



# JAVASCRIPT DEV TEST













## DOCUMENTATION

LE THI NHU Y | [y.contact.here@gmail.com](mailto:y.contact.here@gmail.com) | 2023-07-26

# Phần I: Chuẩn bị

## I. Elasticsearch server và Kibana server

Để có thể chạy được ứng dụng, trước hết cần đảm bảo máy tính của Developer **đã cài đặt** Elasticsearch Docker và **khởi động** server Elasticsearch cũng như Kibana:

<input type="checkbox"/>	Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
<input type="checkbox"/>	 fastify-elasticsearch-js-demo		Running (2/2)	47.63%		47 minutes ago	  
<input type="checkbox"/>	 kb-container b1fc73e06097	docker.elastic.co/kibana/kibana:7.17.10	Running	0.99%	5601:5601	47 minutes ago	  
<input type="checkbox"/>	 es-container 6f7a1368efbd	docker.elastic.co/elasticsearch/elasticsearch	Running	46.64%	9200:9200 <a href="#">Show all ports (2)</a>	47 minutes ago	  

Việc server Elasticsearch đã được khởi động là một trong những giả sử được đưa ra trong đề bài. Nên trong documentation này sẽ không hướng dẫn lại việc cài đặt Elasticsearch và Kibana thông qua Docker.

## II. Clone repository

Clone repository chứa project và toàn bộ file config liên quan ở địa chỉ:

<https://github.com/YLeNhu/jastify-elasticsearch-js-test.git>

Mở folder project bằng IDE chuyên dùng

Sau đó chạy câu lệnh “**npm install**” để build node\_modules

## III. Khởi tạo dữ liệu mẫu

Để phục vụ cho việc truy xuất dữ liệu sách, ta tạo ra một số dữ liệu mẫu trên elastic server bằng cách:

**Bước 1:** Di chuyển đến thư mục src/elasticsearch

**Bước 2:** Tại thư mục này, mở terminal của OS. Và chạy các câu lệnh trong file initialData.md

```
D:\INTERVIEW\js-dev-test\jastify-elasticsearch-js-test\src\elasticsearch>curl "localhost:9200/_cat/indices?v"
health status index      uuid                pri rep docs.count docs.deleted store.size pri.store.size
yellow open   bookstore          TNz4aIlbQfuT2VcPySEphw  1  1         8             0      18.3kb      18.3kb
green  open   .apm-custom-link    Uum2WC6MRUKLwlv1BSuFHg  1  0          0             0       227b       227b
green  open   .apm-agent-configuration DAcXhCq0R9Op2s2Kw5-lWQ  1  0          0             0       227b       227b
green  open   .kibana_7.17.10_001 NmlWEEfdvRy6eyPmgKp4t7g  1  0         47            10       2.4mb       2.4mb
green  open   .kibana_task_manager_7.17.10_001 RzdteYJ8Tvu01jHPDnda1Q  1  0         17           2434       3.2mb       3.2mb
green  open   .tasks              40S3H2u2Sli9x0jG5JUIVw  1  0          6             0      35.1kb      35.1kb
```

**Hình:** Kết quả trả về cho thấy một index “bookstore” đã được tạo mới 8 documents mẫu.

## Phần II: Khởi động project

Sau khi đã hoàn thành xong bước chuẩn bị, ta có thể khởi động dự án theo các bước sau:

**Bước 1:** Chạy lệnh “**npm run server**” để khởi động server. Server được khởi động thành công sẽ có thông báo sau:

```
PS D:\INTERVIEW\js-dev-test\jastify-elasticsearch-js-test> npm run server

> fastify-elasticsearch-js@0.0.0 server
> nodemon src/server/server.js

[nodemon] 2.0.22
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node src/server/server.js`
[21:51:35.369] INFO (28212): Connected to database
● [21:51:35.376] INFO (28212): User routes registered
[21:51:35.481] INFO (28212): Server listening at http://0.0.0.0:8080
```

Địa chỉ server: **http://localhost:8080**

**Bước 2:** Chạy lệnh “**npm run dev**” để khởi động website.

Các chức năng mà trang web cung cấp:

