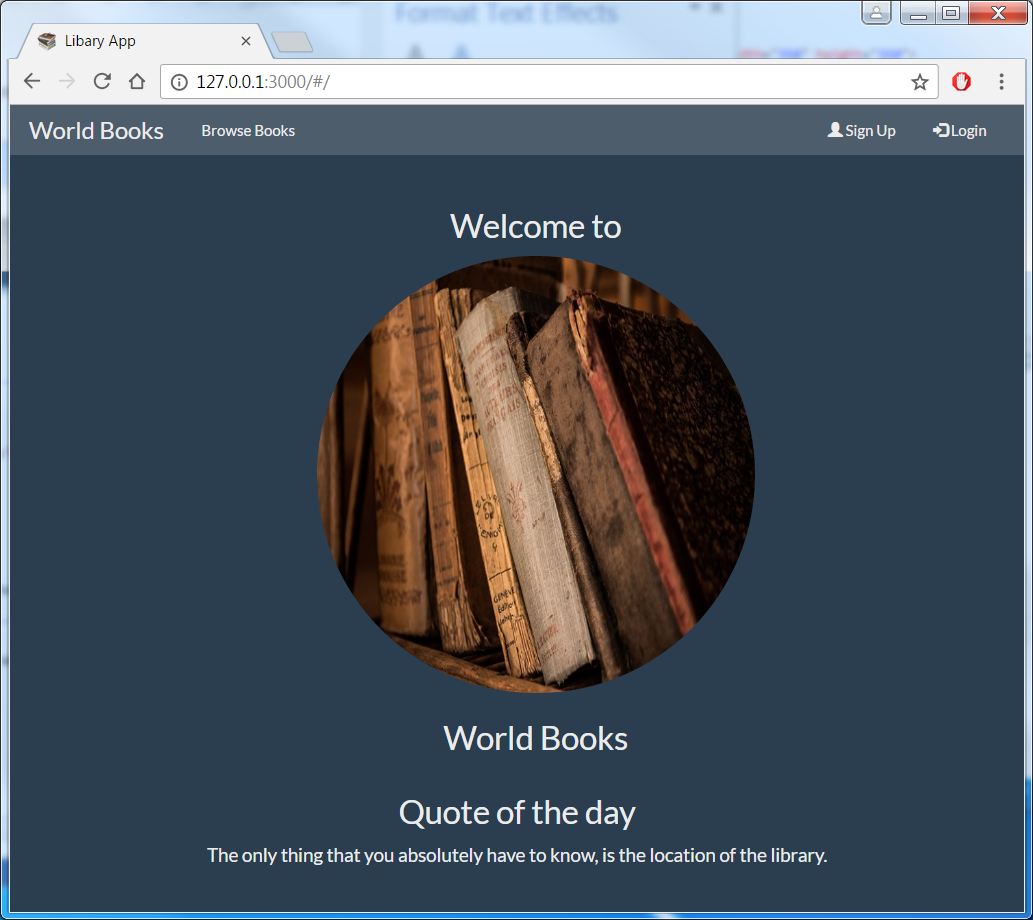
Prerequisite

Make sure you have completed S06 worksheet and have the LibraryApp home page as below or similar depending on your version of Bootswatch:



Simple Hints & Tips to help you work through worksheet.

1. When Word first opens up docs it if often in Read view, switch to Edit view because code is often lost across multiple screens in Read view.
2. Read through each section to understand what is required at the end.
3. Use Visual Studio Code Explorer pane to create directories & files within your app:



1. Change the default ‘Dark’ Colour Theme using File->Preferences->Color Theme if preferred

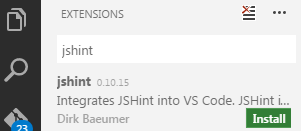
(I change to the Light (Visual Studio) for demos)

1. Let Visual Studio Code help with coding/bugs such as:

* colour syntax
* match brackets
* indent code
* use JSHint (see below)

1. Ensure you are in the correct project directory
2. If you have questions don’t be afraid to speak up.
3. Start mongoDB from outside Visual Studio Code (from Integrated Terminal) because if you close VSC it will close mongoDB too.
4. Try and keep on top of things
5. Note: in browsers localhost is the same as 127.0.0.1
6. Install a JavaScript linting/hinting Extension (this will highlight bad practises/help consistency across the team):

