**Day 1 : 17-05-2021**

**HTML/HTML5,CSS/CSS3, JavaScript Using ES5 and ES6**

**TypeScript**

**https://**[**www.google.com**](http://www.google.com)

http/ https : protocol : hyper text transfer protocol. Secure.

Set of rules which help to communicate more than one machine.

req(https/http)-🡪

**Client Server**

**🡨-----Res(http/https) HTML/HTML5**

**CSS/CSS3**

**JS**

**To do any validation or to read/write/update DOM (Document Object Model) all html tags are known as dom elements using JavaScript become more complex.**

**jQuery : jQuery is external library or external JavaScript file. jQuery contains lot of pre-defined function which internally connected to each other.**

**Using jQuery we can read, write and update DOM element dynamically.**

**jQuery is not a standard. Standard design pattern( best practise).**

**On View side we want to Framework.**

**Framework contains lot of pre-defined API which internally connected to each other to perform specific task.**

**Framework is known as template or protocol. But not final product.**

**If we develop any application using framework 60 to 70 task taken care by framework. Hardly we have to write 20 to 30% core to make final product.**

**Framework internally follow all design pattern. Implementation of all design pattern taken care by framework.**

**Angular is a front side or web application framework.**

**Angular 1.x or Angular JS:**

**Html, css, JavaScript using ES5 and ES6**

**Angular Framework 2 to 11**

**Html, css, JavaScript, TypeScript and Node JS.**

**Node JS 12.x to 16.x**

**VS Code**

**Node JS : Node JS is a run time environment for JavaScript library or framework.**

**Like JRE for Java program.**

**Node JS container lot of pre-defined modules. With the help of those modules we can create Server Side Scripting language.**

**HTML/HTML5, CSS/CSS3 and JavaScript -🡪 front end technologies.**

**Java, Python, Asp.net , php, Node JS etc ---🡪 backend technologies**

**Before node js JavaScript is known as Client side scripting language.**

**Node JS contains lot of pre-defined module may be local module or external module.**

**But After node JS Using Java Script we can create Server side programming language, We can create Rest Full Web Service, we can file handling programing, we can connect database may my sql or mongo db etc.**

**MEAN Stack**

**MERN Stack**

**Mongo DB /MySQL Express Module Angular Node JS**

**Mongo DB /MySQL Express Module ReactJS Node JS**

**npm ( Node Package Manager).**

**Using npm we can download the external module or dependencies.**

**Syntax**

**npm install –g moduleName (install globally)**

**Or**

**npm install moduleName (install locally)**

**In Node JS we can’t use document and window objects.**

**TypeScript : TypeScript is a type of scripting language which support all ES6 features.**

**ES : ECMA : European Computer Manufacture Association**

**ECMA is a concept.**

**JavaScript as well as TypeScript are the implementation of ES5 and ES6.**

**Programming is a concept**

**Using C, Java, Python**

**OOPs is a concept**

**C++, Java, Python are the implementation of OOPS language**

**JavaScript is a partial implementation of ES6.**

**TypeScript, it support all ES6 features.**

**Browser doesn’t support typescript program directly we have to convert TypeScript to JavaScript using transpiler. Converting one format to another format.**

**Compiler C**

**Interpreter JavaScript**

**Ts 🡪 JS Then we have to include js file in html page.**

**tsc (Transpiler)**

**TypeScript Features**

**To declare the variable in JavaScript (ES5) we are using var keyword.**

**From ES6 and TypeScript to declare the variable we are using var, let and const keyword.**

**Using var keyword we can to re-declaration same variable with same value or different values.**

**But using let keyword we can’t to re-declaration.**

**In C or Java**

**int a=10;**

**int a=40; error**

**using var we can declare global scope of that variable.**

**Using let we can declare local scope or block scope.**

**const keyword is use to declare the constant variable we can’t change the value.**

**DataType :**

**Typescript provide data type concept.**

**Syntax**

**let/var variableName:dataTypeName;**

**number, string, boolean,object, any etc.**

**let num:number =100;**

**num = “Ravi”; Error**

**array : array is use to store the same type of values.**

**literal style**

**let arrayName:datatype[];**

**generic style**

**let arrayName:Array<dataType>;**

**TypeScript functions**

**JavaScript function Using ES5**

**function add(a,b){**

**}**

**add(10,20);**

**add(“Ajay”, “Kumar”);**

**add(1);**

**add();**

**In ES5 JavaScript only name of the function must be match not mandatory number of parameter as well as type of parameter.**