- Hiển thị thông tin liên quan đến sách.
- Hiển thị mục bình luận
- Gửi bình luận mới
- Reply bình luận có sẵn (nesting 1 bậc, theo scope của đề bài)
- Xóa bình luận của bản thân

## Phần III: Kiểm thử

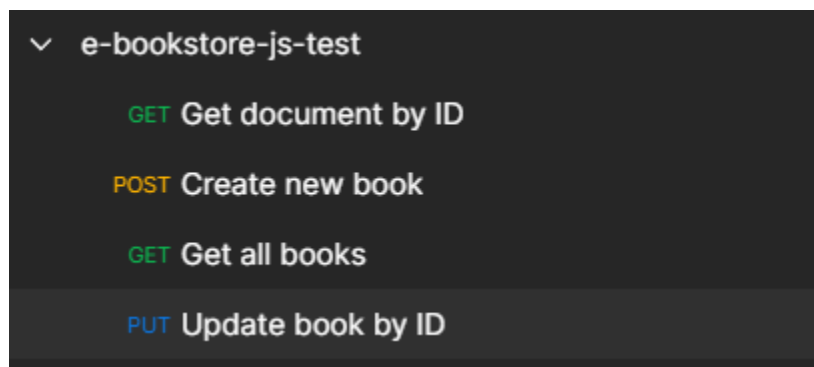
### I. Về phía server

Có thể sử dụng Postman để kiểm thử các API của server:

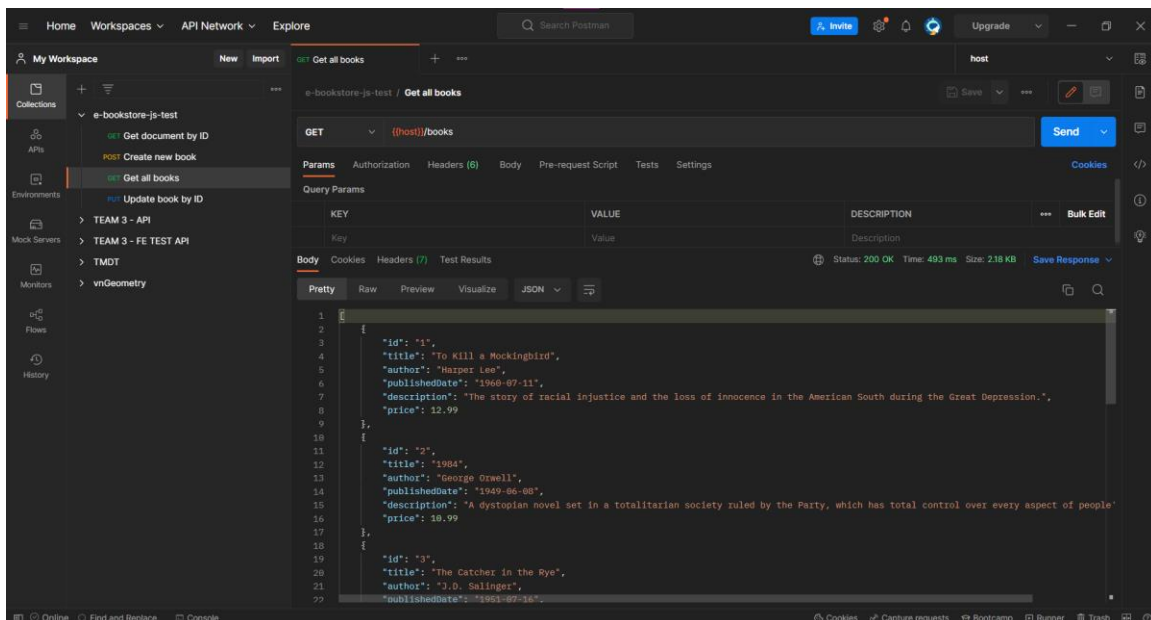
**Bước 1:** Mở Postman, khởi tạo environment mang tên host, giá trị khởi tạo = <http://localhost:8080>

host				
	VARIABLE	TYPE ①	INITIAL VALUE ①	CURRENT VALUE ①
<input checked="" type="checkbox"/>	host	default ▾	http://localhost:8080	http://localhost:8080
	Add a new variable			

