



API

INTRODUCTION

The Goodreads API allows developers access to Goodreads data in order to help websites or applications that deal with books be more personalized, social, and engaging. The API can be used in many ways, including:

[developer key](#)
[api terms](#)
[getting started](#)
[developer forums](#)
[contact us](#)

- **Review Syndication:** Goodreads has 10 million reviews across 700,000 titles - one of the largest and deepest collection of quality book reviews on the internet. Our API makes it easy to display these reviews on your website. Learn more about the [Review Partner Program](#).
- **Goodreads Connect:** Let members connect to their Goodreads accounts, and you'll have full access to the books in their shelves, their ratings, their reviews, and their friends - the social reading graph. Use this to personalize an ecommerce store, power recommendations, show a widget of a member's favorite books, build a mobile or desktop client app, and more. Learn more about [Goodreads Connect](#).

GETTING STARTED

- Most API methods will require you to register for a [developer key](#).
- All developers using the API must adhere to the [Goodreads API Terms & Conditions](#).
- Read examples about [how to use the API](#).
- For questions and discussion, visit the [Developer Group](#) or [contact us](#).

API METHODS

[auth.user](#) — Get id of user who authorized OAuth.

[author.books](#) — Paginate an author's books.

[author.show](#) — Get info about an author by id.

[author_following.create](#) — Follow an author.

[author_following.destroy](#) — Unfollow an author.

[author_following.show](#) — Show author following information.

[book.isbn_to_id](#) — Get Goodreads book IDs given ISBNs.

[book.id_to_work_id](#) — Get Goodreads work IDs given Goodreads book IDs.

[book.review_counts](#) — Get review statistics given a list of ISBNs.

[book.show](#) — Get the reviews for a book given a Goodreads book id.

[book.show_by_isbn](#) — Get the reviews for a book given an ISBN.

[book.title](#) — Get the reviews for a book given a title string.

[comment.create](#) — Create a comment.

[comment.list](#) — List comments on a subject.

[events.list](#) — Events in your area.

[fanship.create](#) — Become fan of an author. DEPRECATED.

[fanship.destroy](#) — Stop being fan of an author. DEPRECATED.

[fanship.show](#) — Show fanship information. DEPRECATED.

[followers.create](#) — Follow a user.

[followers.destroy](#) — Unfollow a user.

[friend.confirm_recommendation](#) — Confirm or decline a friend recommendation.

[friend.confirm_request](#) — Confirm or decline a friend request.

[friend.requests](#) — Get friend requests.

[friends.create](#) — Add a friend.

[group.join](#) — Join a group.

[group.list](#) — List groups for a given user.

[group.members](#) — Return members of a particular group.

My Books

Browse ▾

Community ▾

[list.book](#) — Get the listopia lists for a given book.

[notifications](#) — See the current user's notifications.

[owned_books.create](#) — Add to books owned.

[owned_books.list](#) — List books owned by a user.

[owned_books.show](#) — Show an owned book.

[owned_books.update](#) — Update an owned book.

[owned_books.destroy](#) — Delete an owned book.

[quotes.create](#) — Add a quote.

[rating.create](#) — Like a resource.

[rating.destroy](#) — Unlike a resource.

[read_statuses.show](#) — Get a user's read status.

[recommendations.show](#) — Get a recommendation from a user to another user.

[review.create](#) — Add review.

[review.edit](#) — Edit a review.

[review.destroy](#) — Delete a book review.

[reviews.list](#) — Get the books on a members shelf.

[review.recent_reviews](#) — Recent reviews from all members..

[review.show](#) — Get a review.

[review.show_by_user_and_book](#) — Get a user's review for a given book.

[search.authors](#) — Find an author by name.

[search.books](#) — Find books by title, author, or ISBN.

[series.show](#) — See a series.

[series.list](#) — See all series by an author.

[series.work](#) — See all series a work is in.

[shelves.add_to_shelf](#) — Add a book to a shelf.

[shelves.add_books_to_shelves](#) — Add books to many shelves.

[shelves.list](#) — Get a user's shelves.

[topic.create](#) — Create a new topic via OAuth.

[topic.group_folder](#) — Get list of topics in a group's folder.

[topic.show](#) — Get info about a topic by id.

[topic.unread_group](#) — Get a list of topics with unread comments.

[updates.friends](#) — Get your friend updates.

[user_shelves.create](#) — Add book shelf.

[user_shelves.update](#) — Edit book shelf.

