|  |  |  |  |
| --- | --- | --- | --- |
|  | Bookstore Application Guide | |  |
| Version 1 | |
| 14/09/2024 | |
|  | DXC Technical Assessment | |  |
| Yong Wei |  |  |
|  | |  |

[Pre-Requisite 1](#_Toc1307087146)

[Setting Up 2](#_Toc250344091)

[Running on H2 database 2](#_Toc441608919)

[Running the application 2](#_Toc1085297625)

[Connecting to microsoft sql server 3](#_Toc1775044115)

[Application endpoints 4](#_Toc469755166)

[Find All Book 4](#_Toc1716156297)

[Add Book 5](#_Toc1278852059)

[Update book 6](#_Toc1300051602)

[Find book by title and name 7](#_Toc364806190)

[Delete book 8](#_Toc1538699081)

# Pre-Requisite

* Java version 17
* Mvn version 3.9+
* Microsoft sql server 2022
* Postman – to test the different endpoints
* Optional: Sql Server Management Studio to view the database

# Setting Up

Below is a demonstration on how to set up the applicaition.

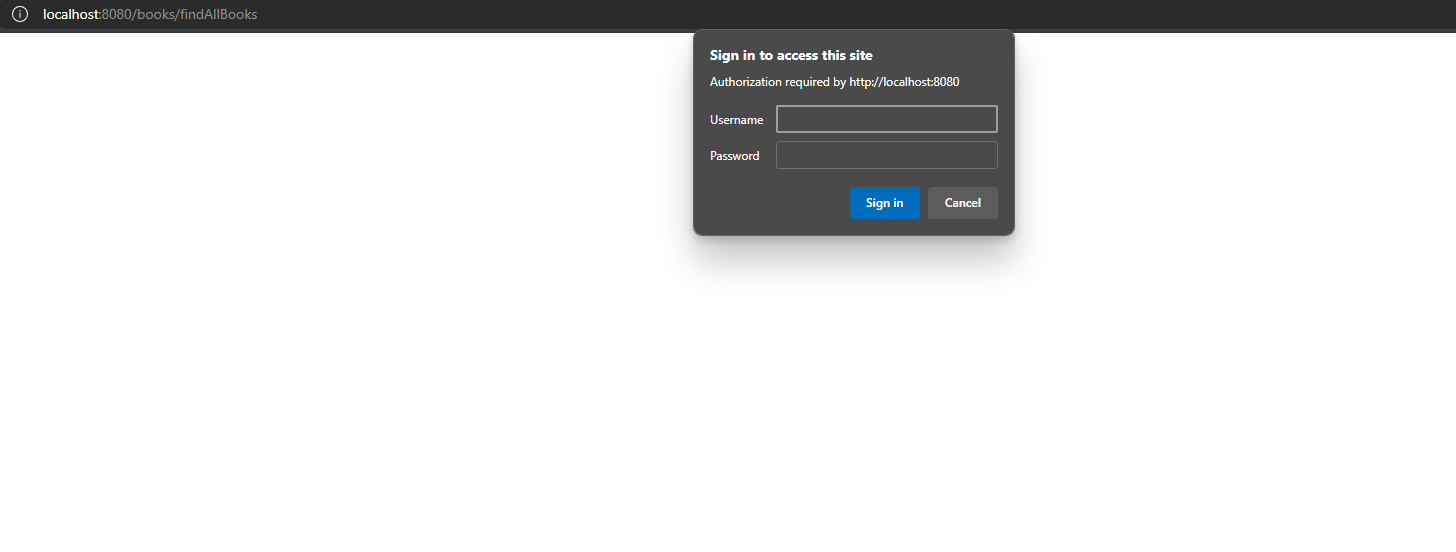
1. Download and Extract Bookstore\_Application to an empty folder
2. Open up command terminal and run the following commands
   1. Mvn install – to install all the dependencies
   2. Mvn compile – to compile and build the source code
3. After installing and compiling the source code, run mvn spring-boot:run to automatically start the server.

