Pipes Lab

Using built-in pipes

1. No offense, but your dates are ugly. Look at your list of orders either in the dashboard or in orders ready to ship. You can also see some ugly dates in ship order and in receive product.
2. All of these can be improved by piping the date through the date pipe. Go ahead and do that now.
3. Run and test. Find any other dates that are being shown and fix them.

A custom location pipe

To make it easier on our new and temp warehouse workers we should decipher the locationID. We're going to create a custom pipe to do that. The locationID has the format **AABSI** where

AA	is the	Aisle
B	is the	Bay
S	is the	Shelf
I	is the	Bin

First, let's see how the pipe will be used before we create it.

4. Open your ShipOrderComponent in an IDE. In the template, find where you're displaying the location. Add a pipe right after it:
```
{{ line.locationID | locationExpand }}
```
5. Run and test. Angular complains about the locationExpand pipe not existing.
6. Create a new pipe called "locationExpand"
```
ng generate pipe locationExpand
```
7. In that pipe you have a transform function that will receve a string ... the locationID. You want it to return a string in this format:
Aisle AA, Bay B, Shelf S, Bin I
8. Write that, then tune it until it is working.
9. And once you've got it working well, let's reuse it! Open your receive-product.component.html page and implement it there also.

Make a ship via custom pipe

Notice that on the Ship Product component there is a ship via property being bound. We'd like that to display the name of the shipper rather than the raw number.

10. Create a new pipe called shipVia. It should translate as follows:

Ship via	Shipper name
1	Federal Express
2	UPS
3	US Postal Service

11. Your goal will be to do an interpolation like this {{ order.shipVia | shipVia }} and have the name of the shipper be displayed.
12. Go ahead and implement it in the ship order component. Once you can see a different shipper for various shipvia values you can be finished.