

readme.md for @apiclient.xyz/bookstack

A fully-typed TypeScript API client for [BookStack](#) — the open-source wiki and documentation platform. Covers the entire BookStack REST API with rich domain objects, auto-pagination, and export support.

Issue Reporting and Security

For reporting bugs, issues, or security vulnerabilities, please visit [community.foss.global/](#). This is the central community hub for all issue reporting. Developers who sign and comply with our contribution agreement and go through identification can also get a [code.foss.global/](#) account to submit Pull Requests directly.

Install

```
pnpm install @apiclient.xyz/bookstack
```

Quick Start

```
import { BookStackAccount } from '@apiclient.xyz/bookstack';

const bookstack = new BookStackAccount(
  'https://your-bookstack-instance.com',
  'your-token-id',
  'your-token-secret',
);

// ☐☐ Test the connection
```

```
const result = await bookstack.testConnection();
console.log(result); // { ok: true }

// List all books
const books = await bookstack.getBooks();

// Create a page
const page = await bookstack.createPage({
  name: 'Getting Started',
  book_id: 1,
  markdown: '# Welcome\nThis is your first page.',
});
```

Authentication

BookStack uses token-based authentication. Generate an API token from your BookStack user profile under **API Tokens**. You'll get a **Token ID** and a **Token Secret** — pass both to the constructor:

```
const bookstack = new BookStackAccount(
  'https://wiki.example.com', // Your BookStack instance URL
  'Fpa7dZR2uJcAT86RmoyLrn5', // Token ID
  '0jrMXAjvxnDbhk2hYAuUJK', // Token Secret
);
```

The user must have the **"Access System API"** role permission enabled in BookStack.

Content Hierarchy

BookStack organizes content in a clear hierarchy. The client mirrors this with rich domain objects:

```
Shelves
├── Books
│   ├── Chapters (optional)
│   │   └── Pages
│   └── Pages (directly in book)
```

Each entity class holds a reference back to the account, so you can navigate the hierarchy fluently.

API Reference

☐☐ Books

```
// List all books (auto-paginated)
const books = await bookstack.getBooks();

// Get a single book
const book = await bookstack.getBook(1);

// Create a book
const newBook = await bookstack.createBook({
  name: 'Engineering Docs',
  description: 'Internal engineering documentation',
  tags: [{ name: 'team', value: 'backend' }],
});

// Update & delete (via the domain object)
const updated = await book.update({ name: 'Updated Title' });
await book.delete();

// Navigate to children
const chapters = await book.getChapters();
const pages = await book.getPages();

// Export
const html = await book.export('html'); // string
const pdf = await book.export('pdf'); // Uint8Array
const markdown = await book.export('markdown'); // string
```

☐☐ Chapters

```
const chapters = await bookstack.getChapters();
const chapter = await bookstack.getChapter(1);
```

```
const newChapter = await bookstack.createChapter({
  book_id: 1,
  name: 'Architecture',
  description: 'System architecture docs',
});

// Navigate to pages within this chapter
const pages = await chapter.getPages();
const page = await chapter.createPage({
  name: 'Overview',
  markdown: '# Architecture Overview',
});

// Export
const text = await chapter.export('plaintext');
```

☐ Pages

```
const pages = await bookstack.getPages();
const page = await bookstack.getPage(1);

// Create in a book or chapter
const page1 = await bookstack.createPage({
  book_id: 1,
  name: 'Quick Start',
  markdown: '# Getting Started\n...',
});

const page2 = await bookstack.createPage({
  chapter_id: 3,
  name: 'API Reference',
  html: '<h1>API Reference</h1><p>...\</p>',
});

// Update content
const updated = await page.update({
  markdown: '# Updated Content',
```

```
});

// Comments
const comments = await page.getComments();
await page.addComment({ html: '<p>Looks good!</p>' });

// Export
const md = await page.export('markdown');
```

📖 Shelves

```
const shelves = await bookstack.getShelves();
const shelf = await bookstack.getShelf(1);

const newShelf = await bookstack.createShelf({
  name: 'Backend Services',
  description: 'All backend service documentation',
  books: [1, 2, 3], // Array of book IDs
});

// Get books on this shelf
const books = await shelf.getBooks();

// Update book assignments
await shelf.update({ books: [1, 2, 3, 4] });
```

🔍 Search

```
const results = await bookstack.search('deployment guide');
// Returns: IBookStackSearchResult[] with id, name, type, url, preview_html
```

