API Doc Template

- How to copy
- Insert endpoint block
- Getting Started
- Widgets
  - List widgets
  - Get widget
  - Create widget
  - Delete widget

How to copy

如何复制使用

这是什么
本书是酷可上编写「产品 HTTP API 文档」的通用模板：一端点一页，章按资源分，样式用甲 ·浅底方章（ .cb-api-*）。系统平台 API 仍在 /api/docs，请勿把产品接口写进那个地址。

最快开写
新建一书（或复制本书，若站点开放复制）。
建概述页：粘贴「开始使用」骨架（ Base URL / 鉴权 / 错误 / 限流）。
按资源建章；每接口一页，源码模式粘贴端点骨架，改方法、路径、参数表与 curl/fetch。
类名表与禁止事项见站内 SPEC（或本书附件 /链到内部文档）。

必须遵守
一端点一页；试调只保留 curl / fetch 文本。
不要嵌入 Scalar、Swagger、Redoc 或自建试调表单。
方法徽章用 cb-api-method--get|post|put|patch|delete，不要用系统 /api/docs 的彩色 api-method 。

以后
OAS → 书页自动生成会对齐同一 HTML；编辑器「插入 API 块」为后续增强。当前以粘贴骨架为准。

参考
端点示例页：本章「获取小部件」
现网甲案 demo（可选）： /books/cocobook-api-jia-pre/page/listusers-get-users
平台 API：/api/docs（标注： BookStack 平台，非本模板）

插入端点 ：详见 插入端点块 （含完整可复制骨架）。

Insert endpoint block

插入端点块
P0 写法：在编辑器里用 源码 / HTML模式粘贴约定骨架，再改文案。编辑器「插入 API 块」按钮是后续增强，当前不依赖。

步骤
1. 新建页（建议页名： 动词 · 路径，一端点一页）。
2. 编辑器切到 源码 （或 HTML）模式。
3. 清空默认空段落，粘贴下方「完整骨架」。
4. 改：方法徽章类名、路径、 operationId、参数表、 curl / fetch、响应 JSON。
5. 保存后在阅读壳预览；对照示例页 获取小部件 。

方法徽章类名

方法 类名

GET cb-api-method cb-api-method--get

POST cb-api-method cb-api-method--post

PUT cb-api-method cb-api-method--put

PATCH cb-api-method cb-api-method--patch

DELETE cb-api-method cb-api-method--delete

必填块顺序
1. 根包裹 div.cb-api（一页一个）
2. h1 操作名
3..cb-api-endpoint（方法徽章 + path）
4..cb-api-meta（可选但推荐）

5. 简述 `p`
6. 参数表 `table.cb-api-table`（有则写）
7. 请求示例：至少 curl，建议再给 fetch（可用 `details.cb-api-fold` 折叠）
8. 响应：状态码芯片 + JSON 示例

禁止

- 不要嵌 Scalar / Swagger / Redoc / 试调表单
- 不要一页堆多个端点
- 不要用系统 `/api/docs` 的 `api-method` 色块
- 不要把产品接口写进 /api/docs

更完整的类名表见本书 如何复制使用_____；概述骨架见 开始使用_____。

完整骨架（复制用）

下面是与「获取小部件」同构的 HTML。粘贴后把示例字段换成你的接口即可。

HTML 源码

```
<!-- 酷可 API 端点页骨架 · 粘贴到 BookStack 页 HTML/源码模式 -->
<!-- 一端点一页 · 甲·浅底方章 · 仅 curl/fetch 静态示例 · 勿嵌试调器 -->
<!-- PRE 20261102apiFoldPre: Header open fold · 请求示例 collapsed fold -->
<div class="cb-api">
  <h1>获取小部件</h1>

  <div class="cb-api-endpoint">
    <span class="cb-api-method cb-api-method--get">GET</span>
    <code class="cb-api-path">/v1/widgets/<var>id</var></code>
  </div>

  <p class="cb-api-meta">operationId: <code>getWidget</code> · 分组: Widgets · 鉴权: Bearer
Token</p>

  <p>按 ID 返回单个小部件。路径参数 <code class="cb-api-param">id</code> 为资源主键；响应体字段以本文示例为准（可替换为你的 OAS 示例）。
</p>

  <h2>Path 参数</h2>
  <table class="cb-api-table">
```

```
    <thead>
      <tr><th>□□</th><th>□□</th><th>□□</th><th>□□</th><th>□□</th></tr>
    </thead>
    <tbody>
      <tr>
        <td><code>id</code></td>
        <td>path</td>
        <td><span class="cb-api-type">string</span></td>
        <td><span class="cb-api-required">□□</span></td>
        <td>□□□ ID</td>
      </tr>
    </tbody>
  </table>

  <details class="cb-api-fold" open>
    <summary>Header □□</summary>
    <table class="cb-api-table">
      <thead>
        <tr><th>□□</th><th>□□</th><th>□□</th><th>□□</th><th>□□</th></tr>
      </thead>
      <tbody>
        <tr>
          <td><code>Authorization</code></td>
          <td>header</td>
          <td><span class="cb-api-type">string</span></td>
          <td><span class="cb-api-required">□□</span></td>
          <td><code>Bearer &amp;lt;token&amp;gt;</code></td>
        </tr>
      </tbody>
    </table>
  </details>

  <details class="cb-api-fold">
    <summary>□□□□</summary>
    <div class="cb-api-example">
      <span class="cb-api-example__label">curl□□□□□□□□□□□□□□□□</span>
      <pre><code>curl -sS 'https://api.example.com/v1/widgets/wdg_42'   -H 'Authorization:
Bearer YOUR_TOKEN'   -H 'Accept: application/json'</code></pre>
    </div>
    <div class="cb-api-example">
      <span class="cb-api-example__label">fetch</span>
```

```
      <pre><code>fetch('https://api.example.com/v1/widgets/wdg_42', {
  headers: {
    'Authorization': 'Bearer YOUR_TOKEN',
    'Accept': 'application/json'
  }
}).then(r =&gt; r.json())</code></pre>
    </div>
  </details>

  <h2>□□</h2>
  <h3><span class="cb-api-status">200</span> □□</h3>
  <div class="cb-api-example">
    <span class="cb-api-example__label">application/json</span>
    <pre><code>{
  "id": "wdg_42",
  "name": "Demo Widget",
  "status": "active",
  "updated_at": "2026-11-02T08:00:00Z"
}</code></pre>
  </div>

  <h3><span class="cb-api-status cb-api-status--err">404</span> □□□</h3>
  <p>ID □□□□□□□□□□□□□□</p>
  <div class="cb-api-example">
    <pre><code>{
  "error": {
    "code": 404,
    "message": "Widget not found"
  }
}</code></pre>
  </div>
</div>
```

