

Postman Library API V2 - Documentation

Postman Library API V2

This project contains a complete Postman collection to interact with a mock Book Library API. The collection demonstrates fundamental REST API operations including GET, POST, PATCH, and DELETE methods ideal for learning, testing, and evaluating Postman skills.

API Endpoints Overview

1. `GET /books`

Retrieve all books in the library.

- **Status**: `200 OK`

- **Example Response**:

```json

[

{

"id": "e2f41dd2-83b6-4223-8c02-e3ea2e827473",

"title": "To Kill a Mockingbird",

"author": "Harper Lee",

"genre": "fiction",

"yearPublished": 1960,

"checkedOut": false,

"isPermanentCollection": false,

"createdAt": "2025-07-30T04:16:22.099Z"

}

]

```

2. `GET /books/:id`

Retrieve a single book by its ID.

- **Status**: `200 OK`
- **Required**: `id` in URL path
- **Header (optional)**: Basic headers like `Accept`, `Connection`

3. `GET /books?genre=fiction&checkedOut=false`

Filter books by query parameters `genre` and `checked out status`.

- **Status**: `200 OK`
- **Query Params**:
 - `genre=fiction`
 - `checkedOut=false`

4. `POST /books`

Add a new book to the library.

- **Status**: `201 Created`
- **Headers Required**:
 - `Content-Type: application/json`
- **Sample Body**:

````json`

`{`

`"id": "7ddb8261-1f3b-4721-85d8-68a904f6c578",`

`"title": "Rich Dad Poor Dad",`

```
"author": "Robert",
"genre": "finance",
"yearPublished": 1997,
"checkedOut": false,
"isPermanentCollection": false
}
...

```

## 5. `PATCH /books/:id`

Checkout a book.

```
- **Status**: `200 OK`
- **Headers Required**:
 - `api-key: postmanrulz`
 - `Content-Type: application/json`
- **Body Example**:
 ``json
 {
 "checkedOut": true
 }
 ...

```

## 6. `POST /skillcheck?movieName=Inception`

Mock test endpoint used for skill assessment.

```
- **Status**: `200 OK`
- **Query Param**: `movieName=Inception`
- **Response Contains**:
 ``json
```

```
"data": {
 "actorName": "Leonardo DiCaprio"
}
```
```

7. `DELETE /books/:id`

Delete a book from the library by ID.

- **Status**: `204 No Content`
- **Headers Required**:
 - `api-key: postmanrulz`
 - `Content-Type: application/json` (if needed)
- **Response**: No content (successful delete)

Test Results

In the **Final Check** test:

- All 20/20 test cases passed.
- Ensured methods, paths, query parameters, and header values are correct.
- Validated baseUrl and request method dynamically using `pm.test()` scripting.

Technologies Used

- Postman (Collection + Tests)
- JavaScript for Postman test scripts
- Mock API baseUrl using `{{baseUrl}}`

- Authorization header with `api-key`

Repository Contents

...

postman-library-api

README.md

Postman Collection (exported JSON)

Screenshots/

GET-Books.png

GET-BookByID.png

GET-FictionBooks.png

POST-AddBooks.png

PATCH-CheckoutBooks.png

POST-SkillCheck.png

FinalCheck-Script.png

FinalCheck-Results.png

...

Author

****Abdulla Al Bayzed****

Hands-on API fundamentals learner with Postman

[Postman Workspace Link](<https://www.postman.com/abdullaalbayzed>)

Notes

- Ensure to set the `api-key` in headers for protected routes like `PATCH`.
- Use `{{baseUrl}}` and `{{id}}` as environment variables for dynamic testing.

Screenshots

See the `/Screenshots` folder for all the step-by-step screenshots.