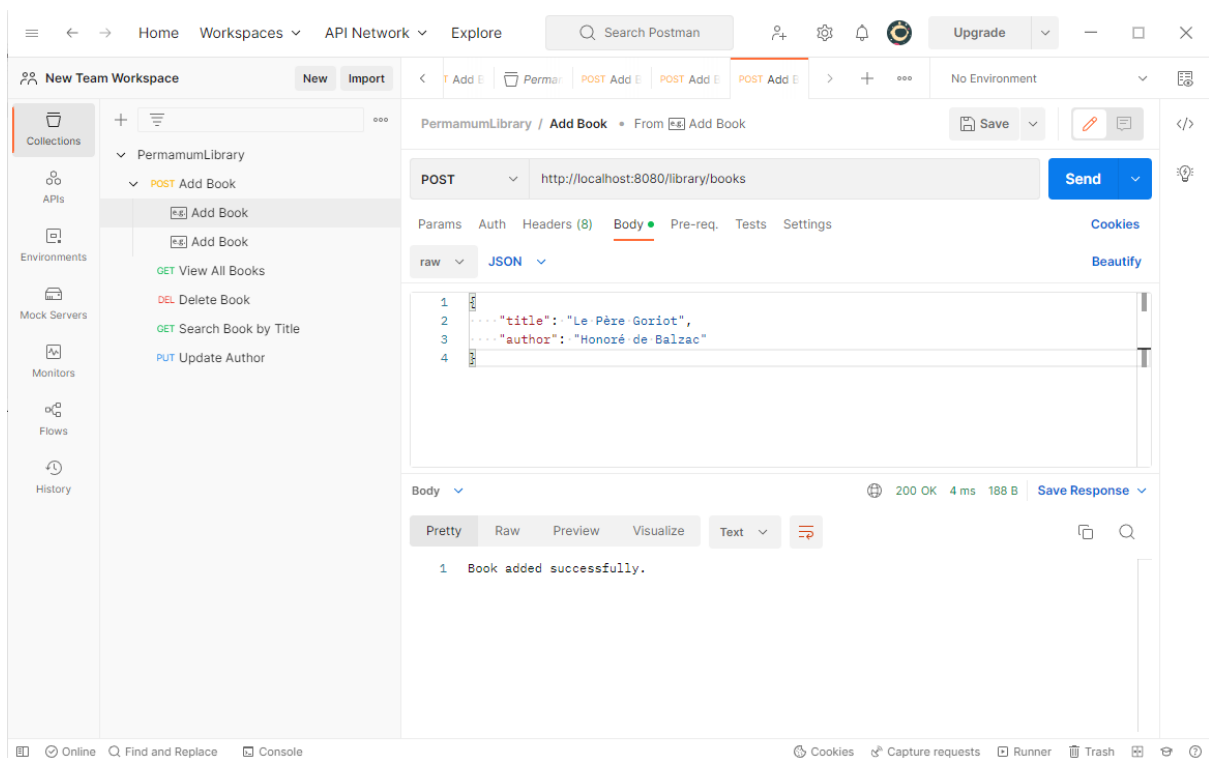
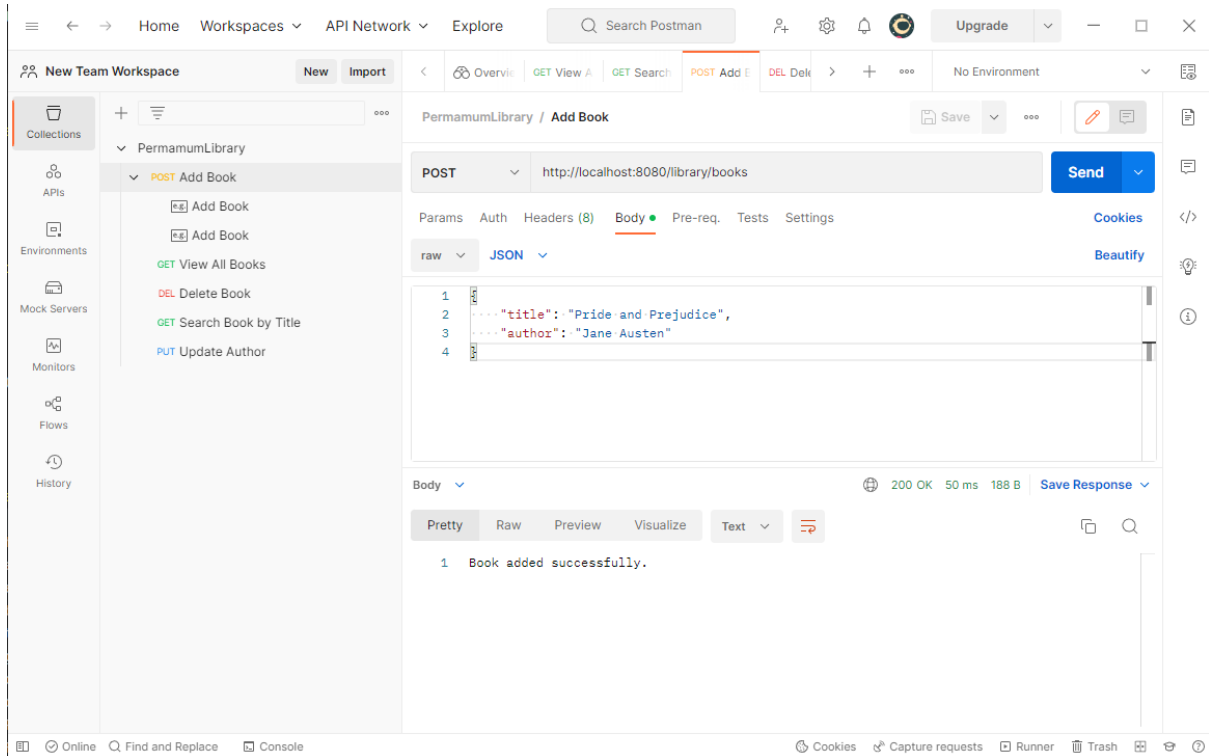