**Bước 2:** Tạo 1 collection mới, set environment cho collection này là host ở bước 1, thêm vào các request phù hợp:

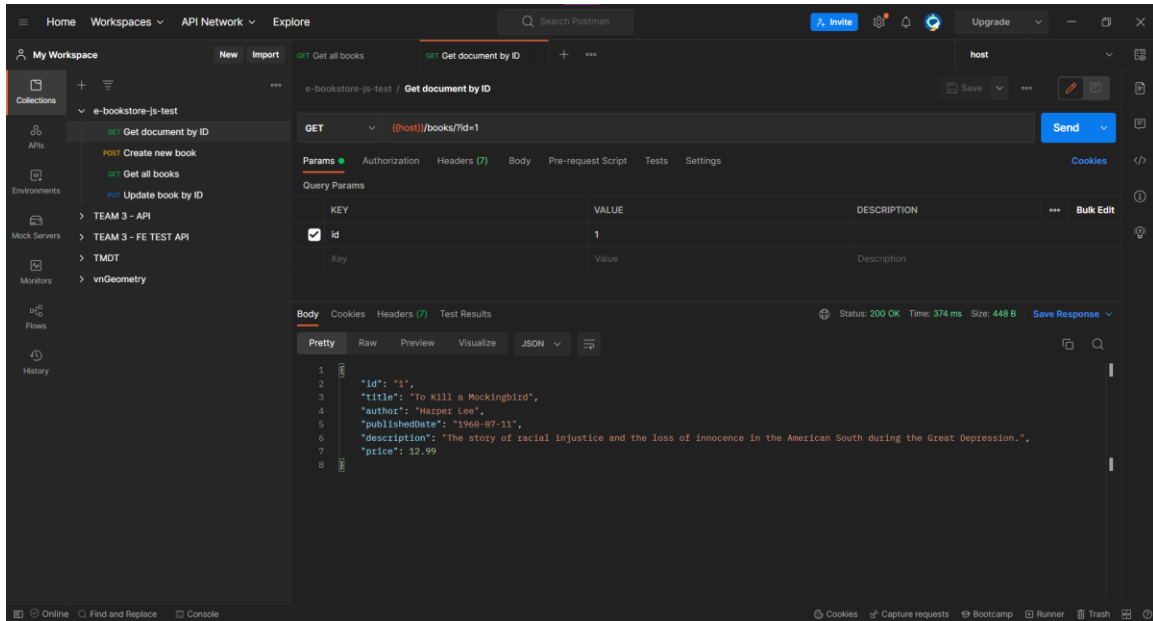


**Bước 3:** Kiểm thử API 'GET/books/':