**In ES5 JavaScript can return any type of value as well as not mandatory to return the value.**

**function info() {**

**//return “Welcome”;**

**//return 100;**

**return true;**

**}**

**Need function parameter which takes 0, 1 or many.**

**Rest operator or parameter is use to receive 0 or 1 or many arguments.**

**Syntax**

**…variableName:datatype[]**

**1,Ramesh,15000**

**2,Ravi,18000,C**

**3,Raju,25000,HTML5,CSS3,JavaScript**

**callback : callback function is use to pass the function body or function name or function itself another function as a parameter.**

**User-defined function**

**Pre-defined function**

**forEach() is a pre-defined function part of array which takes callback function as parameter.**

**OOPs :**

**object : any real world entity . concept**

**properties or state have variables/ fields**

**person**

**behaviour do/does functions / methods**

**bank**

**animal**

**car**

**upto ES5 there no class keyword.**

**function Employee() {**

**}**

**var emp = new Employee();**

**From ES6 and TypeScript we can use class keyword.**

**class : class is know as blue print of object or template of object.**

**constructor : It is a type of special function which help to create the objects.**

1. **To write constructor in class we have to use constructor keyword.**
2. **Constructor get call automatically when we create the object.**

**Constructor short cut initialization**

**Module : Using module we can achieve separation of concern.**

**Module is a collection of variable, function, classes, interfaces or enum etc.**

**Module are divided into different files base upon their functionality.**

**Using export and import we can connect one module to another module.**

**a.ts**

**fun1() {}**

**b.ts**

**fun2() {}**

**main.ts**

**calling fun1 and fun2 functions.**

**Angular Framework 2 to 11**

**Angular is a open source web framework provided by google. It is use to create all type of application like desktop, web as well as mobile application.**

**Angular framework is use to create SPA (Single Page Application).**

**Angular use component which help to control the view or part of the view in web page. Every component work independently. Sometime some component interact with another component depending upon the component relationship.**