# Running on H2 database

The application comes with 2 profiles, dev and prod. By default, it should be running on the dev profile which runs on the n memory h2 database. We can change the profile to prod profile to run the application on Microsoft SQL server. Here we will show you how to run and test the different endpoints on the in memory H2 database.

## Running the application

1. We will run the application by running mvn spring-boot:run to start up the application. this will automatically populate all the dummy data in the schema and data.sql files found in the resource/H2\_ddl folder
2. We can test the resource by going to this link, <http://localhost:8080/books/findAllBooks>



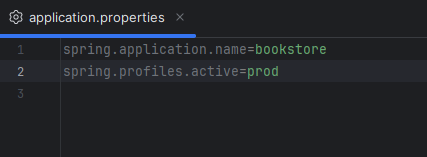
1. It will prompt you to log in since all resources are protected. key in user1 and password1 to log in.
2. After logging in you should be able to see the list of books in populated into the h2 database



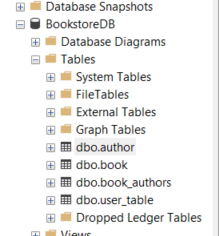
# Connecting to microsoft sql server

Rather than using the in-memory database, we can also configure the application to use Microsoft SQL server database. Before connecting to Microsoft SQLserver, we need to ensure that we set up the users and database first. Go to your Microsoft SQL server and create a username: bookstoreUser, password: password1 and database: BookstoreDB. Grant the user all the rights to this database.

After setting up the above, in the application.properties file, change the active profile from dev to prod.

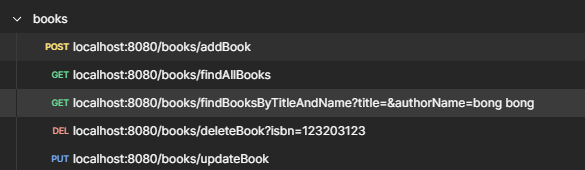


Afterwards you just need to run the application with mvn spring-boot:run. Spring boot will automatically create the tables and populate dummy data into it from the schema and data script found in the resource folder. You can verify if the tables are created by checking the BookstoreDB.



# Application endpoints

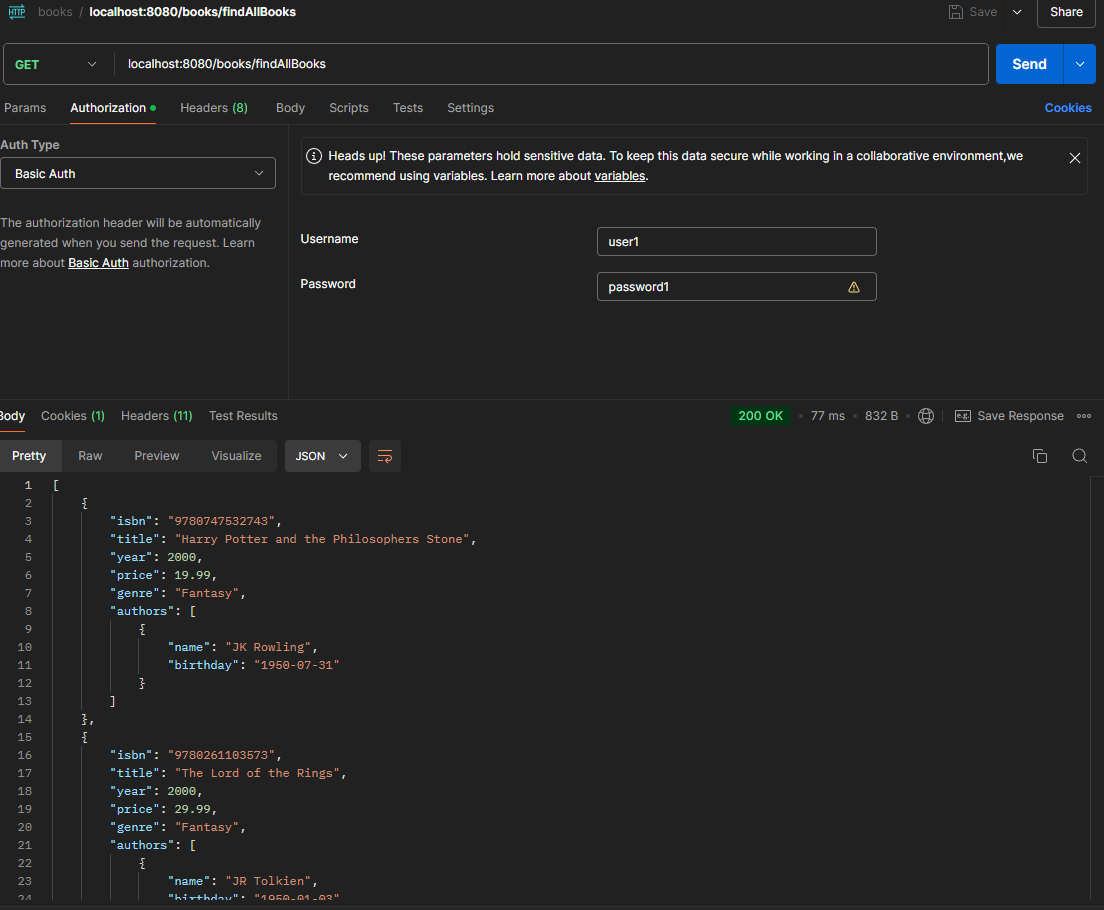
Now we shall test our applications and call all the different endpoints. In your postman app, import the books.postman\_collection file found in the Bookstore\_Application folder. You should be able to see the following endpoints.