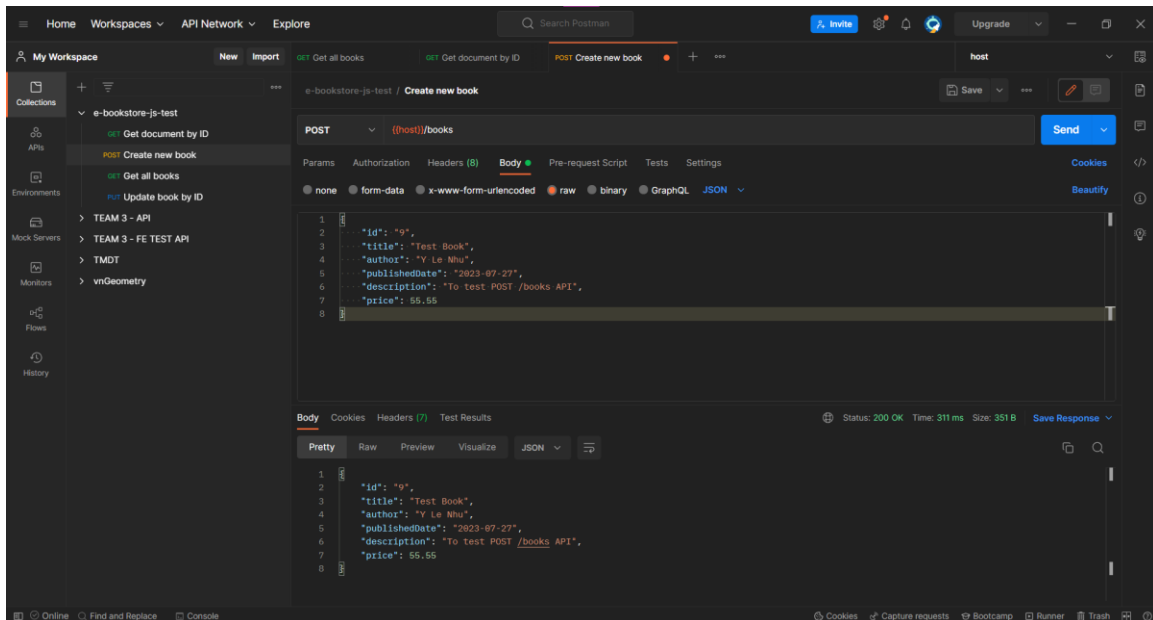
Kết quả trả về toàn bộ dữ liệu sách hiện có.

#### Bước 4: Kiểm thử API 'GET /books/:id':



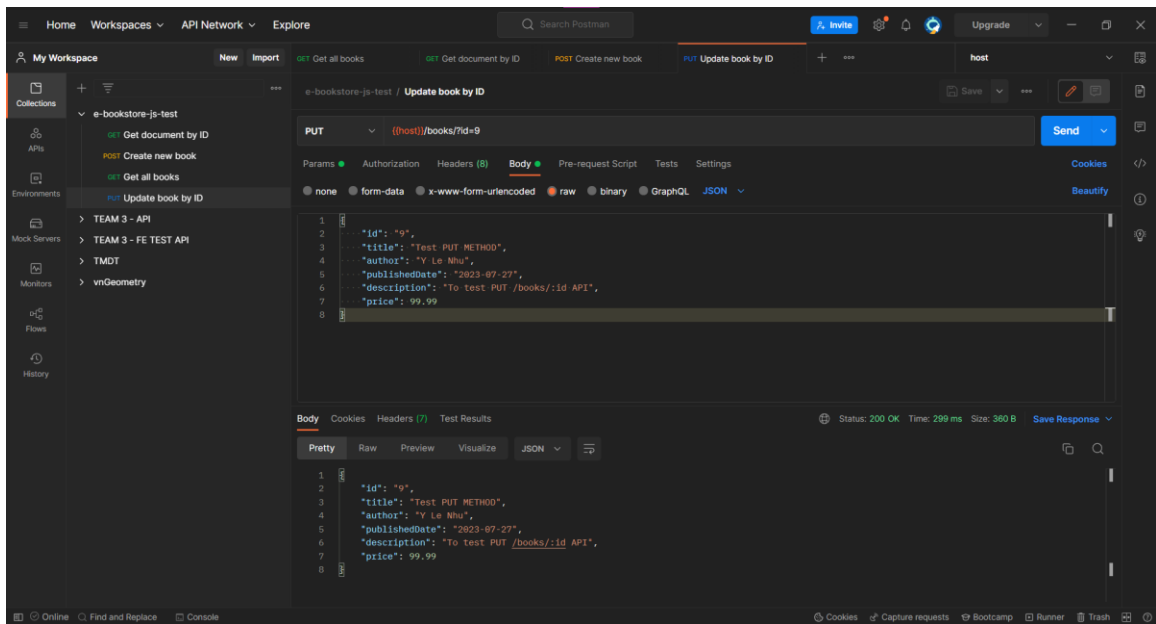
Kết quả trả về sách có ID tương ứng.