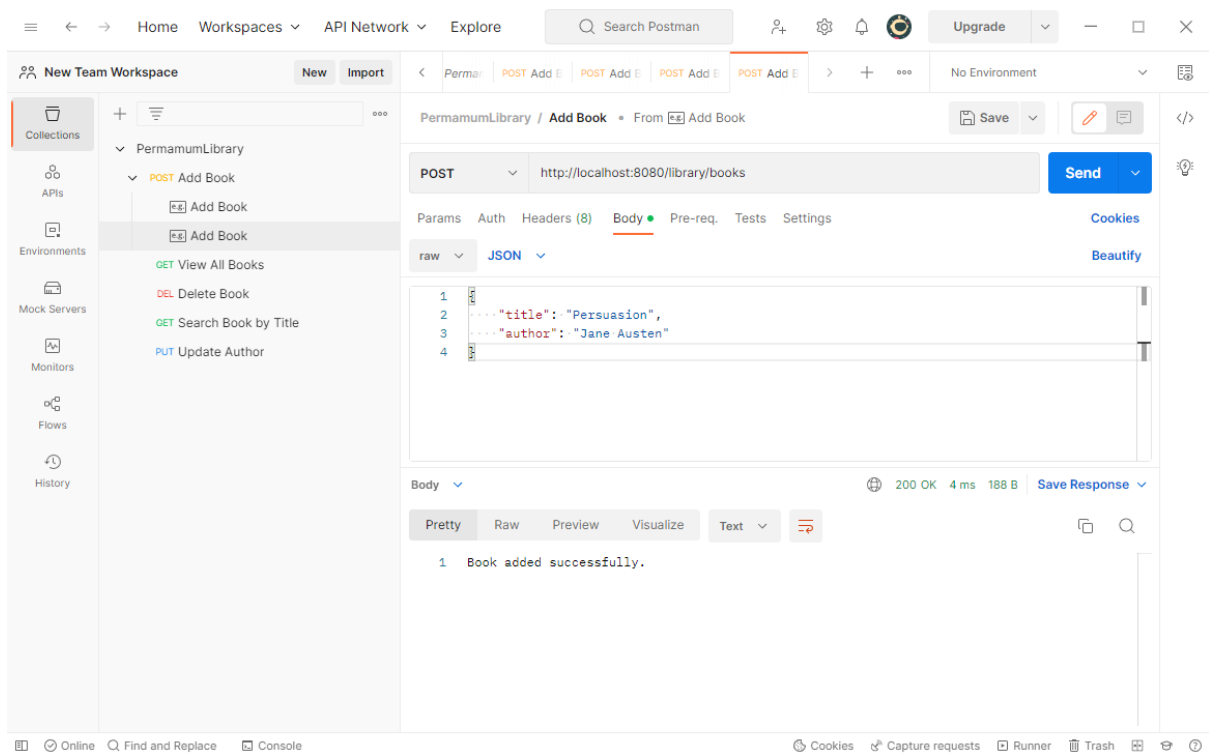


