**Angular Interview questions part 2**

**Question: How Does Angular Router Work?**

A : When a user navigates to a page, **Angular Router** performs the following steps in order: it reads the browser URL the user wants to navigate to. it applies a URL redirect (if one is defined) it figures out which **router** state corresponds to the URL.

### Question: What Are the Router Navigation Events?

**Answer:** Router navigation events help track the lifecycle of a route. These are –

* NavigationStart,
* RouteConfigLoadStart,
* RouteConfigLoadEnd,
* RoutesRecognized,
* GuardsCheckStart,
* ChildActivationStart,
* ActivationStart,
* GuardsCheckEnd,
* ResolveStart,
* ResolveEnd,
* ActivationEnd
* ChildActivationEnd
* NavigationEnd,
* NavigationCancel,
* NavigationError
* Scroll

### Question: Is the Routing Module Mandatory for an Application?

**Answer:** No, routing module can be totally skipped if there are simple configurations.

### Question: What Is a Wildcard Route?

**Answer:** Wildcard route has the path that consists of two asterisks (\*\*) that can match any URL. It is helpful when a URL doesn’t match any of the predefined routes. Instead of throwing error, we can use a wildcard route and defining a component for the same.

### Question: What Are the Special Transition States?

**Answer:** Special transition states are wildcard (\*) and void. Wildcard matches any animation state. The void state is used to configure transitions for elements entering or leaving a page.

### Question: Explain the Lifecycle Hooks

**Answer:** The set of processes that Angular goes through from initiation through end together are called as lifecycle hooks.

|  |  |
| --- | --- |
| ngOnChanges | This method is called when the value of a data-bound property changes |
| ngOnInit | This is called whenever the initialization of the directive/component happens. |
| ngDoCheck | This method is for detecting and taking action on changes that Angular won't detect on its own. |
| ngAfterContentInit | This is called in response after Angular projects any external content into the component's view. |
| ngAfterContentChecked | This is called in response after Angular checks the content projected into the component. |
| ngAfterViewInit | This is called in response after Angular initializes the component's views and child views. |
| ngAfterViewChecked | This is called in response after Angular checks the component's views and child views. |
| ngOnDestroy | This is the clean-up done before Angular destroys the directive/component. |

### Question: How Are Animations Done in Angular?

**Answer:** To use the animation module, it has to be enabled. For this, the BrowserAnimationModule has to be imported.

import { BrowserAnimationsModule } from '@angular/platform-browser/animations'; After this, import the required animation functions into the component files. Example, import { state, animate, transition, // ... } from '@angular/animations';

Next, add the animation metadata property within the @Component() decorator in the component file.

@Component({ selector: 'app-root', templateUrl: 'app.component.html', animations: [ // animation triggers go here ] })

### Question: How Can You Disable All the Animations in Angular?

**Answer:** To disable all the animations, place the @.disabled host binding on the topmost Angular component.

### Question: How Can we reuse the Animations in Angular?

**Answer:** Follow the below steps

<https://angular.io/guide/reusable-animations>

### Question: Mention Some of the Functions That Help Control Complex Animation Sequences

Answer:

|  |  |
| --- | --- |
| query() | finds one or more inner HTML elements within the current element being animated in the sequence |
| stagger() | applies a cascading delay (timing gap) after each animation |
| group() | runs multiple animation steps in parallel. |
| sequence() | runs animation steps one after another (sequentially) |

### Question: Explain the Features of Forms in Angular.

**Answer:**There are two approaches to handle form data (user inputs) – reactive and template-driven.

Reactive forms can be used when you are using reactive patterns in your application and forms are a key part of your application. These forms are scalable, robust and testable.

Template-driven forms are used to add simple forms, for example, a sign-up page. These are not as scalable as reactive forms and should be used only if your form requirements are simple and minimal.

### Question: How Is Metadata Represented in Angular?

**Answer:** Metadata is represented using decorators like class decorators, property decorators, method decorators, property decorators. Example, @Component, @NgModule etc…

### Question: What Are Class Decorators in Angular?

**Answer:**Class decorator contains the metadata of the suitable class type. It appears just before the class definition and declares the class to be of a certain type. Some class decorators are — @Component, @NgModule, @Pipe, @Directive, @Injectable.

### Question: Explain the Difference between Annotations and Decorators in Angular

**Answer:**Annotations are hardcoded features of Angular and store array in it. The compiler creates am attribute of the annotated class and instantiates an object of the same name, passing the metadata to the constructor.

Decorators, on the other hand, are functions that receive the object to be decorated. After receiving, they are free to modify the object in the way it likes. Decorators are implemented by the TypeScript compiler.

### Question: What Is the Difference Between Class Decorators and Class Field Decorators?

**Answer:**Class decorators appear just before class definition, whereas class field decorators appear just before a field in the class definition.

Examples of class decorators are @Component, @NgModule etc… Examples of a class field decorator are @Input, @Output etc…

### Question: What Is Package.json? Explain its Purpose

**Answer:**With json package, it becomes easy to manage the project dependencies. We can mention details like the version, language etc… in package.json. For example, if typescript is used in our project, we can mention typescript and its version in package.json. Examples are metadata.json, tsconfig.json etc…

### Question: What Is a Bootstrapping Module in Angular?

**Answer:** The root module that you bootstrap to launch the application is called as a bootstrapping module. Every Angular application has a bootstrapping module. It is also called as the AppModule. The bootstrapping module is mentioned in the AppModule class.

@NgModule({ declarations: [ AppComponent ], imports: [ BrowserModule, FormsModule, HttpClientModule ], providers: [], bootstrap: [AppComponent] }) export class AppModule { }

### Question: Differences Between Promise and Observable

Answer:

|  |  |
| --- | --- |
| **Promise** | **Observable** |
| return a single value | work with multiple values over time |
| not cancellable | cancellable |
| more readable code with try/catch and async/await | support map, filter, reduce and similar operators |
|  | use Reactive Extensions (RxJS) |
| Executes immediately as soon as created | Executes only when the subscription starts |
| Used with .then() clause | Has chaining and subscription to handle complex applications |
| Errors are pushed to child promises | Centralized and predictable error handling by the use of subscribe() method |
| Provides only one value | Can provide multiple values over time |

### Question: What Is the Single Page Application? How Is It Different From Traditional Web Technology?

**Answer:** In a single page application (SPA), only the home page (index.html) is maintained throughout even though the URL keeps on changing. It is faster and easier to implement when compared with traditional web technology. In traditional technology, every time a user makes a request, the request is passed on to the server. This takes more time.

### Question: What Are the Different Types of Compilations in Angular?

**Answer:** Two types of compilations – AOT (Ahead-of-Time) and JIT (Just-in-Time).

### Question: List the Differences Between Just-In-Time (JIT) Compilation and Ahead-Of-Time (AOT) Compilation

**Answer:** With JIT, the compilation happens during run-time in the browser. It is the default way used by Angular. The commands used for JIT compilation are –

ng build ng serve

In AOT compilation, the compiler compiles the code during the build itself. The CLI command for aot compilation is -

ng build --aot ng server –aot

AOT is more suitable for the production environment whereas JIT is much suited for local development.

### Question: Which One Is Better out of AOT and JIT?

**Answer:** AOT reduces the load and bootstrap time of the application. The pages also load faster. Any errors are also shown during the time of application build itself. Hence, AOT is a better option.

### Question: What Do You Know About the NPM Package?

**Answer:** The components, framework and CLI used by Angular applications are packaged as npm packages. Npm packages can be downloaded using the npm CLI client.