

## ĆWICZENIA WYKONANE PRZEZE MNIE W POSTMANIE W RAMACH PROGRAMU POSTMAN API FUNDAMENTALS STUDENT EXPERT

### Zastosowanie różnych metod HTTP

1. Zastosowanie metody **GET** do pobrania zasobu (w tym przypadku listy książek) – status 200 OK

The screenshot displays the Postman interface for a REST client. The left sidebar shows the 'Collections' panel with a tree view containing 'Collection Test' and 'Postman Library API v2'. The 'Postman Library API v2' collection is expanded, showing several endpoints. The 'GET get books' endpoint is selected. The main panel shows the request details for this endpoint. The URL is 'http://{{baseUrl}}/books'. The 'Params' tab is active, showing a table with two columns: 'Key' and 'Value'. The 'Body' tab is also active, showing the response in 'Pretty' format. The response is a JSON array of two book objects. The status is '200 OK', the time is '787 ms', and the size is '156.2 KB'.

**Request Details:**

- Method: GET
- URL: `http://{{baseUrl}}/books`
- Params: (Empty table)

**Response Details:**

- Status: 200 OK
- Time: 787 ms
- Size: 156.2 KB
- Save as example: (button)

**Response Body (Pretty):**

```
[
  {
    "id": "9b2e6615-511b-4eca-88a1-fe0d16cfdb94",
    "title": "Game of Thrones",
    "author": "James",
    "genre": "fiction",
    "yearPublished": 1997,
    "checkedOut": false,
    "isPermanentCollection": false,
    "createdAt": "2024-02-26T11:02:02.349Z"
  },
  {
    "id": "c7cd0484-aa87-45cb-8e96-473cf81d3f60",
    "title": "Atque ipsam soluta illo laborum est et."
  }
]
```

## 2. Zastosowanie metody **POST** dla stworzenia nowego zasobu (w tym przypadku dodania nowej pozycji – książki) – st. 201 Created

The screenshot displays the Postman API client interface. On the left sidebar, the 'Collections' panel is open, showing a collection named 'Postman Library API v2' with several endpoints. The 'POST add a book' endpoint is selected. The main workspace shows the details of this endpoint. The URL is set to `{{baseUrl}}/books` and the method is **POST**. The 'Body' tab is active, showing a JSON payload with the following structure:

```
1 {
2   "title": "To Kill a Moooooooo",
3   "author": "Harper Lee",
4 }
```

Below the request details, the 'Response' tab is active, showing the status **201 Created**, a time of **329 ms**, and a size of **636 B**. The response body is displayed in a 'Pretty' JSON format:

```
1 {
2   "id": "73fed3c3-3315-4296-bff5-da97b1c18781",
3   "title": "To Kill a Moooooooo",
4   "author": "Harper Lee",
5   "genre": "fiction",
6   "yearPublished": 1960,
7   "checkedOut": false,
8   "isPermanentCollection": false,
9   "createdAt": "2024-02-26T11:09:16.553Z"
10 }
```

The bottom status bar indicates the interface is in 'Online' mode and shows various utility icons like Postbot, Runner, Auto-select agent, Cookies, and Trash.

### 3. Zastosowanie metody **DELETE** by usunąć zasób ( w moim przypadku książkę ) - status 204 no content

The screenshot shows the Postman interface with a workspace named 'Postman API Fundamentals Stud...'. The left sidebar contains a 'Collections' panel with a collection 'Collection Test' by 'Marta Sieniawska-La...'. Under this collection, there are several API endpoints, including 'DELETE delete book' which is currently selected.

The main panel displays the details of the 'DELETE delete book' request. The URL is 'Postman Library API v2 / delete book'. The method is 'DELETE' and the path variable is 'id' with a value of '{{id}}'. The request is saved and ready to be sent.

Below the request details, the 'Body' tab is selected, showing a '204 No Content' status. A tooltip message states: 'The server successfully processed the request, but is not returning any content.' The status bar at the bottom indicates 'Status: 204 No Content', 'Time: 263 ms', and 'Size: 360 B'.

Key	Value	Description
id	{{id}}	Description