📎 Attachments

```
const attachments = await bookstack.getAttachments();
const attachment = await bookstack.getAttachment(1);
```

```
// Create a link attachment
const link = await bookstack.createAttachment({
  name: 'External Reference',
  uploaded_to: 1, // Page ID
  link: 'https://example.com/docs',
});

await bookstack.updateAttachment(1, { name: 'Renamed' });
await bookstack.deleteAttachment(1);
```

☐☐ Comments

```
const comments = await bookstack.getComments();
const comment = await bookstack.getComment(1);

await bookstack.createComment({
  page_id: 1,
  html: '<p>Great article!</p>',
  reply_to: 2, // Reply to comment with local_id 2
});

await bookstack.updateComment(1, { html: '<p>Updated comment</p>' });
await bookstack.deleteComment(1);
```

☐☐ Image Gallery

```
const images = await bookstack.getImages();
const image = await bookstack.getImage(1);
await bookstack.updateImage(1, { name: 'diagram-v2' });
await bookstack.deleteImage(1);
```

☐☐ Users & Roles

```
// Users
const users = await bookstack.getUsers();
const user = await bookstack.getUser(1);
```

```
const newUser = await bookstack.createUser({
  name: 'Jane Doe',
  email: 'jane@example.com',
  roles: [1, 2],
  send_invite: true,
});
await bookstack.updateUser(1, { name: 'Jane Smith' });
await bookstack.deleteUser(1, { migrate_ownership_id: 2 });

// Roles
const roles = await bookstack.getRoles();
const role = await bookstack.getRole(1);
await bookstack.createRole({
  display_name: 'Editor',
  description: 'Can edit content',
  permissions: ['page-update', 'chapter-update'],
});
```

☐☐ Admin Endpoints

```
// System info
const info = await bookstack.getSystemInfo();
// { version, instance_id, app_name, base_url }

// Audit log
const logs = await bookstack.getAuditLog();

// Recycle bin
const deleted = await bookstack.getRecycleBinItems();
await bookstack.restoreRecycleBinItem(1); // { restore_count: 3 }
await bookstack.destroyRecycleBinItem(1); // { delete_count: 3 }

// Content permissions
const perms = await bookstack.getContentPermissions('book', 1);
await bookstack.updateContentPermissions('book', 1, {
  owner_id: 2,
  role_permissions: [
    { role_id: 1, view: true, create: true, update: true, delete: false },
```

```
],  
  fallback_permissions: { inheriting: true },  
});
```

Auto-Pagination

All list methods automatically paginate through results. BookStack uses offset-based pagination (max 500 per request). The client handles this transparently:

```
// Gets ALL books, regardless of how many exist  
const allBooks = await bookstack.getBooks();  
  
// Pass options to control pagination behavior  
const firstPage = await bookstack.getBooks({ offset: 0, count: 10 });  
  
// Sort and filter  
const filtered = await bookstack.getBooks({  
  sort: '+name',  
  filter: { 'name:like': '%Guide%' },  
});
```

Exported Types

All interfaces are exported for use in your own code:

```
import type {  
  IBookStackBook,  
  IBookStackChapter,  
  IBookStackPage,  
  IBookStackShelf,  
  IBookStackAttachment,  
  IBookStackComment,  
  IBookStackUser,  
  IBookStackRole,  
  IBookStackSearchResult,  
  IBookStackTag,  
}
```

```
IBookStackListParams,  
TBookStackExportFormat,  
} from '@apiclient.xyz/bookstack';
```

License and Legal Information

This repository contains open-source code licensed under the MIT License. A copy of the license can be found in the [license](#) file.

Please note: The MIT License does not grant permission to use the trade names, trademarks, service marks, or product names of the project, except as required for reasonable and customary use in describing the origin of the work and reproducing the content of the NOTICE file.

Trademarks

This project is owned and maintained by Task Venture Capital GmbH. The names and logos associated with Task Venture Capital GmbH and any related products or services are trademarks of Task Venture Capital GmbH or third parties, and are not included within the scope of the MIT license granted herein.

Use of these trademarks must comply with Task Venture Capital GmbH's Trademark Guidelines or the guidelines of the respective third-party owners, and any usage must be approved in writing. Third-party trademarks used herein are the property of their respective owners and used only in a descriptive manner, e.g. for an implementation of an API or similar.

Company Information

Task Venture Capital GmbH Registered at District Court Bremen HRB 35230 HB, Germany

For any legal inquiries or further information, please contact us via email at hello@task.vc.

By using this repository, you acknowledge that you have read this section, agree to comply with its terms, and understand that the licensing of the code does not imply endorsement by Task Venture Capital GmbH of any derivative works.

Revision #6

Created 2026-03-28 10:48:22 UTC by foss.global Team

Updated 2026-03-28 12:14:18 UTC by foss.global Team