Meawfy Advanced Search Guide

Introduction

Meawfy is a powerful, specialized search engine designed exclusively for indexing and retrieving files hosted on Mega.nz, the successor to the original Megaupload platform. Built on robust backend technologies inspired by Elasticsearch and MongoDB, Meawfy provides unparalleled precision in searching through vast repositories of shared files. Whether you're seeking text documents, PDFs, videos, compressed archives, or any other file type, Meawfy's advanced query capabilities ensure that you can always pinpoint the exact resources you need. Our system leverages sophisticated indexing techniques to deliver fast, accurate results, empowering users to navigate Mega.nz's ecosystem with confidence and efficiency.

This technical document outlines Meawfy's search syntax, operators, and strategies. By mastering these tools, you will consistently achieve optimal results, regardless of the complexity of your query. All examples provided are general and illustrative, focusing on everyday, non-copyrighted concepts such as educational tutorials, recipes, or public domain resources.

Basic Search Syntax

Meawfy supports a straightforward keyword-based search as the foundation for all queries. Simply enter terms into the search bar, and Meawfy will return relevant files from Mega.nz based on metadata, file names, and content where applicable.

- **Keyword Search**: Enter one or more words to find matching files. Meawfy uses full-text search capabilities similar to Elasticsearch's match queries, ensuring relevance scoring.
    - Example: Searching for `cooking tutorial` will return files related to general cooking guides or instructional materials.

To refine searches, combine keywords with advanced operators. Meawfy's engine interprets these in a manner akin to MongoDB's query language or Elasticsearch's Query DSL, allowing for boolean logic, phrase matching, and field-specific filtering.

Advanced Operators

Meawfy incorporates a suite of operators to fine-tune your searches. These are designed to handle complex logic, ensuring that even the most specific requirements yield precise results. Our system guarantees that with proper operator usage, you can always isolate the desired files from Mega.nz's extensive archives.

Boolean Operators: AND, OR, NOT

These operators allow logical combinations, mirroring Elasticsearch's bool queries or MongoDB's $and/$or/$not operators.

- **AND** (or +): Requires all terms to be present in the results. Use this to narrow down searches effectively.

    - Example: `recipe AND vegetable` (or `recipe + vegetable`) – Finds files containing both "recipe" and "vegetable," such as general vegetable-based meal plans.
- **OR** (or `|`): Returns results matching any of the terms. Ideal for broadening searches while maintaining relevance.
    - Example: `tutorial OR guide` (or `tutorial | guide`) – Retrieves files that are either tutorials or guides, like instructional documents on various topics.
- **NOT** (or `-`): Excludes specific terms from results. This ensures unwanted matches are filtered out, refining your output.
    - Example: `document NOT image` (or `document - image`) – Locates document files while excluding any image-related results, such as text-based reports without visuals.

Phrase Matching

Use double quotes (`" "`) for exact phrase searches, similar to Elasticsearch's match_phrase query. This is perfect for finding files with precise wording.

- Example: `"gardening tips"` – Returns files where "gardening tips" appears exactly as a phrase, such as comprehensive gardening manuals.

Grouping with Parentheses

Combine operators using parentheses `()` for nested logic, akin to complex MongoDB aggregations.

- Example: `(recipe OR tutorial) AND healthy` – Finds files that are either recipes or tutorials, but must include "healthy," like nutritious meal guides.

Wildcards and Fuzzy Matching

Meawfy supports wildcards for partial matches, drawing from Elasticsearch's wildcard queries.

- `*` (Asterisk): Matches any characters.
    - Example: `cook*` – Matches "cooking," "cookbook," etc., retrieving various cooking-related files.
- `?` (Question Mark): Matches a single character.
    - Example: `te?t` – Matches "text" or "test," useful for finding text files with slight variations.

Fuzzy matching (using `~` after a term) allows for approximate matches, tolerating minor spelling errors.

- Example: `recip~` – Finds "recipe" even if misspelled as "recipie," ensuring you always get close matches for recipe files.

Field-Specific Searches

Target specific file attributes like name, extension, or type, similar to MongoDB's field queries.

- **filetype:**: Filters by file extension or type.
    - Example: `tutorial filetype:pdf` – Searches for PDF files related to tutorials, such as downloadable instructional PDFs.
- **filename:**: Searches within file names.
    - Example: `guide filename:beginner` – Finds files with "beginner" in the name that relate to guides.

Searching by File Type

Meawfy excels at categorizing and retrieving files by type, leveraging metadata from Mega.nz. Use the `filetype:` operator or extensions directly in queries to target specific formats. Our indexing ensures comprehensive coverage, so you can always locate files across categories like documents, videos, or archives.

Text Documents and PDFs

Focus on readable, textual content.

- Example: `document filetype:txt` – Retrieves plain text files containing document-related content, such as general notes or logs.
- Example: `"project plan" filetype:pdf` – Finds PDF files with the exact phrase "project plan," like organizational templates.
- Advanced: `(report OR summary) filetype:pdf -draft` – Locates PDF reports or summaries, excluding any drafts.

Videos

Target multimedia files for instructional or demonstrative purposes.

- Example: `tutorial filetype:mp4` – Returns MP4 video files related to tutorials, such as step-by-step video guides.
- Example: `demonstration OR example filetype:avi` – Finds AVI videos that are demonstrations or examples.
- Advanced: `guide filetype:mp4 +beginner -advanced` – Ensures beginner-level video guides in MP4 format, excluding advanced ones.

Compressed Archives

Search for zipped or archived files, often used for bundled resources.

- Example: `collection filetype:zip` – Retrieves ZIP files containing collections, such as archived sets of general resources.
- Example: `"data set" filetype:rar` – Finds RAR archives with the phrase "data set," like compressed data bundles.
- Advanced: `(archive OR bundle) filetype:zip NOT empty` – Locates non-empty ZIP archives related to archives or bundles.

Other File Types

Meawfy supports a wide range, including images, audio, and more.

- Example: `sound filetype:mp3` – Finds MP3 audio files related to sounds, such as general audio clips.
- Example: `image filetype:jpg -blurry` – Retrieves JPG images while excluding blurry ones.

Best Practices for Optimal Results

- **Combine Operators**: Layer boolean logic with field filters for precision. For instance, `(recipe OR meal) filetype:pdf +easy -complex` always yields simple recipe PDFs.
- **Relevance Scoring**: Meawfy ranks results by relevance, similar to Elasticsearch's scoring. Use `boost` (e.g., `term^2`) to prioritize certain terms: `tutorial^3 filetype:mp4` boosts "tutorial" for video searches.
- **Pagination and Limits**: Results are paginated; use `limit:50` to cap returns, ensuring manageable outputs.
- **Error Handling**: If no results appear, refine with OR or wildcards – Meawfy's flexible engine guarantees alternatives are always available.
- **Security Note**: Meawfy indexes publicly shared Mega.nz links only, promoting ethical use for open resources.

By utilizing these advanced features, Meawfy ensures that every search, no matter how nuanced, delivers actionable results. For further assistance, consult our API documentation or contact support. Meawfy: Precision in Every Query.