**Angular 19 preparation document.**

**HINT: in order to use “ng serve or ng s” in terminal use the command "Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser"**

**What is Component?**

In Angular, a **Component** is a fundamental building block of the application. It controls a portion of the UI (User Interface) and is responsible for managing the data and logic associated with that part of the view. Components are reusable and modular, making it easier to build and maintain complex applications.

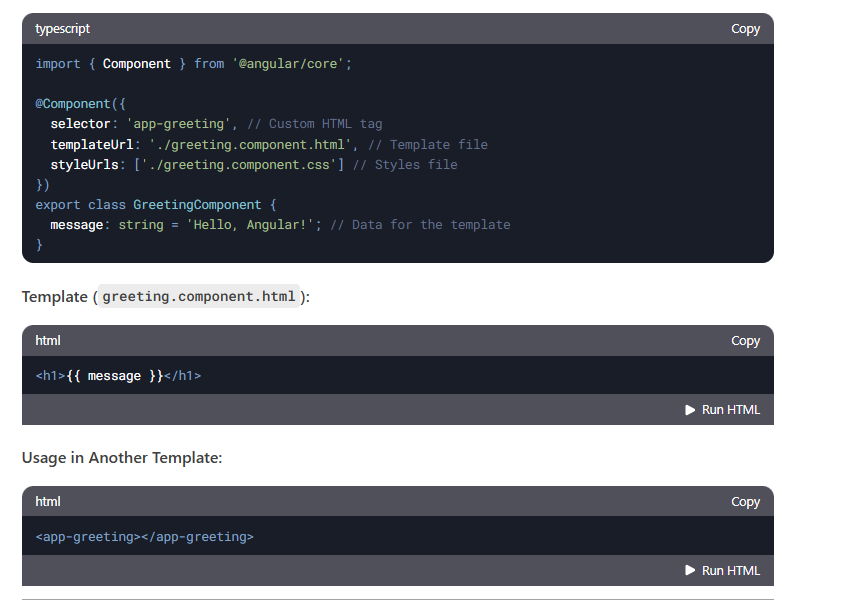
OR

Is a mediator between view and the module.

**Key Features of an Angular Component:**

1. **Template**:
   * The template defines the HTML structure of the component. It determines how the component is rendered in the DOM.
   * Templates can include Angular-specific syntax like data binding, directives, and pipes.
2. **Class**:
   * The class is written in TypeScript and contains the logic and data for the component.
   * It uses the @Component decorator to define metadata about the component (e.g., selector, template, styles).
3. **Metadata**:
   * The @Component decorator provides metadata that tells Angular how to process the component.
   * Metadata includes properties like selector, templateUrl, styleUrls, etc.
4. **Styles**:
   * Styles define the appearance of the component. They can be inline, external, or scoped to the component.
5. **Selector**:
   * The selector is a custom HTML tag (e.g., <app-my-component>) used to embed the component in other templates.

Example of an Angular Component:



**Why Use Components?**

* **Reusability**: Components can be reused across the application.
* **Modularity**: Each component encapsulates its logic, template, and styles.
* **Maintainability**: Breaking the UI into smaller components makes the code easier to manage and test.

**How to create component?**

In order to create component use command   
ng generate component <component name> **(or)** ng g c <component name>

Each component contains below files.

app.component.html--->HTML code with app component should define

app.component.css---> CSS code

app.component.ts--> should import rendering app component

app.component.specs.ts

**What is Component Decorator ?**

The **@Component decorator** is a function that marks a **TypeScript class** as an Angular **component** and provides configuration metadata that determines how the component should be processed, instantiated, and used at runtime.

A screenshot of a computer program

AI-generated content may be incorrect.

**How to Render component ?**

Step 1: Create a Component

Step 2: Use the Component in Another Component (app.html)

Step 3: Ensure the Component is Declared in app.module.ts.

To display UI values from the .html file, they must be defined in the corresponding component (.ts file).

Example:

imports: [DataBindingComponent], in app.component.ts

<app-data-binding></app-data-binding> in app.component.html

**OR**

**create single File component ?**

instead of having html and css, specs.ts and ts we can have only ts file with help of directive.

use template and styles instead of templateURL and styleUrls in Component decorator.



**What is standalone component ?**

component that does not require a module (NgModule) to be declared. . It is a self-contained unit that can be used directly in the application without being registered inside app.module.ts.

ng g c my-component --standalone

**How to Declare Variable / state in Component?**

In Angular, you can declare variables (state) inside the component class using TypeScript properties.

A screenshot of a computer program

AI-generated content may be incorrect.

**What is Data Binding?**

Data Binding in Angular is the mechanism that connects the component's data (TypeScript logic) with the view (HTML template). It ensures that changes in the component reflect in the UI and vice versa.

There are four main types of data binding in Angular:

1. **Interpolation** - {{}} → One-way binding from component to view
2. **Property Binding** - [property] → One-way binding from component to an HTML element
3. **Event Binding** - (event) → One-way binding from UI to the component
4. **Two-Way Binding** - [(ngModel)] → Two-way sync between component and UI

**What is Directive?**

A Directive in Angular is a special type of class that allows you to modify the behavior, appearance, or structure of DOM elements.

Directives are used to extend HTML capabilities and add dynamic behavior to elements in Angular applications.

Hint: Whenever you need to use ngModel, the name attribute is mandatory.

Inside a class, we must declare a variable with a data type so that it can be used for display.

If we need to show data on page load, we must define it in the constructor.

Many events are available, for example: mouseover, dblclick, change, and click.

**What Are Types of Directive ?**

Angular has three types of directives:

1. Component Directives (Most Common) → Directives with a template (@Component)
2. Structural Directives → Modify the DOM structure (\*ngIf, \*ngFor, \*ngSwitch)
3. Attribute Directives → Modify the appearance or behavior of an element ([ngClass], [ngStyle], Custom directives)

To use ngModel, you must import FormsModule in the module file (e.g., app.module.ts) and include it in the imports array: imports: [FormsModule]. Similarly, for \*ngIf, you need to import NgIf, and for \*ngFor, you need to import NgFor.

