

API

Quickstart

🕒 3 min read · Updated 27 September 2026 · Verified against [c761e51](#) on 2026-09-27

Your first calls to the API in five minutes: read an Ang, search, verify a quotation, open a whole composition, cite correctly, and handle errors and caching.

<https://docs.gurbanisoul.com/api/quickstart/>

The API is plain HTTPS and JSON. There is no key, no account and no SDK. Every example below works from a terminal, a server or a script. The base URL is:

```
https://gurbanisoul.com/api/v1
```

Use `/api/v1` for new code. It returns the same bodies as the legacy `/api`, with structured errors (**versioning and caching**). The API does not send CORS headers, so a web page on another origin cannot call it from the browser; call it from your server (**using the API**).

1. Read an Ang

```
curl -s https://gurbanisoul.com/api/v1/ang/1 | jq '{ang, lines: (.lines | length), first: .lines[0] | {id, gurmukhi, translit}}'
```

The response holds the Ang number and its lines, each with the verbatim gurmukhi, a translit reading aid, the structure (raag, section, author, comp_id, line_no, is_header, is_rahao) and, where one exists, the English translation as en. The fields are described on **anatomy of a line record**. An Ang outside 1–1430 is clamped to the nearest valid Ang, not rejected.

2. Search

```
curl -s 'https://gurbanisoul.com/api/v1/search?q=sochai%20soch&limit=5' | jq '{mode, results: [.results[] | {ang, id, translit}]}'
```

`q` can be Gurmukhi, Roman transliteration (with casual spellings), first letters or English; `mode=auto` (the default) finds the right tier. `mode` in the response names the tier that answered, which is useful when debugging but not a contract. `limit` is at most 200; page with `offset`. See **modes and tiers**.

3. Verify a quotation

```
curl -s 'https://gurbanisoul.com/api/v1/verify?q=sochai%20soch%20na%20hovai' | jq
'{verdict, confidence, ang, matched_line_id}'
```

verdict is one of the verification ladder's outcomes, from an exact match to NOT_FOUND, with the canonical line and its Ang. Add &ang=N when the quotation claims an Ang, and the verdict says whether the claim holds.

4. Open a whole composition

Every line carries comp_id, the composition it belongs to, heading included:

```
curl -s https://gurbanisoul.com/api/v1/shabad/2 | jq '{comp_id, lines: (.lines |
length), first_ang: .lines[0].ang}'
```

Some comp_id values are permanent gaps and return 404 (the comp_id rule). Take ids from a line, never by counting.

5. Cite what you show

Show the Gurmukhi exactly as returned, with its dandas and markers, and cite it as **Sri Guru Granth Sahib Ji • Ang N**, using the line's own ang. Label a transliteration as a transliteration and an English line as a translation (Dr. Sant Singh Khalsa's), never as the original. The full rules are in the Answer Protocol for engineers.

6. Handle errors and cache

```
import json, urllib.error, urllib.request

def get(path, etag=None):
    req = urllib.request.Request("https://gurbanisoul.com/api/v1" + path,
                                headers={"If-None-Match": etag} if etag else {})

    try:
        with urllib.request.urlopen(req, timeout=30) as r:
            return r.status, json.loads(r.read()), r.headers.get("ETag")
    except urllib.error.HTTPError as e:
        if e.code == 304:
            return 304, None, etag
        err = json.loads(e.read())["error"]
        raise RuntimeError(f"{e.code} {err['code']}: {err['message']} (request {err['request_id']})")

status, ang, etag = get("/ang/712")
```

The same in JavaScript (Node 18 or later, or any server runtime with fetch):

```
const res = await fetch('https://gurbanisoul.com/api/v1/ang/712');
if (!res.ok) {
  const { error } = await res.json(); // { code, message, request_id }
  throw new Error(`${res.status} ${error.code}: ${error.message} (request
  ${error.request_id})`);
}
const { ang, lines } = await res.json();
```

- **Branch on the status and code**, not the message. Every case is on the [error reference](#).
- **Cache what may be cached.** Readings (ang, shabad, bani, the analytics, timing) come with an ETag and Cache-Control: public; send If-None-Match and a 304 costs nothing. Search, verify, random, meta and health are never cached.
- **Quote the request id.** Every response has an X-Request-Id header; include it when you report a problem.

Next

- Every route with its parameters: [API routes](#), or the full [API reference](#) with response schemas.
- What to expect of the service, and what it expects of you: [using the API](#).
- What changed and when: [API changelog](#).