# 信息检索 API 文档

**版本:** 1.0.0  
**基础URL:** http://server\_address:port，这里以 http://127.0.0.1:8000为例。

## 简介

本API服务提供了一个完整的文档信息检索解决方案，包括文档上传、向量化存储和智能检索功能。用户可以通过上传结构化或非结构化的文本文件来构建知识库，并通过自然语言查询来检索最相关的信息。

后端利用Elasticsearch进行高效的混合检索（关键词+向量），并结合先进的重排模型（Reranker）对召回结果进行二次排序，以确保返回结果的精准性。

## 1. 智能检索

### POST /search

此端点用于执行文档的智能检索。它接收一个自然语言查询，并通过“召回+精排”的两阶段流程返回最相关的文档列表。

#### 请求体 (Request Body)

**Content-Type:** application/json

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必需 | 描述 |
| query | string | 是 | 用户的自然语言查询文本。 |
| top\_k | integer | 否 | 希望返回的重排后结果数量。默认值为 3，取值范围 1-20。 |

**示例:**

{   
 "query": "什么是向量数据库？",   
 "top\_k": 5   
}

#### 响应 (Response)

**200 OK - 成功**

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| results | array | 一个包含检索结果对象的列表，按相关性分数从高到低排序。 |
| results[].score | float | 文档与查询的相关性分数。 |
| results[].text | string | 匹配到的文档内容。 |
| results[].metadata | object | 文档的元数据。 |

**示例:**

{   
 "results": [   
 {   
 "score": 0.987,   
 "text": "第二行介绍了什么是向量数据库。",   
 "metadata": {   
 "source": "sample\_data.txt",   
 "mermaid\_txt": ""   
 }   
 },   
 {   
 "score": 0.85,   
 "text": "Elasticsearch不仅是搜索引擎，还能存储和分析向量数据。",   
 "metadata": {   
 "source": "tech\_blog\_post.html",   
 "mermaid\_txt": ""   
 }   
 }   
 ]   
}

**错误响应**  
\* 503 Service Unavailable: 如果后端的核心模型（如Embedding或Reranker模型）未能成功加载。  
\* 500 Internal Server Error: 如果在检索过程中发生其他内部错误。

#### cURL 示例

curl -X POST "http://127.0.0.1:8000/search" \   
-H "Content-Type: application/json" \   
-d '{   
 "query": "介绍一下图表工具",   
 "top\_k": 2   
}'

## 2. 数据管理

### 2.1 POST /upload\_txt

上传一个纯文本文件 (.txt)。系统会按行读取文件内容，将每一行作为一个独立的文档进行向量化并存入知识库。  
这个接口无法存储文本的附加信息，如mermaid\_txt。只能存储文档本身的内容。

#### 请求体 (Request Body)

**Content-Type:** multipart/form-data

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 描述 |
| file | file | 是 | 要上传的 .txt 文件。 |

#### 响应 (Response)

**200 OK - 成功**

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| message | string | 操作成功的提示信息。 |
| filename | string | 已上传文件的名称。 |
| items\_processed | integer | 文件中被成功处理并入库的非空行数。 |

**示例:**

{   
 "message": "文件 'sample\_data.txt' 上传并处理成功。",   
 "filename": "sample\_data.txt",   
 "items\_processed": 3   
}

**错误响应**  
\* 400 Bad Request: 如果上传的不是 .txt 文件，或者文件中不包含任何有效内容。  
\* 503 Service Unavailable: 如果后端核心模型未加载。  
\* 500 Internal Server Error: 文件处理过程中发生其他错误。

#### cURL 示例

curl -X POST "http://127.0.0.1:8000/upload\_txt" \   
-H "Content-Type: multipart/form-data" \   
-F "file=@/path/to/your/document.txt"

### 2.2 POST /upload\_json

上传一个结构化的 JSON 文件 (.json)。该文件必须包含一个对象列表，每个对象代表一个待入库的文档，包含文本内容和自定义元数据。

#### 请求体 (Request Body)

**Content-Type:** multipart/form-data

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 描述 |
| file | file | 是 | 要上传的 .json 文件。 |

#### JSON 文件内容格式

上传的JSON文件内容必须是一个列表 (array)，列表中的每个元素都是一个具有以下结构的JSON对象 (object)：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 是否必需 | 描述 |
| text | string | 是 | 文档的原始内容。 |
| metadata | object | 是 | 与文档相关的元数据。 |
| metadata.source | string | 是 | (推荐) 文档的来源标识。 |
| metadata.mermaid\_txt | string | 是 | (推荐) 与文档相关的Mermaid图表定义文本。 |

**JSON 文件示例 (my\_data.json):**

[   
 {   
 "text": "Mermaid是一种基于文本的图表绘制工具。",   
 "metadata": {   
 "source": "manual\_doc\_1.md",   
 "mermaid\_txt": "graph TD; A-->B;"   
 }   
 },   
 {   
 "text": "Elasticsearch不仅是搜索引擎，还能存储和分析向量数据。",   
 "metadata": {   
 "source": "tech\_blog\_post.html",   
 "mermaid\_txt": ""   
 }   
 }   
]

#### 响应 (Response)

**200 OK - 成功**

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| message | string | 操作成功的提示信息。 |
| filename | string | 已上传文件的名称。 |
| items\_processed | integer | JSON文件中被成功处理并入库的对象数量。 |

**示例:**

{   
 "message": "文件 'my\_data.json' 上传并处理成功。",   
 "filename": "my\_data.json",   
 "items\_processed": 2   
}

**错误响应**  
\* 400 Bad Request: 如果文件不是 .json 格式、JSON内容无效、或JSON结构不符合规定（如顶层不是列表，或列表元素缺少text/metadata字段）。  
\* 503 Service Unavailable: 如果后端核心模型未加载。  
\* 500 Internal Server Error: 文件处理过程中发生其他错误。

#### cURL 示例

curl -X POST "http://127.0.0.1:8000/upload\_json" \   
-H "Content-Type: multipart/form-data" \   
-F "file=@/path/to/your/my\_data.json"