**How to use ngModel directive?**

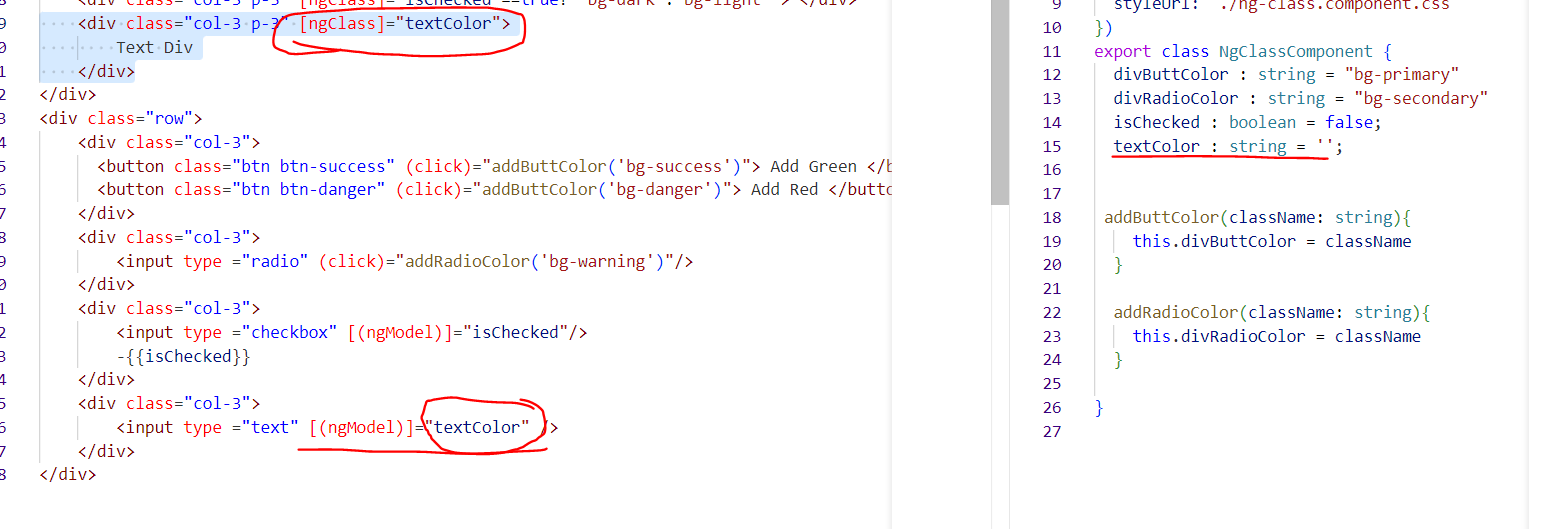
Helps in capturing user input value:

[(ngModel)] = "declaredVariable"

{{ declaredVariable }} → The user input value will be displayed.

[ngClass]="isChecked ==true? ‘’:’’ ">

If you want to assign the value entered in a text box to another field using ngModel, you can do this by updating the second field dynamically in the component.



**What Are Structural Directive ?**

**How to use \*ngIf Directive ? or How to Hide and Show Element in Angular ?**

The \*ngIf directive is used to conditionally display elements based on a boolean condition

A screenshot of a computer

AI-generated content may be incorrect.

**How to use \*ngFor Directive ? or How to Create Dynamic Element in Angular ?**

**Dynamic: dynamic behavior refers to the ability to change content, styles, properties, or data dynamically based on user interactions, API responses, or conditions.**

The \*ngFor directive is used to loop through arrays and dynamically create elements.

A screenshot of a computer program

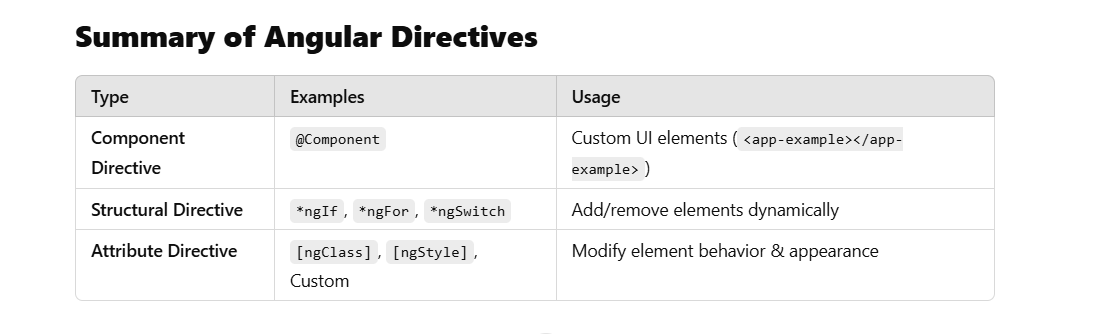
AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.



**Use \*ngIf to conditionally display elements.  
Use \*ngFor to loop through arrays and generate elements.  
Use [hidden] or [ngStyle] to toggle visibility.  
Use Renderer2 for advanced dynamic element creation.**

**In order to use the NgIf directive, structural directives, or attribute directives, we need to import CommonModule or NgIf in component.ts under imports: [NgIf].**

