

NOSIBLE Get Searches - Data Dictionary V2.1.0

As of: 2026-07-20

This document describes every input and output field of the NOSIBLE Get Searches endpoint so that a technical reader can understand the API at a glance. Get Searches ([POST https://www.nosible.ai/search/v2/get-searches](https://www.nosible.ai/search/v2/get-searches)) retrieves all of the saved searches registered against your API key, including the full search configuration of each.

This is a beta endpoint. Access requires an alpha access code, supplied by NOSIBLE, passed as [internal_use.crawl_feed_alpha](#). Requests without it fail with HTTP 401. Old pipe-format API keys are not supported; get a new key at app.nosible.com.

Interactive API documentation: this endpoint is documented live in the NOSIBLE Swagger UI at <https://www.nosible.ai/search/v2/docs/> (OpenAPI spec: <https://www.nosible.ai/search/v2/openapi.json>). The version of this data dictionary tracks the API version published there: this document describes NOSIBLE V2.1.0.

Contents

Request Headers.....	2
Input Fields.....	2
Output Fields.....	2
Example Request and Response.....	3

Request Headers

Every request to this endpoint is an HTTP POST with a JSON body. Authentication is handled by a single header, documented below. You should also send **Content-Type: application/json**, and **Accept-Encoding: gzip** is recommended to cut transfer sizes.

Header	Details
api-key	Definition: Your NOSIBLE Search API key. To get an API key please visit app.nosible.com , where you can manage organizations, users, subscriptions, and API keys. Type: string Length: 42–50 characters Required: Yes - requests without a valid key fail with HTTP 422. Example: "nos_sk_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"

Input Fields

Get Searches takes only the internal-use block carrying your alpha access code.

Input Field	Details
internal_use	Definition: Internal-only. Must carry your alpha access code as <code>crawl_feed_alpha</code> . Type: object Default: {} Example: <pre>{ "crawl_feed_alpha": "YOUR_ALPHA_ACCESS_CODE" }</pre>

Output Fields

A successful call returns HTTP 200 with the list of your saved searches.

Output Field	Details
message	Definition: A human-readable acknowledgement. Type: string

Output Field	Details
	Example: "Retrieved searches."
response	Definition: The wrapper of the result payload. Type: object Fields: searches (list) - all saved searches for your API key Example: <pre>{ "searches": [{...}, ...] }</pre>
response.searches	Definition: One entry per saved search. Type: array of objects Fields per entry: search_id (str) - the ID of the saved search extra_args (dict/null) - the publishing arguments you saved search_args (dict) - the saved search parameters: instruction, question, expansions, sql_filter, algorithm, n_results, n_probes, n_contextify, min_similarity, must_exclude, must_include, brand_safety, language, continent, region, country, sector, industry_group, industry, sub_industry, iab_tier_1..4, companies, collection, deduplicate Example: <pre>{ "search_id": "b3f5c1e2-8a9d-4f60-9c2b-7d41e8a3f0aa", "extra_args": {}, "search_args": { "question": "Conservation efforts for the Queen Alexandra's Birdwing?", "algorithm": "hybrid-3", "n_results": 100, ... } }</pre>

Example Request and Response

The example below retrieves all saved searches end to end. All field values are illustrative.

Example Request:

```
import requests

response = requests.post(
    url="https://www.nosible.ai/search/v2/get-searches",
    json={
        "internal_use": {
            "crawl_feed_alpha": "YOUR_ALPHA_ACCESS_CODE"
        }
    },
    headers={
        "Accept-Encoding": "gzip",
        "Content-Type": "application/json",
        "api-key": "YOUR_NOSIBLE_API_KEY"
    }
)

print(response.json())
```

Example Response:

```
{
  "message": "Retrieved searches.",
  "response": {
    "searches": [
      {
        "search_id": "b3f5c1e2-8a9d-4f60-9c2b-7d41e8a3f0aa",
        "extra_args": {},
        "search_args": {
          "question": "Conservation efforts for the Queen
                      Alexandra's Birdwing?",
          "algorithm": "hybrid-3",
          "n_results": 100,
          "n_probes": 500,
          "n_contextify": 256,
          "sql_filter": "SELECT loc FROM engine",
          ...
        }
      }
    ]
  }
}
```