor directive can even be used with the HTML elements. It also performs various changes in DOM. Several exported values can be aliased to local variables with the help of ngFor directive. It allows us to build data presentation lists and tables in our HTML templates.

**Here’s an example of ngFor directive with the help of HTML:**

<tr \*ngFor=”hero of heroes”>

<td>({hero.name})</td></tr>

1. **Question 25. Explain Property Binding Or One Way Binding In Angular Js?**

**Answer :**

Basically property binding means passing data from the component class and setting the value of a given element in the view. It is a one way binding in which the data is passed from component to the class. It allows us to control the element property values from component to class. Property binding in angular can take place by three ways:

Interpolation can be used to define a value, as long as the value being defined is a string.

Wrapping brackets around the element property and binding it to the component property is the most common type of property binding.

The third way is by adding “bind” before the element property.

1. **Question 26. Explain Ngif Directive With An Example?**

**Answer :**

The ngIf is a built-in template directive which is used to add or remove some parts of DOM. This addition or removal depends on the expression being true or false.

If the expression is evaluated to false, then the ngIf directive removes the HTML element. If the expression is evaluated to be true, then the element gets added to the DOM.

Syntax- \*ngIf=”<condition>”

Example- <ul \*ngFor=”let person of people”

\*ngIf=”person.age < 30”>

<li>{{person.name}}</li></ul>

1. **Question 27. Write The Difference Between Directive And Component In Angular Js?**

**Answer :**

**In angular js, there are differences between the meta-data annotations. Some of the differences are:**

* + A directive is used to add behavior to an existing element. Whereas, a component is used to create a component with attached behavior.
  + “@directive” is used to create a directive. Whereas, “@component” is used to create a component.
  + A directive is used to attach different behaviors to an existing DOM element. Whereas, with the help of component, we can break our application into smaller components.
  + A directive is used to create reusable behavior. Whereas, a component is used to create reusable components.
  + A directive does not require a view. Whereas, a component needs a view via @view.

1. **Question 28. What Do You Understand By Isolated Unit Tests?**

**Answer :**

As the name implies, unit test is all about testing individual units of code. In order to answer some questions, isolating the unit of code under test is really important. When we do this, we are not forced into creating related pieces such as DOM elements for sorting. With the help of isolated unit tests, it becomes easier to implement everything. To simulate the requests, dependency injections are also provided. The individual sort function can be tested in isolation. And not only the sort function, any function can be tested in isolation.

1. **Question 29. What Is A Routing In Angular Js?**

**Answer :**

NgRoute module is used when you want to navigate through different pages of your application but you also want your application to be a single page application. This ngRoute module navigates through different pages of your application without reloading the entire application. The angular js route module should be included to make your application ready for routing. The ngRoute is added as a dependency in the application. The routing engine captures the specific url requested by the user and renders the view based on the defined routing rules.

1. **Question 30. What Do You Understand By Services With Reference To Angular Js?**

**Answer :**

Services in angular js are used to organize and share code across your application. These are the suitable objects which are wired together with the help of dependency injection. The angular js services are lazily instantiated. The service is only instantiated by angular js only when the application component depends on it. In angular js, new services can be made or can even be used in other built-in services. Over 30 built-in services are present in angular js.

#### What is AngularJS 2/4?

[Angular 2/4](http://www.learnangularjs.net/) is an open-source front-end web application platform, which works as a binding framework created by google development team

#### What is the difference between Angular 1(Angularjs) and Angular 2/4?

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

#### What is the difference between AngularJS 1(Angularjs) and Angular 2/4?

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.

#### What is Typescript?

➔ In Simple word Typescript is a Superset of JavaScript meaning any JavaScript code is valid typescript code.  
Typescript is Trans piled to JavaScript behind the scene in the IDE.  
JavaScript is function based which can get confusing to a C# or Java developer while reviewing code So Typescript allows us to use generic programming (classes and modules).  
Typescript follows the ECMAScript 2015 standard which contains features like:

➢Generic  
➢Namespaces  
➢Enumerated type  
➢Interfaces etc.