**To hide or show an element, we must use \*ngIf on that element in older Angular versions.**

**NgFor (for loop, iterator) dynamically creates elements.**

**Structural Directives (modify the DOM): \*ngIf, \*ngFor, \*ngSwitch.**

**Attribute Directives (modify the appearance): [ngClass], [ngStyle].**

**Using the [ngClass] directive, we can dynamically assign CSS classes to an element.**

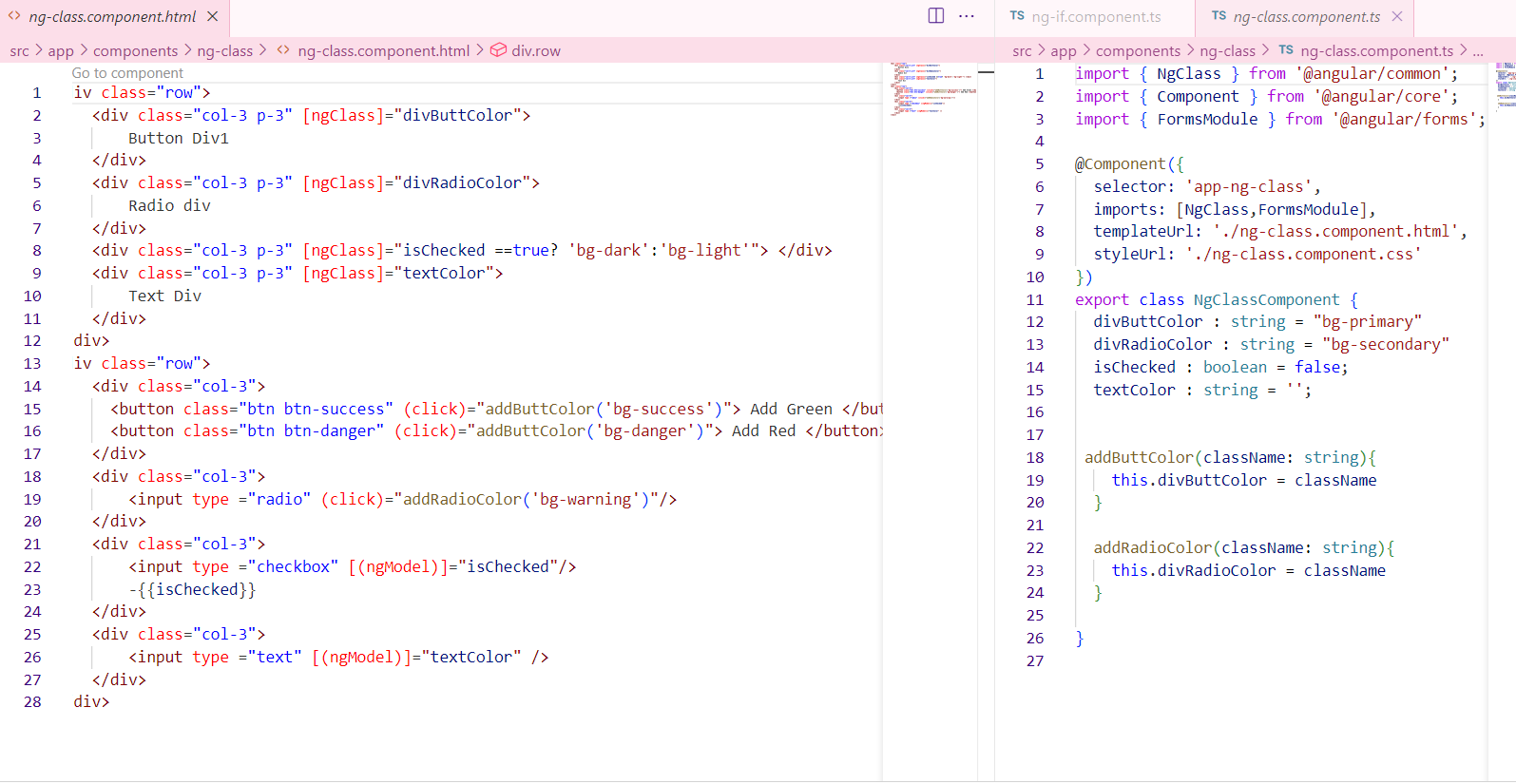
**What Are Attribute Directive ?**

**How to use [ngClass] Directive ? and How to Add Dynamic Class ?**

In order to use ngclass either import**: commonModule** or **NgClass** in .ts file.

The declared variable needs to assign for ngClass.

**[ngClass]=”divColor”**



**How to use [ngStyle] Directive ? and How to Add Dynamic CSS / Style?**

Using of ngStyle added background color to the variable.

A screenshot of a computer

AI-generated content may be incorrect.

**2nd example:**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**What is Control Flow Statement ?**

**ngIf and ngFor needs to import: commonModule** or **NgIf/NgFor**

**How To use @if, @else if , @else ?**

**How to use @for ?**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**What is Signal ?**

When a change is detected, **Zone.js** will detect it and update the UI. Even if **Zone.js** is turned off, signals will still detect changes.

To simplify state management and change detection. They help track and react to changes in data more efficiently.

**How to Create Signal And Update Signal ?**

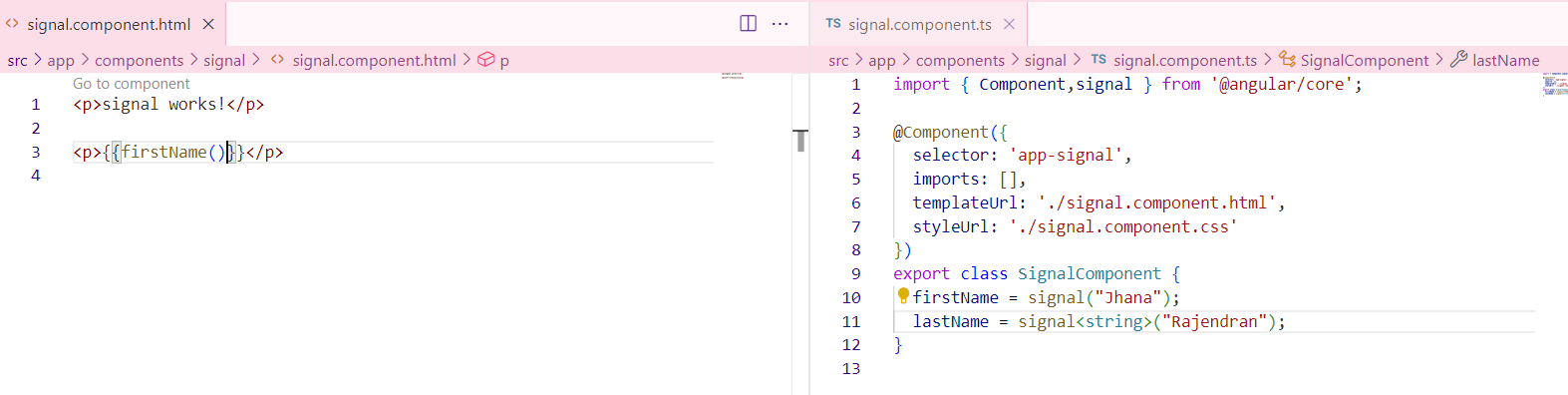
Creation of signal is like same as variable except there is keyword signal.

We can create signal with and without data type.



**How to Access Signal Value ?**

**We can access value as a method.**



**How to use set , update method ?**

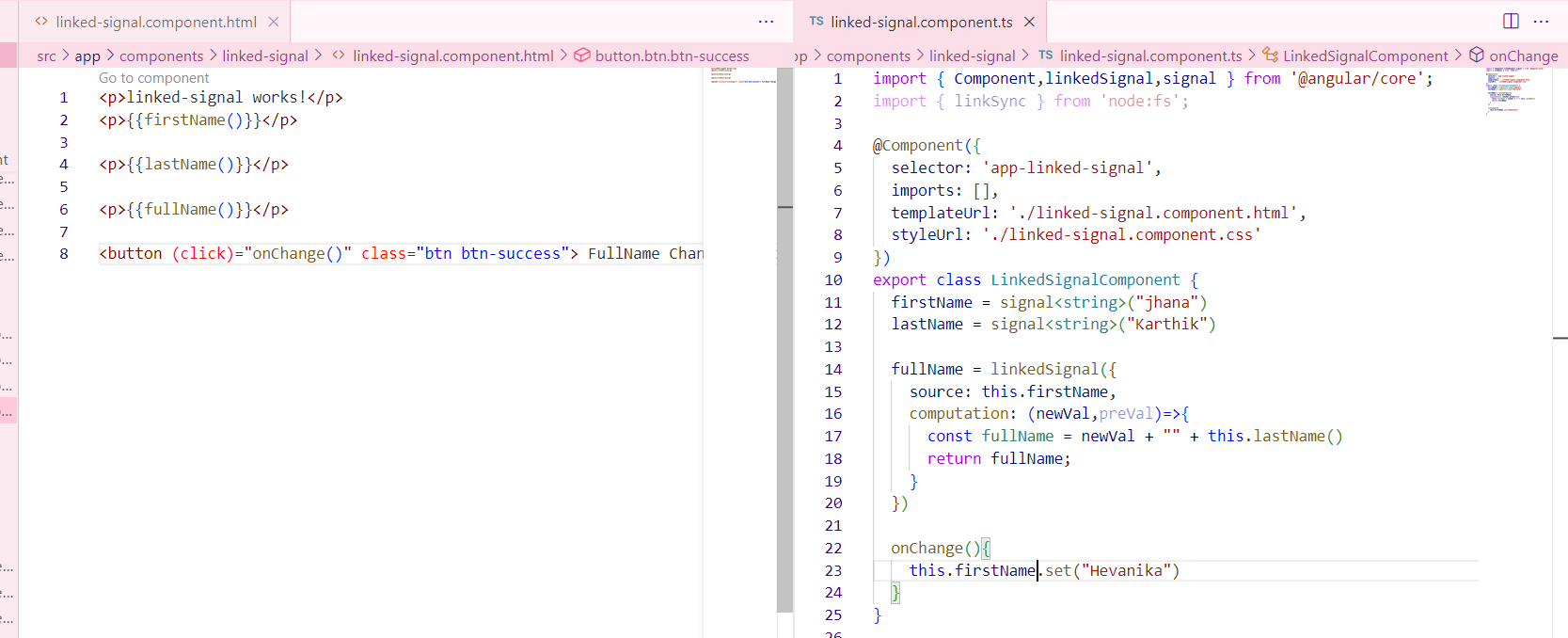
A screenshot of a computer

AI-generated content may be incorrect.