Getting Started

开始使用

本文档说明如何调用本产品的 HTTP API。系统 BookStack 平台 API 请另见 /api/docs（与本书无关）。

Base URL

所有路径均相对于下方根地址（请改成你的环境）。

生产

```
https://api.example.com
```

沙箱（可选）

```
https://api.sandbox.example.com
```

鉴权

在请求头携带访问令牌（按你的实际方案改文案）：

Header

```
Authorization: Bearer <token>
```

curl

```
curl -sS 'https://api.example.com/v1/widgets/wdg_42' \
  -H 'Authorization: Bearer YOUR_TOKEN'
```

请求格式

写操作请设置 `Content-Type: application/json`。各端点接受的字段见对应页的参数表。

错误处理

成功一般为 200 / 201 / 204。失败返回 4xx/5xx，正文建议统一为：

```
{
  "error": {
    "code": 401,
    "message": "Unauthorized"
  }
}
```

HTTP	含义	说明
400	参数错误	校验失败；见 message
401	未鉴权	缺少或无效 Token
403	无权限	Token 有效但无权访问资源
404	未找到	资源不存在
429	限流	见下方 Rate Limits

限流

默认示例：每用户每分钟 180 次（按你的网关改）。触发时返回 429。

建议客户端行为

```
# 尊重 Retry-After（若下发）
# 退避重试；避免突发打满配额
```

本书约定

- 一端点一页；章 = 资源。
- 页内使用 `.cb-api-*` 类；方法徽章为甲 ·浅底方章。

- □□□□□ curl / fetch □□□□□□□□□□□□

Widgets

□□□□□ · Widgets

Widgets

List widgets

□□□□

GET `/v1/widgets`

operationId: `listWidgets` · □□ : Widgets · □□ : Bearer Token

□□□□□□□□ □□□□□ stub□□□□□□□

□□□□

curl

```
curl -sS -X GET 'https://api.example.com/v1/widgets' \
  -H 'Authorization: Bearer YOUR_TOKEN' \
  -H 'Accept: application/json'
```

□□

200

application/json

```
{
  "id": "wdg_42",
  "name": "Demo Widget"
}
```

Widgets

Get widget

□□□□

GET `/v1/widgets/id`

operationId: `getWidget` · □□ : Widgets · □□ : Bearer Token

□ ID □□□□□□□□□□□ `id` □□□□□□□□□□□□□□□□□□□□□□□□□□□ OAS □□□□

Path □□

□□	□□	□□	□□	□□
`id`	path	string	□□	□□□ ID

Header □□

□□	□□	□□	□□	□□
`Authorization`	header	string	□□	`Bearer <token>`

□□□□

curl□□□□□□□□□□□□□□□

```
curl -sS 'https://api.example.com/v1/widgets/wdg_42' \
  -H 'Authorization: Bearer YOUR_TOKEN' \
  -H 'Accept: application/json'
```

fetch

```
fetch('https://api.example.com/v1/widgets/wdg_42', {
  headers: {
```

```
    'Authorization': 'Bearer YOUR_TOKEN',
    'Accept': 'application/json'
  }
}).then(r => r.json())
```

□□

200 □□

application/json

```
{
  "id": "wdg_42",
  "name": "Demo Widget",
  "status": "active",
  "updated_at": "2026-11-02T08:00:00Z"
}
```

404 □□□

ID □□□□□□□□□□□□□

```
{
  "error": {
    "code": 404,
    "message": "Widget not found"
  }
}
```

Widgets

Create widget

□□□□

POST `/v1/widgets`

operationId: `createWidget` · □□ : Widgets · □□ : Bearer Token

□□□□□□□□□□ JSON□ □□□□□ stub□□□□□□□

Body □□

□□	□□	□□	□□	□□
`name`	body	string	□□	□□□

□□□□

curl

```
curl -sS -X POST 'https://api.example.com/v1/widgets' \
  -H 'Authorization: Bearer YOUR_TOKEN' \
  -H 'Accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{"name":"Demo"}'
```

□□

201

application/json

```
{
  "id": "wdg_42",
  "name": "Demo Widget"
}
```

Widgets

Delete widget

□□□□

DELETE `/v1/widgets/id`

operationId: `deleteWidget` · □□ : Widgets · □□ : Bearer Token

□ ID □□□□□□ □□□□□ stub□□□□□□□

Path □□

□□	□□	□□	□□	□□
`id`	path	string	□□	□□□ ID

□□□□

curl

```
curl -sS -X DELETE 'https://api.example.com/v1/widgets/wdg_42' \
  -H 'Authorization: Bearer YOUR_TOKEN' \
  -H 'Accept: application/json'
```

□□

204

application/json

```
(□□□)
```