ArtLair API

Everything on ArtLair that can be read from code: posts, original files, tags, marketplace items, forum threads and public profiles. JSON over HTTPS, one personal key.

This is the complete guide in one file: read it offline, or hand it to a coding assistant and ask it to write the client.

Base URL: https://artlair.eu/api/v1 | Live page: https://artlair.eu/pages/api | Generated 2026-11-19 05:22

```
Base URL:  https://artlair.eu/api/v1
Auth:  X-API-Key header
Format:  JSON, read-only GET
Free limits:  300/hour per key
```

> No key yet? It takes ten seconds and costs nothing: Account Settings -> API -> Generate key. This page assumes you exported it as API_KEY.

Start here: three steps

1. Create your key

Open Account Settings -> API and press
Generate key. This page's key is separate from the one the E1547 app
uses, so generating one never breaks the other.

The key looks like al_live_... and is shown once.
Put it in an environment variable, not in code you publish.

2. Make one request

This asks for the five newest posts. In a shell:

CURL

```
curl -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/posts?limit=5"
```

3. Read the answer

Every response is one JSON object: ok, then data, plus meta on lists.
Never anything else - no envelope to unwrap twice.

RESPONSE


```
{
"ok": true,
"data": [
{
"id": 42,
"title": "Sunset over the lair",
"category": "Digital",
"tags": ["sunset", "landscape"],
"media_type": "image",
"rating": "safe",
"is_ai": false,
"likes": 128,
"views": 940,
"created_at": "2026-09-20 14:32:10",
"url": "https://artlair.eu/media/image?id=42",
"thumb": "https://artlair.eu/media/thumb?f=...&w=400",
"file": {
"url": "https://artlair.eu/media/file?id=42",
"md5": "9f2c1d6b8e4a3f5079c1d2e3b4a59687",
```

```
"size": 1843200,
"ext": "jpg",
"media_type": "image"
},
"author": {
"id": 7,
"username": "nullspark",
"profile_url": "https://artlair.eu/account/profile?u=nullspark"
}
}
],
"meta": { "page": 1, "limit": 5, "total": 412, "total_pages": 83 }
}
```

When something goes wrong you still get one object, with a machine-readable code:

ERROR RESPONSE

```
{
"ok": false,
"error": {
"code": 401,
"message": "Missing or invalid API key."
}
}
```

Your first script

Same request, four ways. Each one prints the title and file hash of the five newest posts, so you can see the shape of a real answer.

Python

PYTHON 3 (NEEDS: PIP INSTALL REQUESTS)

```
import os
import requests

SITE = "https://artlair.eu"
KEY = os.environ["API_KEY"] # export API_KEY=al_live_...

r = requests.get(
f"{SITE}/api/v1/posts",
params={"limit": 5, "sort": "new"},
headers={"X-API-Key": KEY, "Accept": "application/json"},
timeout=20,
)

print("budget left this hour:", r.headers.get("X-RateLimit-Remaining"))

payload = r.json()
if not payload.get("ok"):
raise SystemExit(payload["error"]["message"])

for post in payload["data"]:
print(post["id"], post["title"], post["file"]["md5"])
```

JavaScript (Node 18+, works in a browser too)

JAVASCRIPT (FETCH IS BUILT IN)

```
const SITE = "https://artlair.eu";
const KEY = process.env.API_KEY; // in a browser: paste the key, or proxy through your server

const res = await fetch(`${SITE}/api/v1/posts?limit=5&sort=new`, {
headers: { "X-API-Key": KEY, "Accept": "application/json" },
});

const payload = await res.json();
if (!payload.ok) throw new Error(payload.error.message);

for (const post of payload.data) {
console.log(post.id, post.title, post.file.url);
}
```

PHP

PHP 8

```
<?php
$site = "https://artlair.eu";
$key = getenv("API_KEY");

$ch = curl_init("$site/api/v1/posts?limit=5");
curl_setopt_array($ch, [
CURLOPT_RETURNTRANSFER => true,
CURLOPT_HTTPHEADER => ["X-API-Key: $key", "Accept: application/json"],
CURLOPT_TIMEOUT => 20,
]);

$payload = json_decode((string)curl_exec($ch), true);
curl_close($ch);

if (empty($payload["ok"])) {
exit("API error: " . ($payload["error"]["message"] ?? "unknown"));
}

foreach ($payload["data"] as $post) {
echo $post["id"], " ", $post["title"], " ", $post["file"]["url"], PHP_EOL;
}
```

Command line, for a quick look

CURL + JQ

```
# titles of the five newest posts
curl -s -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/posts?limit=5" | jq -r '.data[].title'

# just the header, to read your remaining budget
curl -s -D - -o /dev/null -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/posts?limit=1" | grep -i
ratelimit
```

Authentication

Every request needs your personal key. Send it in the X-API-Key header - that is the form to build against. Two URL shortcuts exist for quick tests and for tools that cannot set headers.