**What is Linked Signal ?**

If there is a dependent signal, and one of the signals changes, the linked signal (or computed signal) will automatically update. This is because computed signals track dependencies and recalculate when any of their dependencies change.

**How to Create Linked Signal?**



**What is computation and equal in Linked Signal ?**

A screenshot of a computer

AI-generated content may be incorrect.

**What is Routing ?**

Notes: Creating a URL for your components.

Starting from Angular 19, a constant file app.routes.ts is generated by default. In Angular, a route is an object that defines a path and its associated component.

Routing in Angular enables navigation between different views (components/pages) within a Single Page Application (SPA). It allows users to switch between pages without a full-page reload, providing a smoother experience.

**How to Create Routes ? AND How to Create Default Route ?**

Default route will be declared as.

Declare the necessary routes in app.routes.ts and pass this route to app.config.ts.

A screenshot of a computer

AI-generated content may be incorrect.

**What is router-outlet ? AND What is RouterLink ?**

When using navigation links in app.component.html, you must include <router-outlet> to display routed components.

A screenshot of a computer

AI-generated content may be incorrect.

The rule is that whatever you need to use must be imported in app.component.ts. For example, <router-outlet> must be imported in app.component.ts

**How to do Navigation in Angular ?**

By using router link we can achieve navigation.

In order to perform navigation from .html file use routerLink=

A screen shot of a computer

AI-generated content may be incorrect.

In order to perform logical navigation from .ts file must use router services.



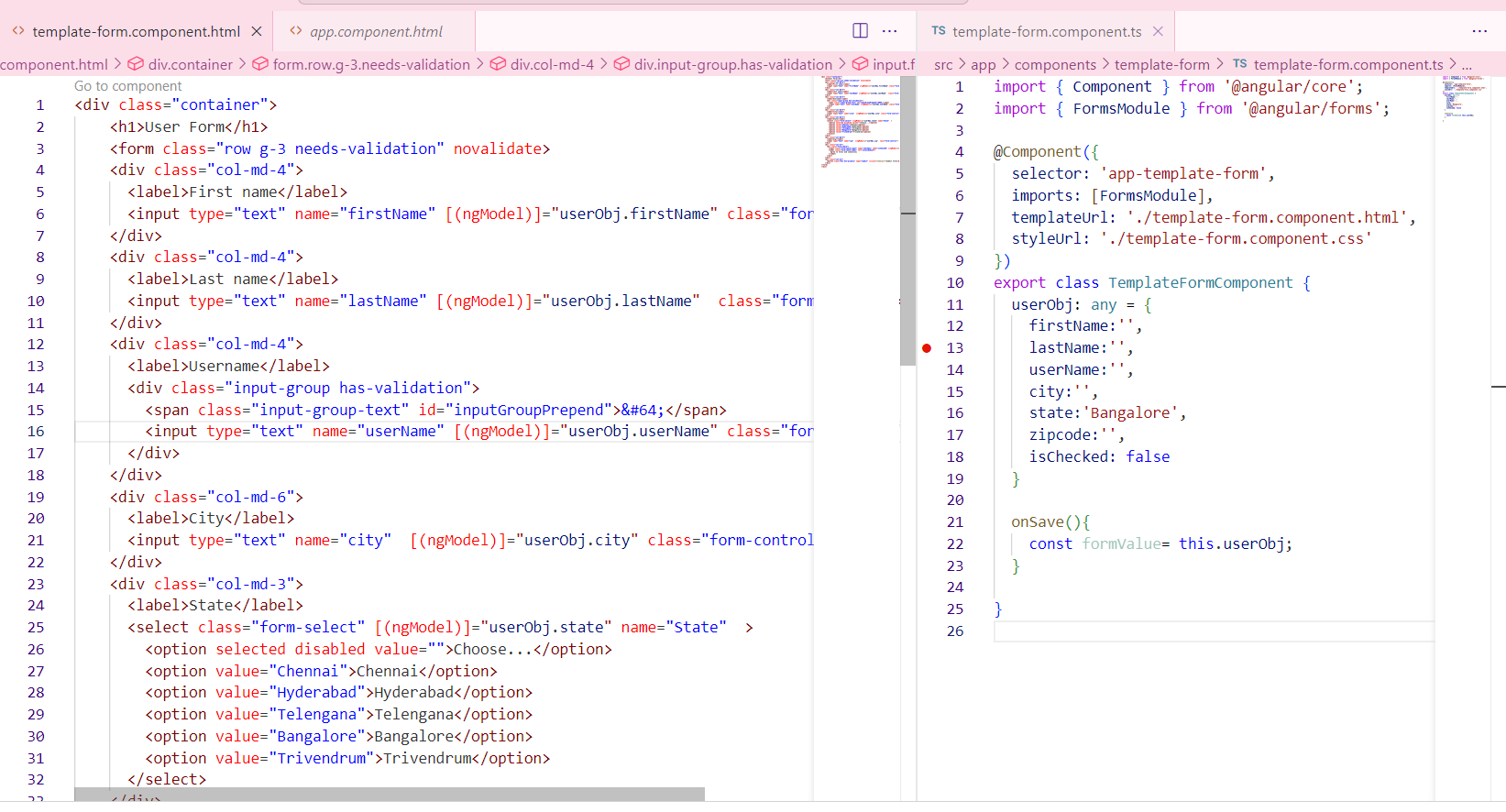
A screenshot of a computer program

AI-generated content may be incorrect.

**Form Types in Angular ?**

Bind an object to a form using ngModel is called Template Form.

**How to Create Template Form ?**



**How to Validate Template Form Use of ngModel in template Form ?**

In template form we need to add validation in HTML.

Creating # property we can validate.



