MDN Web Docs moz://a

Express Tutorial Part 5: Displaying library data

We're now ready to add the pages that display the LocalLibrary website books and other data. The pages will include a home page that shows how many records we have of each model type and list and detail pages for all of our models. Along the way, we'll gain practical experience in getting records from the database, and using templates.

Prerequisites:	Complete previous tutorial topics (including Express Tutorial Part 4: Routes and controllers).
Objective:	To understand how to use the async module and Pug template language, and how to get data from the URL in our controller functions.

Overview

In our previous tutorial articles, we defined Mongoose models that we can use to interact with a database and created some initial library records. We then created all the routes needed for the LocalLibrary website, but with "dummy controller" functions (these are skeleton controller functions that just return a "not implemented" message when a page is accessed).

The next step is to provide proper implementations for the pages that *display* our library information (we'll look at implementing pages featuring forms to create, update, or delete information in later articles). This includes updating the controller functions to fetch records using our models and defining templates to display this information to users.

We will start by providing overview/primer topics explaining how to manage asynchronous operations in controller functions and how to write templates using Pug. Then we'll provide implementations for each of our main "read-only" pages with a brief explanation of any special or new features that they use.

At the end of this article, you should have a good end-to-end understanding of how routes, asynchronous functions, views, and models work in practice.

Displaying library data tutorial subarticles

The following subarticles go through the process of adding the different features required for us to display the required website pages. You need to read and work through each one of these in turn, before moving on to the next one.

Summary

We've now created all the "read-only" pages for our site: a home page that displays counts of instances of each of our models, and list and detail pages for our books, book instances, authors, and genres. Along the way, we've gained a lot of fundamental knowledge about controllers, managing flow control when using asynchronous operations, creating views using *Pug*, querying the site's database using models, passing information to a view, and creating and extending templates. The challenges will also have taught readers a little about date handling using *Luxon*.

In our next article, we'll build on our knowledge, creating HTML forms and form handling code to start modifying the data stored by the site.

See also

- Async module (Async docs)
- Using Template engines with Express (Express docs)
- Pug (Pug docs)
- Luxon (Luxon docs)

In this module

Last modified: Oct 27, 2020, by MDN contributors

Change your language

English (US) Change language