Below we shall go through each of the endpoint. Do note that these endpoints are all protected and you need to pass in user1 and password1 as the username and password in the authorization header, else you will not be able to access the endpoints resource.

## Find All Book

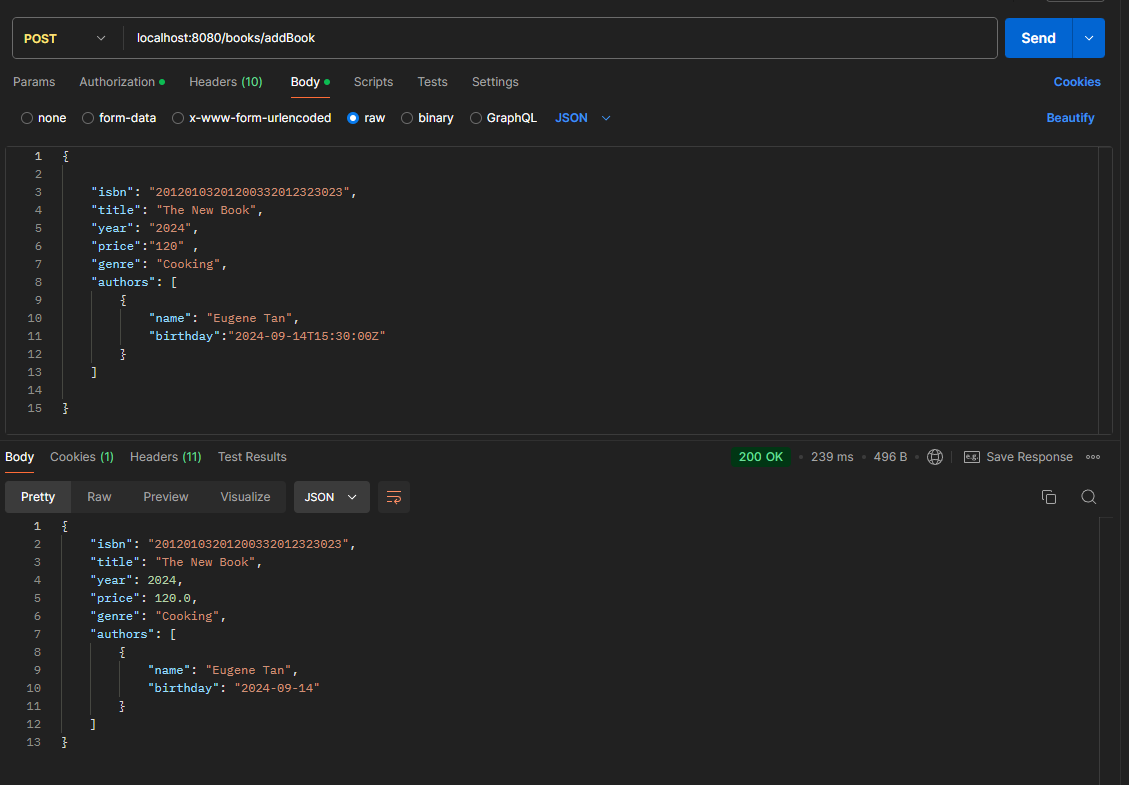
There is one endpoint called find all books. This api will fetch all the books and its corresponding author and returned as a json object.



