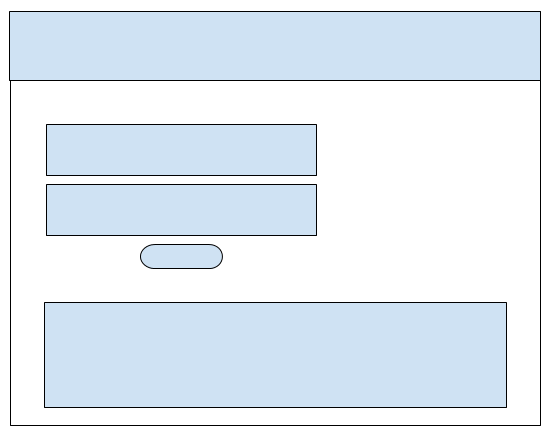
# Instructions – Exercise 6.2 – Output Properties

**Layout**

wish-list



in-n-out-books, part 4

**Instructions**

* Make a copy of the in-n-out-booksp3 from Assignment 5.4 and add it to your week-6 directory
* Rename the application to in-n-out-booksp4
* Delete the node\_modules directory
* Delete the package-lock.json file
* Open the angular.json file and find and replace all “in-n-out-booksp3” entries with “in-n-out-booksp4”
* Open the package.json file and change the name to “in-n-out-booksp4”
* Run npm install and ng serve
  + You are doing this to test the application and confirm there are no errors
* app.component.html
  + Change the title to “Exercise 6.2 - Input/Output Properties, Part 1
* Generate two new components: wishlist and wishlist-create
  + ng g c wishlist
  + ng g c wishlist-create
* app.routing.module.ts
  + Add an import statement for the WishlistComponent
    - import { WishlistComponent } from ‘./wishlist/wishlist.component.ts’;
  + Add a new route for the wishlist component
    - Path: ‘/wishlist’, component: WishlistComponent
* app.component.html
  + Add a link to the top navbar for the Wishlist page and name it “Wish List”
    - routerLink=”/wishlist”
    - Note: the “Wish LIst” link should be to the right of the “Book List” link
* app.module.ts
  + add import statements for FormsModule, MatFormFieldModule, and MatInputModule
    - import { FormsModule } from ‘@angular/forms’;
    - import { MatFormeFieldModule } from ‘@angular/material/form-field’;
    - import { MatInputModule } from ‘@angular/material/input’;
* Run and test the application to verify the component is accessible through the new link
* Add a new interface to the app directory and name it wishlist-item.interface.ts
* wishlist-item.interface.ts
  + Name the interface IWishlistItem
  + Give the interface the following properties
    - title: string
    - authors: string
* wishlist.component.html
  + Using the Flex Layout columns add the wishlist-create
    - <div fxLayout=”column”><app-wishlist-create></app-wishlist-create> <br /> <br /> </div>
* wishlist-create.component.ts
  + Add an import statement for the IWishlistItem
    - Import { IWishlistItem } from ‘../wishlist-item.interface’;
  + Add a variable named item of type IWishlistItem
    - item: IWishlistItem
  + Add Output and EventEmitter to the “Component, OnInit” statement
    - import { Component, OnInit, Output, EventEmitter } from ‘@angular/core’;
  + Add an output statement named addItemEmitter of type EventEmitter<IWishlistItem>
    - @Output() addItemEmitter = new EventEmitter<IWishlistItem>();
  + In the component’s constructor create a new instance of the IWishlistItem object
    - this.wishlistItem = {} as IWishlistItem;
  + Add a new function called “addItem()”
  + In the body of the addItem function create a new object literal of type IWishlistItem and call the addItemEmitter to emit the object
    - this.addItemEmitter.emit({ title: this.item.title, authors: this.item.authors })
    - Next, set the item object to an empty object
      * this.item = {} as IWishlistItem;
* wishlist-create.component.html
  + Note: for this section we are going to use template driven forms to create a form POST.  In later week’s we will cover reactive forms and data validation
  + Using Flex Layout columns, create a <form> with two mat-form-fields and a button.  Each mat-form-field should include a mat-label and input of type text.  Using the “banana in a box” directive assign the wishlistItem properties title and authors to each input field
    - <form fxLayout=”column”</form>
    - <mat-form-field fxFlex appearance=”fill”><mat-label>Title></mat-label><input type=”text” name=”title” matInput [(ngModel)]=”wishlistItem.title” />
    - <button mat-button color=”primary” (click)=”addItem()”></button>
* wishlist.component.ts
  + Add an import statement for the IWishlistItem interface
    - import { IWishlistItem } from ‘../wishlist-item.interface’;
  + Add a new variable named items of type Array<IWishlistItem> and assign it to an empty array
    - items: Array<IWishlistItem> = []
  + Add a new function named updateItemsHandler(item: IWishlistItem)
  + In the body of the updateItemsHandler add the parameter item to the items array
    - this.items.push(item)
* wishlist.component.html
  + Update the <app-wishlist-create> component by adding the “addItemEmitter” as an event and passing-in the updateItemsHandler function
    - <ap-wishlist-create fxFlex (addItemEmitter)=’updateItemsHandler($event)’></app-wishlist-create>
  + Under the app-wishlist-create component, add a mat-list and using the \*ngFor iterate over the items array and map the properties mat-list-item elements
    - <mat-list fxFlex \*ngIf=”items && items.length > 0”><mat-list-item \*ngFor=”let item of items”><div mat-line>Title: {{ item.title }}</div></mat-list-item></mat-list>
  + Note: You will need to show a message to users when the items array is empty.  To do this, add an \*ngIf that checks if the items array is empty
* Run and test the application until you are happy with the results