Angular 8 : Online Class

2-August-2019

By: Sahosoft Solutions

Presented by : Chandan Kumar

@Input and @Output decorator

It is the topic of **Component Interaction** in Angular. As we know, Angular Application is built upon small components so passing the data from Parent component to child component is a little bit tricky, in that scenario, @Input and @Output decorator comes in handy.

Angular components have a better way of notifying parent components that something has changed via events. **Inputs** specify which properties you can set on a component from a parent whereas "**Outputs**" identifies the events a component can fire to send information up the hierarchy to its parent from its child component.

@**Input** and @**Output** decorate input and output properties.

@Input decorator

@Input decorator binds a property within one component (child component) to receive a value from another component (parent component). This is one-way communication from parent to child. The component property should be annotated with @Input decorator to act as input property. A component can receive a value from another component using component property binding. Now we will see how to use **@Input**. It can be annotated at any type of property such as number, string, array or user defined class.

To use alias for the binding property name we need to assign an alias name as @Input(alias).

@Output decorator

@Output decorator binds a property of a component to send data from one component (child component) to calling component (parent component). This is one-way communication from child to parent component. **@Output** binds a property of the type of angular EventEmitter class. This property name becomes custom event name for calling component.

@Output decorator can also alias the property name as @Output(alias) and now this alias name will be used in custom event binding in calling component.
Find the @Output decorator using aliasing.