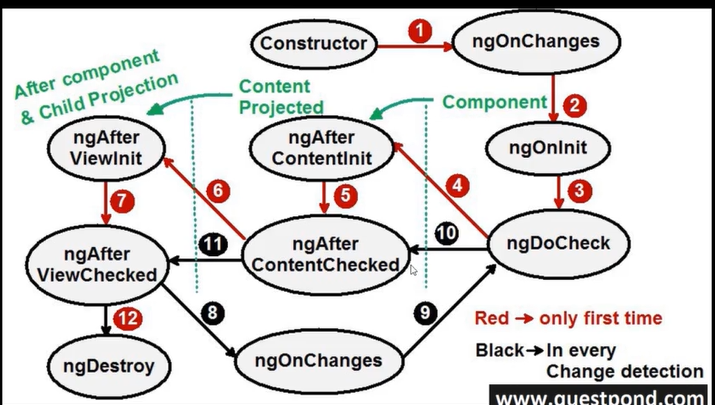
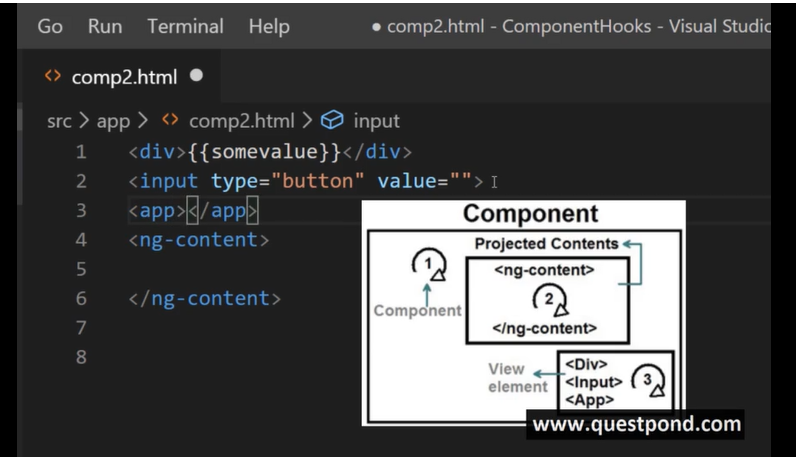
Angular QA Notes

