腾讯云数据库 AI 服务

DeepSearch 应用指南

腾讯云

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

DeepSearch 应用指南
DeepSearch 介绍

最近更新时间：2025-09-17 11:02:21

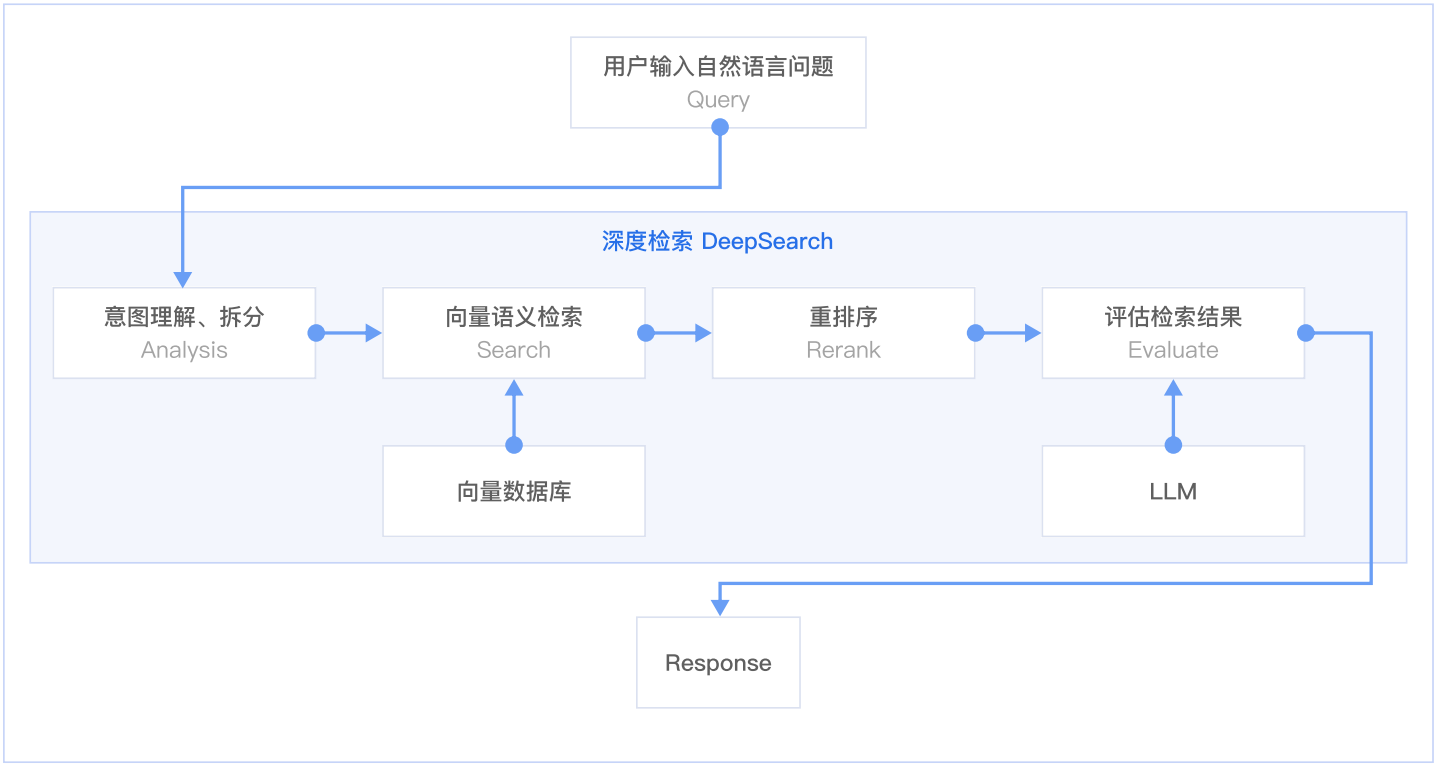
功能介绍

DeepSearch 是一款以自然语言作为查询输入，具备问题理解和评估反馈能力的检索服务，依托深度意图理解与多轮评估机制，实现对上下文的精准感知与知识片段的高效召回，显著提升应答准确性与智能水平。

- 构建“理解-检索-排序-评估”一体化闭环流程，持续优化检索相关性，确保结果稳定可靠。
- 扩展上下文覆盖维度，增强多源信息融合能力，有效提升召回率与知识完整性。
- 生成可解释的检索路径与关联线索，所有结果均具备清晰溯源，保障过程透明、可信。