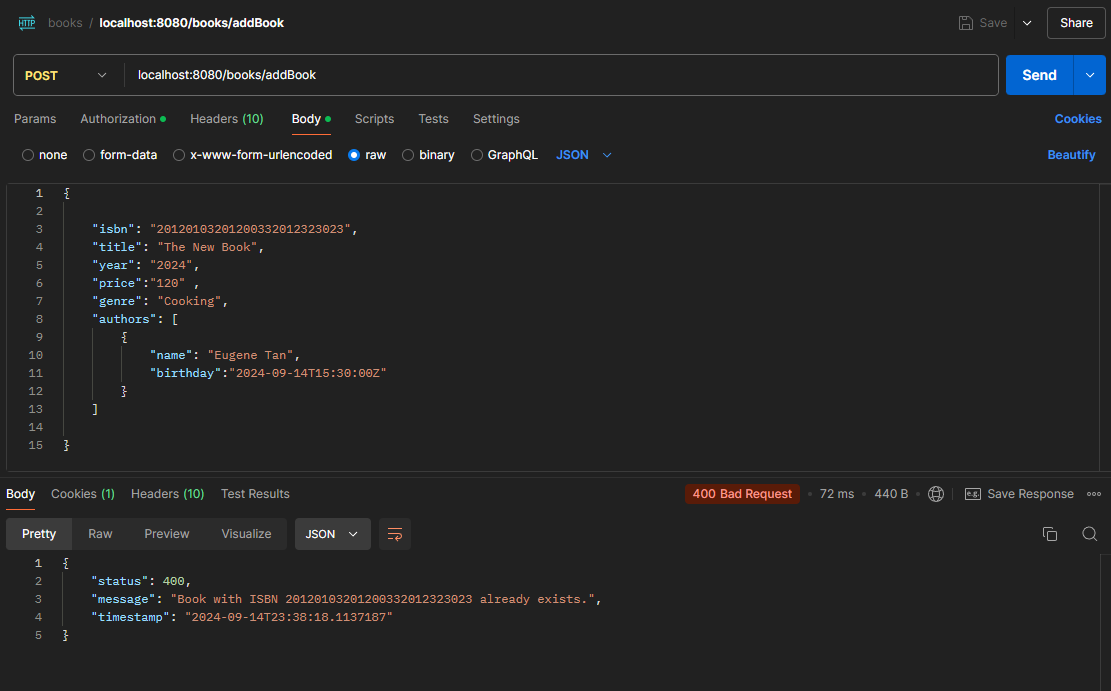
## Add Book

Add Book will take in a json object and check if that book isbn exist. If it exists it will throw an exception saying book already existed in database. Else it will add the book and return the book json object.