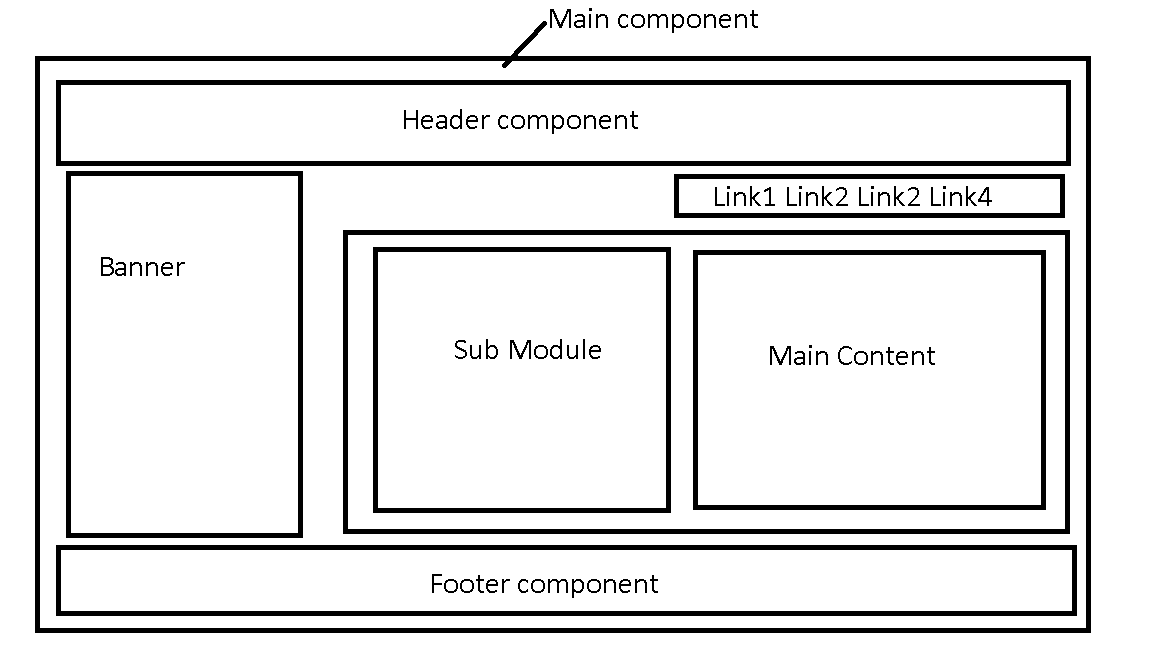
**One.html welcome.html**

**Using hyperlink**

**Using button (submit)**

**Using JavaScript**

**Ajax**

****

**Angular CLI (Command Line Interface). Which help to create the Angular sample project.**

**To crate the sample project it require ng command.**

**ng (next generation for web page).**

**Angular**

**To enable ng command we have to download angular module with the help of npm command.**

**npm install –g @angular/cli**

**ng –-version (it display the angular version details)**

**AnglarProjects folder**

**Please open the vscode or external terminal in AngularProjects**

**ng new project-name**

**Example : ng new welcome-app**

**Then it will ask do you want to routing 🡪 No**

**Styling**

**CSS --🡪 Enter**

**If it ask option security : Y/N**

**After project created successfully**

**Using CD command please move inside a project**

**cd project-name**

**cd welcome-app**

**To run the project we have to run the command as**

**ng serve**

**After project compiled 100%**

**Please open the external browser and type the URL as**

[**http://localhost:4200**](http://localhost:4200) **(default port number is 4200)**

**Open Angular project in VScode**

**and expand the src folder.**

**Then expand the app folder.**

**Please open the app.component.html page**

**Please write the html code which we write in a body tag(don’t write html, head or body tag).**

**Now open the app.component.ts file**

**Decorator : It is a type of special declaration which help to make class, property and function is a type of special function, class or property. It is also known as meta-data. Meta-data data about data.**

**It is like annotation in Java.**

**Decorator is a concept of typescript.**

**@NameOfDecorator**

**@Component**

**@NgModule**

**@Pipe**

**@Input**

**@Output**

**@ViewChild**

**@Injectable**

**Etc**

**Component : Component is use to control the view or part of the view. It is a type of directive. Directive is use to add extra behaviour or functionality to DOM(HTML code).**

**Using Angular we are creating user-defined tags.**

**@Component contains lot of pre-defined attribute**

**selector : “app-root” // it consider as user-defined tags**

**<app-root></app-root>**

**<h1> </h1>**

**<font color=”red”></font>**

**templateUrl : This property is use to connect ts file to html page. In Angular html page is know as template. Because we can use static as well as dynamic contents in that html page.**

**styleUrls : This property is use to link to css file like link tag in normal html page with external css file.**

**Angular Data binding and Angular forms with validation**

**create two new projects.**

**ng new welcome-come**

**ng new angular-data-binding**

**ng new angular-forms**

**Day 2 : 18-05-2021**

**ng serve : This command is use to run the angular project. After executed this command 100% then we have to open the browser and write the url as**

[**http://localhost:4200**](http://localhost:4200)

**or**

**ng server –o : after compiled project 100% it automatically open the default browser.**

**app.module.ts**

**@NgModule is a decorator which help to make the class a angular module.**

**Angular module : It is collection of more than one components.**

**NgModule attribute**

**declaration : in this section we have to provide all component declaration details.**

**import : This attribute is use to hold the details about pre-defined or user-defined modules.**

**provider : This attribute hold details about angular service class.**

**bootstrap : This attribute provide the details about first component to load then angular project start.**

**mainModule (user-defined module)**

**OrderModule CustomerModule LoginModule**

**One or more one or more one or more**

**Component component component**

**main.ts**

**This file provide the main module which going load the then angular project start.**

**index.html**

**we can find selector name provided in component as a user-defined tags.**

**then open angular.json file**

**from here we can get more details like main.js file as well as index.html page.**

**We will create user-defined component as Header component**

**We will create footer component with local template and local css file**

**@Component decorator attribute**

1. **Selector**
2. **templateUrl(external html page) or template (local html code)**
3. **styleUrls(external css page) or styles(local css rules)**