工作机制

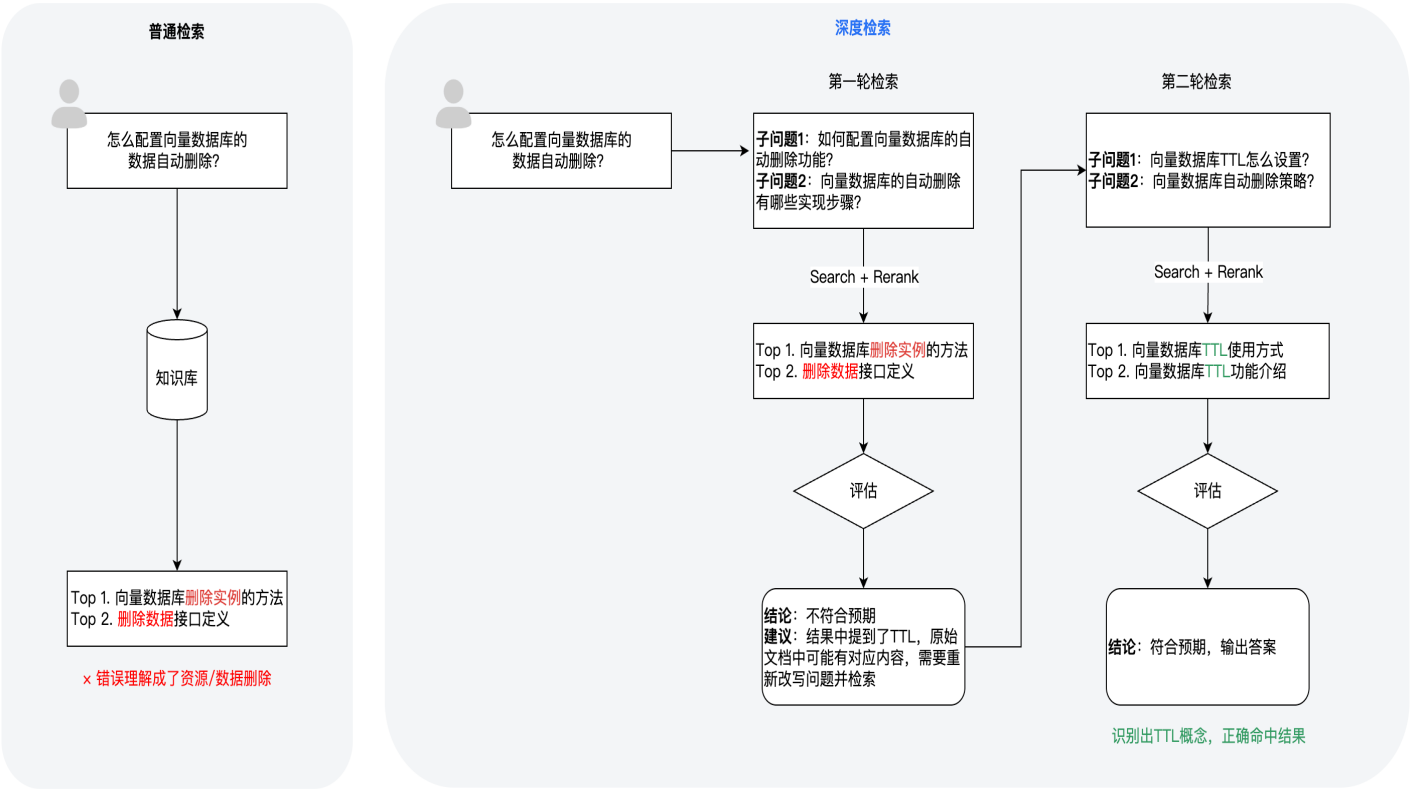
DeepSearch 工作流程始于接收用户自然语言 Query，通过 Analysis 模块解析语义并拆解子任务，随后执行向量与关键词混合 Search，从向量数据库中召回初步相关的文档片段。检索结果进入 Rerank 进行重排序，进入 Evaluate 评估质量与完整性，以决定是否启动多轮优化循环，最终生成 Response 返回用户。整个流程中，LLM 驱动意图理解与答案生成，向量数据库为检索提供核心支持。



关键能力	解释说明
Analysis（查询分析）	接收用户自然语言查询，进行意图识别、关键词提取、语义扩展等分析，明确检索目标。
Search（检索）	将查询语句转换为向量，在向量数据库中进行相似性搜索，找出最相关的文本片段。
Rerank (重排序)	对检索到的结果列表，利用更精细的模型根据其与查询的相关性进行精确排序。
Evaluate（结果评估）	对检索结果进行去重、可信度评估，决定是否需要进入循环（Loop）。
Response（响应生成）	将评估后的结果整合成自然语言响应（如摘要、答案、推荐等），返回给用户。

应用示例

下图以“怎么配置向量数据库的数据自动删除”问题为例，通过 DeepSearch 进行多轮迭代优化实现精准答案定位的过程。整个过程体现了 DeepSearch 在语义理解、检索优化和结果评估方面的显著优势。