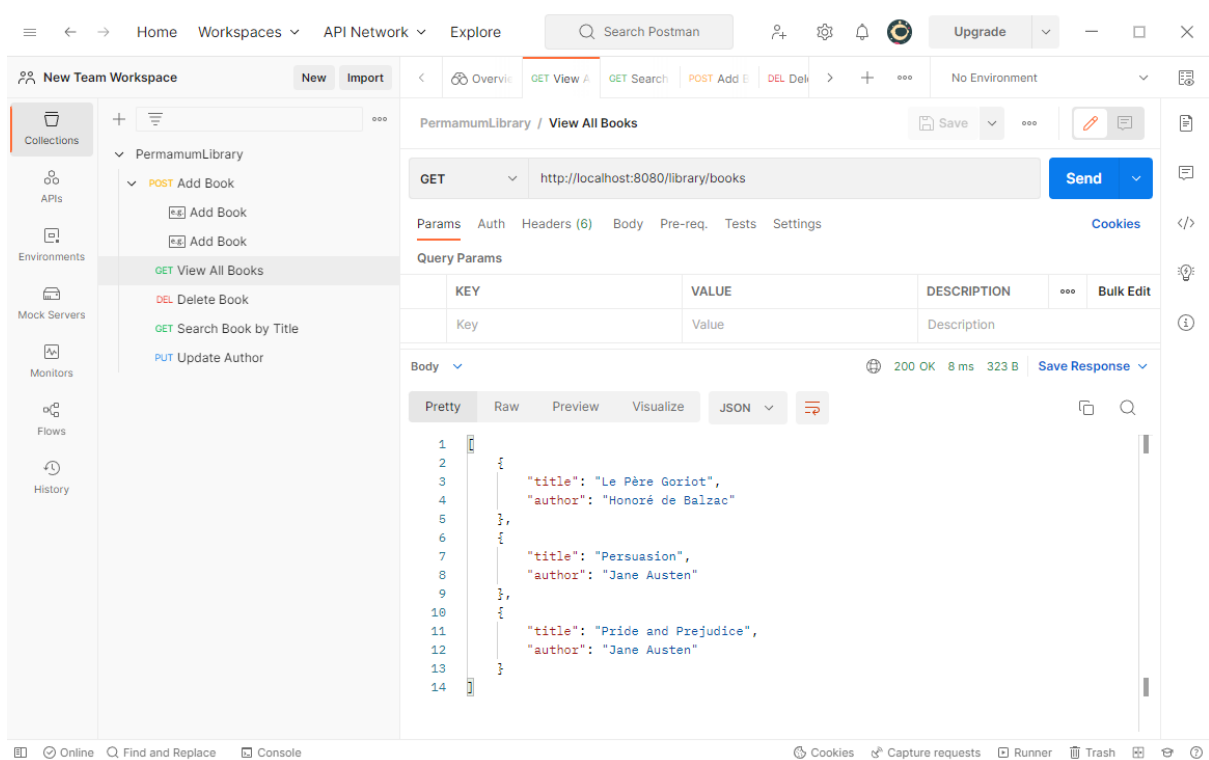
Testing with Postman

1. Adding three books:





2. Showing all the books from the library sorted by their title and author:



NOTICE: The list sorted so that if an author has multiple books, they will be displayed next to each other.

