

向量数据库接口调用文档

环境说明

- 基础 URL: `http://{HOST}:{PORT}`
- HOST: 10.101.100.13 (公司测试环境)
- PORT: 8105 (默认值)

接口列表

1. 创建文档集 (Create Collection)

请求

- 方法: `GET`
- 路径: `/create-collection`
- 参数:
 - `collection_name` (string): 要创建的文档集名称

示例

```
GET http://10.101.100.13:8105/create-collection?collection_name=my_collection
```

响应

```
{
  "ok": true,
  "message": "文档集 my_collection 创建成功。",
  "data": {
    "collection": {
      "name": "my_collection",
      "chunks_count": 0
    }
  }
}
```

2. 列出所有文档集 (List All Collections)

请求

- 方法: `GET`
- 路径: `/list-all-collections`

示例

```
GET http://10.101.100.13:8105/list-all-collections
```

响应

```
{
  "ok": true,
  "message": "文档集列表获取成功。",
  "data": {
    "collections": ["collection1", "collection2", "collection3"]
  }
}
```

3. 获取文档集 (Get Collection)

请求

- 方法: GET
- 路径: /get-collection
- 参数:
 - collection_name (string): 要获取的文档集名称

示例

```
GET http://10.101.100.13:8105/get-collection?collection_name=my_collection
```

响应

```
{
  "ok": true,
  "message": "文档集 my_collection 获取成功。",
  "data": {
    "collection": {
      "name": "my_collection",
      "chunks_count": 3
    }
  }
}
```

4. 删除文档集 (Delete Collection)

请求

- 方法: `GET`
- 路径: `/delete-collection`
- 参数:
 - `collection_name` (string): 要删除的文档集名称

示例

```
GET http://10.101.100.13:8105/delete-collection?collection_name=my_collection
```

响应

```
{
  "ok": true,
  "message": "文档集 my_collection 删除成功。",
  "data": null
}
```

5. 添加文档 (Add Document)

请求

- 方法: `POST`
- 路径: `/add-document`
- 请求体:
 - `collection_name` (string): 文档集名称
 - `document_name` (string): 文档名称
 - `document_id` (string): 文档 ID
 - `document` (string): 文档内容
 - `metadata` (object, 可选): 文档元数据

示例

```
POST http://10.101.100.13:8105/add-document
Content-Type: application/json
```

```
{
  "collection_name": "my_collection",
  "document_name": "合同法",
  "document_id": "12345",
  "document": "合同法的内容...",
  "metadata": {
    "created": "2024-12-02",
    "updated": "2024-12-02"
  }
}
```

```
"metadata": {  
  "source": "中国政府网站"  
}  
}
```

响应

```
{  
  "ok": true,  
  "message": "文档已添加到文档集 my_collection 中。",  
  "collection": null,  
  "data": {  
    "name": "my_collection",  
    "chunks_count": 3  
  }  
}
```

6. 获取文档片段 (Get Chunks)

请求

- 方法: GET
- 路径: /get-chunks
- 参数:
 - collection_name (string): 文档集名称
 - document_name (string, 可选): 文档名称
 - document_id (string, 可选): 文档 ID
 - document_name 和 document_id 只能二选一

示例

```
GET http://10.101.100.13:8105/get-chunks?  
collection_name=my_collection&document_id=12345
```

or

```
GET http://10.101.100.13:8105/get-chunks?  
collection_name=my_collection&document_name=合同法
```

响应

```
{
  "ok": true,
  "message": "文档片段获取成功。",
  "collection": null,
  "data": {
    "ids": ["合同法-12345-#0", "合同法-12345-#1", "合同法-12345-#2"],
    "embeddings": null,
    "metadatas": [
      {
        "document_name": "合同法",
        "document_id": "12345",
        "source": "中国政府网站"
      },
      {
        "document_name": "合同法",
        "document_id": "12345",
        "source": "中国政府网站"
      }
    ],
    "documents": [
      "合同法的内容... #1",
      "合同法的内容... #2",
      "合同法的内容... #3"
    ],
    "chunks_count": 3
  }
}
```

7. 删除文档 (Delete Document)

请求

- 方法: GET
- 路径: `/delete-document`
- 参数:
 - `collection_name` (string): 文档集名称
 - `document_name` (string, 可选): 文档名称
 - `document_id` (string, 可选): 文档 ID
 - `document_name` 和 `document_id` 只能二选一

示例

```
GET http://10.101.100.13:8105/delete-document?
collection_name=my_collection&document_id=12345
```

or

```
GET http://10.101.100.13:8105/delete-document?
collection_name=my_collection&document_name=合同法
```

响应

```
{
  "ok": true,
  "message": "文档片段删除成功。共删除 3 个文档片段",
  "data": {
    "chunks_deleted": 3
  }
}
```

8. 查询 (Query)

请求

- 方法: GET
- 路径: /query
- 参数:
 - collection_name (string): 文档集名称
 - query (string): 查询文本
 - n_results (integer): 返回结果数量
 - rerank (boolean, 可选): 是否重新排序

示例

```
GET http://10.101.100.13:8105/query?collection_name=my_collection&query=未成年人
&n_results=2&rerank=true
```

响应

```
{
  "ok": true,
  "message": "查询成功。",
  "data": {
    "ids": ["合同法-12345-#2", "合同法-12345-#0", "合同法-12345-#1"],
    "distances": [[0.5290875981365002, 0.60670972189909, 0.6192247639104045]],
    "metadatas": [
      [
        { "document_id": "12345", "document_name": "合同法" },
        { "document_id": "12345", "document_name": "合同法" },
        { "document_id": "12345", "document_name": "合同法" }
      ]
    ]
  }
}
```

```
    ]
  ],
  "embeddings": null,
  "documents": [["合同法内容... #2", "合同法内容... #0", "合同法内容... #1"]],
  "uris": null,
  "data": null,
  "included": ["metadatas", "documents", "distances"],
  "rerank_scores": [
    0.15449749029866022, 0.001962731646069681, 0.001160887847174835
  ]
}
}
```

9. 获取文档集所有元数据 (Get all Metadatas in Collection)

请求

- 方法: `GET`
- 路径: `/get-all-metadatas-in-collection`
- 参数:
 - `collection_name` (string): 要获取的文档集名称

示例

```
GET http://10.101.100.13:8105/get-all-metadatas-in-collection?
collection_name=my_collection
```

响应

```
{
  "ok": true,
  "message": "文档集 my_collection 所有元数据获取成功。",
  "data": {
    "metadatas": [
      {
        "document_id": "1788765438918787074",
        "document_name": "郑州商品交易所期货结算管理办法 ( 2023年9月修订 )"
      },
      {
        "document_id": "1788765439107530753",
        "document_name": "上海期货交易所交割细则 ( 2023年8月修订 )"
      }
    ]
  }
}
```

注意事项

1. 所有接口都返回 JSON 格式的响应。
2. 响应中的 `ok` 字段为布尔值，表示操作是否成功，`message` 字段提供操作结果的描述。
3. 部分接口（如添加文档）使用 POST 方法，需要在请求头中设置 `Content-Type: application/json`。
4. 查询接口支持可选的重新排序功能，可以通过 `rerank=true` 参数启用。
5. 先创建文档集，然后添加文档到文档集，再 Query 文档集获取相关文档片段。
6. 删除文档集或文档时要谨慎，操作不可逆。