**If book does not exist**

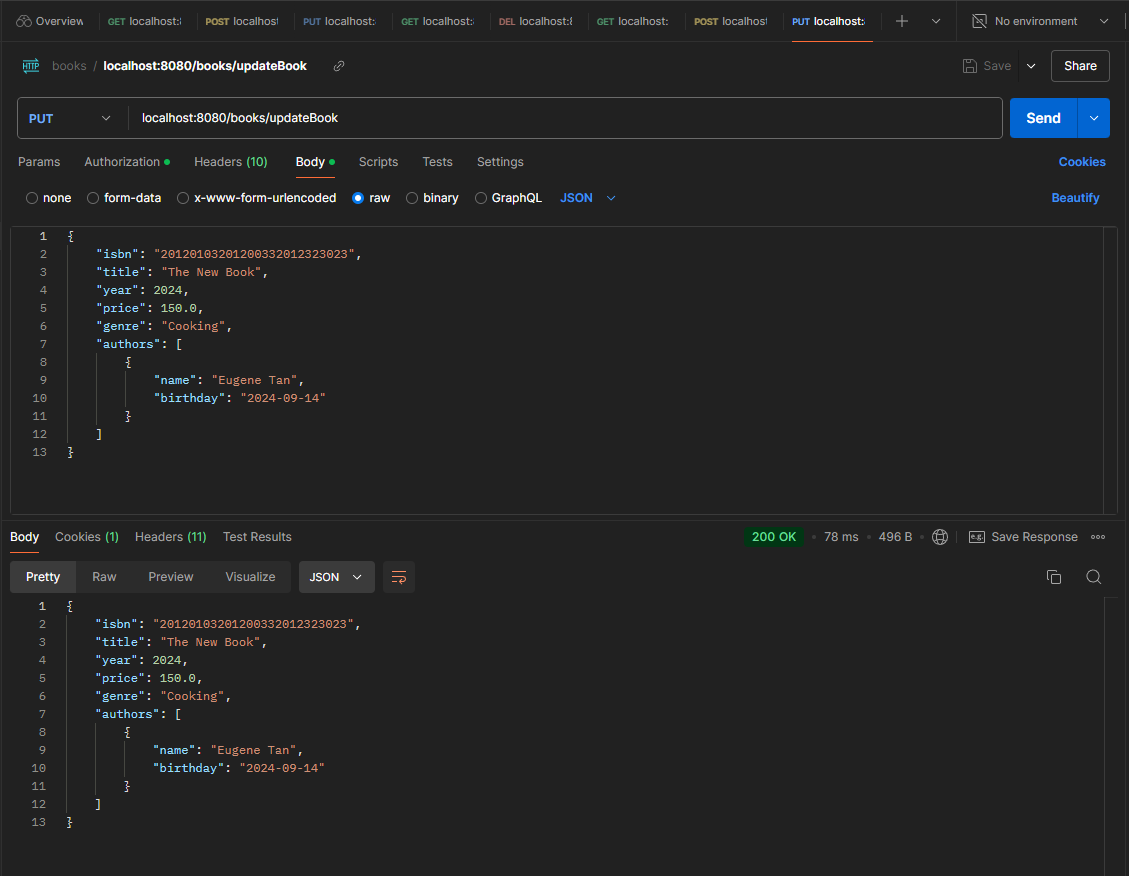


**If book already exists**



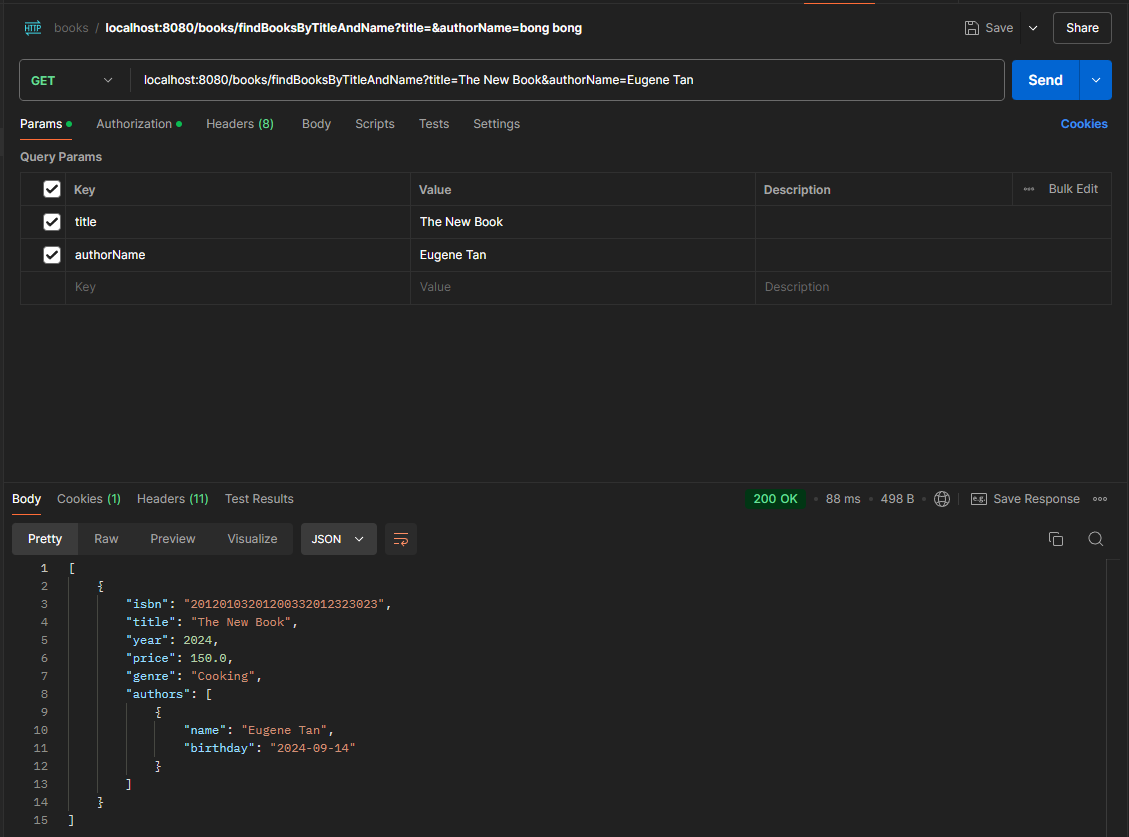
## Update book

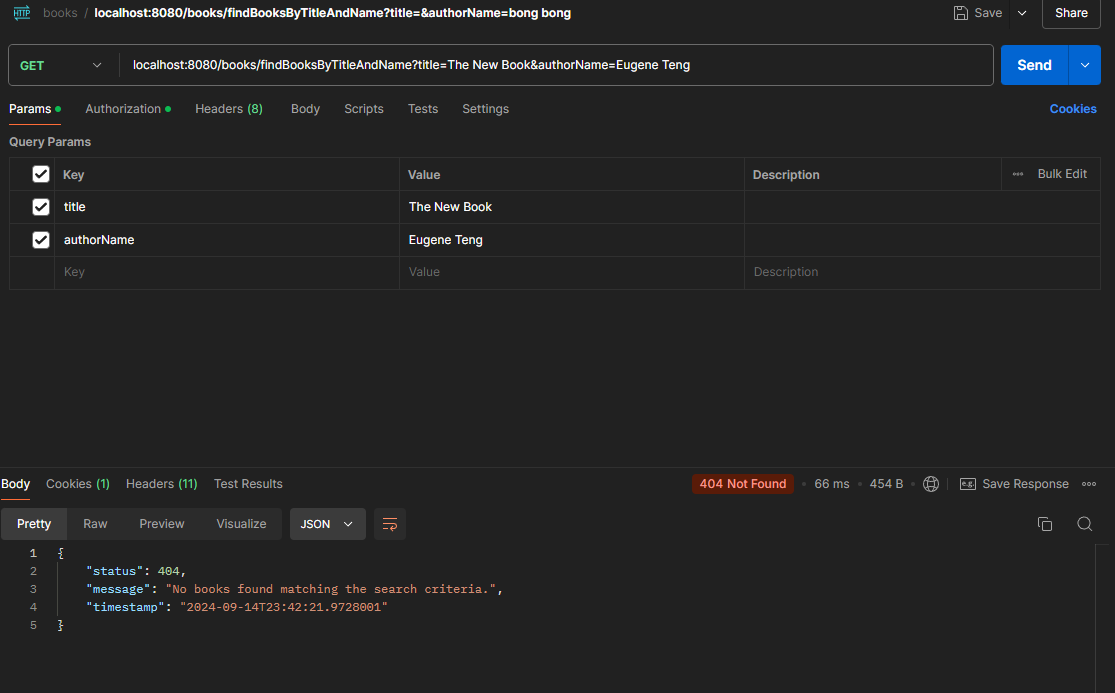
After adding a book, we can update the book by calling the updateBook api. Here we will update the price of the book from 120 to 150. It will return the book response with status code 200 if the book is successfully updated.