[user.show](#) — Get info about a member by id or username.

[user.compare](#) — Compare books with another member.

[user.followers](#) — Get a user's followers.

[user.following](#) — Get people a user is following.

[user.friends](#) — Get a user's friends.

[user_status.create](#) — Update user status.

[user_status.destroy](#) — Delete user status.

[user_status.show](#) — Get a user status.

[user_status.index](#) — View user statuses.

[work.editions](#) — See all editions by work.

LINKS & WIDGETS

[book links](#) — how to build proper links to Goodreads pages.

[add to my books widget](#) — allow users to add books to their Goodreads shelf from your site.

[reviews widget](#) — embed Goodreads book reviews on your site with an iframe.

[share a link](#) — allow members to share a book related link from your site.

API DOCUMENTATION

Get id of user who authorized OAuth

Get an xml response with the Goodreads user_id for the user who authorized access using OAuth. You'll need to [register your app](#) (required).

URL: https://www.goodreads.com/api/auth_user

HTTP method: GET

Paginate an author's books

Get an xml response with a paginated list of an authors books.

URL: <https://www.goodreads.com/author/list.xml> (sample url)

HTTP method: GET

Parameters:

- key: [Developer key](#) (required).
- id: Goodreads Author id (required)
- page: 1-N (default 1)

Get info about an author by id

Get an xml response with info about an author.

URL: <https://www.goodreads.com/author/show.xml> (sample url)

HTTP method: GET

Parameters:

- key: [Developer key](#) (required).
- id: Goodreads Author id.

Follow an author

Make the signed-in user follow an author [using OAuth](#). You'll need to [register your app](#) (required).

URL: https://www.goodreads.com/author_followings?id=AUTHOR_ID&format=xml

HTTP method: POST

Unfollow an author

Unfollow an author [using OAuth](#). You'll need to [register your app](#) (required).

URL: https://www.goodreads.com/author_followings/AUTHOR_FOLLOWING_ID?format=xml

HTTP method: DELETE

Show author following information

Get an xml response [using OAuth](#) describing the association between a user and an author. You'll need to [register your app](#) (required).

URL: https://www.goodreads.com/author_followings/AUTHOR_FOLLOWING_ID?format=xml

HTTP method: GET

Get Goodreads book IDs given ISBNs

Get Goodreads book IDs given one or more ISBNs. Response contains IDs without any markup.

URL: https://www.goodreads.com/book/isbn_to_id (sample url)

HTTP method: GET

Parameters:

- key: [Developer key](#) (required).
- isbn: ISBNs of books to look up.

Get Goodreads work IDs given Goodreads book IDs

Get Goodreads work IDs given one or more Goodreads book IDs. Response contains work IDs without any markup.

URL: https://www.goodreads.com/book/id_to_work_id (sample url)

HTTP method: GET

Parameters:

- key: [Developer key](#) (required).
- id: Book IDs of books to look up.

Get review statistics given a list of ISBNs

Get review statistics for books given a list of ISBNs. ISBNs can be specified as an array (e.g.

My Books

Browse ▾

Community ▾

URL: https://www.goodreads.com/book/review_counts.json (sample url)

HTTP method: GET

Parameters:

- key: [Developer key](#) (required).
- isbn: Array of ISBNs or a comma separated string of ISBNs (1000 ISBNs per request max.)
- format: json
- callback: function to wrap JSON response

Get the reviews for a book given a Goodreads book id

Get an XML or JSON response that contains embed code for the iframe reviews widget. The reviews widget shows an excerpt (first 300 characters) of the most popular reviews of a book for a given internal Goodreads book_id. Reviews of all known editions of the book are included.

XML responses also include shelves and book meta-data (title, author, et cetera). The Goodreads API gives you full access to Goodreads-owned meta-data, but it does not give you full access to book meta-data supplied by third parties such as Ingram. Book cover images, descriptions, and other data from third party sources might be excluded, because we do not have a license to distribute these data via our API.

If you need book meta-data beyond what our API provides, consider signing up for an Amazon developer key.

URL: <https://www.goodreads.com/book/show.FORMAT> (sample url)

HTTP method: GET

Parameters:

- format: xml or json
- key: [Developer key](#) (required).
- id: A Goodreads internal book_id
- text_only: Only show reviews that have text (default false)
- rating: Show only reviews with a particular rating (optional)

Get the reviews for a book given an ISBN

Get an xml or json response that contains embed code for the iframe reviews widget that shows excerpts (first 300 characters) of the most popular reviews of a book for a given ISBN. The reviews are from all known editions of the book.