#### What are the new features of Angular 4 over Angular 2?

➔ The new View Engine (for AOT) in Angular 4 which reduces the generated code from components up to 60%.  
Now in Angular 4 we can use the "if-else" conditional syntax instead of just using "if" like in Angular 2.  
Angular 4 requires a minimum of Typescript 2.1 or higher.  
Animations is removed from @angular/core so as to reduce the size of our code bundle.  
But we can add animations by importing {BrowserAnimationsModule} from @angular/platform-browser/animations our NgModule.  
Renderer is replaced by Renderer 2 in '@angular/core'.  
In Angular 4 there no need of adding pattern for validation we can just define type="email" in our HTML Input which is an Html Dom property and angular will take care of it.

#### What are Components in Angular 2/4?

➔ Angular is based on Components, as Components are the logical piece of code, which consists of Classes, Template and Metadata.  
To say more simplistically Components are used to bind model and template while also providing the logical part.

#### What are modules in Angular 2/4?

➔ Modules are used for organizing components, models, templates and other parts instead of coding everything into a single application.  
This results in a more modular application, which is separated into different modules depending on their functionality and need of our app.

#### What is @inputs In Angular 2?

➔ @Input decorator allows us to pass data into our application controller and templates through html as well as defining custom properties.  
The @Input decorator binds a property with our child component so that it can communicate and pass values from parent to child.

#### What is @outputs In Angular 2?

➔ @Output along with EventEmitter is used to push out events in components.  
The @Output decorator binds a property with our child component so that our child component can call its parent component. We can say it is used for communication between parent and child component.

#### What is npm and why do we need it?

➔ Npm stands for Node package manager. It is an online repository and works as a package manager for JavaScript.  
A package.json file in our application defines the dependencies that we want to download for our application.

#### What is Routing and how does it work in Angular 2/4? How is it done?

➔ The routing mechanism allows us to navigate between different views(via components) and allowing us to configure it to make it more flexible while providing us with features like lazy loading to ease load times and increase performance.

Routing used for the following : -

➢ It is used to create a modular application

➢ To implement role based authorization in our application providing access to certain users to certain parts of the application.

➢ It is used for State management in the application

**Steps :-**

1. Create a module to store routing paths and define different paths and routes in it.
2. Import the routing module to your app.module.ts and add routing module name to the imports of @NgModule.
3. Add routing selector tags to your appcomponent's html.
4. Adding router links for navigation and
5. <base href="/"> to our index.html

#### What is AOT Compilation? Explain its advantages and disadvantages?

➔ AOT stands for Ahead of Time, There is no different compiler used in AOT just the sequence of process is changed.  
Instead of compiling at runtime in the browser like JIT compilation.  
The components and templates are compiled at built time and converted to native JavaScript and html.  
So the browser does not have compile the application so it can directly render it and save time and system resources.

**Advantages:**AOT provides build time error detection so many compile type errors are detected before it running the application.  
Faster download times as the application is compiled and bundled at build time, which eliminates the need of loading the angular compiler and its libraries leading to a lesser runtime.  
Faster Rendering time as the application is already compiled and only needs to be rendered on the browser.

**Disadvantages:** Angular version lower than 4 produces large JavaScript bundles after compilation, which defeats the purpose of AOT.  
It only works only with HTML and CSS, other file types need to be built previously.  
It needs a clean-up step before compiling.

#### What is lazy loading?

➔ Lazy loading modules speeds up our applications startup time.  
Lazy loading creates multiple bundles and loads them on demand in runtime.  
If we had loaded all our components and templates into one big bundle, it would lead to a large performance penalty.

