Text Search Operator ($text)

Text search allows you to perform **full-text search** on string content across text-indexed fields. It’s designed for efficient keyword or phrase searches in text-heavy documents.

Text Search Overview

Operator	Description	Example Usage
$text	Performs a full-text search on fields with a text index	{ $text: { $search: "coffee organic" } }

$text

- Performs a **full-text search** on specified fields.
- Requires creating a **text index** on fields you want to search.
- Finds documents containing **any** of the search terms.
- Supports language-specific rules, stemming, and tokenization.
- Useful for searching words, phrases, or terms in large text data.

How to use $text:

Step 1: Create a text index on one or more fields

```
db.products.createIndex({
 name: "text",
 tags: "text",
 "details.brand": "text"
});
```

- *Creates a text index for name, tags, and details.brand fields.*

Example 1: Find products matching either "coffee" or "organic"

```
db.products.find({
 $text: { $search: "coffee organic" }
});
```

- *Finds products where “coffee” OR “organic” appear in the indexed fields.*

Example 2: Find products containing the exact phrase "dark roast"

```
db.products.find({
 $text: { $search: "\"dark roast\"" }
});
```

- *Finds products that contain the phrase “dark roast” exactly.*

Example 3: Search products excluding the word "decaf"

```
db.products.find({
 $text: { $search: "coffee -decaf" }
});
```

- *Finds products with “coffee” but excludes those that mention “decaf”.*

Example 4: Search with language specification (e.g., English)

```
db.products.find({
 $text: { $search: "organic coffee", $language: "english" }
});
```

- *Performs text search applying English language rules for stemming and stop words.*

Additional Notes:

- You can combine $text with other filters for more precise querying.
- Text search results include a **text score** indicating relevance, accessible via projection:

  ```
  db.products.find(
   { $text: { $search: "coffee" } },
   { score: { $meta: "textScore" } }
  ).sort({ score: { $meta: "textScore" } });
  ```

  - This sorts results by relevance score.