URL: <https://www.goodreads.com/book/isbn/ISBN?format=FORMAT> (sample url)

HTTP method: GET

Parameters:

- format: xml or json
- callback: function to wrap JSON response if format=json
- key: [Developer key](#) (required only for XML).
- user_id: USER_ID (required only for JSON)
- isbn: The ISBN of the book to lookup.
- rating: Show only reviews with a particular rating (optional)

Example code for using json with callback:

```
<script type="text/javascript">
function myCallback(result) {
  alert('nb of reviews for book: ' + result.reviews.length);
}
var scriptTag = document.createElement('script');
scriptTag.src = "https://www.goodreads.com/book/isbn/0441172717?callback=myCallback&format=json&user_id=USER_ID";
document.getElementsByTagName('head')[0].appendChild(scriptTag);
</script>
```

Get the reviews for a book given a title string

Get an xml response that contains embed code for the iframe reviews widget, which shows an excerpt (first 300 characters) of the most popular reviews of a book for a given title/author. The book shown will be the most popular book that matches all the keywords in the input string. The reviews are from all known editions of the book.

URL: <https://www.goodreads.com/book/title.FORMAT> (sample url)

HTTP method: GET

Parameters:

- format: xml or json
- key: [Developer key](#) (required).

- title: The title of the book to lookup.

My Books

Browse ▾

Community ▾

Create a comment

Creates a new comment. You'll need to [register your app](#) (required).

URL: <https://www.goodreads.com/comment.xml>

HTTP method: POST

Parameters:

- type: one of 'author_blog_post', 'author_following', 'blog', 'book_news_post', 'chapter', 'comment', 'community_answer', 'event_response', 'friend', 'giveaway', 'giveaway_request', 'group_user', 'interview', 'librarian_note', 'link_collection', 'list', 'owned_book', 'photo', 'poll', 'poll_vote', 'queued_item', 'question', 'question_user_stat', 'quiz', 'quiz_score', 'rating', 'read_status', 'recommendation', 'recommendation_request', 'reading_year', 'review', 'review_proxy', 'services/models/reading_note', 'sharing', 'topic', 'user', 'user_challenge', 'user_following', 'user_list_challenge', 'user_list_vote', 'user_quote', 'user_status', 'video'
- id: Id of resource given as type param
- comment[body]: This review was really insightful!

List comments on a subject

Lists comments

URL: <https://www.goodreads.com/comment.xml> (sample url)

HTTP method: GET

Parameters:

- type: one of 'author_blog_post', 'author_following', 'blog', 'book_news_post', 'chapter', 'comment', 'community_answer', 'event_response', 'friend', 'giveaway', 'giveaway_request', 'group_user', 'interview', 'librarian_note', 'link_collection', 'list', 'owned_book', 'photo', 'poll', 'poll_vote', 'queued_item', 'question', 'question_user_stat', 'quiz', 'quiz_score', 'rating', 'read_status', 'recommendation', 'recommendation_request', 'reading_year', 'review', 'review_proxy', 'services/models/reading_note', 'sharing', 'topic', 'user', 'user_challenge', 'user_following', 'user_list_challenge', 'user_list_vote', 'user_quote', 'user_status', 'video'
- id: Id of resource given as type param
- page: 1-N (optional, default 1)

Events in your area

Shows events nearby the authenticating user or you can get a list of events near a location by passing lat/lng coordinates

URL: <https://www.goodreads.com/event/index.xml> (sample url)

HTTP method: GET

Parameters:

- key: [Developer key](#) (required).
- lat: Latitude (optional)
- lng: Longitude (optional)
- search[country_code]: 2 characters country code (optional)
- search[postal_code]: ZIP code (optional)

Become fan of an author. DEPRECATED

Make the signed-in user become a fan of an author [using OAuth](#). You'll need to [register your app](#) (required). This method is DEPRECATED and uses author_following data. Please migrate to author_following.create before July 1st, 2015.

URL: <https://www.goodreads.com/fanships?format=xml>

HTTP method: POST

Parameters:

- fanship[author_id]: id of the author (required)

Stop being fan of an author. DEPRECATED

Stop being a fan of an author [using OAuth](#). You'll need to [register your app](#) (required). This method is DEPRECATED and uses author_following data. Please migrate to author_following.destroy before July 1st, 2015.

