The 'Book' object and all of its properties, distributed across tables.

Book = {

id: number (unique book id in the library),

ISBN: number (unique ISBN id),

name: string (book name),

genre: string (book genre),

author: string (author name),

shelfId: number (shelf id in the library),

bookingStatus: boolean (true/false),

bookingDate: date (reservation date),

bookingPerson: string (the person who made the reservation),

outNowStatus: boolean (true/false),

outNowDate: date (the dates on which the book was pick up),

outNowPerson: string (the person who has the book now)

}

|  |  |
| --- | --- |
| 1. Searching for a book | |
| GET /v1/search | |
| Can Access | Librarian and Reader |
| Params description: | id: number (unique book id in the library),  ISBN: number (unique ISBN id),  name: string (book name),  genre: string (book genre),  author: string (author name),  shelfId: number (shelf id in the library). |
| Request | |
| Query params description | ISBN, name, genre, author — search options. None of the parameters are required. However, at least one parameter is required to start the search. |
| Request body | {  ISBN: number,  name: string,  genre: string,  author: string  } |
| Responses | |
| 200 ok | Search successful. Returns all books matching the search parameters. If there are more than 5 such books, then the search returns the 5 most popular ones. Also, the response body includes id and shelfId. |
| 400 bad request | None of the parameters specified or the parameter type is invalid. |
| 404 not found | Nothing with the provided parameters. |
| 405 method not allowed | Wrong HTTP method. |
| 500 internal Server Error | Something went wrong on the server. |
| Response body on success | {  id: number,  ISBN: number,  name: string,  genre: string,  author: string,  shelfId: number  } |

|  |  |
| --- | --- |
| 1. Other manipulations of the book | |
| GET /v1/book | |
| Can Access | Librarian |
| Params description: | id: number (unique book id in the library),  ISBN: number (unique ISBN id),  name: string (book name),  genre: string (book genre),  author: string (author name),  shelfId: number (shelf id in the library). |
| Request | |
| Query params description | id — required and only search parameter. |
| Request body | {  id: number  } |
| Responses | |
| 200 ok | Search successful. Returns book matching the id parameter. |
| 400 bad request | None of the required parameters specified or the parameter type is invalid. |
| 403 forbidden | You don't have permission to access this resource. |
| 404 not found | Nothing with the provided parameter. |
| 405 method not allowed | Wrong HTTP method. |
| 500 internal Server Error | Something went wrong on the server. |
| Response body on success | {  id: number,  ISBN: number,  name: string,  genre: string,  author: string,  shelfId: number  } |
| PUT /v1/book/:id | |
| Can Access | Librarian |
| Params description: | id: number (unique book id in the library),  shelfId: number (shelf id in the library) |
| Request | |
| Query params description | shelfId (required) – new shelfId of shelf to which we want to move the book. |
| Request body | {  shelfId: number  } |
| Responses | |
| 200 ok | The book moved to other shelves. Successfully updated shelfId with the provided new shelfId. Returns the id and new shelfId. |
| 400 bad request | None of the required parameters specified or the parameter type is invalid |
| 403 forbidden | You don't have permission to access this resource. |
| 404 not found | Nothing with the provided parameter. |
| 405 method not allowed | Wrong HTTP method. |
| 500 internal Server Error | Something went wrong on the server. |
| Response body on success | {  shelfId: number  } |
| POST /v1/book | |
| Can Access | Librarian |
| Params description: | id: number (unique book id in the library)  ISBN: number (unique ISBN id)  name: string (book name)  genre: string (book genre)  author: string (author name)  shelfId: number (shelf id in the library) |
| Request | |
| Query params description | ISBN, name, genre, author, shelfId — required parameters of a new book. |
| Request body | {  ISBN: number,  name: string,  genre: string,  author: string,  shelfId: number  } |
| Responses | |
| 201 created | A new book has been added to the shelf. Return new autoincremental id for this book. |
| 400 bad request | None of the required parameters specified or the parameter type is invalid. |
| 404 not found | Nothing with the provided parameters. |
| 403 forbidden | You don't have permission to access this resource. |
| 405 method not allowed | Wrong HTTP method. |
| 500 internal Server Error | Something went wrong on the server. |
| Response body on success | {  id: number  } |