## Find book by title and name

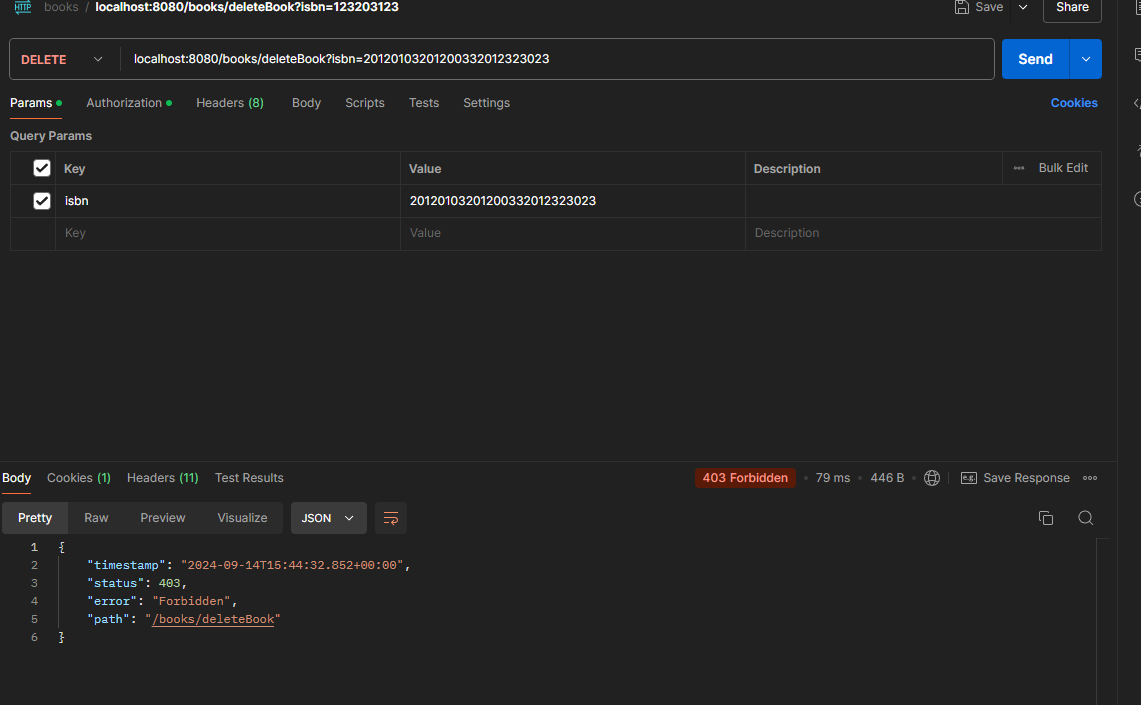
We can also find the book by title and name by passing in these 2 fields into the request parameters of the api call. If the book cannot be found, it will throw a 404 error to warn user the book is not found.





## Delete book

Here delete book can only be done by user holding the admin role. user1 does not have admin role. Trying to run the delete command with user1 will result in forbidden error.



Instead, you need to log in as admin with admin role to delete the book. The credentials for admin are admin with password- adminpass. After using these credentials, you will be able to delete the book.