URL: https://www.goodreads.com/fanships/FANSHIP_ID?format=xml

HTTP method: DELETE

Show fanship information. DEPRECATED

Get an xml response [using OAuth](#) describing the association between a user and an author. You'll need to [register your app](#) (required). This method is DEPRECATED and uses author_following data. Please migrate to author_following.show before July 1st, 2015.

URL: https://www.goodreads.com/fanships/show/FANSHIP_ID?format=xml

HTTP method: GET

My Books

Browse ▾

Community ▾

Start following a user using OAuth. You'll need to register your app (required).

URL: https://www.goodreads.com/user/USER_ID/followers?format=xml

HTTP method: POST

Parameters:

- USER_ID: Goodreads user id of user to follow

Unfollow a user

Stop following a user using OAuth. You'll need to register your app (required).

URL: https://www.goodreads.com/user/USER_ID/followers/stop_following.xml

HTTP method: DELETE

Parameters:

- USER_ID: Goodreads user id of the user you want to stop following

Confirm or decline a friend recommendation

Confirm or decline a friend recommendation for the current user using OAuth. You'll need to register your app (required).

URL: https://www.goodreads.com/friend/confirm_recommendation.xml

HTTP method: POST

Parameters:

- id: friend recommendation id
- response: Y or N

Confirm or decline a friend request

Confirm or decline a friend request for the current user using OAuth. You'll need to register your app (required).

URL: https://www.goodreads.com/friend/confirm_request.xml

HTTP method: POST

Parameters:

- id: friend request id
- response: Y or N

Get friend requests

Returns a XML with the current user's friend requests 'using OAuth. You'll need to register your app (required).

URL: <https://www.goodreads.com/friend/requests.xml>

HTTP method: GET

Parameters:

- page: 1-N page of results to show (optional, default 1)

Add a friend

Sends a friend request to a user using OAuth. You'll need to register your app (required).

URL: https://www.goodreads.com/friend/add_as_friend.xml

HTTP method: POST

Parameters:

- id: Goodreads user id for friend

Join a group

Let the current user join a given group using OAuth. You'll need to register your app (required).

URL: https://www.goodreads.com/group/join?format=xml&id=GROUP_ID

HTTP method: POST

Parameters:

- id: id of the group (required)
- format: xml (required)

List groups for a given user

Returns xml list of groups the user specified by id belongs to

URL: https://www.goodreads.com/group/list/USER_ID.xml (sample url)

HTTP method: GET

Parameters:

- sort: One of 'my_activity', 'members', 'last_activity', 'title' ('members' will sort by number of members in the group)
- key: Developer key (required).

My Books

Browse ▾

Community ▾

URL: https://www.goodreads.com/group/members/GROUP_ID.xml (sample url)**HTTP method:** GET**Parameters:**

- sort: One of 'last_online', 'num_comments', 'date_joined', 'num_books', 'first_name'
- q: List of names to search for, separating each name with a space character. Optional, will find all members by default
- page: Which page of results to show (default 1)
- key: [Developer key](#) (required).

Find a group

Search group titles and descriptions for the given string

URL: https://www.goodreads.com/group/search.xml?q=SEARCH_STRING (sample url)**HTTP method:** GET**Parameters:**

- q: The query string
- page: page number (optional, default 1)
- key: [Developer key](#) (required).

Get info about a group by id

XML version of group/show

URL: https://www.goodreads.com/group/show/GROUP_ID.xml (sample url)**HTTP method:** GET**Parameters:**

- sort: Field to sort topics by. One of 'comments_count', 'title', 'updated_at', 'views'
- order: 'a' for ascending, 'd' for descending
- key: [Developer key](#) (required).

Get the listopia lists for a given book

XML version of list/book. This API requires extra permission please [contact us](#)**URL:** https://www.goodreads.com/list/book/BOOK_ID.xml (sample url)**HTTP method:** GET**Parameters:**

- key: [Developer key](#) (required).

