**Week 6 Discussion Angular Input and Output Properties**

*Becca Buechle*

*October 18, 2020*

Angular @input and @output are both independent from one another so they can be used totally separate from one another if needed of you can use both. If the nested component is such that it only needs to send data to its parent, you wouldn’t need the @input you’d only need the @output. (Angular.) However, the same is true if it were to be reversed if the child only needs to get data from the parent, you’d only need the @input. (Angular.)

The @input decorator is telling the Angular code that when you find this property binding with this name then it should be mapped to the component property of this other name. (Yadav, N.) But it hasn’t been assigned an alias the just to use the component property name. Even though newer versions of Angular support the name alias it should be avoided, and you should instead call the property name. (Yadav, N.) The decoder is the one that will mark the class field as the @input property and will then supply the configuration meta-data. The DOM and @input property are bound in the templet. (Yadav, N.) During the detection Angular will update data property so that it will make with the DOMs property value. (Yadav, N.)

The @output property should be initialized to Angular EventEmitter with its values flowing out of the component as events. (Yadav, N.) When using the @output it is marking that it is a property child component and is being used as a doorway which is helping flow the data from the child to the parent property. (Yadav, N.) When an event needs to be raised the @output will work with the EventEmitter with will be used to emit any kind of custom events. (Yadav, N.)

Bibliography

Angular. (2020). Retrieved October 18, 2020, from https://angular.io/guide/inputs-outputs

Yadav, N. (2020, April 13). Angular Input Output Properties. Retrieved October 18, 2020, from https://medium.com/javascript-in-plain-english/angular-input-output-properties-cb02a1a543f8