21 May 2024

14:13

* What is angular?
  + Angular is framework and single page application.
  + JavaScript binding framework which binds HTML UI and java script model.
  + Features : Http, Dependency injection, Routing
* Diff b/w angular js vs angular
  + Angular js 1.x is javascript
    - Javascript
    - Controller
    - Not lazy loading
    - No CLI
  + Angular 2,to 17 framework
    - Type script
    - Component
    - Lazy loading
    - CLI
* What are directives in angular ?
  + Angular syntax inside html behaviour change
    - [(ngModel)] , {{value}}, [hidden]
* Types of angular directives?
  + Structural
    - Change the structure of the DOM elements (\*ngFor)
  + Attribute
    - [hidden]="hide()"
    - Change behaviour of html(colour, visibility)
  + Component
    - Directives with template. Its like a user control.
* Npm and Node\_modules folder?
  + NPM: Node package manager
    - Its help us to install the packages
  + Node\_modules
    - That installed packages are available on this folder
* Importance of package. jSon?
  + It’s a file.it hold a project related packages.
* What is type script?
  + Its strongly typed. Do avoid the error.
  + Super set of java script.
  + Its provide the oops concepts
* Angular CLI?
  + Its provide the readymade template project files.
* What is component?
  + Component is mediator component vs html
* Decorator in angular?
  + @Component is one decorator its when component create it will come on component's file.
  + @NgModule
* What is annotation or metadata? Ans is above question
* What is a template?
  + Html view of angular
  + Two way of binding:
    - In component :
      * We can provide the templateurl
      * And we can write html code using template :
* Types of bindings in angular?
  + View and component communicate each other.
  + Expression: if for {{}}
  + Property binding: [(ngmodel)]="username" data flows component to view
  + Event binding:( click) => view to component
  + Two ways binding : data flows from component to view vice versa.
* Architecture of angular?
  + Template(view)
  + Component
  + Module(collection of component)
  + Bindings {{}},[()],()
  + Directives (interpolation, event)
  + Service(common logic)
  + Dependency injection(service inject on component)
* Explain SPA:
  + Single page application
  + Common ui load once, other Ui load based on user selection
* Routing :
  + Navigation off application
* Routing working:
  + Routing.ts => routes
  + {path:'login', component: "logincomonent"}
  + <router-outlet> </router-outlet> we can write this code where the page UI change based on URL
* Lazy Loading in angular?
  + The loading what is necessary
* How to implement lazy loading?
  + Divide in our project as different module.
* Service in angular?
  + Share common functionality for all our modules.
  + Validation , logging, http
* Dependency injection:
  + We can use the provide attribute for to inject the service.
  + App.module.ts => provider
  + {provider : baseclass , userclass: httplogger}
  + decoupling the class dependencies, and when you add new dependencies changes required only one places.
* Ng serve: doing dev ng serve is good , its build onin-memory
* Ng build: build on hard disk
* What does --prod param in ng build?
  + Ng build --prod
  + In above comment remove unwanted code and create minify files.
* Explain view child vs view children ?
  + View child: references one object(individual elements)
    - help us to reference view objects in the component which is connected.
    - <div #div></div>
  + View children: reference the collection
    - <com2><p>hai</p> <p>hello</p> <com2>
* Why template reference variable in angular?
  + #div1
  + {{div1.textcontext}}
  + Refer the DOM elements, angular components
* Explain content Projection?
  + In normal scenario child component html are showing but this content projection whatever we want we can show them using <ng-content> tag.
  + parent
    - <div>
    - <child>
      * <p> this is child component content</p>
    - </chid>
    - </div>
  + Child:
    - <ng-content></ng-content>
* What content project slot in angular?
  + If child component have multiple <ng-content> tag we can use slots in parent
  + Parent:
    - <child> <p slot1><hello slots </child>
  + Eg: child:
    - <ng-content select="slot1"></ng-content>
* Contents child : access single content on child from parent
* content children: access collection content child from parent
* **View child & view children**: help us to reference view elements which belongs to **his own view**.
* C**ontents child & content children** : help us to reference view elements which is **projected by the paren**t.
* **Components of Life cycle:**
  + **Importance of component of life cycle:**
  + 
  + ctor:when obj created its called
  + ngOnChanges: if any value changes on input
  + ngOnInit: when data bound and display the value
  + ngDoCheck: when angular changes detection check
  + ngAfter content init: its more related to content projection
  + ngAfter content checked:
  + Ngafterview Init: Child view
  + Ngafterviewchecked:

* 
* Constructor: its typescript concept
* ngOnint: angular concept
* What kind on code write ng oninit and ctor?
  + Constructor :
    - used to initialize the variables and do dependency injection
    - Dom not Initialized
  + Ng oninit:
    - after Ui is bind, we have the access to Dom elements
* How to make http call in angular?
  + Httpclient import from angular /common/http
  + Create object on http client or dependency injection
  + Import http module in ap.module.ts
  + Post=> url, data,
    - Subscripe=> success ,error
* How to handle success and failed?
  + Using subscribe function we can use success and error methods
* How to data b/w components?
  + Parent child => input/output/event emitter
  + View chid can use to refer UI and pass data.
* Navigating from one url to another url:
  + Pass data using query params
* What is need Angular Pipes?
  + Pipe help you to transform data on angular UI expression
  + {word | **uppercase**}}
* Inbuild pipes?
  + Async pipe
* How to create custom pipe?
  + Class Implements Pipetransform
  + Write the logic on transfrom method => syntax look like extension method
* **Rxjs**
  + What is fullform RxJs:
    - Reactive extension for java script.
  + Why do we need Rxjs:
    - To handle the asyc data stream
    - Data come into within seconds/some time to handle.
  + What are observables and observer?
    - observable => async data
    - Observer=> listener
    - Both are Rxjs objects
* What is importance of subscribe method in oservable?
  + **Import observable from rxjs**
  + Attach the listener to observable using subscribe