See the current user's notifications

Viewing any new notifications here will mark them as 'viewed'. using [OAuth](#)**URL:** </notifications.xml> (sample url)**HTTP method:** GET**Parameters:**

- page: 1-N page number (optional, default 1)

Add to books owned

Adds a book to user's list of owned books using [OAuth](#). You'll need to [register your app](#) (required).**URL:** https://www.goodreads.com/owned_books.xml**HTTP method:** POST**Parameters:**

- owned_book[book_id]: id of the book (required)
- owned_book[condition_code]: one of 10 (brand new), 20 (like new), 30 (very good), 40 (good), 50 (acceptable), 60 (poor)
- owned_book[condition_description]: description of book's condition
- owned_book[original_purchase_date]: when book was purchased
- owned_book[original_purchase_location]: where this book was purchased
- owned_book[unique_code]: BookCrossing id (BCID)

List books owned by a user

Get an xml response with a list of owned books using [OAuth](#). You'll need to [register your app](#) (required).**URL:** https://www.goodreads.com/owned_books/user?format=xml (sample url)**HTTP method:** GET**Parameters:**

- id: Goodreads user_id
- page: 1-N (optional, default 1)

My Books

Browse ▾

Community ▾

it's BookCrossing ID (bcid), number of times traded, etc. OWNED_BOOK_ID is a unique identifier for the owned book (not a book_id). You'll need to [register your app](#) (required).

URL: https://www.goodreads.com/owned_books/show/OWNED_BOOK_ID?format=xml

HTTP method: GET

Update an owned book

Updates a book a user owns [using OAuth](#). You'll need to [register your app](#) (required).

URL: https://www.goodreads.com/owned_books/update/ID?format=xml

HTTP method: PUT

Parameters:

- id: id of the owned book record
- owned_book[book_id]: id of the book (required)
- owned_book[condition_code]: one of 10 (brand new), 20 (like new), 30 (very good), 40 (good), 50 (acceptable), 60 (poor), (optional)
- owned_book[condition_description]: description of book's condition (optional)
- owned_book[original_purchase_date]: when book was purchased (optional)
- owned_book[original_purchase_location]: where this book was purchased (optional)
- owned_book[unique_code]: BookCrossing id (BCID) (optional)

Delete an owned book

Deletes a book a user owns [using OAuth](#). You'll need to [register your app](#) (required).

URL: https://www.goodreads.com/owned_books/destroy/ID?format=xml

HTTP method: POST

Parameters:

- id: id of the owned book record

Add a quote

Add a quote [using OAuth](#). If you don't specify an author_id, it will try to look one up based on the author_name you provide. You'll need to [register your app](#) (required).

URL: <https://www.goodreads.com/quotes?format=xml>

HTTP method: POST

Parameters:

- quote[author_name]: Name of the quote author (required)
- quote[author_id]: id of the author
- quote[book_id]: id of the book from which the quote was taken
- quote[body]: The quote! (required)
- quote[tags]: Comma-separated tags
- isbn: ISBN of the book from which the quote was taken. This will not override the book_id if it was provided

Like a resource

Like a resource (e.g. review or status update) [using OAuth](#). You'll need to [register your app](#) (required).

URL: <https://www.goodreads.com/rating?format=xml>

HTTP method: POST

Parameters:

- rating[rating]: 1 (required)
- rating[resource_id]: id of the resource being liked (required)
- rating[resource_type]: camel case name of the resource type (e.g. UserStatus, Review) (required)

Unlike a resource

Unlike a resource (e.g. review or status update) [using OAuth](#). You'll need to [register your app](#) (required).

URL: <https://www.goodreads.com/rating?format=xml>

HTTP method: DELETE

Parameters:

- id: rating id

Get a user's read status

Get information about a read status update.

URL: https://www.goodreads.com/read_statuses/ID?format=xml&key=KEY

HTTP method: GET

Parameters:

- key: [Developer key](#) (required).

- id: read status id

My Books

Browse ▾

Community ▾

Get information about a particular recommendation that one user made for another using [OAuth](#). Includes comments and likes. You'll need to [register your app](#) (required).

URL: <https://www.goodreads.com/recommendations/ID?format=xml>

HTTP method: GET

Parameters:

- id: recommendation id

Add review

Add book reviews for members using [OAuth](#). You'll need to [register your app](#) (required).

URL: <https://www.goodreads.com/review.xml>

HTTP method: POST

Parameters:

- book_id: Goodreads book_id (required)
- review[review]: Text of the review (optional)
- review[rating]: Rating (0-5) (optional, default is 0 (No rating))
- review[read_at]: Date (YYYY-MM-DD format, e.g. 2008-02-01) (optional)
- shelf: read|currently-reading|to-read|<USER SHELF NAME> (optional, must exist, see: [shelves.list](#))

Edit a review

Edit a book review using [OAuth](#). You'll need to [register your app](#) (required). This method should be called with a PUT request but we support POST as well.