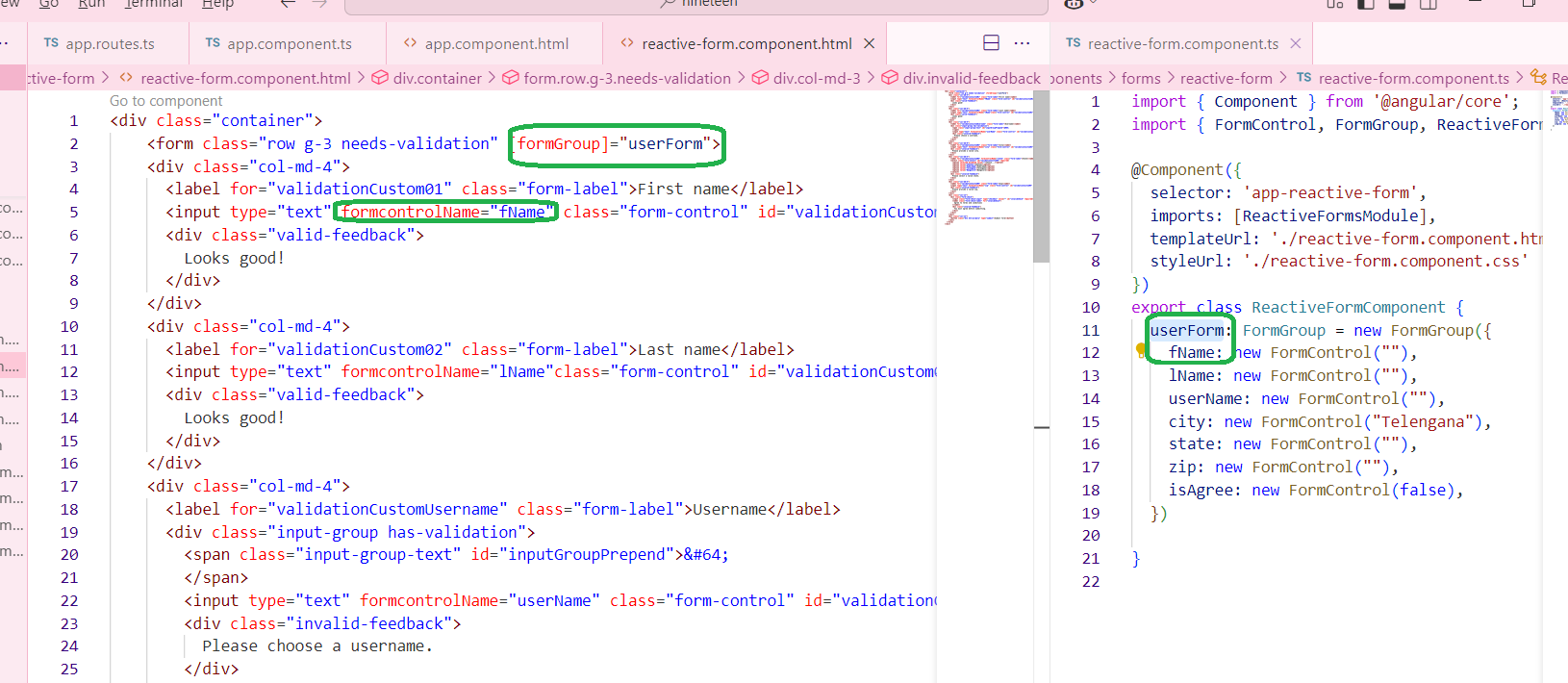
**How to Create Reactive Form in Angular ?**

Import FormGroup and FormControl from @angular/forms.

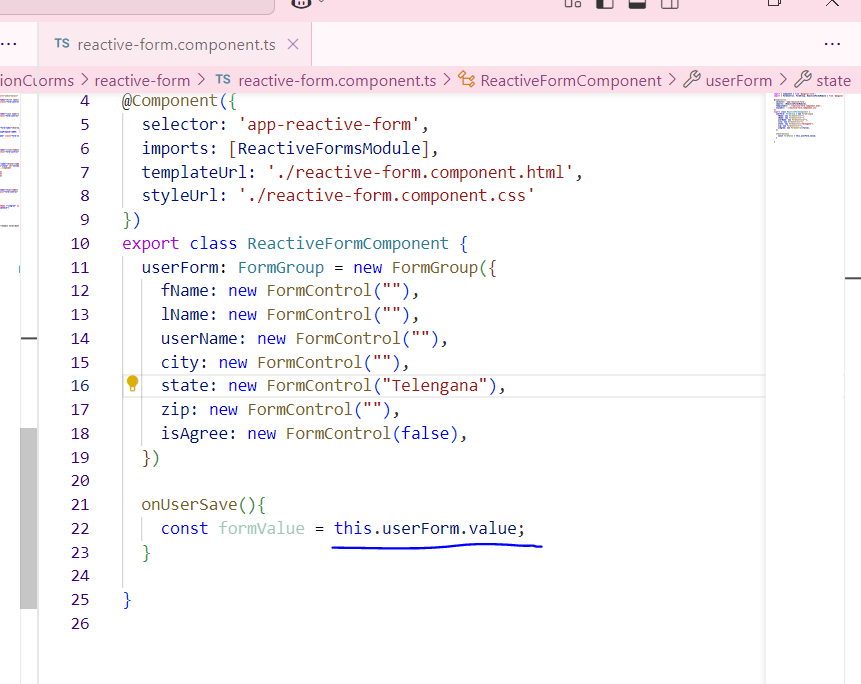
Initialize the bind form with fields inside the constructor.

All input name should bind in formcontrolName

Reactive form FormGroup is best approach.



We can get complete form value on submit, so that we can send this object in API.



**How to Validate Reactive Form in Angular ?**

In reactive form we need to add validation while creating form.

Validation must pass as an array because there is multiple of validations avail.

Can validate using form. Control example: 

A screenshot of a computer program

AI-generated content may be incorrect.

**What is Form Control ?**

**How to Disable Form Control ?**

If need to disable the control for the field.

1.create a constructor.

2.disable that particular field.

A close-up of a computer code

AI-generated content may be incorrect.

We have an enable option also.

A screen shot of a computer code

AI-generated content may be incorrect.

Const isValid =this.userForm.valid;-🡪 Whole form valida..

A screenshot of a computer

AI-generated content may be incorrect.

**What is API ?**

Application programming interface, a set of protocols and definitions that allow different software components or programs to communicate with each other and share data.

In order to use API first need to add provideHTTPclient() in providers array avail in app.config.ts file.

A screenshot of a computer program

AI-generated content may be incorrect.

**Hint: Free api’s avail in: https://beeceptor.com/mock-server/explore/**A diagram of a cloud and a gear

AI-generated content may be incorrect.

**What is HttpClient ?**

**What is GET API ?**

**GET RECORDS**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**How to Integrate GET API call ?**

**How to Store api Response ?**

**What are Fake API ?**

**What is POST API ?**

**TO CREATE RECORDS**

**How to Integrate POST API ?**

**POST API With Template form ?**

**What is PUT API ?**

**TO UPDATE EXISTING RECORDS**

**How to Integrate PUT API ?**

**PUT API With Template form ?**

**What is DELETE API ?**

**TO DELETE EXISTING RECORDS:**

**How to Integrate DELETE API ?**

**Show confirmation before Deleting Record**

**What is Resource API ?**

It provides a declarative way to manage API calls, making it easier to fetch, store, and reuse data across components without manually managing state.

A screenshot of a web page

AI-generated content may be incorrect.

**For which Purpose We have to use Resource API in angular 19?**

**How to make api call with Resource API in angular 19? and What are loader and reload function in Resource API in angular 19?**



**What is Service ?**

A Service is a self-contained, reusable piece of code that performs a specific task, making it easily shareable across multiple components within your application.

**ng g s serviceName**

This service generates 2 files:

**serviceName.spec.ts**

**serviceName.service.ts** -🡪 **This contains the basic service skeleton; the service logic needs to be added here**

**services will use @Injectable decorator to enable dependency injector.**

**@Injectable decorator tells angular to inject constructor arguments through its depency injection system.**

**What is Dependency Injection?**

**Allows components to receive dependencies rather than creating them manually by using dependency injection we can make our code more modular and reusability.**

**Dependency Injection (DI)** is a fundamental concept in Angular. It is built into the Angular framework and allows classes with Angular decorators—such as components, directives, pipes, and injectables—to receive dependencies.

When a dependency is requested, the injector checks its registry for an existing instance. If the instance is not available, it creates a new one.