3. Deleting a book from the library:

The screenshot shows the Postman interface with a workspace named "New Team Workspace". The left sidebar lists collections, including "PermamumLibrary" which contains several endpoints. The "DELETE Delete Book" endpoint is selected. The main panel shows the request configuration for the "DELETE" method at the URL "http://localhost:8080/library/books". The "Body" tab is active, showing a JSON payload:

```
{  "title": "Le Père Goriot",  "author": "Honoré de Balzac"}
```

. The "Send" button is visible. Below the request configuration, the response is displayed in the "Body" tab, showing a 200 OK status and the message "Book deleted successfully".

NOTICE: To verify if the deletion worked, we can submit again the request for “View all Books”:

The screenshot shows the Postman interface with the same workspace. The "GET View All Books" endpoint is selected. The main panel shows the request configuration for the "GET" method at the URL "http://localhost:8080/library/books". The "Send" button is visible. Below the request configuration, the response is displayed in the "Body" tab, showing a 200 OK status and a JSON array of book objects:

```
[  {    "title": "Persuasion",    "author": "Jane Austen"  },  {    "title": "Pride and Prejudice",    "author": "Jane Austen"  }]
```

4. Searching a book by a title:

The screenshot shows the Postman interface with a workspace named "New Team Workspace". The left sidebar contains a "Collections" panel with a collection named "PermamumLibrary". Under this collection, there is a folder "POST Add Book" and a folder "GET Search Book by Title". The "GET Search Book by Title" folder is selected, and its endpoint is "http://localhost:8080/library/books/title?title=Pride and Prejudice". The request method is "GET". The response status is "200 OK" with a response time of "11 ms" and a body size of "218 B". The response body is displayed in "Pretty" format, showing a JSON object with the following structure:

```
1 {
2   "title": "Pride and Prejudice",
3   "author": "Jane Austen"
4 }
```

NOTICE: If we try to find the book we have just deleted, we won't find it, therefore our display box will be empty.

The screenshot shows the Postman interface with the same workspace and collection as the previous screenshot. The request method is still "GET", but the endpoint is now "http://localhost:8080/library/books/title?title=Le Père Goriot". The response status is "404 Not Found" with a response time of "4 ms" and a body size of "130 B". The response body is displayed in "Text" format, showing a single line:

```
1
```

5. Updating the author of a book:

The screenshot shows the Postman interface with a PUT request selected. The request is for the endpoint `http://localhost:8080/library/books/author?title=Pride and Prejudice&author=REPLA`. The 'Query Params' section contains two entries: 'title' with value 'Pride and Prejudice' and 'author' with value 'REPLACED AUTHOR'. The 'Body' section is currently empty, showing a status of 200 OK.

| KEY | VALUE | DESCRIPTION |
|--|---------------------|-------------|
| <input checked="" type="checkbox"/> title | Pride and Prejudice | |
| <input checked="" type="checkbox"/> author | REPLACED AUTHOR | |
| Key | Value | Description |

NOTICE: To verify if the update worked, we can submit again the request for “View all Books”:

The screenshot shows the Postman interface with a GET request selected. The request is for the endpoint `http://localhost:8080/library/books`. The 'Query Params' section is empty. The 'Body' section shows a JSON response with a status of 200 OK. The response contains an array of two book objects.

```
1 {
2   "title": "Persuasion",
3   "author": "Jane Austen"
4 },
5 {
6   "title": "Pride and Prejudice",
7   "author": "REPLACED AUTHOR"
8 }
9
10
```