* In VSC click Extensions Icon (5th icon on LHS)
* In search box type jshint (should highlight Dirk Baeumer)
* Click Install



* Using VSC create file .jshintrc (Note file create in any VSC directory because you will move it later, also note file name starts with a dot)
* Add this to its content (including curly brackets):

{

"bitwise": true,

"camelcase": true,

"curly": true,

"eqeqeq": true,

"esversion": 6,

"forin": true,

"freeze": true,

"immed": true,

"indent": 4,

"latedef": "nofunc",

"newcap": true,

"noarg": true,

"noempty": true,

"nonbsp": true,

"nonew": true,

"plusplus": false,

"quotmark": "single",

"undef": true,

"unused": false,

"strict": false,

"maxparams": 10,

"maxdepth": 5,

"maxstatements": 40,

"maxcomplexity": 8,

"maxlen": 120,

"asi": false,

"boss": false,

"debug": false,

"eqnull": true,

"esnext": false,

"evil": false,

"expr": false,

"funcscope": false,

"globalstrict": false,

"iterator": false,

"lastsemic": false,

"laxbreak": false,

"laxcomma": false,

"loopfunc": true,

"maxerr": 50,

"moz": false,

"multistr": false,

"notypeof": false,

"proto": false,

"scripturl": false,

"shadow": false,

"sub": true,

"supernew": false,

"validthis": false,

"noyield": false,

"browser": true,

"node": true,

"globals": {

"angular": false,

"$": false

}

}

* Close VSC.
* Move this file .jshintrc to your default Windows user workspace ie:

C:\Users\<your corp id>

* Finally from a command prompt enter:

npm install jshint -g (note make sure you use

correct ‘-‘, otherwise errors)

* Re-open VSC to ensure it picks up the options file (from any project)

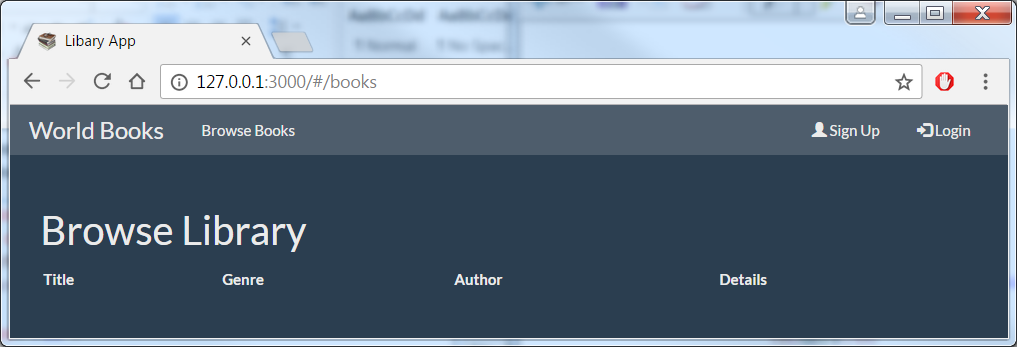
getAllBooks in Angular – 1st Approach

You can get all books in db using <http://127.0.0.1:3000/allBooks> in plain JSON format. We will use Angular next to present results on web page books.html (Angular will make a http request to Express server which will query the DB).

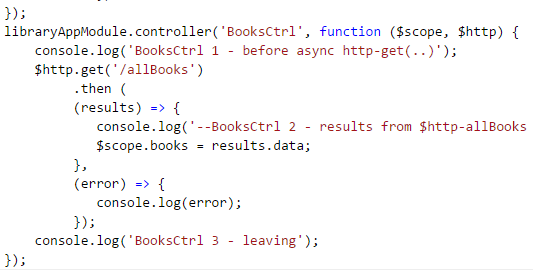
1. Make sure mongoDB is running.
2. Make sure you are getting book details back in JSON format using <http://127.0.0.1:3000/allBooks>
3. Open Visual Studio Code in D:\Projects\angular1-library (S06 workspace)
4. Within app/views create file books.html
5. Paste following into books.html



