Angular Interview questions part 3

Question:- How to implement SPA in Angular ?

Answer: Perform below steps

1. Install single-spa-angular. Using below command

npm install --save single-spa-angular

1. Generate a main.single-spa.ts in your project src/.
2. Generate single-spa-props.ts in src/single-spa/
3. Generate asset-url.ts in src/single-spa/
4. Generate an EmptyRouteComponent in src/app/empty-route/, to be used in app-routing.module.ts.
5. Add an npm script npm run build:single-spa.
6. Add an npm script npm run serve:single-spa.
7. For Angular 7 only, create a new entry in the project's architect called single-spa, which is a preconfigured Angular Builder.

Question:- How to implement routing in Angular ?

Answer: Follow below steps

1. We can user angular CLI to generate it by using below command.

ng generate module app-routing --flat --module=app

1. Now in app.routing.module.ts add below statements
   1. import { [RouterModule](https://angular.io/api/router/RouterModule), [Routes](https://angular.io/api/router/Routes-0) } from '@angular/router';
   2. const routes: [Routes](https://angular.io/api/router/Routes-0) = [ { path: 'heroes', component: HeroesComponent } ];
   3. @[NgModule](https://angular.io/api/core/NgModule)({ imports: [RouterModule.forRoot(routes)],

exports: [[RouterModule](https://angular.io/api/router/RouterModule)] })

Question no: 17 :- Explain Lazy Loading ?

Answer: Lazy loading provides features like loading of the required component and not loading the entire application at once.

Lazy loading is a technique in the Angular framework that allows you to load JavaScript components asynchronously when a specific route is activated. Generally all the component are loaded at the starting of the app and making it slow.

With Lazy loading, you can only load the required component at the starting of the angular app and call further components when their specific routes are called. This can add some initial performance during the initial load, especially if you have many components with complex routing.

## Question : Advantage of Lazy Loading in Angular?

## Answer:

1. High performance in bootstrap time on initial load.
2. Modules are grouped according to their functionality.
3. Smaller code bundles to download on initial load.
4. Activate/download a code module by navigating to a route.

Question no: 18 :- How to implement Lazy Loading in Angular ?

Question no: 19 :- Define Services ?

Answer: Angular services are single objects that normally get instantiated only once during the lifetime of the Angular application. This Angular service maintains data throughout the life of an application. It means data does not get replaced or refreshed and is available all the time.

Question no: 23 :- Whats the benefit of Depedency Injection ?

Question no: 24 :- Differentiate between ng serve and ng build ?

Question no: 25 :- Explain the --prod parameter in ng build ?

Questions No:- 26 :- Explain ViewChild and ViewChildren?

Questions No:- 27 :- Why do we need Template reference variables?

Questions No:- 28 :- What is ContentProjection?

Questions No:- 29 :- Explain Content projection Slot?

Questions No:- 30 :- What is ContentChild and ContentChildren?

Questions No:- 31 :- ViewChild vs ViewChildren vs ContentChild vs ContentrChildren?

Questions No:- 32 :- Explain the importance of Component life cycle ?

Questions No:- 33 :- Explain events and sequence of component life cycle ?

Questions No:- 34 :- Constructor vs ngOnInit() ?

Questions No:- 35 :- How to make HTTP calls using Angular ?

Questions No:- 36 :- What is the need of Subscribe function ?

Questions No:- 37 :- How to handle errors when HTTP fails ?

Questions No:- 38 :- How to pass data between components ?

Questions No:- 39 :- Explain importance of input, output & event emitters ?

Questions No:- 40 :- How to pass during routing ?

Questions No:- 41 :- Is it a good practice to pass data using services ?

Question no: 45 :- Whats the full form of RxJs?

Question no: 46 :- What is the purpose of RxJs?

Question no: 47 :- What are observables and observers?

Question no: 48 :- Explain the use of Subscribe with sample code.

Question no: 49 :- How to unsbscribe in RxJs?

Question no: 50 :- Explain concept of operators with sample code.

Question no: 51 :- How to install RxJs?

Question no: 52 :- Differentiate between promise and RxJs?

Question no: 53 :- In Angular where have you used RxJs?

Question no: 54 :- Which operators have you used from RxJs?

Question no: 55 :- What is Push/reactive vs Pull/Imperative?