使用场景

DeepSearch 作为一个统一的智能检索底座，通过其强大的理解、检索、分析与反馈能力，能够为各行各业的应用场景注入“智能”，驱动业务创新和效率提升。

企业知识问答 / 知识库搜索

DeepSearch 能够对企业内部的文档、FAQ、流程制度等进行高效的语义理解和检索，不仅支持自然语言提问，还能结合上下文给出可信答案，显著降低信息检索门槛，减少重复咨询，提升团队协作与知识复用效率。无论是新员工培训还是日常业务查询，DeepSeek 都能提供一致、可靠的信息支持。

研发与运维助手

DeepSearch 可深度接入代码仓库、变更记录、监控日志和故障报告，为开发和运维团队提供智能检索与问题分析支持，能够快速定位错误来源、关联历史故障，并给出修复建议，极大提升排查效率。同时，DeepSearch 还能自动梳理和沉淀解决方案，形成可复用的知识库，促进团队经验积累和技术成长。

客服与服务台辅助

通过整合知识库、工单系统、公告和操作指南等多源信息，DeepSearch 为客服人员提供实时、结构化的应答建议，并确保每一条回复具备参考来源和可追溯性。这不仅大幅提升客服响应速度与准确性，也增强了应答的一致性和可靠性，从而提高客户满意度和服务品质。

合规与审计检索

DeepSearch 帮助企业合规、法务及审计部门高效处理大量结构化与非结构化文档，如法规条文、合同约定和企业制度等，能够精准匹配条款内容、识别潜在风险点，并支持多维度关联分析，从而提升审查全面性和准确性，降低合规风险，支持企业稳健运营。

行业资料研读

面对研报、专利、标准文档等大量专业资料，DeepSearch 可进行关键信息提取和片段精准召回，帮助用户快速获取核心观点、技术要点和趋势分析。不仅支持长文档的语义检索和摘要生成，还能够关联多文档内容，辅助用户高效完成研读、分析和决策，极大提升信息利用效率和业务洞察力。

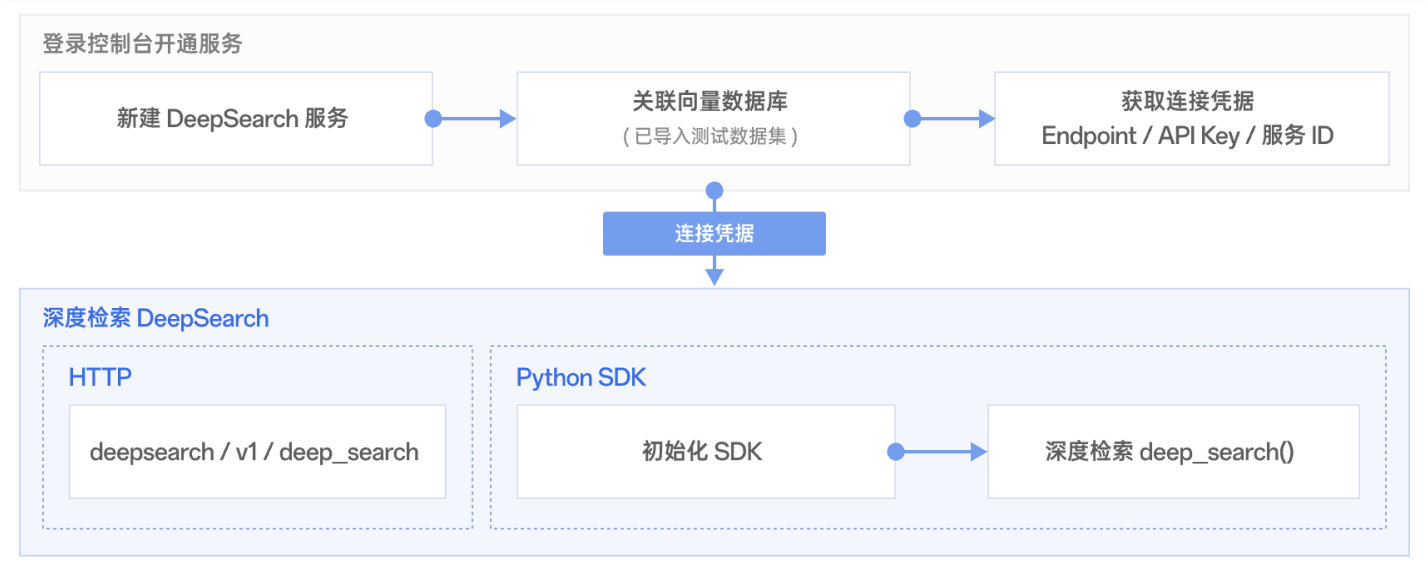
快速体验

最近更新时间：2025-09-17 16:17:35

本文旨在指导您如何快速部署并使用 DeepSearch，包括开通服务、导入测试数据、深度检索等。

流程解读

DeepSearch 服务的完整使用流程包括：登录控制台开通服务后，获取连接凭据（包括 Endpoint、API Key 和服务 ID），便可通过 HTTP 接口或 Python SDK 进行深度检索体验。



