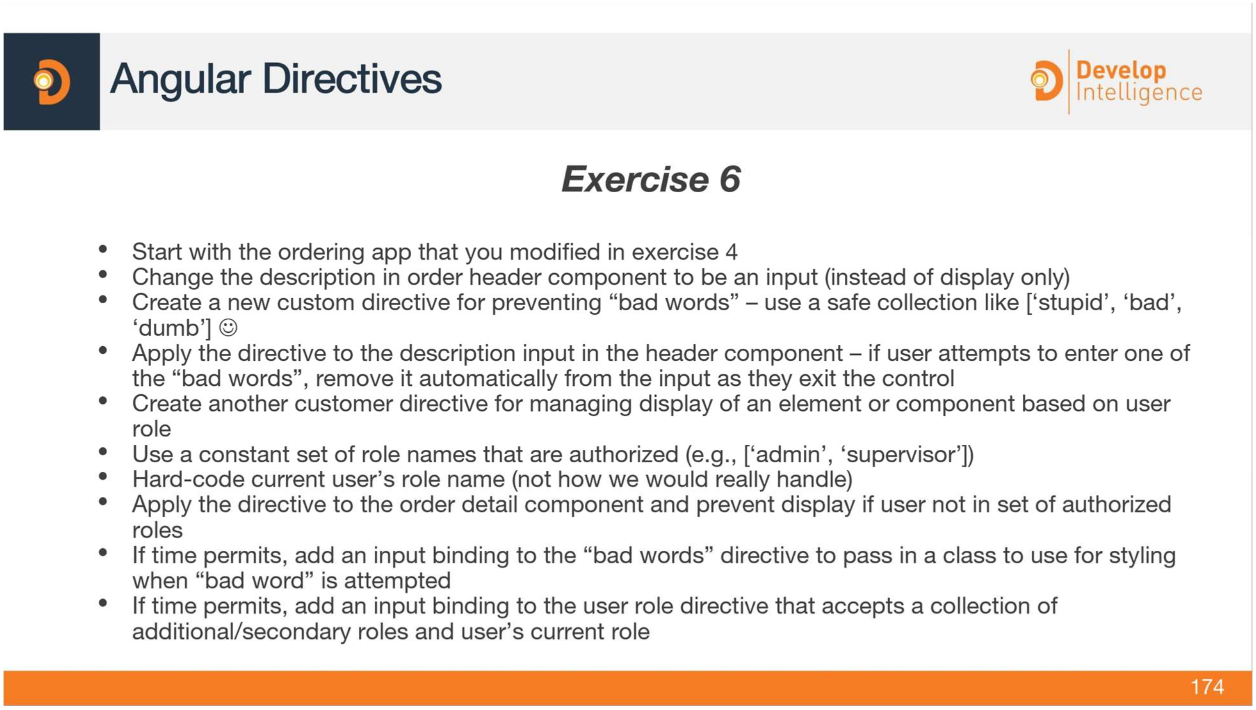
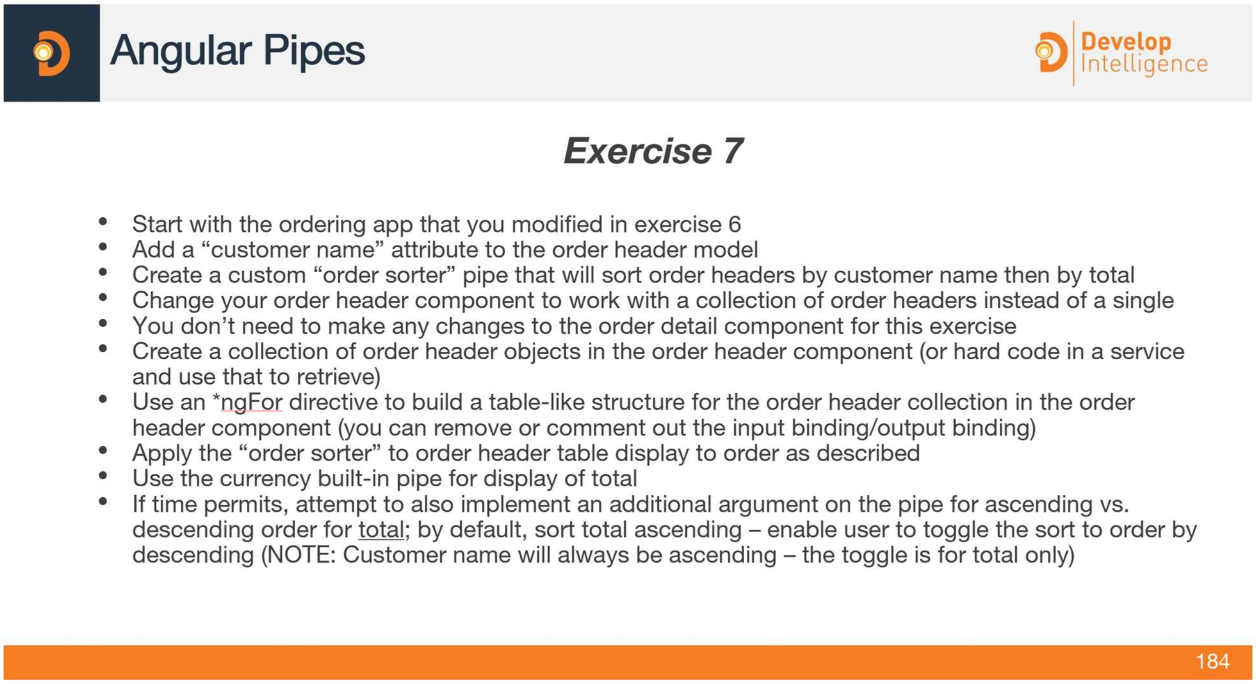