URL: https://www.goodreads.com/review/REVIEW_ID.xml

HTTP method: PUT

Parameters:

- id: Review Id
- review[review]: Text of the review (optional)
- review[rating]: Rating (0-5) (optional, default is 0 (No rating))
- review[read_at]: Date (YYYY-MM-DD format, e.g. 2008-02-01) (optional)
- finished: true to mark finished reading (optional)
- shelf: read|currently-reading|to-read|<USER SHELF NAME> (optional, must exist, see: [shelves.list](#))

Delete a book review

Deletes a book review using [OAuth](#). You'll need to [register your app](#) (required). This method should be called with a DELETE request but we support POST as well.

URL: <https://www.goodreads.com/review/destroy/12345?format=xml>

HTTP method: DELETE

Parameters:

- id: Review Id

Get the books on a members shelf

Get the books on a members shelf. Customize the feed with the below variables. Viewing members with profiles who have set them as visible to members only or just their friends requires using [OAuth](#).

URL: <https://www.goodreads.com/review/list?v=2> (sample url)

HTTP method: GET

Parameters:

- v: 2
- id: Goodreads id of the user
- shelf: read, currently-reading, to-read, etc. (optional)
- sort: title, author, cover, rating, year_pub, date_pub, date_pub_edition, date_started, date_read, date_updated, date_added, recommender, avg_rating, num_ratings, review, read_count, votes, random, comments, notes, isbn, isbn13, asin, num_pages, format, position, shelves, owned, date_purchased, purchase_location, condition (optional)
- search[query]: query text to match against member's books (optional)
- order: a, d (optional)
- page: 1-N (optional)
- per_page: 1-200 (optional)
- key: [Developer key](#) (required).

Recent reviews from all members.

Get an xml response with the most recently added reviews from all members.

URL: https://www.goodreads.com/review/recent_reviews.xml (sample url)

HTTP method: GET

My Books

Browse ▾

Community ▾

Get a review

Get an xml response that contains the review and rating

URL: <https://www.goodreads.com/review/show.xml> (sample url)

HTTP method: GET

Parameters:

- key: [Developer key](#) (required).
- id: id of the review
- page: 1-N. The page number of comments. (default 1, optional)

Get a user's review for a given book

Get an xml response that contains the review and rating for the specified book and user

URL: https://www.goodreads.com/review/show_by_user_and_book.xml (sample url)

HTTP method: GET

Parameters:

- key: [Developer key](#) (required).
- user_id: id of the user
- book_id: id of the book
- include_review_on_work: 'true' or 'false' indicating whether to return a review for another book in the same work if review not found for the specified book (default 'false', optional)

Find an author by name

Get an xml response with the Goodreads url for the given author name.

URL: https://www.goodreads.com/api/author_url/<ID> (sample url)

HTTP method: GET

Parameters:

- id: Author name
- key: [Developer key](#) (required).

Find books by title, author, or ISBN

Get an xml response with the most popular books for the given query. This will search all books in the title/author/ISBN fields and show matches, sorted by popularity on Goodreads. There will be cases where a result is shown on the Goodreads site, but not through the API. This happens when the result is an Amazon-only edition and we have to honor Amazon's terms of service.

URL: <https://www.goodreads.com/search/index.xml> (sample url)

HTTP method: GET

Parameters:

- q: The query text to match against book title, author, and ISBN fields. Supports boolean operators and phrase searching.
- page: Which page to return (default 1, optional)
- key: [Developer key](#) (required).
- search[field]: Field to search, one of 'title', 'author', or 'all' (default is 'all')

See a series

Info on a series

URL: <https://www.goodreads.com/series/show/ID.xml> (sample url)

HTTP method: GET

See all series by an author

List of all series by an author

URL: /series/list?format=xml&id=AUTHOR_ID (sample url)

HTTP method: GET

See all series a work is in

List of all series a work is in

URL: https://www.goodreads.com/series/work/WORK_ID?format=xml (sample url)

HTTP method: GET

Add a book to a shelf

Add a book to a shelf [using OAuth](#). This method can also be used to remove from shelf. You'll need to [register your app](#)

My Books

Browse ▾

Community ▾

Parameters:

- name: Name of the shelf (see: [shelves.list](#))
- book_id: id of the book to add to the shelf
- a: Leave this blank unless you're removing from a shelf. If removing, set this to 'remove'. (optional)

Add books to many shelves

Add a list of books to many current user's shelves [using OAuth](#). You'll need to [register your app](#) (required).

