Index.js

const express = require("express");

const mongoose = require("mongoose");

const cors = require("cors");

const bodyParser = require("body-parser");

const handlers = require("./controller/bookCartHandlers");

const server = express();

server.use(cors());

server.use(bodyParser());

server.get("/", (req, res, next) => {

res.send("hello world");

})

server.get("/books", handlers.getBooks);

server.get("/book/:isbn", handlers.getBook);

server.post("/book", handlers.createBook);

server.put("/book", handlers.updateBook);

server.delete("/book/:isbn", handlers.deleteBook);

mongoose.connect("mongodb://127.0.0.1:27017/bookdb")

.then((d) => {

server.listen(3000, () => {console.log("Server listening at port: 3000")});

})

.catch((err) => {

console.log("ERROR: " + err);

});

controller/bookCartHandlers.js

const Books = require("../models/book")

const handlers = {

getBooks: async (req, res, next) => {

let data = [];

try{

data = await Books.find();

res.status(200).json(data);

} catch(err) {

res.status(500).json({

error: err

});

}

},

getBook: async (req, res, next) => {

try{

await Books.findOne({isbn: req.params.isbn})

.then((book) => {

if(book !== null)

res.json(book);

else

res.json({

"message": "No book found with ISBN: " + req.params.isbn

});

});

} catch (err) {

res.status(500).json({

error: err

});

}

},

createBook: async (req, res, next) => {

await Books.create(req.body)

.then((data) => {

res.json(data);

})

.catch((err) => {

if(err.errors)

res.json({

error: err.errors.name.message

});

else

res.json({

error: "ISBN must be unique"

});

})

},

deleteBook: async (req, res, next) => {

try{

await Books.findOne({isbn: req.params.isbn})

.then((book) => {

if(book !== null)

Books.deleteOne({isbn: req.params.isbn})

.then((data) => {

res.json({

"message": "Deleted the book with ISBN: " + req.params.isbn + " successfully"

})

});

else

res.json({

"message": "No book found with ISBN: " + req.params.isbn

});

});

} catch(err) {

res.status(500).json({

error: err

});

}

},

updateBook: async (req, res, next) => {

try{

await Books.findOne({isbn: req.body.isbn})

.then((book) => {

if(book !== null)

Books.updateOne({isbn: req.body.isbn}, req.body)

.then((data) => {

res.json({

message: "Edited the book with ISBN: " + req.body.isbn + " successfully"

});

});

else

res.json({

"message": "No book found with ISBN: " + req.params.isbn

});

});

} catch(err) {

res.status(500).json({

error: err

});

}

}

}

module.exports = handlers;

models/book.js

const { Schema, default: mongoose } = require("mongoose");

const Book = new Schema({

isbn: {type: String, required: true, unique: true},

name: {type: String, required: true},

author: {type: String, required: true},

price: {type: Number, required: true}

});

module.exports = mongoose.model("book", Book);