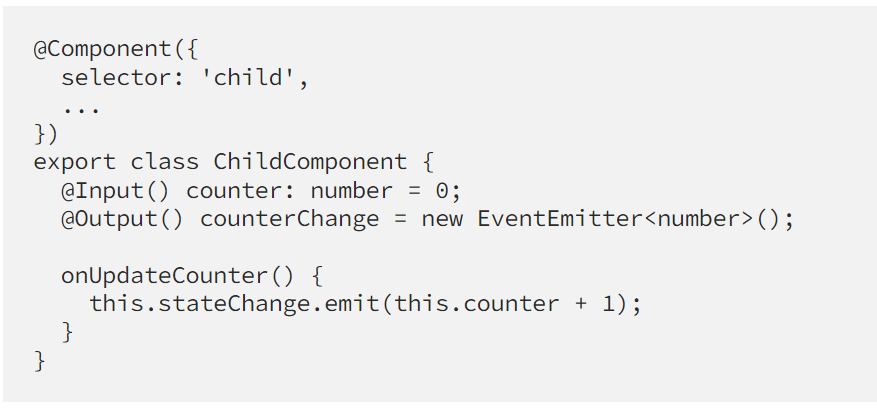
11

**Component** **Communication**

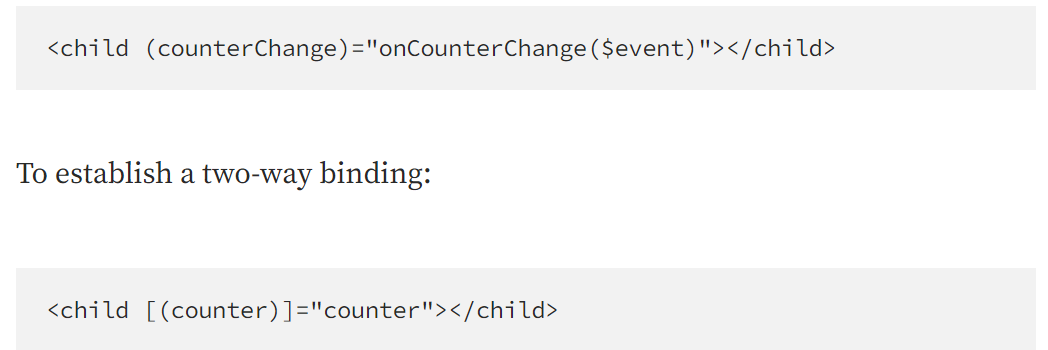
* **Parent Child**
  + parent to child (@Input)
  + child to parent (@output)
  + Reference (@ViewChild & @ContentChild)
* between the different components using service

1. **Binding (@Input & @Output)**

* the parent component can pass the data to the child using the @input Property.

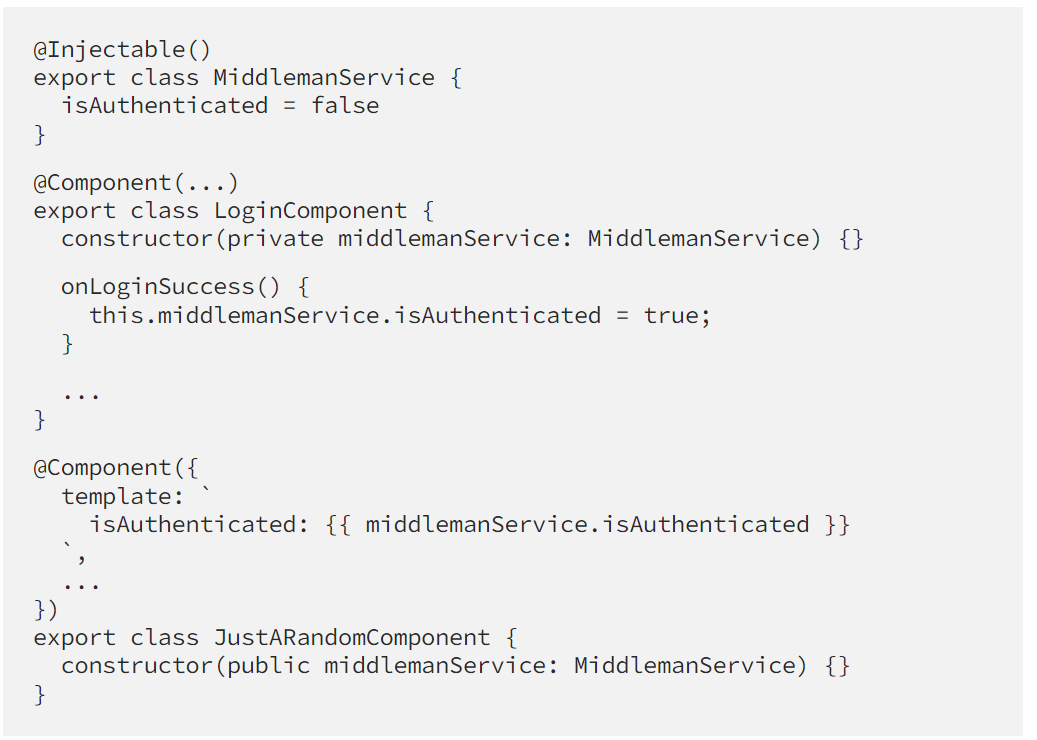


To bind a value to the input:



**2. Provider (Service**)

Provider is a more high-level alternative for communication powered by the Angular **dependency injection**. Instead of establishing a direct linkage between components, a standalone injectable (service) is used as a middleman between them.



**3.Reference (@ViewChild & @ContentChild)**

 In simple, to directly grab the instance reference of the specific component and manipulate it programmatically.

@ViewChild() allows a component to access a child component or DOM element. For example, a parent component might use @ViewChild() to access a child component like this:

In order to use the @ViewChild() decorator, we need to reference the child component in the parent component's class, and then we can use to query the child component's properties and methods or manipulate the DOM directly.

ContentChild

* used to query or helps to get a reference to the projected content.
* Projected content will be accessible for the first time in the ngAfterContentInit lifecycle hook method.

ALL THE BEST!!!