How	Example	Use it when
Header	X-API-Key: al_live_...	Always, for real clients. Keys in headers never land in browser history or server logs.
Query	?api_key=al_live_...	A one-off test in a browser tab or from a tool that cannot set headers.
Path	/api/v1/al_live_.../posts/api/v1/posts/al_live_...	The same, when appending a query string is awkward. The key is accepted at the start or the end of the path.

ALL THREE FORMS

```
curl -H "X-API-Key: al_live_your_key_here" "https://artlair.eu/api/v1/posts?limit=5"
curl "https://artlair.eu/api/v1/posts?limit=5&api_key=al_live_your_key_here"
curl "https://artlair.eu/api/v1/al_live_your_key_here/posts?limit=5"
```

A key belongs to one account. Reads are public data only; the key identifies you so that limits, favourites and blocks stay yours. If a key leaks, revoke it in Settings and generate a new one - nothing else changes.

Response format

Field	What it holds
ok	true or false. Check this before reading anything else.
data	An object for one item, an array for a list. Present when ok is true.
meta	Only on lists: page, limit, total, total_pages.
error	Only when ok is false: code (same as the HTTP status) and a readable message.

Numbers arrive as numbers ("likes": 128, not "128") and booleans as real booleans, so you rarely need to cast anything. Timestamps are site time, YYYY-MM-DD HH:MM:SS. Relative URLs are always absolute and start with https://artlair.eu, so they work from any device.

Common jobs, solved

The requests people actually start with. Every one is copy-paste ready.

Newest posts, most liked, most viewed

CURL

```
curl -s -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/posts?sort=new&limit=20"
curl -s -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/posts?sort=popular&limit=20"
curl -s -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/posts?sort=views&limit=20"
```

Filter by tag, category, user, or search words

CURL

```
curl -s -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/posts?tag=dragon"
curl -s -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/posts?category=Digital&limit=10"
curl -s -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/posts?user=nullspark"
curl -s -H "X-API-Key: $API_KEY" --get --data-urlencode "q=armour study"
"https://artlair.eu/api/v1/posts"
```

One post, with its description

CURL

```
curl -s -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/posts/42" | jq '.data | {title, description,
tags, file}'
```

Download an original file (no key needed)

CURL

```
# straight to disk, keeping the site's filename
curl -OJ "https://artlair.eu/media/file?id=42"

# or name it after the content hash, which is how duplicate-safe libraries name files
curl -s -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/posts/42" \
| jq -r '.data.file | "\(.md5).\(.ext)\t\(.url)"' \
| while IFS=$'\t' read -r name url; do curl -s -o "$name" "$url"; done
```

Everything by one tag, downloading as you page

PYTHON

```
import os, time, pathlib, requests

SITE = "https://artlair.eu"
KEY = os.environ["API_KEY"]
OUT = pathlib.Path("./downloads"); OUT.mkdir(exist_ok=True)

page = 1
while True:
r = requests.get(f"{SITE}/api/v1/posts",
params={"tag": "dragon", "page": page, "limit": 50},
headers={"X-API-Key": KEY}, timeout=30)
payload = r.json()
if not payload.get("ok"):
raise SystemExit(payload["error"]["message"])

for post in payload["data"]:
f = post["file"]
name = f"{f['md5'] or post['id']}.{f['ext']}"
target = OUT / name
if target.exists():
continue
data = requests.get(f["url"], timeout=60).content
target.write_bytes(data)
print(name, len(data), "bytes")

if page >= payload["meta"]["total_pages"]:
break

page += 1
time.sleep(0.3) # stay polite; 300 requests/hour is the budget
```

Follow the budget instead of being blocked by it

PYTHON

```
r = requests.get(f"{SITE}/api/v1/posts", headers={"X-API-Key": KEY}, timeout=20)

left = int(r.headers.get("X-RateLimit-Remaining", 100))
reset = int(r.headers.get("X-RateLimit-Reset", 0))

if r.status_code == 429:
wait = max(1, reset - int(time.time()))
print(f"out of budget, sleeping {wait}s")
time.sleep(wait)
elif left < 10:
time.sleep(2) # slow down before the limit, not after
```

Marketplace, forum and profiles

CURL

```
curl -s -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/marketplace?sort=price&limit=10"
curl -s -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/marketplace/12"
curl -s -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/forum/boards"
curl -s -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/forum/threads?board_id=3&limit=10"
curl -s -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/users/nullspark"
```

Reference: posts

GET `https://artlair.eu/api/v1/posts`

Approved posts, newest first. Combines with every parameter in the table, so one call can do "dragon art by this user from this year".