准备工作

- 已注册腾讯云账号并完成实名认证。
  - 如需注册腾讯云账号：请单击 注册腾讯云账号 。
  - 如需完成实名认证：请单击 实名认证 。
- 申请一台与 DeepSearch 服务在同一地域（目前仅开放**广州**）的云服务器，并确保该服务器与 DeepSearch 网络互通，以获得安全且低延迟的访问体验。

操作步骤

步骤1：开通 DeepSearch 服务

1. 使用腾讯云账号登录 DeepSearch 控制台 页面，配置 DeepSearch 的基本信息，并关联向量数据库实例。若无向量数据库，请根据页面指引，新建向量数据库。具体操作，请参见 新建 DeepSearch 。
2. 获取连接凭据：在 **DeepSearch 服务列表**页面，可看到新建的 DeepSearch 卡片，如下图所示。

- **DeepSearch ID**：将鼠标悬停在 ID 上，单击 ID 可一键复制 DeepSearch ID。
- **访问地址及密钥**：单击卡片，可在**服务概览**的 **API 接入**区域获取 DeepSearch **访问地址**与**访问密钥**。

步骤2：准备测试数据

- 方式1：使用已有库表数据进行深度检索。

> **说明：**
> 其集合中需要包含检索对应的文本字段，深度检索需要基于原始文本进行匹配、评估。

- 方式2：使用 Demo 数据进行深度检索。

使用脚本 batch_upload_files.py 导入测试数据集 向量数据库产品文档.zip 。具体示例，如下所示。

```
python batch_upload_files.py \
--vdb_url http://{访问域名}:{端口} \
--vdb_key {访问密钥} \
--user_name root \
--db_name {数据库名称} \
--coll_name {集合名称} \
--data_dir {数据文件夹路径}
```

- 方式2：基于已有的向量数据库深度检索，其集合中需要包含检索对应的文本字段，深度检索需要基于原始文本进行匹配、评估。

步骤3：使用 DeepSearch 深度检索

以“怎么配置向量数据库的数据自动删除”为例，通过申请的 CVM 连接并调用 DeepSearch 服务后，依托“Query 拆分→检索→重排序→结果评估”的迭代机制，经过两轮检索优化，最终返回了与用户意图高度匹配的精准答案。

- Authorization：配置 DeepSearch 访问密钥。
- x-tdai-service-id：配置 DeepSearch ID。

```
curl -i -k -X POST \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer ****************************************' \
```

```
  -H "x-tdai-service-id: tdai-dps-9f3w****" \
  https://deepsearch.tdai.tencentyun.com/deepsearch/v1/deep_search \
  -d '{
    "database": "dps_test_db",
    "collection": "dps_test_coll",
    "search": {
      "query": "怎么配置向量数据库的数据自动删除",
      "query_preprocessing": {
        "method": "sub_query_split"
      },
      "retrieval_config": {
        "mode": "dense",
        "limit": 3,
        "retrieval_params": {
          "ef": 200
        }
      },
      "rerank": {
        "rerank_model": "bge-reranker-large"
      },
      "evaluate_config": {
        "max_iterations": 3
      },
      "limit": 3,
      "return_execution_details": true
    }
  }'
```

基于测试数据集，深度检索的结果如下所示。


```
{
    "id": "1409808654987071492",
    "score": 0.98535156,
    "text": "## 适用场景\n\n某些业务场景下，业务数据的增长很快，并且业务数据的热度随着时间推移会有明显的降低，可以使用 TTL 将热度降低的数据在特定的过期时间时自动清理。例如：在电子商务平台中，用户搜索和浏览行为产生的向量数据，如点击流和用户偏好向量，通常具有很高的时效性。这些数据在短期内对于个性化推荐至关重要，但随着时间的推移，其相关性会迅速降低。在向量数据库集合中，设置TTL，便会自动删除过时的用户行为数据，确保推荐系统能够基于最新的用户互动提供实时、相关的推荐，从而提升用户体验和业务效率。\n\n"
},
{
    "id": "1409808654987071491",
    "score": 0.46704102,
```

```
    "text": "## 功能介绍\n\nTTL（Time to Live，生存周期）功能，指数据项从创建或更新后开始，到自动被系统删除的时间期限。在创建数据库集合时，通过为文档设置TTL，数据库能够自动清理过期的数据，从而优化存储空间的使用，保持数据的时效性，并确保查询结果的准确性，特别是在数据具有时效性要求的场景中，如实时推荐系统等。\n\n"
},
{
    "id": "1409808654987071496",
    "score": 0.27807617,
    "text": "
.withTtlConfig(TTLConfig.newBuilder().WithEnable(true).WithTimeField(\"expired_at\").build())\n            .build();\n    Collection collection = db.createCollection(collectionParam);\n    ```\n2. 插入向量数据时，指定 TTL 字段 expired_at 过期时间。\n    \n\n    > **说明：**\n    > \n\n    > 数据库插入数据，过期时间允许有1个小时的误差。\n    > \n\n    ``` java\n"
}
```

