

1. Explain the Installation process of Angular 8.

Angular 8 Installation :-

Angular 8 CLI installation is based on every simple steps. It will take not more than 5 minutes to install.

NPM is used to install Angular 8 CLI. Node.js is installed, NPM, is also installed. If you want verify it, type the following command.

npm -v

You could see the version below:

6.94.4

Let's install Angular 8 CLI using NPM as follow.

npm install -g @angular/cli@18.0.0

To verify, Angular 8 is properly installed on your machine, type the below command:

ng version.

Node: 14.2.0

OS: Win32 x64

Angular:

.....

Package	Version
@angular-devkit/architect	0.803.26
@angular-devkit/core	8.3.26
@angular-devkit/schematics	8.3.26
@schematics/angular	8.3.26
@schematics/update	0.803.26
vxjs	6.4.0

- Explain various Angular version briefly.

Angular versions:-

Angular JS:

Angular JS is very powerful Javascript framework. It was released in October 2010.

Angular JS based on Model View Controller (MVC) architecture and automatically handles Javascript code suitable for each browser.

Angular 2.0:-

Angular 2.0 was released in September 2016. It is re-engineered and rewritten version of Angular JS. Angular JS had a focus on controllers but version 2 has changed focus on components. Components are the main building blocks of applications.

Angular 4.0:-

Angular 4.0 was released in march 2017. It is updated to typescript 2.2, supports ng if else conditions whereas Angular 2 supported only if conditions.

Angular 4.0 introduces animation packages, HTTP search parameters and finally angular 4 applications are smaller and faster.

Angular 5.0:-

Angular 5.0 was released in November 2017. It supported some of the salient features such as HttpClient API, lambda support, improved Compiler and "build" optimizer.

Angular 6.0:-

Angular 6.0 was released in May 2018. Features added to this version are updated Angular CLI, updated CDK, updated Angular material, multiple validators and JS library.

Angular 7.0 :-

- Angular 7.0 was released in October 2018.
- Some of salient features are google supported community, POJO based development, modular structure, declarative user interface and modular structure.

Angular 8.0 :-

Angular 8 comes up with the following new attractive features:

- Bazel support

- Lazy loading

- Differential loading

- Web Worker

3. Define template and Explain how to attach a template to angular application.

Template :-
Template is basically a superset of HTML.
Template includes all the features of HTML and provides additional functionality to bind the component data into the HTML and to dynamically generate DOM elements.

Used to bind the data from the component to the template.

Here, title is a property in App component and it is bind to template using interpolation.

Directives used to include a logic as well as enable creation of complex HTML DOM elements.

<P *ngIf="conshow">

This section will be shown only when the "conshow" property value in the corresponding components is true.

</P>

<P [showToolTip] = 'tips'>

Here `ngIf` and `showToolTip` are directives,

`ngIf` create the Paragraph DOM element
only when `canShow` is true

Similarly `showToolTip` is Attribute derivatives,
which adds the tooltip functionality to the paragraph
element.

4. Explain various types of one way binding
technique.

One way Binding:

One way data Binding is a one-way interaction
between component and its template

String Interpolation:

In general, string interpolation is the process
of formatting or manipulating strings. In
Angular, interpolation is used to display data
from component to view (DOM).

It is denoted by the expression of `{}{}`
and also known as mustache syntax.

Add below code test-components.ts file as
follows:

```
export class TestComponent implements OnInit  
  {  
    appname = "My first app in Angular 8";  
  }
```

Move To test-component.html file and below code:
`<h1> {{appname}} </h1>`.

Event binding :-

Events are actions like mouse, click, double click, hover or any keyboard and mouse actions.

It is denoted by either parenthesis () or on component to view binding.

Add the following code in test-component.ts files as follows:

```
export class TestComponent {
  showData ($event: any) {
    console.log("button is clicked");
    if ($event) {
      console.log($event.target);
      console.log($event.target.value);
    }
  }
}
```

We have two approaches to call the component method to view:-

(test-component.html) • First one is defined below:

```
<h2> Event Binding </h2>
<button (click)="showData($event)"> Click here </button>
```

Property Binding :-

Property Binding is used to bind the data from property of a component to DOM elements. It is denoted by [].

Add the below code in test-component.html

```
export class TestComponent {
  UserName : string = "Peter";
}
```

Add the below code in test.component.ts file

```
<input type="text" [value] = "user Name">
```

Attribute Binding:

Attribute binding is used to bind the data from component to HTML attributes. The syntax is as follows:

```
<HTML Tag [attr.name] = "Component data">
```

For eg:

```
<td [attr.colspan] = "column Span"> ... </td>
```

Add the below code in test.component.ts file

```
export class TestComponent {
}
```

```
  UserName : string = "Peter";
```

}

Add the below changes in view test.component.html.

```
<input type = "text" [value] = "user name">
```

Q) Explain reactive programming briefly? - Considerable operators.