URL: https://www.goodreads.com/shelf/add_books_to_shelves.xml

HTTP method: POST

Parameters:

- bookids: comma-separated list of book ids
- shelves: comma-separated list of shelf names

Get a user's shelves

Lists shelves for a user

URL: <https://www.goodreads.com/shelf/list.xml> (sample url)

HTTP method: GET

Parameters:

- key: [Developer key](#) (required).
- user_id: Goodreads user id (required)
- page: 1-N (default 1)

Create a new topic via OAuth

Create a new topic [using OAuth](#). You'll need to [register your app](#) (required).

URL: <https://www.goodreads.com/topic.xml>

HTTP method: POST

Parameters:

- topic[subject_type]: Either 'Book' or 'Group'. If 'Book', the book the topic is about. If 'Group', the group that the topic belongs to. (required)
- topic[subject_id]: The id for the subject the topic belongs to, either book_id or group_id, as appropriate (required)
- topic[folder_id]: If the subject is a group, you can supply a folder id to add the topic to. Be sure that the folder belongs to the group. By default, if the subject_type is 'Group', then the topic will be added to the 'general' folder
- topic[title]: Title for the topic (required)
- topic[question_flag]: Indicates whether the topic is a discussion question ('1') or not ('0'). Default is 0 (non-question)
- comment[body_usertext]: The text of the comment that starts the topic thread. Can include Goodreads book/author tags of the form [book:Title|ID] (required)
- update_feed: Indicates whether the comment for the new topic should be added to the user's update feed. To enable, set to 'on'; otherwise, default is not to add to update feed
- digest: Indicates whether the user would like to receive an email when someone replies to the topic (user will get one email only). To enable, set to 'on'; otherwise, default is not to add to update feed

Get list of topics in a group's folder

Returns a list of topics in a group's folder specified either by folder id or by group id.

URL: https://www.goodreads.com/topic/group_folder/FOLDER_ID?format=xml (sample url)

HTTP method: GET

Parameters:

- group_id: If supplied and id is set to 0, then will return topics from the general folder for the group indicated by group_id. If id is non-zero, this param is ignored. Note: may return 404 if there are no topics in the general folder for the specified group
- page: Page in results to show, 1-N (default 1)
- sort: One of 'comments_count', 'title', 'updated_at', 'views'
- order: 'a' for ascending, 'd' for descending
- key: [Developer key](#) (required).

Get info about a topic by id

XML version of topic/show

URL: https://www.goodreads.com/topic/show.xml?id=TOPIC_ID (sample url)

HTTP method: GET

Parameters:

- key: [Developer key](#) (required).

My Books

Browse ▾

Community ▾

Get a list of topics from a specified group that have comments added since the last time the user viewed the topic using [OAuth](#). You'll need to [register your app](#) (required).

URL: https://www.goodreads.com/topic/unread_group/GROUP_ID?format=xml (sample url)

HTTP method: GET

Parameters:

- viewed: Indicates whether to show topics user has viewed before or not. Default is to include all topics; set this param to 'true' or '1' to restrict to only topics the user has already viewed
- page: Page in results to show, 1-N (default 1)
- sort: One of 'comments_count', 'title', 'updated_at', 'views'
- order: 'a' for ascending, 'd' for descending

Get your friend updates

Get your friend updates (the same data you see on your homepage) using [OAuth](#). You'll need to [register your app](#) (required).

URL: <https://www.goodreads.com/updates/friends.xml>

HTTP method: GET

Parameters:

- update: Type of update. Valid values are: books, reviews, statuses. (optional, default all)
- update_filter: Which updates to show. Options are: friends (default - includes followers), following, top_friends. (optional)
- max_updates: The max limit of updates. (optional)

Add book shelf

Add book shelves for members using [OAuth](#). You'll need to [register your app](#) (required).

URL: https://www.goodreads.com/user_shelves.xml

HTTP method: POST

Parameters:

- user_shelf[name]: Name of the new shelf
- user_shelf[exclusive_flag]: 'true' or 'false' (optional, default false)
- user_shelf[sortable_flag]: 'true' or 'false' (optional, default false)
- user_shelf[featured]: 'true' or 'false' (optional, default false)

Edit book shelf

Edit a shelf using [OAuth](#). You'll need to [register your app](#) (required).