控制台指南
新建 DeepSearch

最近更新时间：2025-09-17 11:02:21

操作场景

本文旨在指导您快速新建并配置一个 DeepSearch 服务。从设置基本信息到关联相关实例，逐步引导您完成全部构建流程。

前提条件

已注册腾讯云账号并完成实名认证。

- 如需注册腾讯云账号：请单击 注册腾讯云账号 。
- 如需完成实名认证：请单击 实名认证 。

操作步骤

1. 使用腾讯云账号登录 DeepSearch 控制台 页面。
2. 单击 **新建 DeepSearch**，进入DeepSearch 配置页面。
3. 参考如下表的信息，配置 DeepSearch 的基本信息与关联实例。

界面区域	界面参数	解释	限制与建议
基本信息	地域	服务数据存储的物理位置。选择离你的用户群体最近的地域可以获得更低的访问延迟和更好的性能。	目前仅开放广州地域。

	服务名称	DeepSearch 的唯一标识名称，用于管理和区分不同的 DeepSearch 服务。	• 限制：60个字符以内，支持中英文、数字及下划线。 • 建议：名称应能直接反映 DeepSearch 的用途。
	描述	对 DeepSearch 功能的描述说明。	–
	关联大模型	指定该记忆体服务关联的大模型。模型决定了信息处理（如总结、提取）的质量和能力。	目前默认且仅支持 DeepSeek-V3。
	标签	以键值对的形式为资源添加标识，用于项目管理、环境区分（如开发/测试/生产）等。	–
关联实例	关联产品	当前仅支持关联向量数据库。	–
	关联实例	选择 DeepSearch 所关联的向量数据库实例。	仅可选择与 DeepSearch 在同一地域的向量数据库。
	数据库账号	指定向量数据库访问账号。	–
	数据库密码	指定向量数据库访问账号的密码。	–

4. 单击**连通性测试**，测试 DeepSearch 与向量数据库的网络连通性。
5. 单击**立即试用**，等待任务执行完成，可在 DeepSearch 控制台 页面查看到已创建的 DeepSearch，如下所示。具体信息，请参见 查看 DeepSearch 。