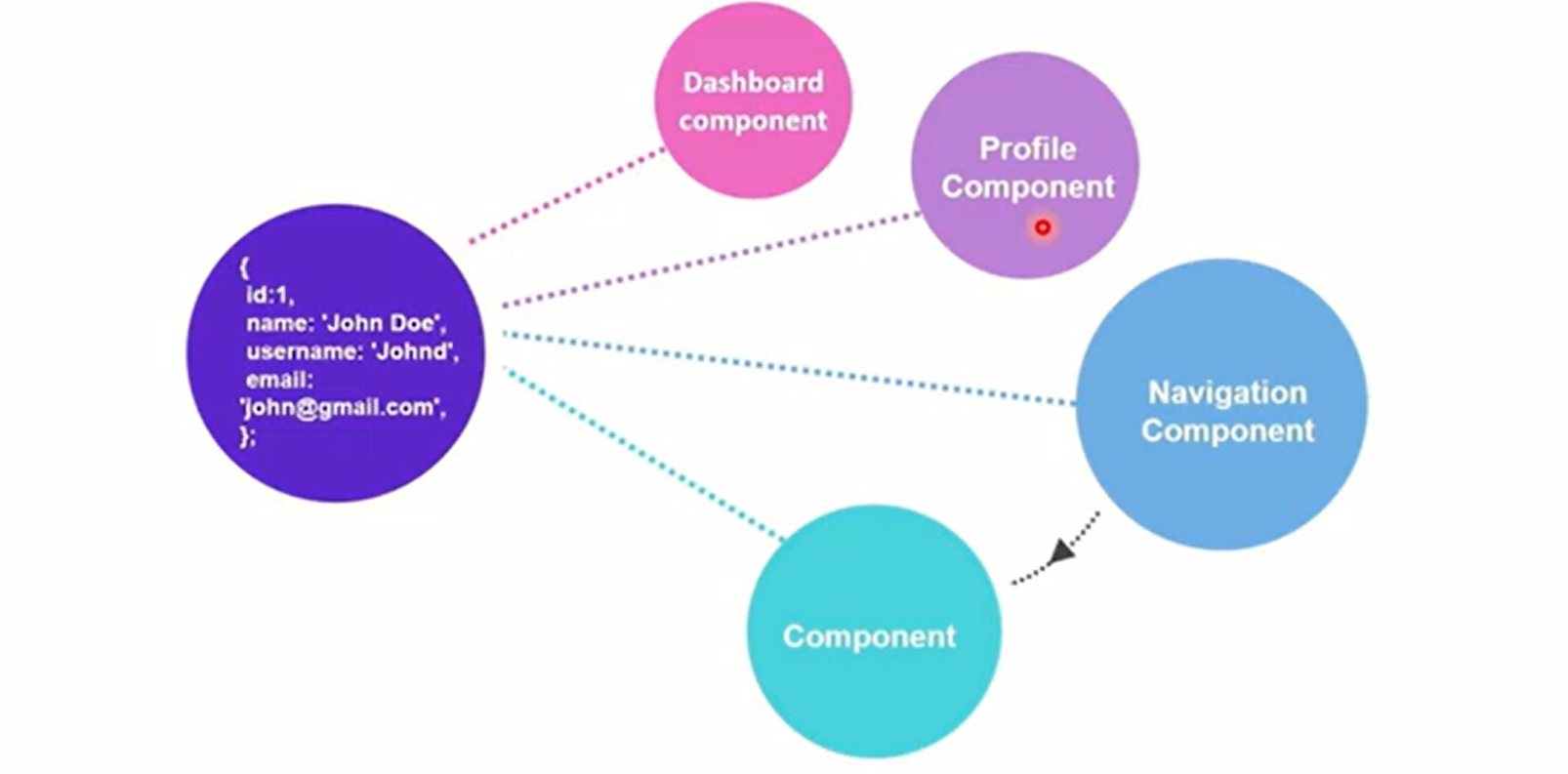
We can inject dependencies at the **app-root level**, making them accessible to all components. If a dependency needs to be accessible only to a specific component, we can use a **provider** within that component.

**For which Purpose We have to use Service ?**

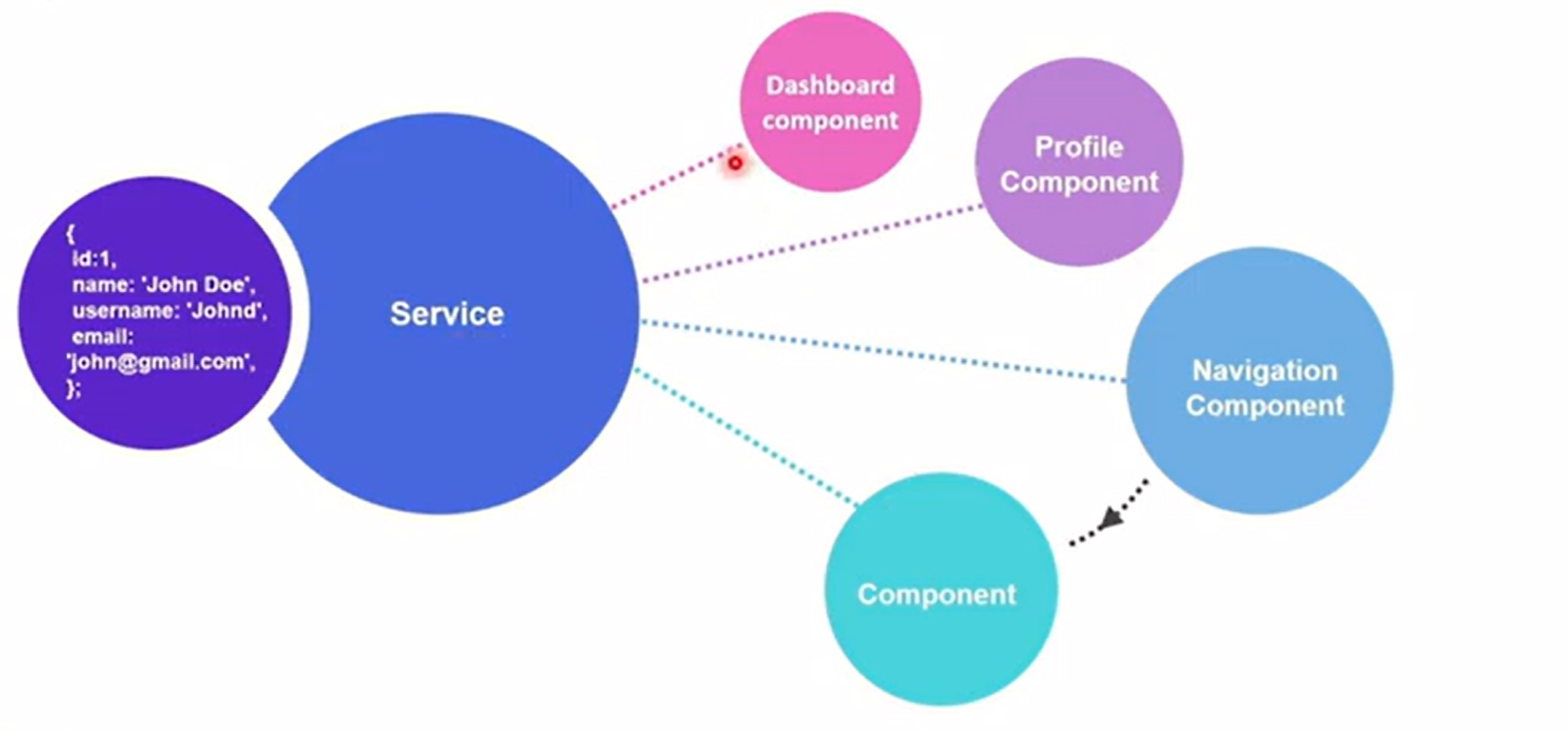
A diagram of a software process

AI-generated content may be incorrect.

