

 NIMBUS CLOUD STORAGE

API Reference

Version 2.3 • REST / JSON over HTTPS

 **Base URL** api.nimbus.dev/v2

 **Format** application/json

 **Auth** API key or OAuth 2.0 bearer token

 **Rate limit** 1,000 req/hour (standard tier)

1 Overview

The Nimbus API is a resource-oriented REST interface for managing files, folders, and event webhooks inside a Nimbus workspace. All requests and responses use JSON, all endpoints require HTTPS, and all timestamps are ISO 8601 UTC. Path-based versioning (/v2); /v1 is deprecated and sunsets 2026-12-31.

- Resource IDs (file_id, folder_id, ...) are opaque, URL-safe strings prefixed by type, e.g. fl_8a2c... for files.
- List endpoints are paginated with cursor and limit; see Section 7.
- Fields omitted from a PATCH body are left unchanged; send null explicitly to clear a field.

2 Authentication

Server-to-server integrations should use API keys; user-facing apps should use OAuth 2.0 bearer tokens. Both are sent via the **Authorization** header.

API KEY

Authorization: ApiKey
nb_live_7f3c9a2e1d4b8f0a

OAUTH 2.0 BEARER

Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5c...

 API keys are generated per-workspace from **Settings** → **Developer**. OAuth access tokens expire after 3,600 seconds; use the refresh_token grant to renew without re-prompting the user.

3 Rate Limiting & Errors

Every response carries rate-limit headers, enforced per credential on a rolling one-hour window. Exceeding the limit returns 429 with a **Retry-After** header.

Header	Example	Meaning
X-RateLimit-Limit	1000	Requests allowed in the current window
X-RateLimit-Remaining	842	Requests left before throttling
X-RateLimit-Reset	1750939200	Unix timestamp the window resets

Errors share one JSON shape across every endpoint:

ERROR — 404 NOT FOUND

```
{
  "error": {
    "code": "resource_not_found",
    "message": "No file 'fl_8a2c9f1e'.",
    "request_id": "req_4d1f7b2a"
  }
}
```

Status	Error code
400	invalid_request
401	unauthenticated
403	insufficient_scope
404	resource_not_found
409	conflict
422	validation_failed
429	rate_limited
500	internal_error

4 Files

Endpoints for uploading, retrieving, updating, and removing files.

4.1 List & retrieve files

GET /v2/files **GET** /v2/files/{file_id}

List files visible to the caller (most recently modified first), or fetch one file's full metadata, including `download_url` and `checksum_sha256`.

Query parameters (list only)

Name	Type	Required	Description
folder_id	string	No	Restrict results to one folder
limit	integer	No	Page size, 1–100, default 20
cursor	string	No	Opaque pagination cursor from a prior response

RESPONSE — 200 OK (single file object)

```
{
  "id": "f1_8a2c9f1e6d4b",
  "name": "q3-roadmap.pdf",
  "size_bytes": 482913,
  "mime_type": "application/pdf",
  "folder_id": "fd_2b7e1c9a",
  "modified_at": "2026-06-18T15:41:22Z"
}
```

4.2 Upload, update, and delete

POST /v2/files

Creates a file record and returns a pre-signed upload URL. Send the raw bytes to `upload_url` within 15 minutes.

REQUEST BODY

```
{
  "name": "q3-roadmap.pdf",
  "folder_id": "fd_2b7e1c9a",
  "mime_type": "application/pdf",
  "size_bytes": 482913
}
```

RESPONSE — 201 CREATED

```
{
  "id": "f1_8a2c9f1e6d4b",
  "upload_url": "https://upload...",
  "expires_at": "2026-06-18T16:00Z"
}
```

PATCH /v2/files/{file_id} **DELETE** /v2/files/{file_id}

PATCH renames or moves a file (`{"name": ..., "folder_id": ...}`), returning 200 with the updated object. **DELETE** moves the file to the trash (purged after 30 days); returns 204 **No Content**. Pass `?permanent=true` to skip the trash (requires `files:hard_delete`).

5 Folders

5.1 Create & list contents

POST /v2/folders **GET** /v2/folders/{id}/children

Create a folder, optionally nested under a parent, or list its immediate child files and subfolders (accepts the same `limit/cursor` parameters as Section 4.1, plus `type=file|folder` to filter).

REQUEST BODY

```
{
  "name": "Client Deliverables",
  "parent_id": null
}
```

RESPONSE — 201 CREATED

```
{
  "id": "fd_2b7e1c9a",
  "name": "Client Deliverables",
  "parent_id": null
}
```

6 Webhooks

Subscribe to workspace events instead of polling. Nimbus signs every delivery with an HMAC-SHA256 signature in the `X-Nimbus-Signature` header.

POST `/v2/webhooks` **DELETE** `/v2/webhooks/{webhook_id}`

Register an HTTPS endpoint and event list, or unregister one (204 No Content).

REQUEST BODY

```
{
  "url": "https://hooks.example-crm.io/nimbus",
  "events": ["file.created", "file.deleted"],
  "secret": "whsec_3f9a7c2e"
}
```

DELIVERY — file.created

```
{
  "event": "file.created",
  "occurred_at": "2026-06-18T15:41Z",
  "data": { "id": "f1_8a2c9f1e6d4b" }
}
```

Event	Fires when...
<code>file.created</code>	A new file finishes uploading
<code>file.deleted</code>	A file is trashed or permanently deleted
<code>folder.shared</code>	A folder's sharing settings change
<code>quota.exceeded</code>	The workspace storage quota is reached

7 Pagination

List endpoints use cursor-based pagination. Request the first page without a `cursor`; if `next_cursor` is present in the response, pass it back as the `cursor` query parameter to fetch the next page. A response with no `next_cursor` field marks the end of the list.

REQUEST

```
GET /v2/files?folder_id=fd_2b7e1c9a&limit=20&cursor=eyJvZmZzZXQ1QjIwfQ==
```

8 Changelog

Version	Date	Changes
v2.3	2026-06-01	Added <code>quota.exceeded</code> webhook event; <code>folder_id</code> filter on list files
v2.2	2026-03-14	Introduced cursor-based pagination, replacing offset pagination
v2.1	2025-12-02	Added OAuth 2.0 authorization-code flow alongside API keys
v2.0	2025-09-19	Initial /v2 release; /v1 entered deprecation

 Full changelog, SDKs (`nimbus-python`, `nimbus-node`), and a Postman collection are available at api.nimbus.dev/docs.