

COMP 2406

Winter 2020 - Tutorials #9

Objectives

- Add a validation function to an existing Mongoose schema
 - Add a static method to an existing Mongoose schema
 - Practice forming basic queries and processing data using Mongoose
-

Problem Background

This tutorial will involve adding more features to an existing Mongoose schema. Download the T09-BaseCode.zip file from cuLearn to get started. This zip file contains the following files:

1. `books.json` – A file containing the necessary book data for the database-initializer.js file to run.
2. `BookModel.js` – A file defining a Mongoose schema for books.
3. `database-initializer.js`: Run this file while the Mongo daemon is running and it will create a database called 't9' containing the dataset of books that will be used for this tutorial.

Problem 1 (Required Fields)

Currently, the schema in `BookModel.js` contains only a type definition for each field. Modify this schema so that all fields, except for ratings, are marked as required fields.

Problem 2 (ISBN Validation)

For this problem, you must implement a validation function and add it to the book schema so that ISBN values are validated before being saved. The rules for ISBN validation are outlined below:

1. If the string representing the book's ISBN does not have 10 characters, then the ISBN should be considered invalid.
2. If the book's ISBN contains 10 characters, then the ISBN should be considered valid if:
 - a. The ISBN contains all digits (0-9), except for the last character, which may be a digit or the character 'X'.
 - b. The checksum value divided by 11 produces no remainder. To calculate the checksum value for an ISBN, multiply the first digit in the

ISBN by 10. Add to that the second digit multiplied by 9, third digit by 8, and so on until you multiply the last digit by 1. If the last character in the ISBN is an X, it should be given the value 10. So, if the ISBN is 039925675X, the checksum would be $(10*0) + (9*3) + (8*9) + (7*9) + (6*2) + (5*5) + (4*6) + (3*7) + (2*5) + (1*10) = 264$. Since 264 is evenly divisible by 11, this is a valid ISBN.

Once you have implemented the validation function for the schema, run the database-initializer.js file and it will try to add all books to the database. If your implementation is correct, you should get 2689 books added successfully and 10 books that failed the validation check.

Problem 3 (Adding a Static Method)

Add a static method to the book schema called `getTopBooks`. Before getting started, it is worth noting that it may be necessary to implement this function in a seemingly inefficient way (e.g., finding the average rating of all books and sorting).

This method must query the database and return an array of the ten highest rated books. The rating for a book should be calculated as the average of all ratings in the ratings array for that book. Since we are also interested in popular books, any books with 10 or less ratings should be ignored. If you run this method with the provided dataset, you should get something similar to the following books (note: there is a multi-way tie for the later books, so as long as you have something with the same average rating, everything should be okay):

1. The Narrow Road to the Deep North - 4.181818181818182
2. The Magical Worlds of Harry Potter: A Treasury of Myths, Legends, and Fascinating Facts - 4.125
3. The Return of the Native - 4.071428571428571
4. An Ember in the Ashes (An Ember in the Ashes, #1) - 4
5. Dog on It (A Chet and Bernie Mystery #1) - 3.9473684210526314
6. Identical - 3.9285714285714284
7. The Motorcycle Diaries: Notes on a Latin American Journey - 3.9166666666666665
8. Metaphysics - 3.9166666666666665
9. The Bloodletter's Daughter: A Novel of Old Bohemia - 3.909090909090909
10. Kabul Beauty School: An American Woman Goes Behind the Veil - 3.888888888888889

The method should also accept a callback function as an input parameter. When the method terminates, either because it has found the correct result data or experienced

an error, it should call the callback function with the appropriate data (the selected books) as a parameter.