**Modeling API for Library**

1. Participants:

* librarian;
* readers;

2. Activities:

* a reader/librarian can search a shelf where a book is located by query params:
* id;
* ISBN;
* name;
* genre;
* author;
* a librarian can add new book to a shelf;
* a librarian can change a shelf param of a book;
* a librarian can get full info on a book by its id;
* a reader can check out a book temporarily;
* a librarian can get the information about a reader that checked out a book;
* a librarian can get information about checked out books and their readers;
* a librarian can get information about all books checked out by a reader;
* a reader can get the information about the due date for a book which has been checked out by another reader;
* a reader can reject checking out a book;
* system should reject checking out a book by a reader if it it’s not available for a specified period;
* a reader can see a list of books he checked out;
* a reader/librarian can see a list of books located on a shelf by its id;

3. Resources:

* list books (public access);
* list checked out books (private access);
* add/remove a book to/from a private list;
* deny adding a non-available book to a private list;
* add/remove a book to/from a shelf;
* view full book info (private access);
* view brief book info (public access);

|  |  |
| --- | --- |
| **GET**/books / | |
| Can Access: | Librarian or Reader |
| Params description: | - |
| Query params description: | Available query params:   * book\_id (optional): string – book ID; * isbn (optional): string – book ISBN code; * name (required): string – book name; * genre (optional): string – book genre; * author (optional): string – book author; |
| Responses: | |
| 200 Ok | An array of objects ( e.g. IBook[] ) with an interface of an object like: IBook { book\_id: number, name: string, author: string, isbn: string, genre: string, shelf\_id: number }. This array contains list of books corresponding to query request params. Array may have zero length. |
| 400 Bad Request | Required parameter name was not provided / incorrect type of provided parameter (e.g. boolean instead of a string); |
| 500 Internal Server Error | Something went wrong on the server. |
| Response body on success: | |
| [  {  book\_id: number,  name: string,  author: string,  isbn: string,  genre: string,  shelf\_id: number,  status: string  }  ] | |

|  |  |
| --- | --- |
| **POST**/books /shelves / :shelf\_id /items | |
| Can Access: | Librarian |
| Params description: | shelf\_id – ID of a shelf in a library. |
| Request body: (required): user, book\_id, name, author;  (optional): isbn, genre, status, hasPatron, patron. | |
| {     user: string;  book\_id: number,  name: string,  author: string,  isbn: string,  genre: string,  status: string,  isDue: boolean,  dueDate: date string,  hasPatron: boolean,  patron: string  } | |
| Responses: | |
| 201 Created | Added a new book to a specified shelf. |
| 404 Not Found | Shelf with a specified ID doesn’t exist. |
| 400 Bad Request | Required parameters book\_id, name or/and author were provided |
| 401 | Required parameter user was not provided (authentication failed). |
| 403 | User is not a librarian, but a reader e.g. (not authorized access). |
| 500 Internal Server Error | Something went wrong on the server. |
| Response body on success: | |
| {      message: “book is added”  } | |

|  |  |
| --- | --- |
| **PATCH**/books/ :book\_id | |
| Can Access: | Librarian |
| Params description: | book\_id – ID of a book in a library. |
| Request body: | |
| (required): user, new\_shelf\_id;  {  user: string,  new\_shelf\_id: number  } | |
| Responses: | |
| 200 Ok | Updated a book with a new shelf ID. |
| 400 Bad Request | Required parameter new\_shelf\_id was not provided / incorrect type of provided parameter (e.g. Boolean instead of a number); |
| 404 Not Found | Such a book wasn’t found. |
| 401 | Required parameter user was not provided (authentication failed). |
| 403 | User is not a librarian, but a reader e.g. (not authorized access). |
| 500 Internal Server Error | Something went wrong on the server. |
| Response body on success: | |
| {      message: “book is transferred to a new shelf”  } | |

|  |  |
| --- | --- |
| **GET**/books / :book\_id | |
| Can Access: | Librarian or Reader |
| Params description: | book\_id – ID of a book in a library. |
| Responses: | |
| 200 Ok | Information about a book: full (for a librarian), brief (for a reader). |
| 404 Not Found | Such a book wasn’t found. |
| 500 Internal Server Error | Something went wrong on the server . |
| Response body on success (librarian): | |
| {  book\_id: number,  name: string,  author: string,  isbn: string,  genre: string,  status: string,  isDue: boolean,  dueDate: date string,  hasPatron: boolean,  patron: string  } | |
| Response body on success (reader): | |
| {  book\_id: number,  name: string,  author: string,  isbn: string,  genre: string,  status: string,  isDue: boolean,  dueDate: date string  } | |

|  |  |
| --- | --- |
| **PUT**/books /:book\_id | |
| Can Access: | Reader |
| Params description: | book\_id – ID of a book in a library. |
| Request body: (required): user; (optional) dueDate – date upon which a reader has to return a book, default: + 90 days from a current date of request. | |
| {  user: string,      dueDate: date string  } | |
| Responses: | |
| 200 Ok | A reader successfully checked out (took) a book. |
| 401 | Required parameter user was not provided (authentication failed). |
| 403 | User is not a reader (not authorized access). |
| 422 | A book for a specified period is not available (booked by another reader) |
| 500 Internal Server Error | Something went wrong on the server. |
| Response body on success: | |
| {  book\_id: number,  name: string,  author: string,  isbn: string,  genre: string,  status: “checked out”,  dueDate: date string  } | |

|  |  |
| --- | --- |
| **GET**/books /checkedout | |
| Can Access: | Librarian |
| Request body:  (required): user | |
| {  user: string,  } | |
| Responses: | |
| 200 Ok | A librarian got the list of booked books and their readers (patrons).  A list (array) may have zero length. |
| 401 | Required parameter user was not provided (authentication failed). |
| 403 | User is not a librarian (not authorized access). |
| 500 Internal Server Error | Something went wrong on the server. |
| Response body on success: | |
| [  {  book\_id: number,  patron: string,  dueDate: date string  }  ] | |

|  |  |
| --- | --- |
| **GET**/books /readers / :reader\_id | |
| Can Access: | Librarian |
| Params description: | reader\_id – ID of a reader. |
| Request body:  (required): user | |
| {  user: string,  } | |
| Responses: | |
| 200 Ok | A librarian got the list of booked books by a specified reader (patron).  A list (array) may have zero length. |
| 401 | Required parameter user was not provided (authentication failed). |
| 403 | User is not a librarian (not authorized access). |
| 500 Internal Server Error | Something went wrong on the server. |
| Response body on success: | |
| [  {  book\_id: number,  dueDate: date string  }  ] | |