Parameter	Default	What it does
page	1	Page number, starting at 1. Read meta.total_pages to know where it ends.
limit	20	Items per page, 1-50. Use 50 for bulk work, 20-30 for a feed.
sort	new	new, popular (most likes) or views (most views).
category	-	Exact category name, e.g. Digital.
tag	-	One tag, e.g. dragon. Use q to search several words.
user	-	Only posts by that username.
q	-	Free-text search across title and tags.

REQUEST

```
curl -s -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/posts?tag=dragon&sort=popular&limit=2"
```

What each post field means

Field	Use
id	Post ID. Everything else (file, page, comments) is addressed by it.
title / description	Description is only in the single-post response, to keep lists small.
tags	Array of strings, already split for you.
rating	safe, questionable or explicit. Filter a feed with it before showing anything.
media_type	image or video (GIFs count as images).
likes / views	Numbers. Handy for sorting client-side within a page.
thumb	400px preview. Cheap enough for a grid - use it, not file.url, in lists.
file	The original: url, md5, size, ext, media_type. See files.
author	id, username, profile_url.
url	The site page for the post, for a "view on site" button.

GET `https://artlair.eu/api/v1/posts/{id}`

One post with its description. Answers 404 when the post is gone or not public, never a half-filled object.

REQUEST

```
curl -s -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/posts/42"
```

Files and downloads

GET **https://artlair.eu/media/file?id={id}**

The original file, byte for byte: images, GIFs and videos alike. This is what file.url points at and the only file URL worth storing - the storage behind it can change without breaking your code. No API key is needed for approved posts, so a downloader can fetch files directly, and there is no per-file limit. Range requests and HEAD work on locally stored files; files mirrored elsewhere redirect once to their source.

REQUEST

```
curl -O -J "https://artlair.eu/media/file?id=42" # keep the site's filename
curl -I "https://artlair.eu/media/file?id=42" # size and type without the bytes
```

Field	Meaning
file.url	Where the original lives, for good. Store this, not a preview or storage link.
file.md5	Lowercase MD5 of the original - the same digest e621 publishes. Name a download after it to de-duplicate across libraries. null until the file has been hashed once.
file.size	Exact bytes, so you can preallocate or show progress.
file.ext	Extension without the dot. GIFs stay gif; videos report mp4, webm or mov.
file.media_type	image or video.

thumb is the other URL worth knowing: a small preview that is cheap to render in a grid. A list screen only needs thumb; fetch file.url when someone actually opens or saves a post.

Reference: marketplace

GET **https://artlair.eu/api/v1/marketplace**

Active marketplace items, with prices in tokens.

Parameter	Default	What it does
page / limit	1 / 20	Pagination, exactly as on posts.
sort	new	new, popular (most purchases) or price (cheapest first).
category	-	Exact category name.
q	-	Free-text search across title and tags.

REQUEST

```
curl -s -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/marketplace?sort=price&limit=10"
```

RESPONSE

```
{
"ok": true,
"data": [
{
"id": 12,
"title": "Custom dragon banner",
"category": "Banners",
"tags": ["dragon", "banner"],
"price_tokens": 500,
"purchase_count": 3,
"created_at": "2026-09-18 10:05:44",
"preview_url": "https://artlair.eu/uploads/marketplace/previews/...",
```

```
"url": "https://artlair.eu/marketplace_item?id=12",
"seller": "nullspark"
}
],
"meta": { "page": 1, "limit": 10, "total": 27, "total_pages": 3 }
}
```


GET https://artlair.eu/api/v1/marketplace/{id}

One item with its full description, seller and current owner.

REQUEST

```
curl -s -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/marketplace/12"
```

Reference: forum

GET https://artlair.eu/api/v1/forum/boards

Boards with their categories and thread counts. Start here: the id values feed board_id below.

REQUEST

```
curl -s -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/forum/boards"
```

GET https://artlair.eu/api/v1/forum/threads

Threads, pinned first, with reply counts and last activity. Leave out board_id to get every board.

Parameter	Default	What it does
board_id	all	Only threads in this board.
page / limit	1 / 20	Pagination, as on posts.

REQUEST

```
curl -s -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/forum/threads?board_id=3&limit=10"
```

RESPONSE


```
{
"ok": true,
"data": [
{
"id": 96,
"title": "Welcome to the forum!",
"board_id": 3,
"author": "admin",
"is_pinned": true,
"is_locked": false,
"reply_count": 14,
"views": 302,
"created_at": "2026-08-01 09:00:00",
"last_post_at": "2026-09-25 18:44:02",
"url": "https://artlair.eu/forum/thread?id=96"
}
],
"meta": { "page": 1, "limit": 10, "total": 57, "total_pages": 6 }
}
```


Reference: users

GET https://artlair.eu/api/v1/users/{username}

Public profile and stats. Never an email address, never a private setting.