|  |  |
| --- | --- |
| 1. Taking a book | |
| PATCH /v1/user/:userId/booking/:bookId | |
| Can Access | Reader |
| Params description: | userId: string (reader id)  name: string (book name)  outNowDate: date (date when the book must be return) |
| Request | |
| Query params description | outNowDate — required parameter. |
| Request body | {  outNowDate: date  } |
| Responses | |
| 200 ok | The book is issued to the reader. |
| 400 bad request | None of the parameters specified or the parameter type is invalid. |
| 403 forbidden | You don't have permission to access this resource. |
| 404 not found | Nothing with the provided parameters. |
| 405 method not allowed | Wrong HTTP method. |
| 500 internal Server Error | Something went wrong on the server. |
| Response body on success | {  outNowDate: date  } |
| GET /v1/booking | |
| Can Access | Librarian |
| Params description: | name: string (book name),  outNowStatus: boolean (true/false),  outNowPerson: string (the person who has the book now),  outNowDate: date (date when the book must be return) |
| Request | |
| Query params description | name, outNowStatus — none of the parameters are required. However, at least one parameter is required to start the search. |
| Request body | {  name: string,  outNowStatus: boolean  } |
| Responses | |
| 200 ok | Return all items matching the search parameters. |
| 400 bad request | None of the parameters specified or the parameter type is invalid |
| 403 forbidden | You don't have permission to access this resource or book not available now. |
| 404 not found | Nothing with the provided parameters. |
| 405 method not allowed | Wrong HTTP method. |
| 500 internal Server Error | Something went wrong on the server. |
| Response body on success | {  name: string,  outNowStatus: boolean,  outNowPerson: string,  outNowDate: date  } |

|  |  |
| --- | --- |
| 1. Reserving a book | |
| PATCH /v1/reservation/:userId | |
| Can Access | Reader |
| Params description: | userId: string (reader id),  name: string (book name),  bookingDate: date (date when the book must be return) |
| Request | |
| Query params description | name — required parameter. |
| Request body | {  name: string  } |
| Responses | |
| 200 ok | The book is reserved to the reader. Return finish date. |
| 400 bad request | None of the parameters specified or the parameter type is invalid |
| 403 forbidden | You don't have permission to access this resource or book is booked or not available. |
| 404 not found | Nothing with the provided parameters. |
| 405 method not allowed | Wrong HTTP method. |
| 500 internal Server Error | Something went wrong on the server. |
| Response body on success | {  bookingDate: date  } |
| DELETE /v1/reservation/:userId | |
| Can Access | Reader |
| Params description: | userId: string (reader id),  name: string (book name), |
| Request | |
| Query params description | name — required parameter. |
| Request body | {  name: string  } |
| Responses | |
| 200 ok | The book is unreserved. |
| 400 bad request | None of the parameters specified or the parameter type is invalid |
| 403 forbidden | You don't have permission to access this resource. |
| 404 not found | Nothing with the provided parameters. |
| 405 method not allowed | Wrong HTTP method. |
| 500 internal Server Error | Something went wrong on the server. |
| Response body on success |  |
| GET /v1/reservation/:userId | |
| Can Access | Reader |
| Params description: | login: string (reader login),  name: string (book name), |
| Request | |
| Query params description |  |
| Request body |  |
| Responses | |
| 200 ok | Return all books that are reserved for the reader. |
| 403 forbidden | You don't have permission to access this resource. |
| 404 not found | Nothing with the provided parameters. |
| 405 method not allowed | Wrong HTTP method. |
| 500 internal Server Error | Something went wrong on the server. |
| Response body on success | {  name: string  } |

|  |  |
| --- | --- |
| 1. Bookshelf information | |
| GET /v1/:shelfId | |
| Can Access | Reader and Librarian |
| Params description: | name: string (book name),  shelfId: number (shelf id in the library). |
| Request | |
| Query params description | shelfId — required parameter. |
| Request body | {  shelfId: number  } |
| Responses | |
| 200 ok | Return all the books on this shelf. |
| 400 bad request | None of the parameters specified or the parameter type is invalid |
| 404 not found | Nothing with the provided parameters. |
| 405 method not allowed | Wrong HTTP method. |
| 500 internal Server Error | Something went wrong on the server. |
| Response body on success | {  name: string  } |