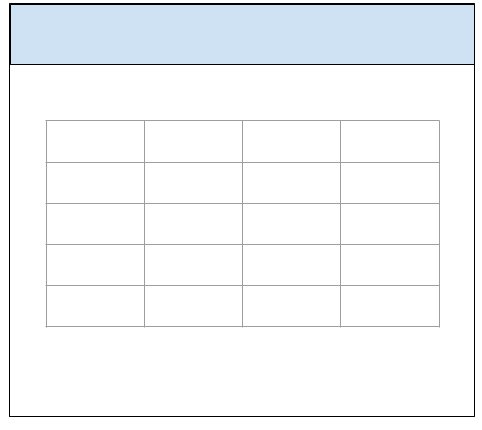
# Instructions – Exercise 5.3 – Data Tables

**Layout**

book-list



in-n-out-books, part 2

**Instructions**

* Make a copy of the in-n-out-booksp1 from Exercise 5.2 and add it to your week-5 directory
* Rename the application to in-n-out-booksp2
* Delete the node\_modules directory
* Delete the package-lock.json file
* Open the angular.json file and find and replace all “in-n-out-booksp1” entries with “in-n-out-booksp2”
* Open the package.json file and change the name to “in-n-out-booksp2”
* 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 “Exercise 5.3 - Data Tables”
* app.module.ts
  + Add an import statement for MatTableModule
    - import { MatTableModule } from ‘@angular/material/table’;
  + Add the MatTableModule to the imports array
* Generate a new Angular service and name it books
  + ng g s books
* books.service.ts
  + Import the IBook interface
    - Import { IBook } from ‘./book.interface’;
  + Import the Observable class from rxjs
    - import { Observable } form ‘rxjs’;
  + Import the of operator from rxjs
    - import { of } from ‘rxjs’;
  + Import the map operator from rxjs
    - import { map } from ‘rxjs/operators’;
  + Add a books variable of type Observable<IBook>
    - books: Array<IBook>
  + In the services constructor, create 5 new book objects and prefill them with your favorite books.
  + Create two new functions: getBooks() and getBook(isbn: string)
  + getBooks()
    - Set the return type to an Observable Array of IBook objects
      * getBooks(): Observable<IBook[]>
    - Cast the books array to an observable and return them
  + getBook(isbn)
    - Set the return type to IBook
      * getBook(isbn: string): IBook
    - Loop over the array of books and return a book matching the passed-in isbn number
* book-list.component.ts
  + Add an import statement for the BooksService
    - import { BooksService } from ‘../books.service’;
  + Add an import statement for IBook interface
    - import { IBook } from ‘../book.interface’;
  + Add an import statement for Observable
    - import { Observable } from ‘rxjs’;
  + Create a books variable of type Observable<IBook[]>
  + Add the BooksService to the components constructor
    - constructor(private booksService: BooksService)
  + In the components constructor call the booksService.getBooks() function and assign the results to the books variable
  + Create a variable named header of type Array<string> and assign it the following values:
    - Isbn, title, numOfPages, authors
  + Create a variable named book of type IBook
  + Create a function called showBookDetails(isbn: string)
    - In the body of the showBookDetails(isbn: string) function make a call to the booksService.getBook(isbn: string) and map the return object to the book variable
    - console.log() the book object (we are doing this to test that the service is returning the correct book object)
* Book-list.component.html
  + Create a Flex row div and wrap the content
    - Using the header and books as data sources, create an Angular data table
    - Give the data table a header <h2> and title of “Top 5 Favorite Books” and a font color of green
    - Using JavaScripts built-in join() function, join the authors array into a comma delimited string
      * {{ element.authors.join(‘, ‘)  }}
  + Wrap the isbn row with an anchor tag and Angular’s click directive with a call to the showBooksDetails(isbn: string) function
    - Add CSS styling for text-decoration: none; color: blue
    - Add CSS styling for hover cursor: pointer; text-decoration: underline