**Share data between components**



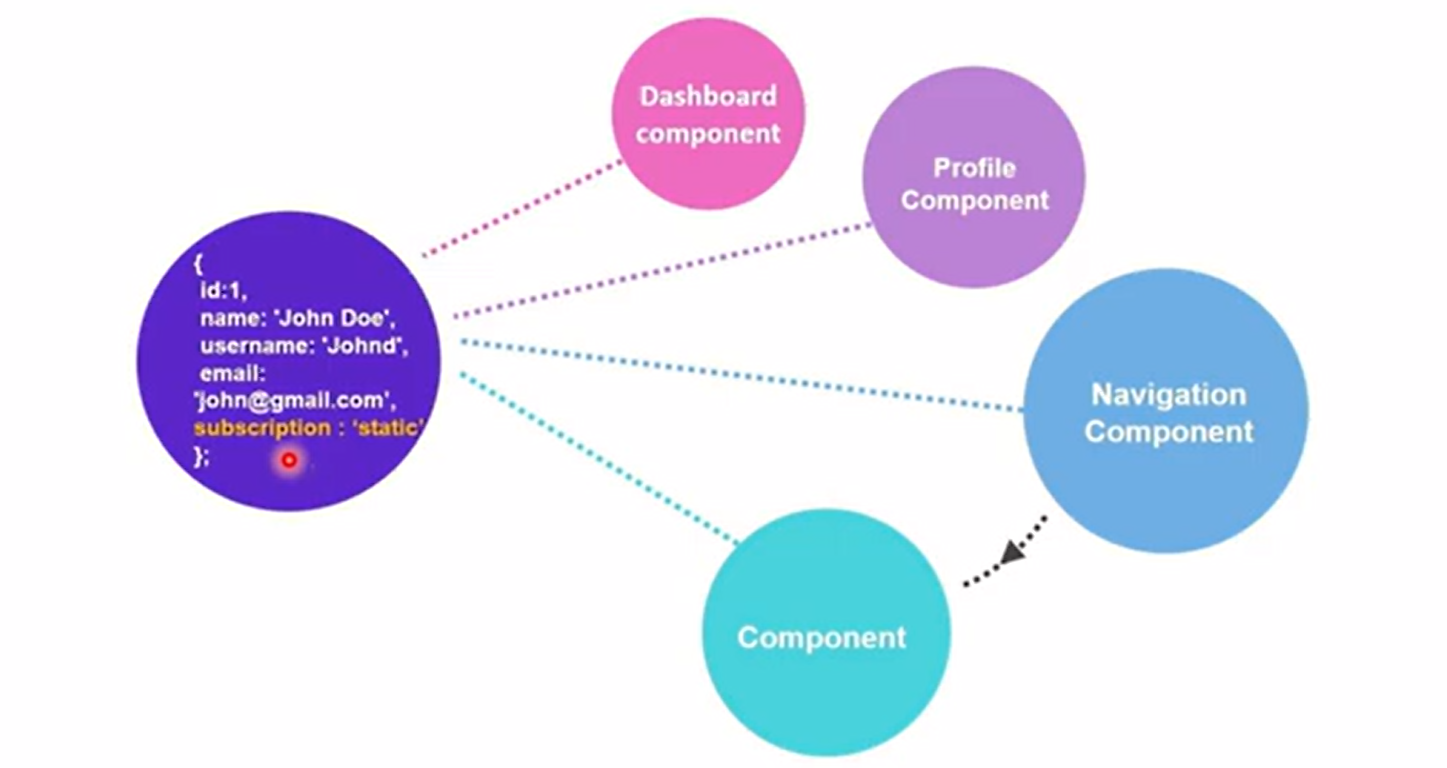
To avoid fetching all user data and duplicating it across different components, we can use a service to access the user's data in all components.



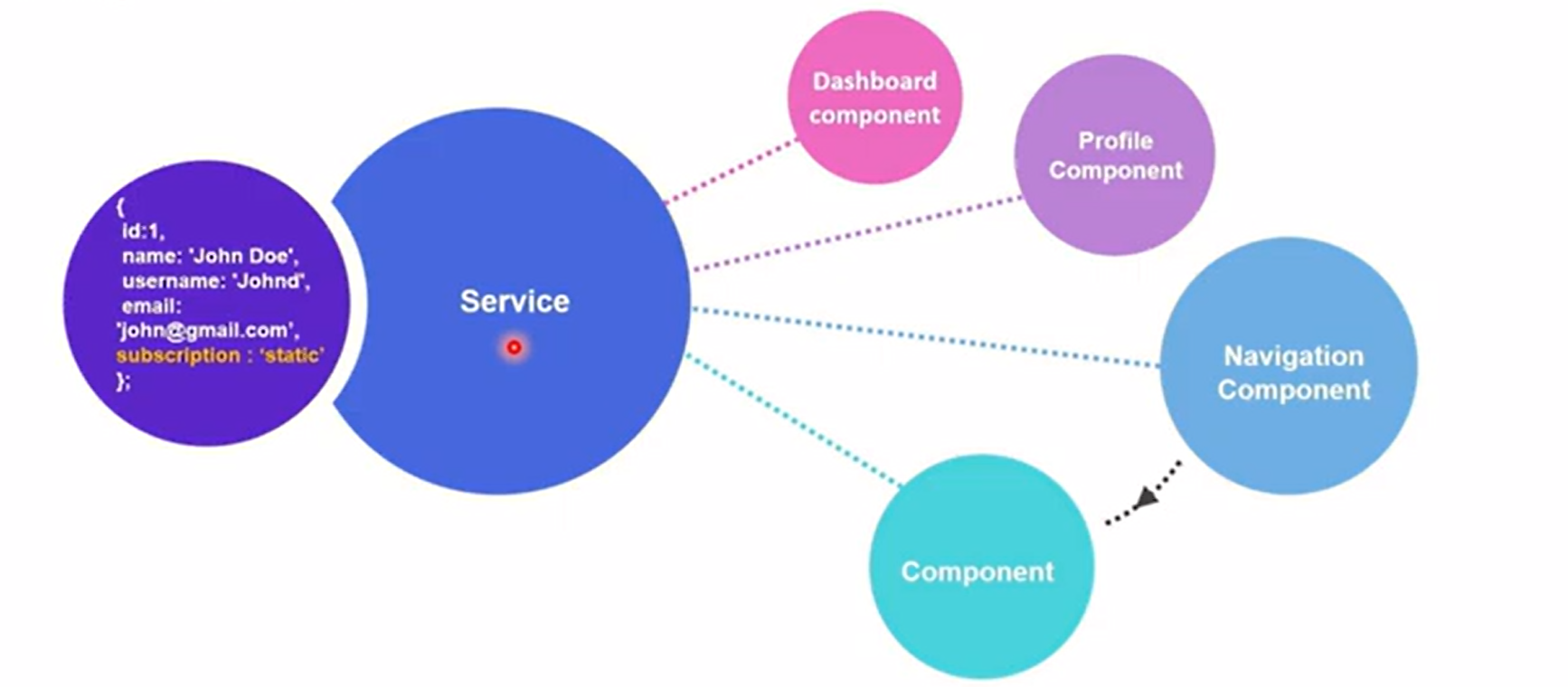
A screen shot of a computer

AI-generated content may be incorrect.