Angular Directives

Exercise 6

- Start with the ordering app that you modified in exercise 4
- Change the description in order header component to be an input (instead of display only)
- Create a new custom directive for preventing “bad words” – use a safe collection like [‘stupid’, ‘bad’, ‘dumb’] ☺
- Apply the directive to the description input in the header component – if user attempts to enter one of the “bad words”, remove it automatically from the input as they exit the control
- Create another customer directive for managing display of an element or component based on user role
- Use a constant set of role names that are authorized (e.g., [‘admin’, ‘supervisor’])
- Hard-code current user’s role name (not how we would really handle)
- Apply the directive to the order detail component and prevent display if user not in set of authorized roles
- If time permits, add an input binding to the “bad words” directive to pass in a class to use for styling when “bad word” is attempted
- If time permits, add an input binding to the user role directive that accepts a collection of additional/secondary roles and user’s current role



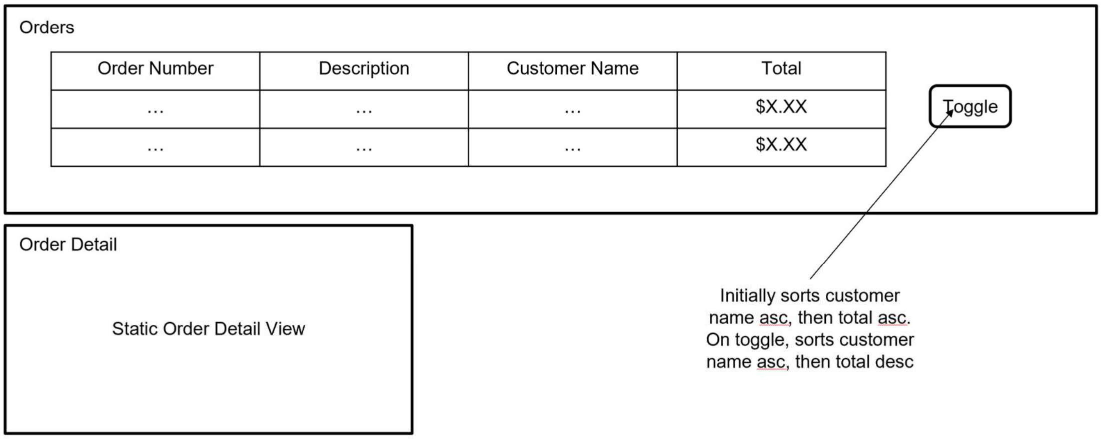
Angular Pipes

Exercise 7

- Start with the ordering app that you modified in exercise 6
- Add a “customer name” attribute to the order header model
- Create a custom “order sorter” pipe that will sort order headers by customer name then by total
- Change your order header component to work with a collection of order headers instead of a single
- You don’t need to make any changes to the order detail component for this exercise
- Create a collection of order header objects in the order header component (or hard code in a service and use that to retrieve)
- Use an *ngFor directive to build a table-like structure for the order header collection in the order header component (you can remove or comment out the input binding/output binding)
- Apply the “order sorter” to order header table display to order as described
- Use the currency built-in pipe for display of total
- If time permits, attempt to also implement an additional argument on the pipe for ascending vs. descending order for total; by default, sort total ascending – enable user to toggle the sort to order by descending (NOTE: Customer name will always be ascending – the toggle is for total only)

Angular Pipes

Exercise 7



Orders

Order Number	Description	Customer Name	Total
…	…	…	$X.XX
…	…	…	$X.XX

Toggle

Initially sorts customer name asc, then total asc. On toggle, sorts customer name asc, then total desc

Order Detail

Static Order Detail View