#### Bước 5: Kiểm thử API 'POST /books':

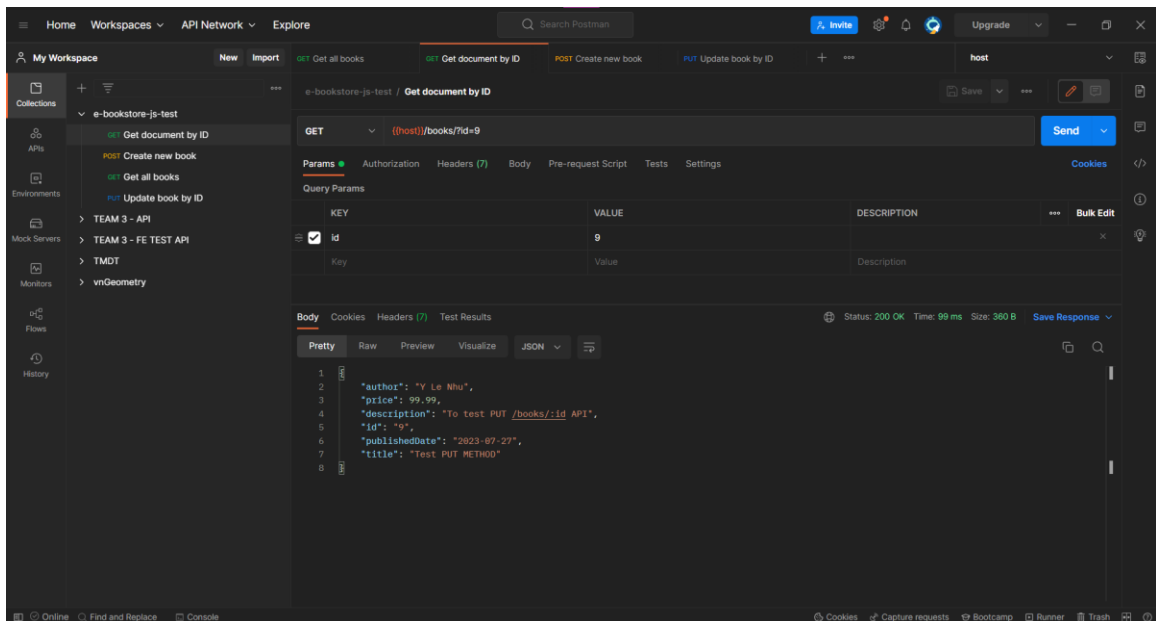


Kết quả trả về body đã gửi đi

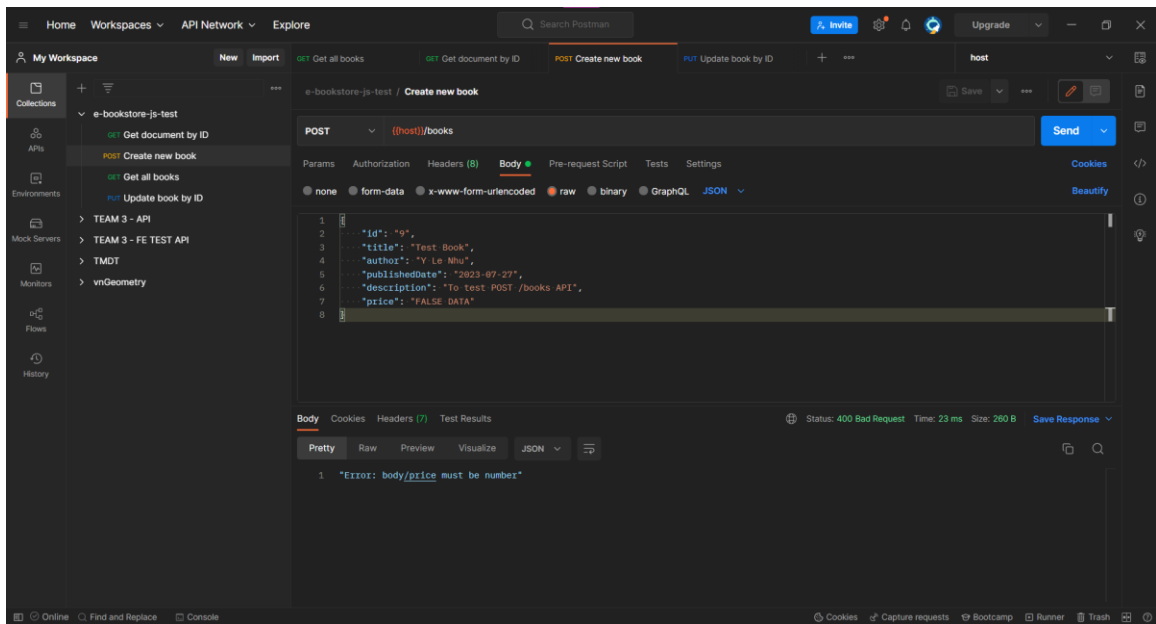
#### Bước 6: Kiểm thử API 'PUT /books/:id':



