Testing Book Exchange API with Visual Studio Code's HTTP Client:

**1. Setting Up:**

* Launch **Visual Studio Code**.
* Navigate to **Extensions** using Ctrl + Shift + X.
* Search and install the **REST Client** extension developed by Huachao Mao.

**2. Initiating a New Request File:**

* Construct a new file within your workspace and save it with a .http extension. Example: BookExchangeTest.http.
* Use this file to script your HTTP requests.

**3. POST Request: Adding a New Book:**

**Requirement:** Add 5 books and supply before-and-after screenshots, including screenshots of Atlas MongoDB Collections.

**Important Note on Rest Client:**

##If you already have the token and you want to add it as an Authorization header in your requests using the REST Client extension in Visual Studio Code, you can do so by adding the Authorization header manually in your .http file. Here’s how you can do that:

**@token = "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY1NDNlOTExNWIwYjExOTA2ZDllZWU0YiIsImlhdCI6MTY5ODk0OTM5NSwiZXhwIjoxNjk5ODEzMzk1fQ.EhcVrKPFca7sm01sAraa0GTnQngfF7aQJWAhRoXp\_Co"**

**@baseUrl = http://localhost:4000**

**###**

**GET {{baseUrl}}/books**

**Authorization: Bearer {{token}}**

**# In this example, @token is a custom variable that holds your actual token. Replace your\_actual\_token\_here with the token you have.**

**###**

**POST {{baseUrl}}/books**

**Content-Type: application/json**

**Cookie: jwt={{token}}**

**{**

**"title": "The Lord of the Rings",**

**"author": "J. R. R. Tolkien"**

**}**

**# In this case, make sure to include the .env file in your .gitignore or equivalent to prevent committing sensitive information into version control.**

**# For an even more secure approach, you can use system environment variables or secret management tools to handle sensitive data like tokens, especially if you're working in a team or a shared environment. These tokens should be secured and only made available to the environments that require them, such as your local development environment or a secure CI/CD pipeline.**

**Test Code:**

### Sample Test: Add a new book

POST http://localhost:4000/books/add HTTP/1.1

Content-Type: application/json

{ "title": "Sample Book", "author": "Author Name", "description": "Book description here", "exchangeType": "borrow", "owner": "5fa12345678abcdef9012345", "status": "available" }

* To run the request, click on Send Request above the POST command.

**4. GET Request: Retrieve All Books:**

**Test Code:**

### Sample Test: Retrieve all books  
GET http://localhost:4000/books HTTP/1.1

* Click Send Request to obtain a list of all books.

**5. GET Request: Fetch Specific Book by ID:**

**Requirement:** Acquire 3 books by ID and submit before-and-after screenshots, and screenshots of Atlas MongoDB Collections.

Replace YOUR\_BOOK\_ID with the desired book ID:

**Test Code:**

### Sample Test: Fetch book by ID

GET http://localhost:4000/books/YOUR\_BOOK\_ID HTTP/1.1

**6. PUT Request: Modify a Specific Book by ID:**

**Note:** ***Prior to executing, confirm the presence of a route and controller for the HTTP PUT method. You will need to add code in the booRouter.js and bookController.js. For guidance look at the code the UI uses for Updating and Deleting the Books.***

**Requirement:** Modify 3 books and furnish before-and-after UI screenshots, as well as Atlas MongoDB Collection screenshots.

Substitute YOUR\_BOOK\_ID with the intended book ID:

**Test Code:**

### Sample Test: Modify a book by ID

PUT http://localhost:4000/books/YOUR\_BOOK\_ID HTTP/1.1

Content-Type: application/json

{ "title": "Updated Book Title", "author": "Updated Author Name" }

**7. DELETE Request: Erase a Specific Book by ID:**

**Note: *Prior to executing, confirm the presence of a route and controller for the HTTP PUT method. You will need to add code in the booRouter.js and bookController.js. For guidance look at the code the UI uses for Updating and Deleting the Books.***

Replace YOUR\_BOOK\_ID with the ID of the book slated for deletion:

**Requirement:** Erase 3 books and produce before-and-after screenshots of both the UI and Atlas MongoDB Collections.

**Test Code:**

### Sample Test: Delete book by ID

DELETE http://localhost:4000/books/YOUR\_BOOK\_ID HTTP/1.1

**General Instructions:**

* Whenever a request needs to be executed, select Send Request situated above the relevant HTTP command (GET, POST, PUT, DELETE) in your .http file.
* Adapt or extend the given requests based on the specific functions of your API.
* Ensure to submit the test cases in a TEXT file format.
* Also, include screenshots of the Atlas MongoDB Collections.

**General Instructions:**

* + Whenever a request needs to be executed, select Send Request situated above the relevant HTTP command (GET, POST, PUT, DELETE) in your .http file.
  + **Adapt or extend the given requests based on the specific functions of your API.**
  + Ensure to submit the test cases in a TEXT file format.
  + Also, include screenshots of the Atlas MongoDB Collections.