查看 DeepSearch

最近更新时间：2025-09-12 21:41:52

操作场景

您可以前往管理 DeepSearch 的控制台，查看 DeepSearch 的基本信息，关联实例、访问地址等信息。

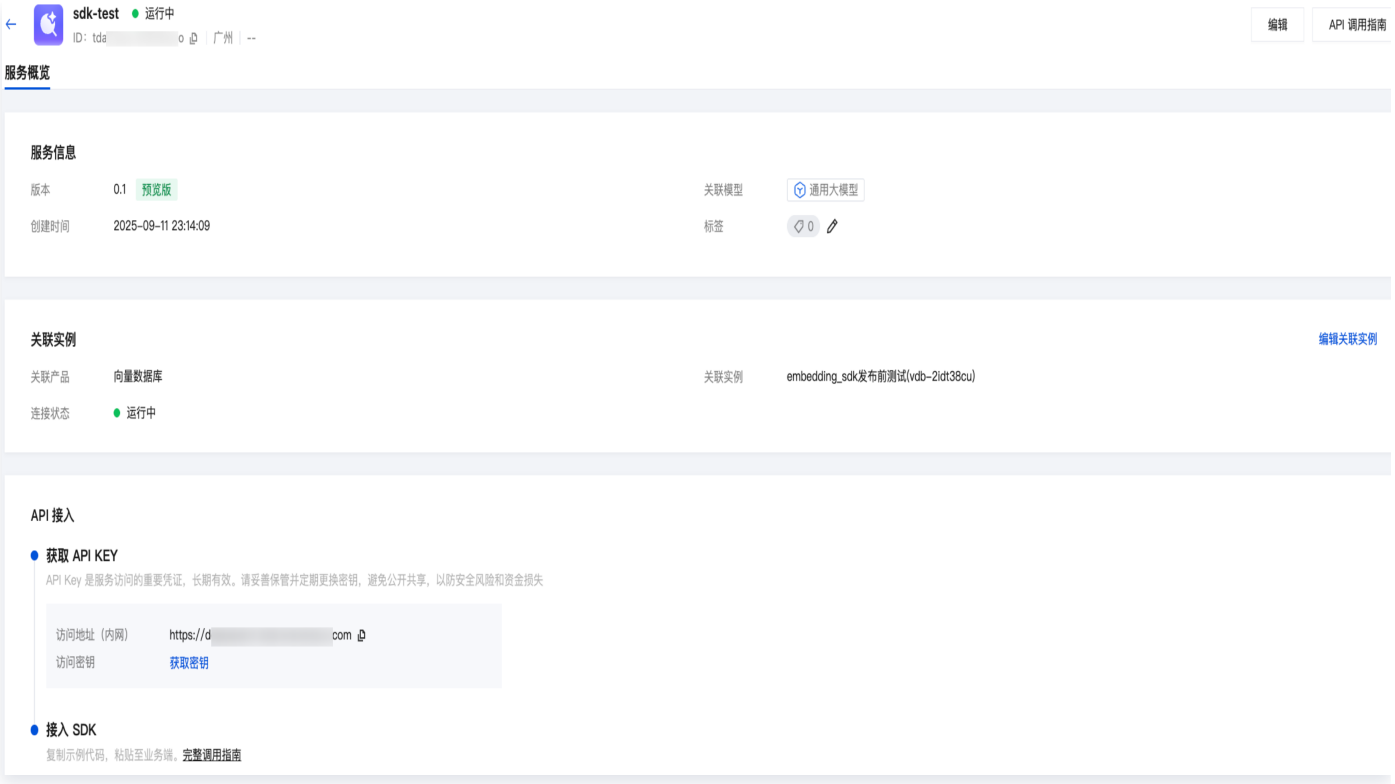
操作步骤

1. 使用腾讯云账号登录 DeepSearch 控制台 。

2. 在每一个 DeepSearch 卡片上，您可以查看到 DeepSearch 名称、ID、关联的大模型、 运行状态及创建时间等信息。

> **说明：**
> DeepSearch ID 由系统随机分配，具有唯一性，将鼠标悬停在 ID 上，点击 ID 可一键复制 ID。

3. 在右上角的搜索框，可根据**服务名称**、**服务 ID**、**服务状态**、**标签**搜索需查看的 DeepSearch。
4. 单击 DeepSearch 卡片，进入查看 DeepSearch 详情页面，如下所示。
   - 在**服务信息**区域，可获取 DeepSearch 的版本信息、关联模型、创建时间及标签。
   - 在**关联实例**区域，可获取 DeepSearch 关联的产品信息、实例信息、以及数据库的连接状态。
   - 在 **API 接入**区域，可获取 DeepSearch 的访问地址与访问密钥。单击**接入 SDK** 的**完整调用指南**可跳转至 SDK 接口文档，可使用 SDK 接口进行深度检索。



5. （可选）单击右上角的**编辑**，可修改 DeepSearch 的名称。
6. （可选）单击右上角的 **API 调用指南**，跳转至 API 接口文档，可使用 HTTP API 接口进行深度检索。

开发者工具
Python SDK
创建 Client

最近更新时间：2025-09-12 21:41:52

接口定义

DeepSearchClient() 用于通过 HTTP I/O 请求方式的请求方式创建一个 DeepSearch 的客户端对象。

```
class DeepSearchClient(
    endpoint: str,
    api_key: str,
    service_id: str,
    timeout: int = 60,
    stub: Stub | None = None
)
```

使用示例

```
from tdaideepsearch import DeepSearchClient
EndPoint = "https://deepsearch.tdai.tencentyun.com"
ApiKey = "************************"
ServiceId = "tdai-dps-9f3w****"

client = DeepSearchClient(endpoint=EndPoint, api_key=ApiKey,
service_id=ServiceId)
```

参数名	参数含义	是否必须	获取方式
endpoint	客户端所需连接 DeepSearch 服务端的访问地址。	是	登录 DeepSearch 控制台，在详情页面的 API 接入区域，可获取 DeepSearch 的访问地址与访问密钥。
api_key	客户端访问 DeepSearch 服务端的	是	

	API 密钥，用于进行身份认证。		
service_id	DeepSearch 服务端具有唯一标识的 ID。	是	登录 DeepSearch 控制台，在 DeepSearch 的卡片上或详情页面，可复制 ID。
timeout	请求超时时间。	否	• 单位：秒。 • 默认值：10。 • 取值范围：大于等于0。若 timeout 设置为小于0或为 null，系统会自动赋值为默认值。

deep_search

最近更新时间：2025-09-17 16:17:35

接口定义

deep_search() 提供端到端的 DeepSearch 检索服务，其完整流程包括：查询预处理、相似性检索、模型重排序（Rerank）以及基于 LLM 的评估与迭代。

```
def deep_search(
    database: str,
    collection: str,
    query: str,
    query_preprocessing: dict | QueryPreprocessingParams | None = None,
    retrieval_config: dict | RetrievalConfigParams | None = None,
    rerank: dict | SearchRerankParams | None = None,
    evaluate_config: dict | EvaluateConfigParams | None = None,
    limit: int | None = None,
    filter: str | None = None,
    output_fields: List[str] | None = None,
    return_execution_details: bool | None = None
) -> Dict[str, Any]
```

说明：
DeepSearch 每秒最多能处理10,000个 token，涵盖了 Embedding（向量化）、LLM（大语言模型生成）和 Rerank（重排序）三个核心组件的消耗。