* How to un subscripe?
  + Get the object of observable subscribtion and un subscribe.
* What are operators in Rxjs?
  + Operator create one more observable to filter the data using pipe.
  + Chain of logic
* rxJx Operator?
  + Map=> transform the data into different format
  + Filter => filter the data like where condition
  + Merge => combine multiple observables into one
* **Promise vs observable:**
  + Observable
    - Return stream of data
    - Un subscribe the stream
  + Promise:
    - Return single value
    - You cannot cancel a promise
* **Observable used maximum of time http call:**
* **Interceptor:**
  + To execute pre-processing logic before any http call is made from angular application
  + Ng g inercept --skip-tests
  + Maximum we are using set bearer token set header
  + In app.modul.ts level we can add this interceptor under "provider"
* Interceptor use cases:
  + Authentication , logging , caching, URL transformation, modifying header
* Can we provide multiple interceptor?
  + Yes its possible , we can add that interceptor on provider section under app.module.ts.
* **Validation:**
  + **Type of validation:**
    - Template driven form=> html part of angular
    - Reactive form
  + **Template Drive from:**
    - Validation inside the template (name ngmodel required)
    - Its more declarative
    - Its easy to write
    - Difficult to write unit test
  + Reactive from:
    - Written programmatically in ts file
    - Its more imperative
    - It take more control dynamic validation we can add easy
    - Unit test easy
  + Template reference variable:
    - Help us to access DOM elements inside our angular template
  + Template structure for template driven form:
    - Form group=> form control=> validation

||

* Form tag => input=> required
* We can use form template reference variable
* Reactive Form:
  + Formbuilder,formgroup,validatator
  + Dynamic validation => we can use formarray
* Inbuild validator:
  + Requires,minlength, max length, email.
* Custom validator:
  + Use validatorfn interface from "angular/forms"
* Without form tag we are able to implement the validation
* What is [ngModelOptons]="{standalone:true}"
  + When input under from and not participate on the validation we can use above tag

