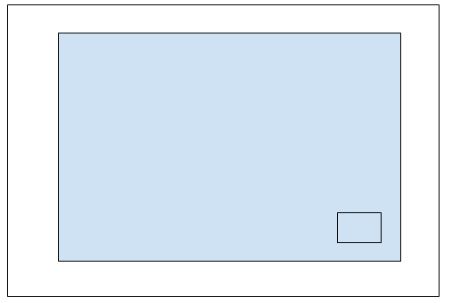
# Instructions – Assignment 5.4 - Dialogs

**Layout**

book-details-dialog



in-n-out-books, part 3

**Instructions**

* ~~Make a copy of the in-n-out-booksp2 from Exercise 5.3 and add it to your week-5 directory~~
* ~~Rename the application to in-n-out-booksp3~~
* ~~Delete the node\_modules directory~~
* ~~Delete the package-lock.json file~~
* Open the angular.json file and find and replace all “in-n-out-booksp2” entries with “in-n-out-booksp3”
* Open the package.json file and change the name to “in-n-out-booksp3”
* 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 assignment name to “Assignment 5.4 - Dialogs”
* Generate a new component and name it book-details-dialog
  + ng g c book-details-dialog
* app.module.ts
  + Add an import statement for the MatDialogModule
    - import { MatDialogModule } from ‘@angular/dialog’;
  + Add the MatDialogModule to the imports array
  + Note: Starting with Angular 9 it is no longer necessary to add dialogs to the entryComponents: []
* book-list.component.ts
  + Add an import statement for the MatDialog
    - import { MatDialog } from ‘@angular/material/dialog’;
  + Add an import statement for the BookDetailsDialogComponent
    - import { BookDetailsDialogComponent } from ‘../book-details-dialog/book-details-dialog.component’;
  + Add the MatDialog class to the components constructor
    - constructor(private booksService: BooksService, private dialog: MatDialog)
  + showBookDetails(isbn: string)
    - Underneath the book object assignment (booksService.getBook(isbn), create a dialogRef object and assign it to the dialog.open() function
      * const dialogRef = this.dialog.open(BookDetailsDialogComponent, {})
      * In the body of the open() function create a new object literal called data and a nested object literal called book and assign it the book variable
        + data: { book: this.book }
      * Set the disableClose to true
      * Set the width to 800px
    - Call the afterClosed() function and set the book variable to null
      * dialogRef.afterClosed().subscribe(result => { if (result === ‘confirm’) { this.book = null; } });
* book-details-dialog.component.ts
  + Add an import statement for MAT\_DIALOG\_DATA and MatDialogRef
    - import { MAT\_DIALOG\_DATA, MatDialogRef } from ‘@angular/material/dialog’;
  + Add an import statement for Inject
    - import { Inject } from ‘@angular/core’;
  + Add an import statement for IBook
    - import { IBook } from ‘../book.interface.ts’;
  + Add the MatDialogRef and MAT\_DIALOG\_DATA to the components constructor
    - constructor(private dialogRef: MatDialogRef<BookDetailsDialogComponent>, @Inject(MAT\_DIALOG\_DATA) data) {}
  + Add a variable named book of type IBook
  + In the components constructor map the data.book value to the book variable
    - this.book = data.book;
* Book-details-dialog.component.html
  + Add mat-card with a title, content and actions
    - <mat-card><mat-card-title></mat-card-title><mat-card-content></mat-card-content><mat-card-actions></mat-card-actions></mat-card>
  + The title should display the book’s title and set the font color to blue
    - {{ book.title }}
  + For the mat-card-content add an Flex column layout and use paragraph tags to output each value of the book object
    - <div fxLayout=”column”>
  + For the mat-card-actions, add a single mat-raised button, set the color to accent, and add the matDialogClose directive to ‘confirm’
    - <button mat-raised-button color=”accent” matDialogClose=”confirm”>OK</button>
* Run and test the application
  + Verify you can select books and a dialog appears with the books details
  + Verify the header of the dialog is blue, button is green, and clicking the button closes the dialog window