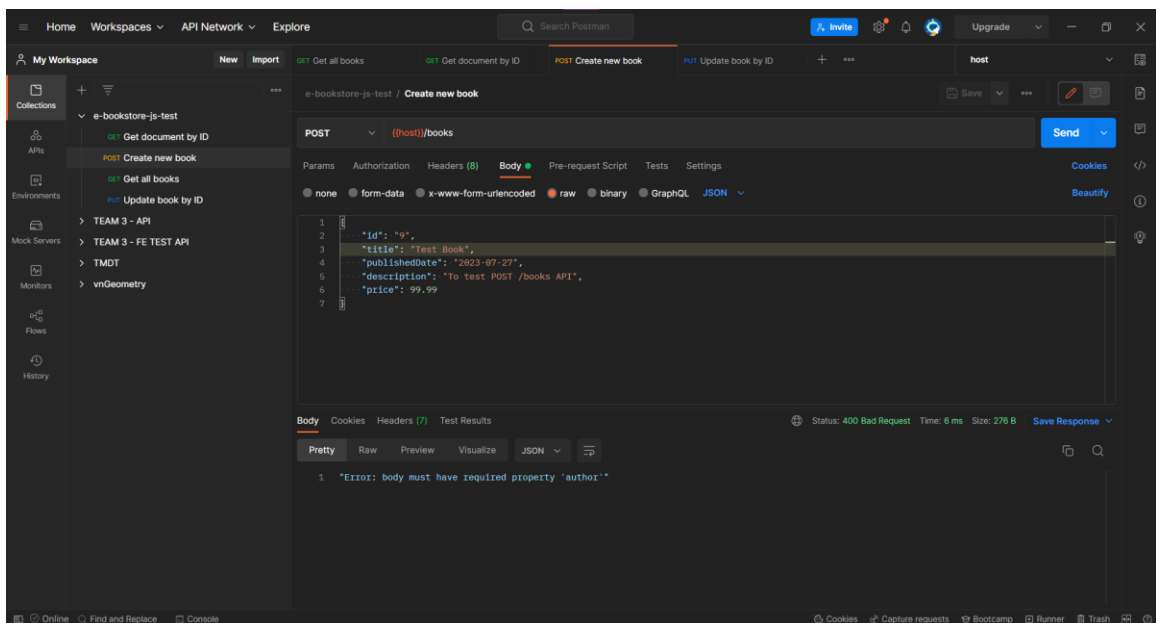
Kiểm tra kết quả bằng API 'GET /books/:id' với id = 9