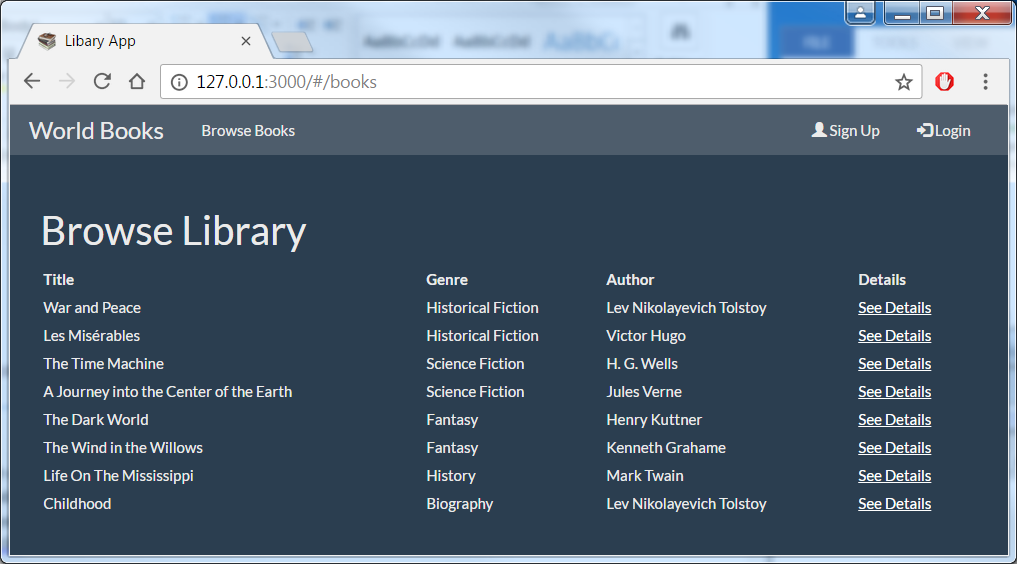
1. Confirm $routeProvider for ‘books’ (from S06 worksheet) in .config of Angular module ‘libraryApp’ is routing correctly by clicking on ‘Browse Books’ on navigation bar, as below (no books returned) NOTE: ‘#’ indicating this is Angular using SPA (not Express)



1. In index.html add the Controller version doing a direct $http call to server just below .config(..});



1. Refresh your browser and confirm you are now getting books back and showing using Angular:



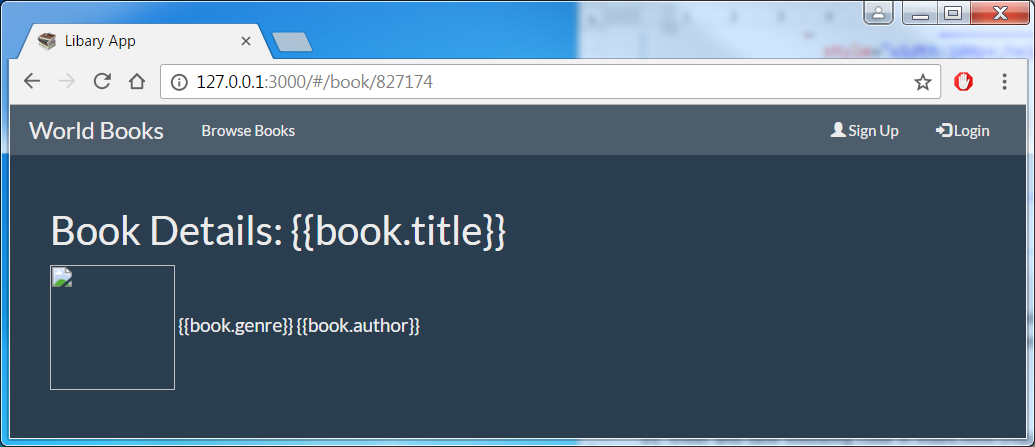
getBook(id) – Delegate to Service Layer (user Promise)

You can get information about a particular book from Goodreads by for example <http://127.0.0.1:3000/book/99152> to get back image location and further information about the book in JSON format. This section will integrate this into the app by delegating a call to the Service Layer call from Angular’s Controller and display in book.html.

