Create the LibraryApp Service

In this section create and move the hardcoded data retrieved from a view-component into a Service class.

1. Open VS Code and LibraryApp2.
2. In the integrated terminal start the app so the LibaryApp Welcome page opens in a browser (and it is also in watch mode).

Ensure that Browse Bools link brings back a list of books.

1. Inside books directory create file book.service.ts and paste in following code:



1. Now remove the list of books from book-list.component.ts:



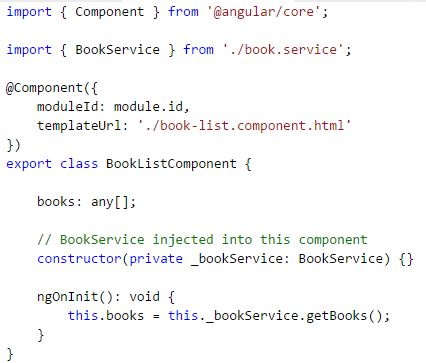
Confirm that ‘Browse Books‘ link does not bring any books back, because BookListComponent has not been linked to BookService.getBooks.

(Note: Cut out the list of books from book-list.component.ts to save on typing)

1. Update app.module.ts to indicate that BookService is a provider (BookService is created here) of injectable components:



1. Modify book-list.component.ts so that the BookService is injected into BookListComponet and calls getBooks() to get all books:

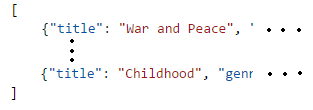


Confirm that ‘Browse Books‘ link does bring back all books – this time via the service (this time hardcoded in the service)

Create Http Observable

This section: move the hardcoded list of books out of the BookService code into a separate mock/test file and us an Http Observable to retrieve the data.

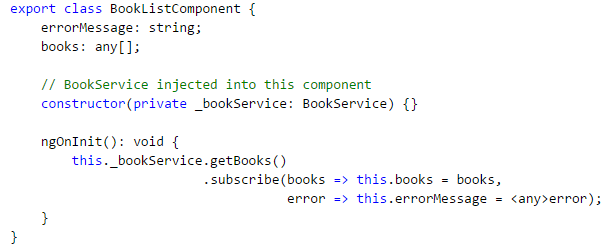
1. Create file/directory src/mock/books/books.json and copy list of books into it:



1. Add the two lines of code in app.module.ts to import the HttpModule (from '@angular/http')
2. Update book.service.ts so that getBooks() calls an Http Observable (similar to Promises) - to the mock file books.json:



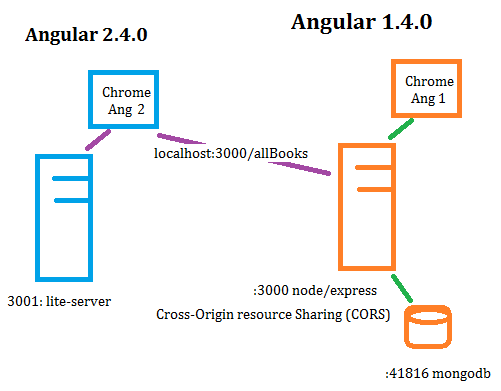
1. Update book-list.component.ts to subscribe to the getBooks() Observable:



Confirm that Browse Books brings back list of books

Connect to Existing DB Service

Next want to access the LibaryApp DB held in our MongoDB. Rather than add a Node-Express-Mongo layer another option is to make use of existing Angular 1.4 LibraryApp’s http service <http://localhost:3000/allBooks>. To do this you need to change the new Angular 2.4.0 LibraryApp to point to url=//[localhost:3000/allBooks](http://localhost:3000/allBooks):



1. In book.service change URL location

private \_bookUrl = '//localhost:3000/allBooks';

1. Navigate to ‘Browse Books’. There should be no books shown.
2. Stop the application (Ctrl C followed by Y)
3. Start mongoDB (mongod)
4. Back In VSC open another VSC window (for Angular 1.4.0 LibraryApp)

File->New Window

1. In newly opened VSC open Angular 1.4 LibraryApp (File->Open Folder…)
2. In terminal windows start the old application:

node app

(Should see message listening on port 3000. Test getting back books in JSON format by typing http://localhost:3000/allBooks)

1. Now switch back to Angular 2.4.0 LibraryApp window
2. Restart Angular 2.4.0

npm start

Angular 2.4.0 LibraryApp should now appear and ‘Browse Books’ is getting back books from DB (in mongod window an update message is displayed)

Note: Application should start but from a different port (3001), as start script detect’s 3000 already in use so assigns next available port.