使用示例

```
import json
from tdaideepsearch import DeepSearchClient

res = client.deep_search(
    database="dps_test_db",
    collection="dps_test_coll",
    query="怎么配置向量数据库的数据自动删除",
    query_preprocessing={
        "method": "sub_query_split",
    },
    retrieval_config={
        "mode": "dense",
```

```
        "params": {
            "ef": 200
        },
        "limit": 3
    },
    rerank={
        "rerank_model": "bge-reranker-large"
    },
    evaluate_config={
        "max_iterations": 3,
    },
    limit=3,
    # filter="productId=1",
    # output_fields=["id", "text"],
    return_execution_details=True,
)
print(json.dumps(res, indent=2, ensure_ascii=False))
```

请求参数

参数（一级）	参数（二级）	参数（三级）	是否必选	参数含义	配置方法及要求
database	-	-	是	指定 DeepSearch 检索所关联的向量数据库名称。	在控制台关联的向量数据库。
collection	-	-	是	指定 DeepSearch 检索的向量数据库的集合名称。	所关联向量数据库中已经创建好的向量集合。
search	query	-	是	用户传入的自然语言问题。	数据类型：String。
	query_preprocessing	method	否	指定输入问题的预处理方法。	当前仅支持指定为 sub_query_split，表示系统会使用 LLM 将复杂的主查询语句拆解为多个更易回答的子问题，并行

					检索后再合并结果，以提高复杂问题的检索效果。
	retrieval_config	embedding_model	否	指定用于检索的向量化模型。 **说明：** 如果创建集合时已经绑定了模型，则以集合配置为准，此处的指定会被忽略。	您需根据业务的语言类型、数据维度要求等综合选择合适的模型。取值如下所示： • **bge-large-zh-v1.5**：适用中文，1024维，推荐使用。 • **bge-base-zh-v1.5**：适用中文，768维。 • **bge-large-zh**：适用中文，1024维。 • **bge-base-zh**：适用中文，768维。 • **m3e-base**：适用中文，768维。 • **e5-large-v2**：适用英文，1024维。 • **text2vec-large-chinese**：适用中文，1024维。 • **multilingual-e5-base**：适用于多种语言类型，768维。 • **BAAI/bge-m3**：适用于多种语言类型，1024维。
		text_field	否	指定向量数据库集合中所检索的文本字段。	以向量数据库集合中配置的文本字段为准。
		mode	否	指定检索模式。	支持如下检索模式。 • dense：单路稠密型向量检索。默认为：dense。 • sparse：单路稀疏型向量检索。 • hybrid：稠密向量与稀疏向量两路混合检索。

		fusion_mode	否	若 mode 为 hybrid，则需指定混合检索的重排序（rerank）的方式。	支持的重排序（rerank）方式如下所示。
					• weighted：基于稠密向量与稀疏向量的加权组合来进行排序。
					  ○ dense_weight：指定稠密向量的权重。例如，0.9。
					  ○ sparse_vector：指定稀疏向量的权重。例如，0.1。
					• rrf：一种融合稠密向量与稀疏向量各自相关性评分体系的检索结果排序算法。参数 k 是一个平滑因子，主要用于调节低排名文档对最终结果的影响程度。默认值为60。

		params	否	指定向量数据库集合索引类型所对应的检索参数。	• FLAT：无需指定参数。 • HNSW 类型：需配置参数 ef，指定需要访问向量的数目。取值范围[1,32768]，默认为10。 • IVF 系列：需设置参数 nprobe，指定所需查询的单位数量。取值范围[1,nlist]，其中 nlist 在创建 Collection 时已设置。
		limit	否	指定检索向量数据库返回的最大数量。	• 默认值：最终 DeepSearch 检索返回数量的3倍。 • 最大值：100。 • 取值：向量数据库检索返回的数量与最终DeepSearch 检索返回的数量，二者取较大者作为实际取值。
	rerank	rerank_model	否	指定重排序配置模型。	当前仅支持 bge-reranker-large 模型，评估检索出来的结果和原始问题的关联程度去做重排序。
	evaluate_config	max_iterations	否	指定多轮计算的次数。	正整数，最小值为1，最大值为5。

	limit	-	是	最终返回的文档数量。	正整数，取值范围[1,10]。
	filter	-	否	设置检索的过滤条件。	对向量数据库集合的标量字段设置 FIlter 表达式。具体信息，请参见 Filter 表达式 。
	output_fields	-	否	指定返回结果中应包含哪些字段。	返回字段为所检索的向量数据库集合中的的字段。
	return_execution_details	-	否	指定是否返回检索过程信息。	• false：不返回检索过程。默认 false。 • true：返回检索过程。