URL: https://www.goodreads.com/user_shelves/USER_SHELF_ID.xml

HTTP method: PUT

Parameters:

- user_shelf[name]: Name of the new shelf
- user_shelf[exclusive_flag]: 'true' or 'false' (optional, default false)
- user_shelf[sortable_flag]: 'true' or 'false' (optional, default false)
- user_shelf[featured]: 'true' or 'false' (optional, default false)

Get info about a member by id or username

Get an xml response with the public information about the given Goodreads user.

URL: <https://www.goodreads.com/user/show.xml?id=false> (sample url)

HTTP method: GET

Parameters:

- key: [Developer key](#) (required).
- id: Goodreads user id.
- username: Goodreads user name (not first name). Usernames are optional on Goodreads.

Compare books with another member

Get an xml response with stats comparing your books to another member's using [OAuth](#). You'll need to [register your app](#) (required).

URL: <https://www.goodreads.com/user/compare/1.xml>

HTTP method: GET

Parameters:

- id: Goodreads user_id you want to compare books to

Get a user's followers

Get an xml response with the given user's followers using OAuth.

My Books

Browse ▾

Community ▾

- key: [Developer key](#) (required).
- page: 1-N (optional, default 1)

Get people a user is following

Get an xml response with people the given user is following using OAuth.

URL: https://www.goodreads.com/user/USER_ID/following.xml (sample url)

HTTP method: GET

Parameters:

- key: [Developer key](#) (required).
- page: 1-N (optional, default 1)

Get a user's friends

Get an xml response with the given user's friends using OAuth.

URL: <https://www.goodreads.com/friend/user.xml> (sample url)

HTTP method: GET

Parameters:

- id: Goodreads user_id (required)
- page: 1-N (optional, default 1)
- sort: first_name|date_added|last_online (optional)

Update user status

Add status updates for members using OAuth. You'll need to [register your app](#) (required).

URL: https://www.goodreads.com/user_status.xml

HTTP method: POST

Parameters:

- user_status[book_id]: id of the book being reviewed (optional)
- user_status[page]: page of the book (optional)
- user_status[percent]: percent complete (use instead of page if appropriate)
- user_status[body]: status update (required, unless page or percent is present, then it is optional)

Delete user status

Delete a status update for a member using OAuth. You'll need to [register your app](#) (required).

URL: https://www.goodreads.com/user_status/destroy/USER_STATUS_ID?format=xml

HTTP method: POST

Get a user status

Get information about a user status update.

URL: https://www.goodreads.com/user_status/show/ID?format=xml&key=KEY

HTTP method: GET

Parameters:

- key: [Developer key](#) (required).
- id: user status id

View user statuses

View most recent user statuses on the site.

URL: https://www.goodreads.com/user_status/index.xml

HTTP method: GET

See all editions by work

List of all the available editions of a particular work. This API requires extra permission please [contact us](#)

URL: /work/editions/WORK_ID?format=xml (sample url)

HTTP method: GET

Parameters:

- key: [Developer key](#) (required).

BOOK LINKS

- by ISBN: <https://www.goodreads.com/book/isbn/0590353403>


• add a review: <https://www.goodreads.com/review/isbn/0590353403>

ADD TO MY BOOKS WIDGET

g | Add book

97278 people read it (3.98 average rating)

6 friends have added this.



Allows visitors to your site to add the book to their shelf on Goodreads with one easy click.

Add Widget »

REVIEWS WIDGET

This is a widget designed for authors, publishers, or online book catalogues (booksellers, libraries, etc) that will display the most popular reviews given an ISBN. The reviews are from all editions of the book.

[Add the reviews widget »](#)

SHARE A LINK

Allows members to share links from around the internet about books on their Goodreads profiles. You can pass in the url and the ISBN to the link.

```
<a target="_blank"
href="https://www.goodreads.com/update_status?
isbn=0316015849&url=https%3A%2F%2Fwww.goodreads.com%2Fbo
ok%2Fshow%2F19501"><img alt="Share on Goodreads" border="0"
```

LOGOS AND IMAGES



Mark Twain Bookmark: [PDF](#)



Madeleine L'Engle Bookmark: [PDF](#)



Patricia McKillip Bookmark: [PDF](#)



Charles Eliot Bookmark: [PDF](#)

Goo
drea
ds
bad
ges

My Books

Browse ▾

Community ▾

g

g

g

g

g

g huge

+ Add to

goodreads

Read reviews on

goodreads

g

Login with goodreads