**Angular Data binding**

**ng new angular-data-binding**

**Data Binding : Data Binding provide the bridge between component to template(html page) and vice-versa.**

**Using Data binding we can share the data between component to template and vice-versa.**

**Angular CLI provide the command with the help of those command we can create component**

**ng generate component componentname**

**or**

**ng g c componentname**

**2 types**

1. **1 way data binding**
2. **String interpolation :**

**It is component to view (template/html)**

**Component ---------------🡪View or Template**

**Syntax**

**{{variableName}}**

**Ex : {{fname}}**

**{{5+10}}**

**{{display()}}**

1. **Property binding :**

**Component --------------🡪 Template**

**[]**

**Form tags like textfield**

**Name:**

**<input type=”text” [value]=”variableName”/>**

**Non form tag like p, div, span etc**

**<p [innerText]=”variablename”></p>**

**<img [src]=”imageVariable”/>**

1. **Event binding :**

**Template ----------🡪Component**

**Event : event provide bridge between html to JavaScript**

**onClick**

**onDblclick**

**onMouseOver**

**onMouseOut**

**Angular using same event provided by JavaScript but removed pre-fix from event name and event name wrap with ().**

**Angular event**

**(click)**

**(dblclick)**

**(mouseover)**

**(ngSubmit)**

**Using Event binding and String interpolation we are achieve two way data binding.**

**Event binding : template ---🡪 component**

**String interpolation : component -🡪 template**

**Both combined = 2 way data binding**

**Pass the textfield, password etc value from template to component.**

**Template reference.**

**<input type=”text” #nameRef/>**

**#nameRef : it is use to create the reference of text field which help to retrieve the value in components.**

1. **2 way data binding**

**Two way data binding : if we do any changes in component automatically update on template and vice-versa.**

**Component 🡨-----------🡪 Template**

**To achieve two data binding angular provided ngModel attribute.**

**Syntax**

**[()]**

**<input type=”text” [(ngModel)]=”variableName”/>**

**When we use ngModel attribute we will get the error. Because ngModel attribute is a part of FormsModule. So we have to import FormsModule in app.module.ts in import section to use ngModel attribute to achieve two way data binding.**

**Create new project**

**ng new angular-forms**

**Angular Forms**

**Using Template reference we are passing the value of textfield or other field to components.**

**Using Angular forms we can pass more than one html forms values as a reference to components.**

**2 types of angular forms**

1. **Template Driven Forms**
2. **This type of form is easy to create.**
3. **It suitable for simple forms.**
4. **The flow of the application from template (html) to components**

**Template ----------------🡪 Component**

1. **Model Driven Form or Reactive Forms** 
   1. **This type of form is complex.**
   2. **It suitable for create complex forms.**
   3. **The flow of the application from component to template**

**Component -----------------🡪Template**

**ng new angular-forms**

**You have to create two**

**ng g c tdf-login-page**

**ng g c mdf-login-page**

**Template Driven Form**

**Syntax to create the form reference.**

**<form #loginRef=”ngForm”>**

**<input type=”text” name=”user” ngModel/>**

**</form>**

**ngForm is a pre-defined attribute which help to create the Form reference.**

**#loginRef = “ngForm”**

**ngForm is a pre-defined attribute part of FormsModule. So we have to import FormsModule in app.module.ts in import section.**

**If we write ngModel attribute in textfield or password. The value of textfield and password field pass through reference in the form of JSON.**

**Model Driven Form**

**According Model Driven form every textfield, passwordfield, radiobutton, checkbox, dropdown etc is known as FormControl.**

**FormGroup is a collection of more than one FormControl as well as another FormGroup.**

**FormGroup**

**FirstName-🡪FormControl**

**LastName-🡪FormControl**

**Age--🡪FormControl**

**Address FromGroup**

**City 🡪 FormControl**

**State 🡪FormControl**

**PinCode 🡪FormControl**

**First in Component we have to create FormGroup and FormControl.**

**When we bind FormGroup and FormControl in template side using attribute as formGroup and formControlName we will get the error. Because formGroup and formControlName attribute is a part of ReactiveFormsModule. So we have to import ReactiveFormsModule in app.module.ts in import section file to avoid the error.**

