According to angular.io you are able to share data with the parent component to at least one or many child components through the @input() and @output() decorators. (*Angular*, n.d.-d) The @input() allows the parent to refresh its data in the component of the child, where as the @output() component will allow the child component to send data to its parent. (*Angular*, n.d.-d) In order for us to user @input() in the child component we would first have to “import Input and then decorate the property with @input()”. (*Angular*, n.d.-d) @input may have numbers, strings, Booleans or objects. (*Angular*, n.d.-d) For example:

import { Component, Input } from '@angular/core';

Then below we would need to decorate the property:

export class ItemDetailComponent {

@Input() product = '';

}

(*Angular*, n.d.-d)

Then we can add the child:

<p>

Huge savings: {{product}}

</p>

(*Angular*, n.d.-d)

In order to send our data to a parent we would need to user @input() decorator. This will allow the data to be processed to the parent. It will “raise an event” (*Angular*, n.d.-d) so that the parent component will know of the change and it will also need to have EventEmmitter imported from the class @angular/core. (*Angular*, n.d.-d) For example, in order to user @output() we would first import the following classes:

import { Output, EventEmitter } from '@angular/core';

Then we would follow it up with something like this:

@output() newItemEvent = new EventEmitter<string>();

The newItemEvent is the name of the @output(), and the EventEmitter determines the type of @output(). (*Angular*, n.d.-d)

References:

*Angular*. (n.d.-d). angular.io. Retrieved June 26, 2023, from https://angular.io/guide/inputs-outputs

Kim, G., Debois, P., Willis, J. O., & Humble, J. (2016). *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. https://dl.acm.org/citation.cfm?id=3044729