#### What is ECMA Script?

➔To say as simple as possible ECMAScript is a subset of JavaScript.  
ECMAScript is at the core of JavaScript and JavaScript just builds up on it.  
ActionScript, JavaScript, Jscript and all other languages have ECMAScript at its core.

#### What are pipes in Angular 2/4?

➔ Pipes are used in Angular 2/4 to edit/transform and format our data in the template itself.  
Angular has built-in pipes for dates, currency, percentage and character cases.  
In Addition to this we can create custom pipes ourselves and use them in angular.  
Therefore, we can say that pipes take in data as input and transforms it to output we desire.

#### What is Change detection?

➔ Change detection basically is the process of detecting the changes in the internal state of our application which might be caused due to Events, XHR and Timers and make it visible in the user interface.  
Change detection work on the principle of zone.js in Angular 2/4.

#### What are Directives in Angular 2/4?

➔ Directives in Angular are used to add behavior to our existing DOM elements.  
We use @Directive meta-data annotation to register directives.  
Directives in angular are used to design reusable components.  
Directives do not have their own views.

#### What Are Event Emitters in Angular 2/4?

➔ If the change in one of the child components needs to be reflected to any of its parent component, we can emit the event by using Event Emitter API in Angular.  
EventEmitter is class present in @angular/core module, which is used by components and directives to emit custom events.

#### What is dependency injection in angular 2/4?

➔ In Angular 2/4, dependency injection gives us the ability to add functionality of a component at runtime.  
It also provides us a way to create a service that is a reusable piece of code, which can be used across our application to perform a particular function.

#### What are typings in Angular 2/4?

➔ Typings is a way of installing the Typescript definitions using **typings.json** file, as it is necessary as the browser does not understand typescript natively and hence it must be transpiled first before rendering.

#### What are module loaders in Angular 2/4?

➔ Module loaders in Angular are used to bundle different modules that contain their dependencies along with angular components into one bundle or multiple bundles (For lazy loading) and load them in the browser.

#### What is the use of systemjs? How is webpack better to use in Angular 2/4?

➔ Systemjs is a client side module bundler in angular as it loads modules (components and other files) on demand instead of loading an entire application at startup.  
This largely reduces load times while starting up the app.  
The up side of Webpack over Systemjs is that it bundles and creates a single file called bundle.js, which contains HTML, CSS and JS etc.  
While the initial load time might take a few seconds once the app is cached it becomes lightning fast and will lead to a large boost in performance.

#### What are services in Angular 2/4?

➔ In Angular 2/4 services are reusable function which include their properties used to perform some common functionality which can be used by different components instead of writing the same code again and again for different components a data service can be used by multiple components (via dependency injection).

#### How to enable lazy loading in Angular 2/4?

➔ Steps to enable lazy loading in Angular 2/4 are as follows:  
Add loadChildren in place of components in the routes in the exported class in the Main Routing module of your application.  
Now change forRoot to forChild in the module or feature level router config module and you are done.

#### Is it good to use JQuery in Angular 2/4?

Angular 2/4 and JQuery at its core are very different concept all together.  
JQuery deals with manipulating DOM directly while Angular is used primarily for binding data.  
In some scenarios, using JQuery libraries can be used as quick solutions but doing this we might face problems in the future in terms of pre-rendering.

#### How can you use JQuery in Angular 2/4?

➔ Steps to add and use jQuery in Angular 2/4 are as follows:  
In your node.js terminal type **npm install --save jQuery** and hit enter.  
Then install jQuery Declaration files by typing **npm install -D @types/jquery** and hitting enter.  
Next Import into your component and then provide a typed reference for jQuery.

#### What is viewchild?

➔ @ViewChild decorator is used when a parent component needs to communicate with and pass data to the child component.  
We need to pass the class name of the child component to @Viewchild decorator function.

#### How to share global data across components?

➔ In Angular 2/4, services are used to communicate data between different components.  
As one component could pass, the data to the service while another component could read from it.  
Therefore, this data service could pass the data from one components to multiple components.