Reactive Programming:

Reactive programming is a programming paradigm dealing with datastreams and the propagation of changes. Data streams may be dynamic or static.

Reactive programming enables the datastream to be emitted from source called observable and emitted data stream to be caught by other source called observer through a process called subscription.

Javascript does not have the built-in functions

For Reactive Prog. Angular uses Rxjs library extensively to do below mentioned advanced concepts:

- Data transfer - between components
- HTTP client
- Router
- Reactive forms.

Observable:

Observables are data sources and they may be static or dynamic. Rxjs provides lot of method to create observable from common Javascript objects.

const numbers = of(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);

Subscribing process:

Subscribing to an observable is quite easy. Every observable object will have method

~~subscription~~ pi. Subscribe for the subscription process.

Observer need to implement three call back function to subscribe.

- next - Receive and process the value emitted from the observable
- error - Error handling call back.
- complete - call back function called when all data from observable are emitted.

Operations:-

Rxjs library provide some of the operators to process the data stream.

Filter: Enable the filter the data stream using callback function.

```
const filterFn = filter((num : number) => num > 5);
const filteredNumbers$ = filterFn(numbers$);
filteredNumbers$.subscribe((num : number) =>
  this.filteredNumbers.push(num));
this.vol2 += num});
```

map: Enables to map the data stream using callback function and to change the data stream itself.

```
const mapFn = map((num : number) => num * num);
const mappedNumbers$ = mapFn(numbers$)
```

Pipe: Enable two or more operators to be combined.

```
const filterFn = filter((num : number) => num > 5);
```

```
const filteredNumbers$ = filterFn(numbers$);
filteredNumbers$.subscribe((num: number) =>
  this.filteredNumbers.push(num);
  this.value += num));

```

- 2) Explain Http client programming and briefly discuss expense Rest API.

Http client programming is a must needed feature in every modern web applications. Now a days, lot of applications exposes their functionality through Rest API.

Angular provides a separate module, HttpClient module and a service.

Http client to do HTTP Programming

Expense Rest API:-

The Prerequisite to do HTTP programming is the basic knowledge of HTTP protocol and REST API technique.

HTTP Programming involves two part, server and client.

Angular provides support to create client side application.

Express a popular web framework provides support to create sever side application.

Open a command prompt and create a new folder, express -rest -api

cd / go / to / workspace
mkdir express - rest - api
cd expense - rest - api
- Initialise a new node application using below command:
npm init.

3. Explain the process of configure routing create and accessing routes with examples.

Configure routing:

Angular CLI provides complete support to setup routing during the application creation process as well as during working on application.

Let us create a new application with routes enabled using below command ng new routing app --routing.

Angular CLI generate a new module, APP Module for routing purpose. The generate code as follows:

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
const routes: Routes = [ ];
```

```
@NgModule({})
```

```
imports: [RouterModule.forRoot(routes)],
```

```
exports: [RouterModule].
```

```
}
```

```
export class AppRoutingModule {}
```

Creating routes:

Creating a route is simple and easy.
The basic information to create a route is given below:

- Target component to be called.
- The path to access the target component.

The code to create a simple route is mentioned below:

```
const routes = Route = [
```

```
  {path: 'about', Component: AboutComponent}
```

```
];
```

Accessing routes:

Accessing the route is a two step process.

Include router-outlet tag in the root component template

```
<router-outlet></router-outlet>
```

Use routerLink and routerLinkActive property in the required place.

```
<a routerLink="/about" routerLinkActive="active">First  
component</a>
```

• RouterLink set the route to be called using the path.

• routerLinkActive set the css class to be used when the route is activated.

4) Explain how to figure configure forms, create forms.

Configure Forms :-

To enable template driven forms, let first we need to import forms module in app.module.ts . It is given below:

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import {FormsModule} from '@angular/forms';
imports: [
  BrowserModule,
  AppRoutingModule,
  FormsModule]
```

Once forms module is imported, the application will be ready for programming.

Create simple form

Let us create a sample application (template-form-app) in Angular 8 to learn the template driven form

Open command prompt and create new Angular application using below

command : cd/go/to/workspace

ng new template-form-app

cd template-form-app

Configure Forms Module in App Component as shown below:

```
import {FormsModule} from '@angular/forms';
```

```
@NgModule({
```

```
  declarations: [
```

```
    AppComponent,
```

```
    TestComponent
```

```
],
```

```
  imports: [
```

```
    BrowserModule,
```

```
    FormsModule
```

```
],
```

```
  providers: [
```

```
    bootstrap: [AppComponent].
```

```
]
```

```
  export class AppModule {}
```

Create a test component using Angular CLI as mentioned below:

```
ng generate component test.
```