Barleta, Denesis

Borja, Nikko

Syfu, Jonah

**MyBookList API Documentation**

**Introduction:**

The MyBookList API allows the developers access to the data stored in the MBL database which can be used for websites that deal with books. The API can be used to rate a collection of titles, and makes it easy to display on the website. The API can also be used to have users connect with their common books or interests.

**API methods:**

**Get author info by id**

Get an xml response with info about an author.

URL:

HTTP method: GET

Parameters:

\* id: database author id

**Get book info by id**

Get an xml response with info about a book.

URL:

HTTP method: GET

Parameters:

\* id: database book id

**Add book to shelf**

Adds a book to a user's shelf.

URL:

HTTP method: POST

Parameters:

\* book[book\_id]: id of the book (required)

\* book[original\_purchase\_date]: when book was bought

\* book[reading\_state]: reading, on-hold, etc

**Edit books on shelf**

Edits a book on user's shelf.

URL:

HTTP method: PUT

Parameters:

\* book[book\_id]: id of the book (required)

\* book[original\_purchase\_date]: when book was bought

\* book[reading\_state]: reading, on-hold, etc

**Remove a book from shelf**

Removes a book from user's shelf.

URL:

HTTP method: DELETE

Parameters:

\* id: id of the book

**Add rating**

Adds a rating of a book from the user.

URL:

HTTP method: POST

Parameters:

\* book[book\_id]: id of the book (required)

\* book[date\_rated]: when book was rated (mm/dd/yy)

\* book[rating]: book user rating

**Edit rating**

Edits a rating of a book from the user.

URL:

HTTP method: PUT

Parameters:

\* book[book\_id]: id of the book (required)

\* book[read\_at]: when book was read

\* book[rating]: book user rating (1-10, 0 is default)

**Delete rating**

Deletes a rating of a book from the user.

URL:

HTTP method: DELETE

Parameters:

\* book[book\_id]: id of the book

**Get books from user's shelf**

Get the books on a user's shelf.

URL:

HTTP method: GET

Parameters:

\* id: id of the user (required)

\* state: read, currently-reading, to-read, dropped, on-hold

\* sort: title, author, rating, etc

**Find author by name**

Get an xml response for the given author name.

URL:

HTTP method: GET

Parameters:

\* id: author name

**Find book by name or author**

Get an xml response for the given book or author name.

URL:

HTTP method: GET

Parameters:

\* id: book or author name

\* field: field to search (title, author, both)