**Angular Validation Using Template Driven Form as well as Model Driven Form**

**Angular provided totally 6 type of classes to do validation using template driven or model driven form.**

**ng-valid : if condition satisfies**

**ng-invalid : if condition doesn’t satisfies**

**ng-touched : if user visit textfield, password it come true**

**ng-untouched : if user doesn’t visit it become true**

**ng-dirty : if user change the value**

**ng-pristine : if use doesn’t change the value.**

**ng new angular-directive**

**Angular Directives**

**Angular Directives is use to add extra behaviour to HTML or DOM elements.**

**3 types of directives**

1. **Components directive : Using component directive we are creating user defined tags.**

**@Component({**

**selector:”my-tag”,**

**template:”<div>Welcome to My App</div>”**

**})**

**export class MyComponent {**

**}**

1. **Structure directive**

**Using structure directive we can add or remove dom element or html tags.**

**\*ngIf**

**\*ngFor**

**Syntax**

**<tagName**

**\*ngFor=”let variableName of arrayName”>**

**{{variableName}}**

**</tagName>**

**Display User-defined array objects using \*ngFor**

1. **Attribute directive**

**Attribute directive is use to apply css styling for angular.**

**ngStyle and ngClass**

**it a style attribute in CSS and class selector in CSS.**

**Bootstrap : Bootstrap is a open source Responsive Web Framework for CSS.**

**Bootstrap provide lot of pre-defined css class with respective html tags.**

**Like**

**Button css classes**

**Form css classes**

**Table css**

**Alter css class**

**So we can add bootstrap features of Angular project lot of ways**

1. **Add bootstrap URL in index.html page.**
2. **We can download bootstrap module using npm command.**

**Create two projects**

**ng new angular-service**

**Angular Service :**

**If we write any business logic in component it may be simple or complex.**

**That logic become local to that component means we can all that code from that component’s template only.**

**Angular service help to provide global business logic for more than one components.**

**Using Angular we can achieve separation of concern.**

**Template -🡪 View (html)**

**Component -🡪Controller (Component)**

**Business Logic -🡪 Model**

**Model view Controller (MVC)**

**Angular Service divided into types**

1. **User-defined service**
2. **User-defined service creating object using new word.**
3. **User-defined service creating object with help to DI (Dependency Injection).**

1. **Pre-defined service**

**User defined service using new word**

**Create two component**

**ng g c first**

**ng g c second**

**Angular create the object of those classes if class is a type of special class with some decorator like @Component or @NgModel etc.**

**If class is normal class we have to create the object as well as we have to maintain the life the of the objects.**

**IOC : Inversion of control**

**DI : In place creating any resources or object explicitly allow to create by container. Pull from container whenever you required. If container create then it create only one object as a singleton design pattern (only one memory). The life of the object maintain by container.**

**Container means engine. It is a part of server.**

**To create Angular Service class with DI concept**

**First we have to create normal class with one or more function.**

**On that class we have to use @Injectable decorator**

**Then we have to provide that class details (register) in app.module.ts file in provider attribute field.**

**In which component you want service class object we have to pull the object in that component with the help of constructor**

**Angular Pre-defined service.**

**Angular Provide pre-defined API ie HttpClient. HttpClient API which help to call backend Restfull web service written any language like Java(Spring boot), Aps.net, Python, Php or Node JS.**

**Web Service**

**Req xml/json req**

**Client sbi hsbc client**

**Res**

**Java .net res**

**Python**

**Node js**

**Web Service : It is use to give the service for web application when both application running using different technologies(language), or may be in different platform.**

**2 types**

1. **SOAP Web Service : only xml (old web service)**
2. **RESTFull Web Service : xml as well as json. As well as any other format according to application requirements.**

**Representational State Transfer**

**SOAP : Simple Object Access Protocol**

**XML and JSON help to pass the data between two technologies.**

**HttpClient provide pre-defined methods like get, post, put and delete which help to call RestFull service written in any language.**

**Get : Get The resources : get employee details, customer details.**

**Post : Create the resource : Store data in database**

**Account Creation, Registration etc.**

**Put : Update the Resources : Update the customer info using id**

**Password update, age update, phone number etc**

**Delete : Delete the Resource : Delete the customer info using id**

**Delete customer order details.**

**Delete employee details.**

**Backend technologies like Java(Spring boot), Asp.net, Php, Python or Node create the application as expose the application as a Rest API or Service in the JSON format.**