**Share business logic across multiple components**



We can share business logic using methods in a component, but when there's more data to handle or the logic needs to be reused across components, it's better to use a service.



The premium subscription eligibility and all related operations are handled by the service.

**Interact with a database or external data source**

**How to create api call Function in Service ?**

**Angular CRUD with api with Service call Functions ?**

**What are Life Cycle Events ?, Types of Life cycle Events ? Which one We mostly Use and Their UseCase ?**

Life Cycle Events are function which called automatically when they are needed.

**Lifecycle of a Component:**

Angular components have a lifecycle managed by Angular. Key lifecycle hooks include:

ngOnInit(): Called after the component is initialized. (We can use to trigger Api call function)

constructor(): is to initialize the variable.



ngOnChanges(): Called when input properties change.

ngAfterViewInit():view child, getting reference of element..(how much our UI taking time to load)

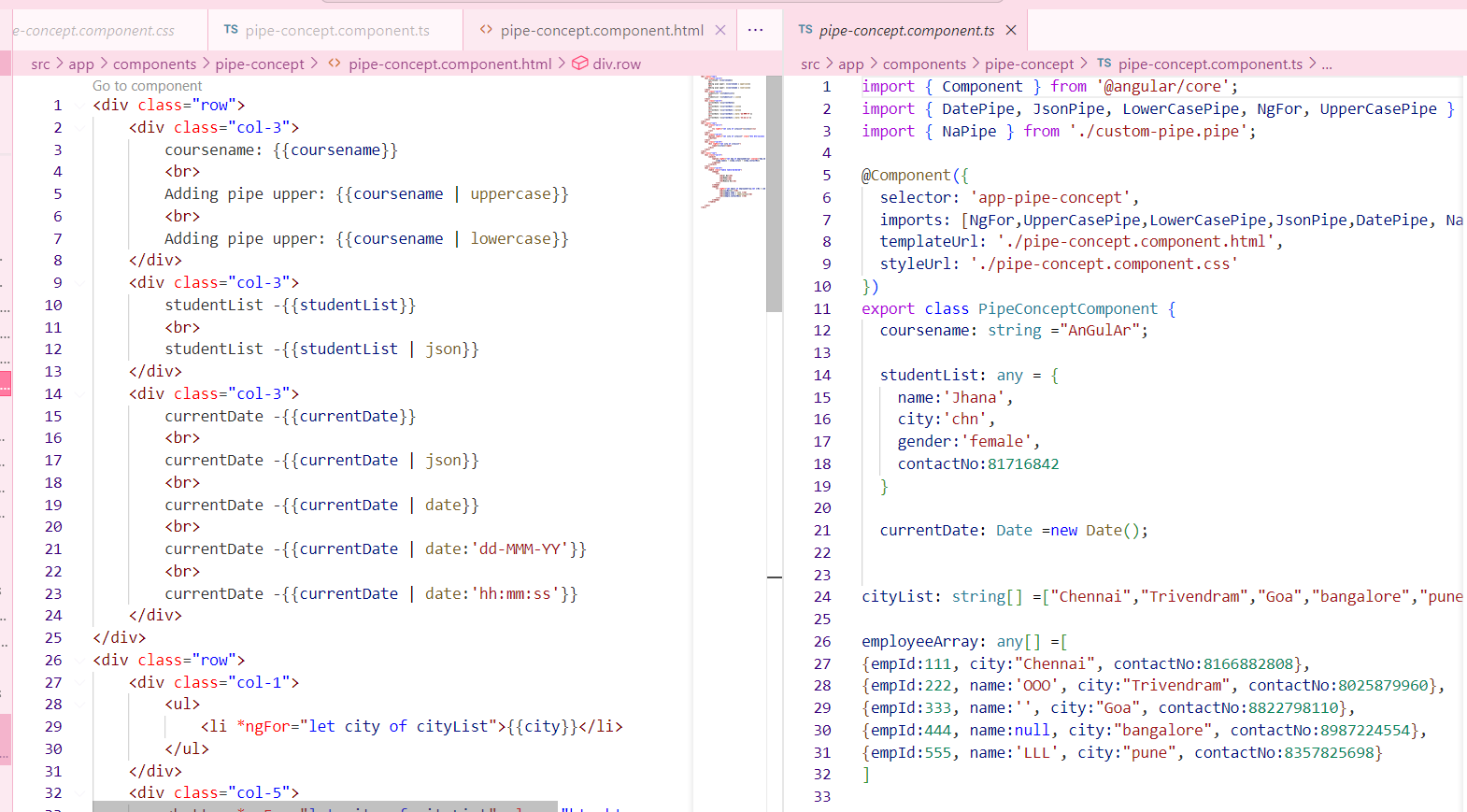
ngOnDestroy(): Called before the component is destroyed.. (while leaving the page).



**What are Pipes ?**

Pipes are simple function by default provided by angular, Used to format the data before rendering on UI.

**Default Pipe in angular ?**



**How to Create Custom Pipe ?**

A screenshot of a computer

AI-generated content may be incorrect.

**What is @Input in angular?**

Input is also a decorator.

Sending of data from parent to child.

Example using a progress bar.

1.Using the selector name app-progress-bar, we can implement it in any component by adding it to the component's HTML file after importing it in the component's TypeScript file.

2. To pass data from parent to child we must import input in child component.

By using the @Input decorator, we can make this acting as dynamic.

export class ProgressBarComponent {

 @Input() progressValue: number = 0;

}

HINT: It allows to group a series of buttons together. (.btn-group)

**What is @Output() in angular ?**

Is a decorator

Get of data from child to parent.

**What is EventEmitter in angular?**

**How to Create Reusable component in angular ?**

**How to create interface ?**

**How to create class ?**

**When to use class or interface ?**

**How to Implement Login Page angular ?**

**What is Hard Code Login angular ?**

**How to Validate User With API angular ?**

**What is guard in angular ?**

**How to Create canActivate guard in angular ?**

**What is ViewChild ?**

**How to create viewChild for Element Reference ?**

**How to create viewChild for Component Reference ?**

**How to use signal with Forms.**

**Creating individual FormControl instead of creating Form.**

**Use of async Pipe with API call.**

**How to use resource api to make api calls.**

**use of Subject in Filter textbox**

**How to Use ngTemplate ?**

**How to Use ng-container ?**

**How to Use ngTemplateOutlet with ngTemplate ?**

**How to Use Interceptor?**

**How to Do JWT token Login ?**

**How to send token in headers in api ?**

**CRUD**

**Lazy loading**

**$event is used to pass the data which is in the input to the typescript.**

**Event Emitter is used to discharge the events to the other components.**