

Event Binding Lab

When our customers order a product we have to pick the products, pack them, and ship them. Our pickers have to know where to get the product from our warehouse. Let's create a method for them to get the best location for that product.

The ship order component

1. Edit your ship-order.component.ts file. Add a new method called `getBestLocation(orderLine)`. Note that it will receive an `orderLine` object. For now it should just set `orderLine.locationID` to "01A1A" and then `console.log` it.
2. Now edit your ship-order.component.html and find the 'Get best location' button. Add an angular click event handler to it. When fired, it should call the `"getBestLocation()"` method you just wrote. Don't forget to pass in the current `orderLine`.
3. Run and test. When you click your button, the button should be replaced by a location.

Cool! Do you see the "Mark as shipped" and "Problem" buttons? Let's work on them.

4. Wire up the "Mark as shipped" button to run a method called `markAsShipped(order)` which receives in an order object. This method should change the order's status from 0 to 1.
5. Wire up the "Problem" button. It should run a `markWithProblem()` method which also receives an order object. It should set the order's status to 2. And it should add this message to the top of the instructions: "Don't pick this order! It needs to be looked at by a supervisor."
6. Run and test. Look at the orders in the console and make sure that the status is indeed changing.

Receive inventory

At this point we have worked on several components but one has been neglected, the receive product component. Let's fill that in now.

7. First, notice that in `setup/assets/html`, you have a starter for this component. Copy the html that is in there.
8. Open `receive-product-component.html` and paste the html from that other file.
9. Run and test. You should see a fine-looking component but it doesn't really do anything. It is just static HTML.

The managers of the warehouse have decided that the users should enter the package tracking number for every order we receive. We've decided that we will hide everything else on the page until the tracking number is entered.

10. Create a boolean class variable called `showForm`.
11. Add a structural directive so that only the tracking number section is drawn until this `showForm` variable is true.
12. Create an event handler for the button click. It should call a new method named `saveTrackingNumber()` which merely sets `showForm` to true. (Later on we can actually save the tracking number.)
13. Run and test. Did you make the rest of the component show when you click the button?
14. Notice that there's a button labeled "Receive product". When the user clicks it, call a method named `receiveProduct()`. For now it can just `console.log()`. We'll make it do something meaningful later.
15. Run and test. Make sure that when you click the button something is logged to the console.

16. Do the same for the "Finished receiving" button. It should call a `finishedReceiving()` method which just logs something to the console.
17. Run and test that as well.

When you have your buttons working, you can be finished. See how simple event handling can be?