**Front end technologies like JavaScript, Angular or React JS consume the data and display in a proper format.**

[**https://jsonplaceholder.typicode.com/todos**](https://jsonplaceholder.typicode.com/todos)

**So We will create fake component and fake service**

**To create the service using command prompt**

**ng g s servicename**

**In service class we have to do the DI for HttpClient API provide by Angular.**

**HttpClient pre-defined API part of HttpClientModule. So we have to import HttpClientModule in app.module.ts file in import section.**

**HttpClient get(), post(), put() and delete() method use rxjs (ReactiveJS) concept to consume and produce the data ie Observable and Subscribe.**

**HttpClient methods return type is Observable.**

**To load the data from Observable we have to call subscribe.**

**Subscribe method take 3 parameter as a callback function.**

**1st parameter is next : which help to load the data one by one.**

**2nd parameter is error: This parameter get call if any error generated at the beginning or middle or at last.**

**3rd parameter completed :This parameter get call after all data loaded successfully.**

**2nd and 3rd parameter are optional.**

**But advisable we have to write 2nd parameter to handle the error.**

**If we want to display the rest api data in template (html). So we have to call subscribe method in components.**

**In Angular to map the json data we have to create the model class.**

**this.http.get(“url”) : retrieve**

**this.http.post(“url”,jsonData); : store**

**this.http.put(“url”,jsonData) : update**

**this.http.delete(“url”) : delete**

**ng new angular-pipe**

**ng new angular-life-cycle**

**RxJS : Reactive JS programming**

**Observable : It represents collection of values or events Or Observable are stream of data or flow data.**

**Subscriber : It use to execute the Observable.**

**First install the rxjs plugin using command as**

**npm install rxjs**

**npm install esm (embedded system module)**

**or**

**npm install rxjs –g**

**npm install esm –g**

import {Observable} from 'rxjs';

let sub = Observable.create((sub)=> {

    sub.next("Hi")

    sub.next("Hello");

    sub.next("How r you")

    sub.complete()

});

sub.subscribe(data=>console.log(data),

err=>console.log(err),

()=>console.log("completed"))

**To run the program**

**npm –r esm MyObservable.js**

**Angular Pipe is use to filter or modified the data in string interpolation.**

**Angular Pre-defined pipes**

**uppercase**

**lowercase**

**currency**

**date**

**json**

**etc**

**Day 4 : 20-05-2021**

1. **ng new product-crud-operation**

**Product CRUD Operation**

**If we want to create REST API end point we have to use backend technologies like Java, python, asp.net, node etc.**

**Node js provide external module which help to run the static json file as a REST API.**

**We have to install json-server external module**

1. **product.json file**

{

    "products":

        [

            {"id":100,"pname":"Tv","price":55000},

            {"id":101,"pname":"Computer","price":42000}

        ]

}

**npm install –g json-server**

**or**

**npm install json-server**

**After installation we have to run the command as**

1. **run the file using json-server**

**json-server product.json**

1. **step create component, service and model**

**Create the component**

**ng g c product**

**Create the service**

**ng g s product**

**create the model typescript class**

**ng g class product --type=model**

1. **open app.module.ts and import the modules as ReactiveFormsModule and HttpClientModule in import section.**
2. **Open product.component.ts file and copy selector and paste in main component html page ie app.component.html**
3. **Open product.model.js file and write three property ie id,pname,price in constructor.**
4. **Open the app.product.ts file and create FormGroup with three FormControl for id,pname and price.**
5. **Open the app.prodcut.html page create form with three fields as id,pname and price with bind with FormGroup and FormController reference.**
6. **After receive the value in component ie id, pname and price**
7. **Now create the DI for ProductService and pass the product object to service class functions.**
8. **In Service class create the DI for HttpClient API.**