* **Interview Happy Angular Questions:** [Top 50 Angular Interview Questions](https://www.youtube.com/watch?v=4KBVkQ7b6yk&t=6s)
  + What is Angular?
    - Angular is a component based framework for building structured, scalable and single page application for client side.
  + Angular Advantages:
    - Its simple to build single page application with help of component
    - Oops friendly(to make flexible and structured)
    - Its cross platform and open source
    - Reuseable code(Services)
    - Good for testability
  + Differ b/w angular js vs angular
    - Angular js support javascript angular suport type script and java script
    - Angular JS don't support type Type script
    - Angular Js don't have cli
    - Angular js don't have dependency injection
  + NPM: node package manager
    - Is online repository
    - In angular project => node modules provide the all packages
  + ClI Tool:
    - Command line interface
    - Cli use to initialize and develop angular application
  + What are the components in angular?
    - Menu component , login component , list component
    - Component are the most basic UI building block of an angular app
  + What is selector and template:
    - Selector :used to identify the component
    - Template: templateURL : HTMl view of angular
  + What is module in angular? What is app.module in angular?
    - Modules is place where you can group the component, directives, pipes and service which are related to the application.
  + How an angular loaded and started?
    - Index.html => main.ts=> app.module.ts=> app.component.
  + What is bootstrap module and bootstrap component?
    - Angular application start then the first module launched is the bootstrap module and same as bootstrap component.
    - In which component when run the first to run the application first.
  + What is data binding?
    - Communicate b/w the type script of your component and html code.
    - Output data:
      * String interpolation {{name}}
      * property binding : [property]
    - Input data
      * Event binding (event)="functionname"
    - Both ways
      * Two way data binding :[(ngmodel.name)]="data"
  + String interpolation:(one way data binding)
    - Data pass from component from view
    - Its represent by {{}}
    - Only play with string
  + Property binding:(one way data binding)
    - Its allow Boolean and string
    - It change the Html property
    - <div[innertext]='title' ></div>
  + Event Binding:
    - User action on Ui like button click
    - (click)="onclick()" mouse hover
    - Data pass from html to component
  + Two way data binding:
    - Exchange the data from html to view and view to html
    - [("ngmodel"]="data"
    - Form module include on imports on app.module.ts
  + Directives?
    - Add the additional behaviour to html elements
    - Types:
      * Structural: \*ngIF, \*ngFor,\*ngSwitch => add remove elements on html
      * Attribute: change appearance/behaviour of element
        + [ngclass] =>[ngClass]="classname"
        + [ngstyle] =>[ngStyle]={'backround-color : colorname"}
      * Component =>with own template
  + Decorator?
    - Its store metadata about a class, method or property
    - Metadata=> data that provides information about the data
    - All decorators represent with @ symbol , @compoent @Ngmodule
  + Type of decorators:
    - Class => @NgModule @component @injectable @pipe
    - Property =>@input @output @view child @viewchildern @contenntchild @contentchildren
    - Method =>@hostlistener
    - Parameter=> @inject @self @host @skipself @optional
  + What are pipes ? Types of pipe?
    - Accept input value and return the transformed value
    - types
      * Build-in pipe => lowercase uppercase date percentage currency decimal slice json
      * Custom pipe
    - Examples:
      * {title | uppercase}
      * {123.45 | currency}
      * {123.45 | currency :'INR'} => parametrized pipe
  + What is chaining pipe?
    - Multiple pipes on input
    - {dbo | date | uppercase}
  + Explain services wit example:
    - Service is a typescript class and reusable code in multiple components
    - Ng g service userservicename
    - Add service name in ngmodule => provider section
    - In constructor => we can inject the service
  + Hierarchical dependency injections?
    - In component => ts file under component => provider : service name
  + Provider ?
    - Inject our service name and used on entire application.
  + What is role of @injectable decorator? How to use one service in another service?
    - @injectable is very imp on service class or else its throw the error.
    - Its normal declaration of inject service on component constructor part same way we can use one service to another service.
  + Lifecycle hook in angular?
    - A component from creation to destruction goes through several level stages and these stages are the life cycle hooks.
      * Component initiation
      * Rendering the component to html view
      * Creating the child component
      * Destroying the component
    - Constructor:
    - Ngonchanges : called when input changes
    - Ngonint : called when component creation
    - ngDocheck : when component is creation after we can check the status component
      * Ngaftercontentinit
      * Ngaftercontentchecked
      * Ngafterviewinit
      * Ngafterviewchecked
    - ngonDestroy : component destroy
  + Constructor in angular?
    - Constructor is method in a typescript class
    - Constructor is not part of life cycle
    - It's used to inject the dependencies into the component
  + Ng oninit life cycle ?
    - NgonInit signals the activation of the created component
    - This is second hook and called after ngonchanges
    - Its called only once during the lifecycle
    - By default inside the component
    - Use to perform the business logic
  + What are asynchronous operation?
    - Observable are used to perform the async operation
  + Difference b/w promise and observable
    - Async pass the data we can use both
    - Promise:
      * Once whole data is ready then only it will show data to UI
      * Emit single value at a time.
      * Are not lazy: execute immediately after creation
      * Are not cancellable.
    - Observable:
      * Stream data => continuously processing the data
      * Emit multiple values over a period of the time
      * Are lazy : they are not executed until we subscript to them using subscribe method.
      * Cancellable=> using unsubscribe()
  + What is Rxjs?
    - Rxjs Is father of observable
    - Reactive extension of javascript
    - Observable => stream of data
    - Observer: subscriber
  + What is observable?
    - Stream data to multiple components
    - Import observable from Rxjs library
    - Create observable and emit the data => myobservable = new oabervable();
    - myobservable .subscribe() => receiving and sowing the data.
  + How to get the data from http client using observable?
    - Http client build-in service class in angular.
    - @anular/common/http package