**Bước 7:** Kiểm tra thao tác validation:



Hình: Trường hợp sai định dạng dữ liệu

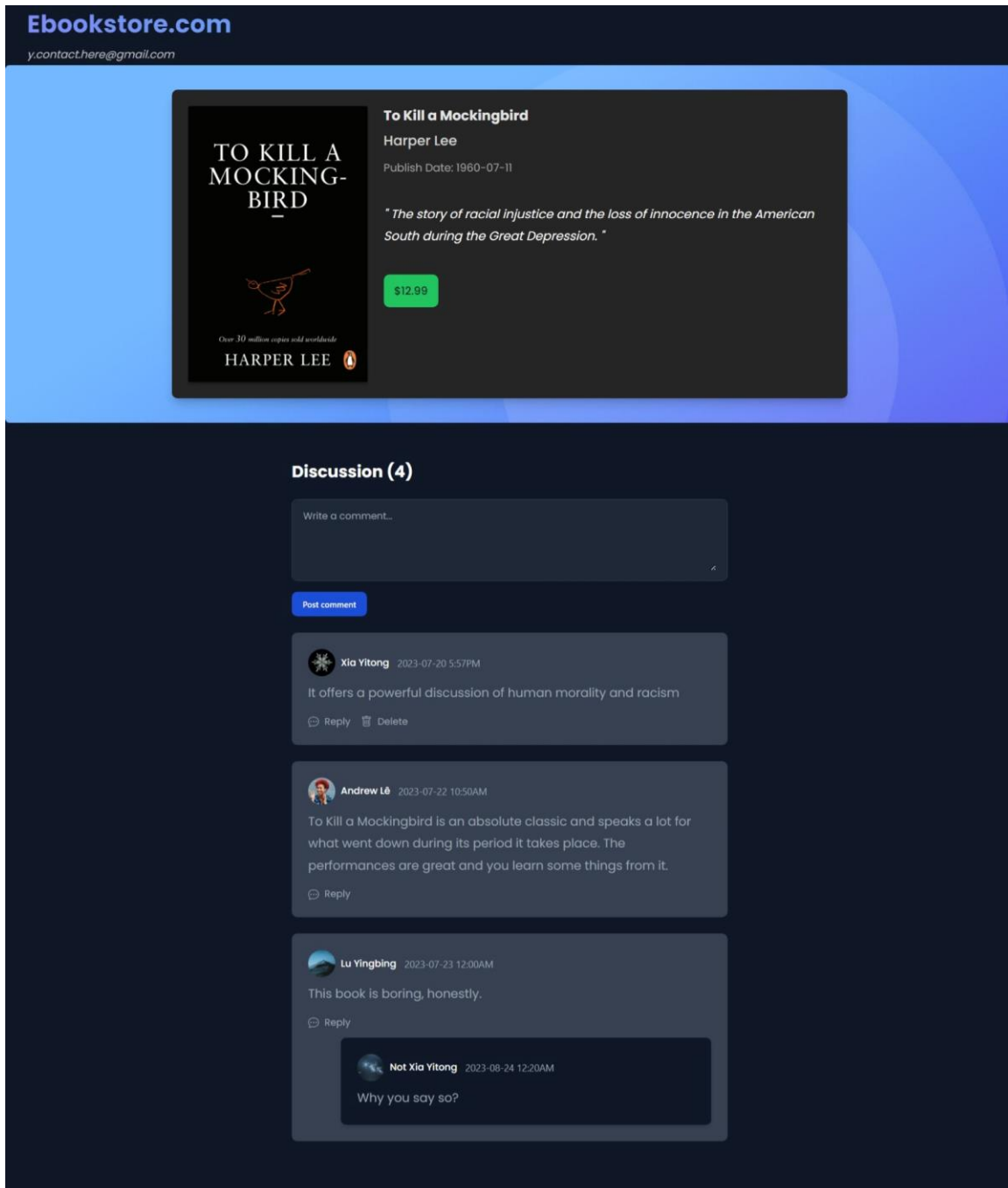


Hình: Trường hợp sai thiếu trường dữ liệu.

Kết quả trả về lỗi tương ứng.

## I. Về phía website

Website được khởi tạo thành công mang giao diện sau:



Website cung cấp những tính năng sau:

- Hiển thị thông tin về sách
- Hiển thị khu vực bình luận



- Cho phép đăng bình luận mới
- Cho phép xóa bình luận của bản thân đã đăng
- Reply bình luận sẵn có (1 level nesting theo scope của đề bài)

*Chân thành cảm ơn quý công ty đã tạo cơ hội cho lần test này!*