**Next operation retrieve the records from product.json file.**

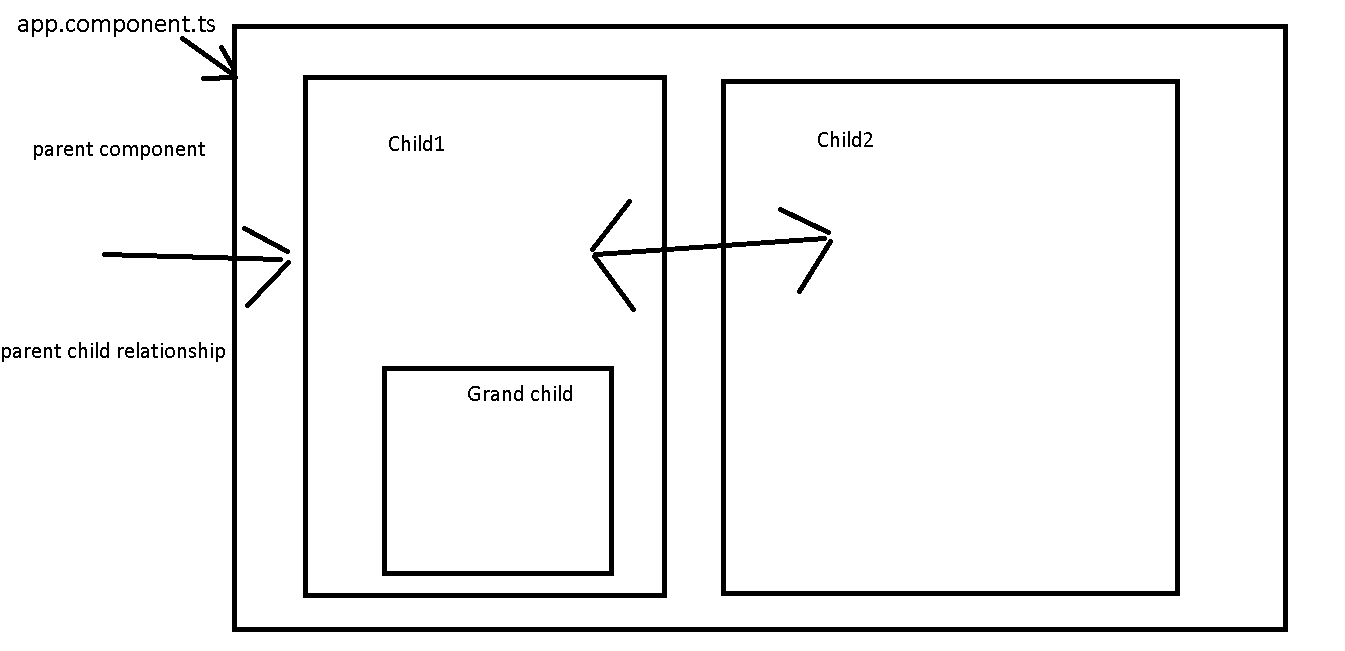
1. **Open the product.service.ts file and write a function which help to call the get method to retrieve json data from a file.**

**Sharing the value between two components**

**ng new component-communication**

**Component Communication**

**Inside one component template if we use another component’s selector then it is known as parent and child relationship.**

****

**If any component contains any property (variable) of any types of number, Boolean, string, array type or object that. That property we can access within that component template only. We can’t access that property in another component template.**

1. **Parent – Child : @Input decorator**
2. **Child – parent : @Output decorator with help of EvenEmitter.**

**EventEmitter is a pre-defined API which help to emit (pass the value) from child component to parent component. So we have to create this reference with @Output decorator.**

1. **Sibling : a. We can take the help of sessionStorage or localStorage part of HTML5 features.**

**Using Shared Service.**

**sessionStorage and localStorage**

**app.component.ts**

**name:string=”Ram”**

**// store the value in session storage**

**sessionStorage.setItem(“key”,”value”);**

**sessionStotage.setItem(“obj”,name);**

**app.child1.ts**

**let obj = sessionStorage.getItem(“key”);**

**app.item.ts**

**let name = sessionStorage.getItem(“obj”);**

**sessionStorage.removeItem(“obj”);**

**app.order.ts**

**let obj = sessionStorage.getItem(“key”);**

**Now create two components**

**ng g c child1**

**ng g c child2**

**app.component -🡪 parent component**

**child1.component 🡪Child**

**child2.component 🡪Child**

**Parent Component add number one by one through text field :**

**Parent component is use to display sumOfEven and sumOfOdd.**

**Child component display those number in proper format and do the calculation sum of even or odd number.**

**Child component is responsible to display number in proper format which receive from parent component and find sumOfEven and sumOfOdd.**

**Day 5 :**

**ng new angular-routing --🡪 Routing yes**

**ng new angular-testing 🡪 No Routing**