**REST API**

**myLibraryAPI**: REST API to manage books and borrows of a library

version: 1.0

host: myLibraryAPI/v1.com

media types: only supports JSON

security mechanisms: JWT (JSON Web Tokens) - open endpoints requires no authentication; protected endpoints require a valid Token to be included in the header of the request

API PATHS

[USERS 2](#_Toc161232110)

[POST /users/login 2](#_Toc161232111)

[BOOKS 3](#_Toc161232112)

[GET /books 3](#_Toc161232113)

[DELETE /books/:bookID 5](#_Toc161232114)

## 

# USERS

## POST /users/login

Authenticate the user and obtain a JSON Web Token (authentication key).

### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Parameters** | **Format** | **Required** |
| BODY  BODY | username  password | string  string | yes  yes |

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 OK | {"accessToken": <token>}  token (**string**) - all further API calls to protected routes must have  this token in authentication header |
| 400  Bad request | {"message":"Please provide username and password"} |
| 401  Unauthorized | {"message":"Invalid credentials"} |
| 500 Internal server error | {"message":"Something went wrong. Please try again later"} |

# BOOKS

## GET /books

Obtain the general information about all books in the library. The list can be filtered by title and/or author(s) name. Results can be paginated by defining the number of books per page and the current page in the request.

### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Parameters** | **Format** | **Required** |
| QUERY  QUERY QUERY  QUERY | title  author limit  page | string  string  number  number | no  no  no  no |

**title, author**

Strings to search from the complete list of books, books that contain the title substring and/or the author’s name substring.

**limit**

Integer (higher than 5) that indicates the number of books retrieved per page.

**page**

Integer (equals or higher than 0) that indicates the page to retrieve.

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 OK | Response will be an object containing the list of books (array)  as well as some metadata, like pagination, and HATEOS navigation links.  An example response is:  {  "pagination": {  "total": 235,"pages": 24,"current": 0,"limit": 10  },  "data": [  {  "book\_id": 10,  "title": " Desenvolvimento Avançado para a Web",  "authors": ["Ricardo Queirós","Filipe Portela"],  "links":[ {"rel":"self","href":"/books/10","method":"GET"}]  }  ],  "links": [  {"rel":"add-book","href":"/books","method":"POST"},  {"rel":"next-page","href":"/books?limit=10&page=1","method":"GET"}  ]  } |
| 400  Bad request | {"message":"Limit must be a positive integer, greater than 5"} |
| 400  Bad request | {"message":"Page must be 0 or a positive integer"} |
| 500 Internal server error | {"message":"Something went wrong. Please try again later"} |

## DELETE /books/:bookID

Delete a book. A book can only be deleted if there are no borrows for it. Route only available for librarians (authentication token must be provided in header).

### Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Parameters** | **Format** | **Required** |
| PATH  HEADER | bookID  token | integer  string | yes  yes |

**bookID**

Unique identifier of the book to delete.

**token**

The authentication token that was given in response to /users/login.

### Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 204  No Content | The book was found and successfully deleted. No data is retrieved. |
| 401  Unauthorized | {"message":"No access token provided"} {"message":"Your token has expired! Please login again."} |
| 403  Forbidden | {"message":"This request requires administrator privileges"} |
| 404  Not Found | {"message":"Book with ID <bookID> was not found"} |
| 409  Conflict | {"message":"Book with ID <bookID> cannot be deleted because it is currently borrowed"} |
| 500 Internal server error | {"message":"Something went wrong. Please try again later"} |