REQUEST

```
curl -s -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/users/nullspark"
```

RESPONSE


```
{
"ok": true,
"data": {
"id": 7,
"username": "nullspark",
"avatar_url": "https://artlair.eu/uploads/avatars/...",
"joined": "2026-05-14 12:00:00",
"is_banned": false,
"stats": { "posts": 18, "likes": 512, "followers": 27, "following": 9 },
"url": "https://artlair.eu/account/profile?u=nullspark"
}
}
```


To list that person's posts, call https://artlair.eu/api/v1/posts?user=nullspark - profiles and their galleries are two calls by design.

Rate limits

Three budgets apply at once, and every response tells you where you stand before you hit one: 120 requests/minute per IP address, 300/hour per key, 2,000/day per key. The hourly budget is the one normal clients touch.

Header	Meaning
X-RateLimit-Limit	Requests allowed in the current hour.
X-RateLimit-Remaining	Left right now. This is the smallest of the three budgets, so it is the number to watch.
X-RateLimit-Reset	Unix seconds when the hourly window frees up.
X-RateLimit-Window	Length of that window in seconds (3600).
X-RateLimit-Day-Limit	Requests allowed in the current day.
X-RateLimit-Day-Remaining	Left today.
X-RateLimit-Day-Reset	Unix seconds when the daily window resets.
X-RateLimit-Ip-Limit	Requests allowed per minute from your IP address.
X-RateLimit-Ip-Remaining	Left this minute. Shared connections (offices, VPNs, mobile carriers) share this one.

READ YOUR BUDGET IN ANY LANGUAGE

```
curl -s -D - -o /dev/null -H "X-API-Key: $API_KEY" "https://artlair.eu/api/v1/posts?limit=1" | grep -i ratelimit
```

Errors

Status	What happened	Fix
400	The request could not be read (bad parameter or body).	Check the parameter names in this page.
401	Missing or invalid API key.	Send X-API-Key, or generate a fresh key in Settings.
403	Key blocked, or the account is suspended.	Contact support; a new key from the same account will not help.
404	Unknown endpoint, or the post/item is not public.	Check the path; unapproved posts are invisible to the API by design.
405	Wrong HTTP method.	This API is read-only: use GET.

429	Rate limit exceeded.	Wait for Retry-After, then resume more slowly.
500	Something broke on our side.	Retry once after a pause; if it repeats, tell us which call.

A good client checks ok, not only the status code, and shows error.message to a human instead of guessing.

Details worth knowing

Topic	Details
Pagination	Every list takes page and limit (max 50) and returns meta.total and meta.total_pages. Newest-first lists shift as new posts arrive, so for a stable crawl work from the highest page down, or de-duplicate by id.
CORS	Access-Control-Allow-Origin: *, so browser code can call the API directly. That still puts your key in the page - proxy through your server for anything public.
Caching	Responses may be cached for up to 60 seconds. Errors never are. No need to poll faster than that.
Read-only	This API answers GET. Uploads, likes and comments happen on the site, or through the e621-compatible API used by the E1547 app.
Fallback URL	If clean URLs misbehave on a weird proxy: /api/v1/index.php?endpoint=posts&id=42.
Discovery	GET https://artlair.eu/api/v1/ returns the endpoint list as JSON, so a client can check what this instance supports instead of hard-coding.

The other API: E1547 app

This site also speaks the e621 API, which is what the E1547 app expects.
It is a different surface: e621-style paths, HTTP Basic auth with your site username and a separate app key, and it can post comments and file reports as well as read. If you are building a client for desktop, use the JSON API above; if you want ArtLair inside the E1547 app, use this.

E621-STYLE REQUESTS (BASIC AUTH: USERNAME + APP KEY)

```
# any user's posts, newest first
curl -u "your_username:your_app_key" "https://artlair.eu/posts.json?tags=order:new&limit=20"

# one post
curl -u "your_username:your_app_key" "https://artlair.eu/posts/42.json"

# profiles, tags, favourites, forum, reports
curl -u "your_username:your_app_key" "https://artlair.eu/users/nullspark.json"
curl -u "your_username:your_app_key" "https://artlair.eu/tags.json?search=drag&limit=10"
curl -u "your_username:your_app_key" "https://artlair.eu/favorites.json"
curl -u "your_username:your_app_key" "https://artlair.eu/forum_topics.json"
```

Generate that app key under Account Settings -> E1547,
which also shows the exact instance URL and sign-in details to paste into the app. Responses carry the same X-RateLimit-* headers, and posts include file.md5 with the same meaning as above.

Need an endpoint that does not exist yet, or found a bug? Use the
contact page or email support@artlair.eu.
Tell us the URL you called and we can usually answer the same day.