

Optimize the query

The image shows a dual-screen setup. The left screen displays the MongoDB Intro Lab page for 'Optimize the Query'. The right screen shows a code editor with a TypeScript controller and its runtime errors.

MongoDB Intro Lab: Optimize the Query

Introduction

MongoDB Atlas >

Importing Data >

Open the Application >

Schema Validation >

Indexing >

Optimize the Application >

Intro

Patterns Used

Optimize the Query

Troubleshooting the application >

Summary

Now that you know of the patterns that we used in our data schema, you can rewrite the `getbook` method to retrieve the book information.

Open the `server/src/controllers/books.ts` file, and look for the `getbook` method. How could you rewrite this query to be blazing fast, and meet the requirements that we defined for this application?

TIP

How many calls do you need to do to your database now? Do you still need an aggregation pipeline? Look in the documentation for the `findOne` method.

[Click here to see the answer](#)

[Edit this page](#)

Previous [Patterns Used](#) Next [Troubleshooting the application](#)

Try MongoDB Atlas · Forums · Developer Center · MongoDB University · This lab in GitHub · © 2024 MongoDB, Inc.

books - library-management

```
server > src > controllers > books > getbook
1 import { DeleteResult, InsertOneResult, UpdateResult } from 'mongodb';
2 import { Book } from '../models/book';
3 import { collections } from '../database.js';
4 // eslint-disable-next-line @typescript-eslint/no-unused-vars
5 import getEmbeddings from '../embeddings/index.js';
6
7 class BookController {
8   errors = {
9     UNKNOWN_INSERT_ERROR: 'Unable to create book',
10    UNKNOWN_UPDATE_ERROR: 'Unable to update book',
11    UNKNOWN_DELETE_ERROR: 'Unable to delete book',
12    NOT_FOUND: 'Book not found',
13    DETAILS_MISSING: 'Book details are missing',
14    BOOK_ID_MISSING: 'Book id is missing',
15    ADMIN_ONLY: 'This operation is only allowed for admins',
16    NOT_AVAILABLE: 'Book is not available'
17  };
18
19   success = {
20     CREATED: 'Book created',
21     UPDATED: 'Book updated',
22     DELETED: 'Book deleted'
23   };
24
25   public async getBook(bookId: string): Promise<Book> {
26     /**
27      * Initial Code
28      * Task: Optimise the query to take advantage of the already computed field.
29      * Hint: Take a look at the shape of the Book documents using MongoDB Compass.
30     */
31   }
```

PROBLEMAS SAÍDA CONSOLE DE DEBUGAÇÃO TERMINAL PORTAS COMENTÁRIOS

```
at resolvePath (file:///home/codespaces/.npm/_npx/645672645557e6/node_modules/tsx/dist/esm/index.mjs:17367968:17996:2:4073)
at resolve (file:///home/codespaces/.npm/_npx/645672645557e6/node_modules/tsx/dist/esm/index.mjs:17367968:17996:2:4447)
at nextResolve (node:internal/modules/esm/hooks:366:38)
at Hooks.resolve (node:internal/modules/esm/hooks:384:38) {
  code: 'ERR_MODULE_NOT_FOUND',
  url: 'file:///workspaces/library-management-system/src/indexing/borrowed-books-index.ts'
}

Node.js v20.17.0
@RafaelGallo → /workspaces/library-management-system (main) $
```