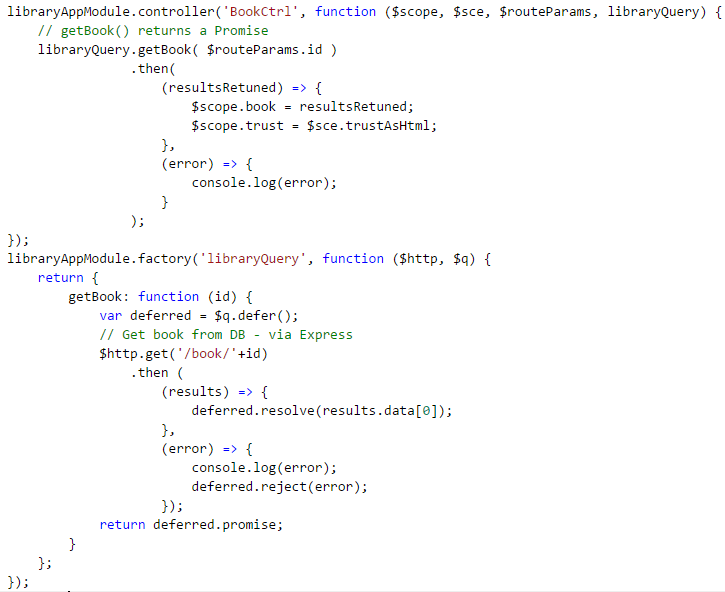
1. Confirm you are getting image and detail information back in JSON format from Goodreads: <http://127.0.0.1:3000/book/99152>
2. Within app/views create file book.html
3. Paste following code into book.html:



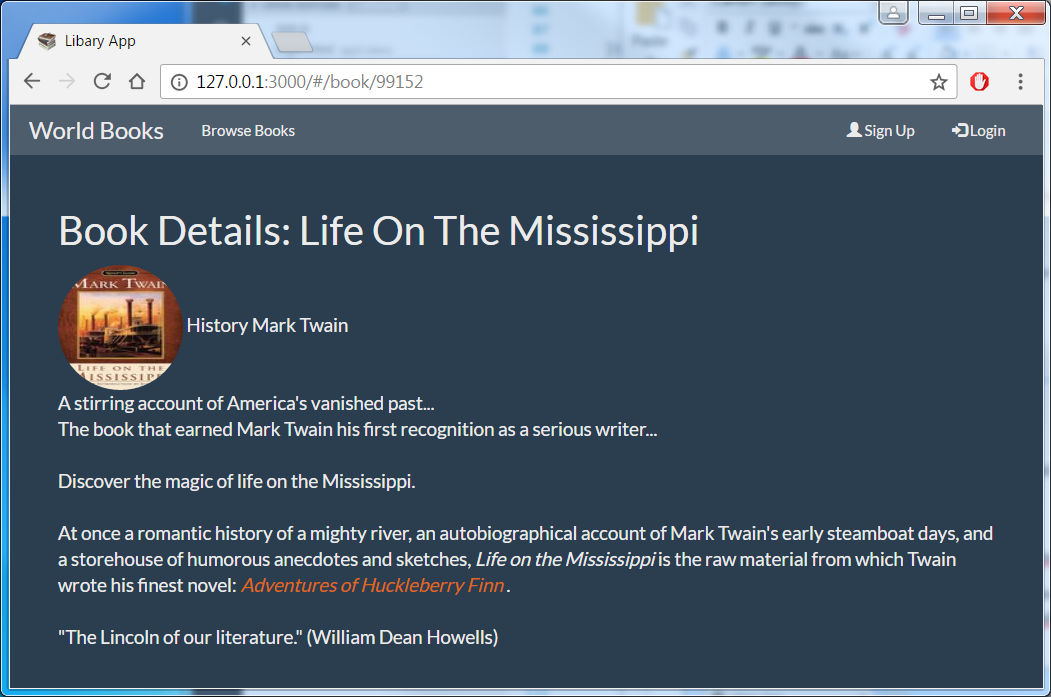
1. Confirm $routeProvider ‘book’ in .config of Angular’s libraryApp module routing as expected - although no data pulled back yet - by clicking on any of the ‘See Details’ links from books page:



1. Add improved design of Controller and Service Layer delegating function that returns a user defined Promise as well as $http Promise in index.html:



1. Confirm Controller and Service Layer are working together by clicking on the Detail View links.



1. Notice how the detail section had been formatted because of using $sce.trustedAsHtml

getAllBooks in Angular – Refactor BooksCtrl

1. Using your understanding of getBook(id) refactor getAllBooks() so that ‘Book**s**Ctrl’ makes a Service Layer call to get all books rather than a direct $http call.

* Hints/Tips: Comment out your existing getAllBook()/BooksCtrl code as a backup
* Use section 13 above as a template
* In factory ‘libraryQuery’ (in section 13) add your own getAllBooks