出参描述


```
{
  "code": 0,
  "message": "operation success",
  "documents": [
    {
      "id": "1415908106070560773",
      "score": 0.98535156,
      "chunk_num": 2,
      "file_name": "VDB TTL.md",
      "section_num": 2,
      "text": "## 适用场景\n\n某些业务场景下，业务数据的增长很快，并且业务数据的热度随着时间推移会有明显的降低，可以使用 TTL 将热度降低的数据在特定的过期时间时自动清理。例如：在电子商务平台中，用户搜索和浏览行为产生的向量数据，如点击流和用户偏好向量，通常具有很高的时效性。这些数据在短期内对于个性化推荐至关重要，但随着时间的推移，其相关性会迅速降低。在向量数据库集合中，设置TTL，便会自动删除过时的用户行为数据，确保推荐系统能够基于最新的用户互动提供实时、相关的推荐，从而提升用户体验和业务效率。\n\n"
    },
    {
      "id": "1415908106070560772",
      "score": 0.46704102,
      "chunk_num": 1,
      "file_name": "VDB TTL.md",
      "section_num": 1,
      "text": "## 功能介绍\n\nTTL（Time to Live，生存周期）功能，指数据项从创建或更新后开始，到自动被系统删除的时间期限。在创建数据库集合时，通过为文档设置TTL，数据
```

```
库能够自动清理过期的数据，从而优化存储空间的使用，保持数据的时效性，并确保查询结果的准确性，特别是在数据具有时效性要求的场景中，如实时推荐系统等。\n\n"
    },
    {
      "id": "14159082057897410 90",
      "score": 0.07116699,
      "chunk_num": 1,
      "file_name": "VDB备份&恢复技术方案.pdf",
      "section_num": 1,
      "text": "# 向量数据库备份&恢复技术方案\n\n## 使用场景\n\n主要有以下两个场景：\n\n数据安全是云上数据库产品的生命线，备份回档作为保障数据安全的核心能力，是云上数据库产品的基本功能，提供后也能够为客户误操作导1.\n\n致的数据丢失、数据错乱等现象提供兜底。\n\n部分客户存在克隆数据到新实例的需求。2.\n\n具体到某个场景\n\n在数据库出现性能问题或故障时，离线备份提供了一种安全且可控的方式来排查问题。通过将数据库实例迁移到新的节点，可问题排查与性能验证：1.\n\n以在不影响生产环境的情况下进行详细的性能分析和故障排查。\n\n定期进行离线备份是防止数据丢失的关键措施。"
    }
  ],
  "execution_details": {
    "original_query": "怎么配置向量数据库的数据自动删除",
    "total_iterations": 2,
    "max_iterations": 3,
    "iterations": [
      {
        "iteration_no": 1,
        "query_processing": {
          "method": "sub_query_split",
          "input_query": "怎么配置向量数据库的数据自动删除",
          "sub_queries": [
            {
              "query": "向量数据库支持哪些数据自动删除机制",
              "purpose": "识别实体与核心功能"
            },
            {
              "query": "常见向量数据库（如Milvus、Pinecone、Weaviate）如何配置自动删除策略",
              "purpose": "获取具体产品的配置方法"
            },
            {
              "query": "配置向量数据库自动删除策略时的注意事项与最佳实践",
              "purpose": "补充操作细节与限制条件"
            }
          ],
```

```
            "token_usage": 351
          },
          "retrieval": {
            "mode": "dense",
            "sub_query_results": [
              {
                "query": "向量数据库支持哪些数据自动删除机制",
                "retrieved_docs": 3,
                "top_docs": [
                  "14159081108478730 59",
                  "14159081073036861 48",
                  "14159082057897410 90"
                ],
                "top_doc_texts": [
                  " | [TTL]
(https://cloud.tencent.com/document/product/****) | \nSDK 功能更新 |
Python、GO、Java SDK 支持对 Database、Collection 的存在判断。 | - |
\n\n\n\n",
                  "腾讯云向量数据库提供数据备份与回档服务，支持预设备份时间点并周期性执行主动备份，同时允许随时手动触发即时备份操作；所有备份均以高可靠快照形式存储，通过实例克隆功能可实现数据回档，恢复业务状态。\n\n> **说明：**\n> \n\n>  V2.4 版本起，支持产品化的备份克隆功能，若需要使用该功能，请 [提交工单]
(https://console.cloud.tencent.com/workorder/category) 升级实例版本。\n>
\n\n\n",
                  "# 向量数据库备份&恢复技术方案\n\n## 使用场景\n\n主要有以下两个场景：\n\n数据安全是云上数据库产品的生命线，备份回档作为保障数据安全的核心能力，是云上数据库产品的基本功能，提供后也能够为客户误操作导1.\n\n致的数据丢失、数据错乱等现象提供兜底。\n\n部分客户存在克隆数据到新实例的需求。2.\n\n具体到某个场景\n\n在数据库出现性能问题或故障时，离线备份提供了一种安全且可控的方式来排查问题。通过将数据库实例迁移到新的节点，可问题排查与性能验证：1.\n\n以在不影响生产环境的情况下进行详细的性能分析和故障排查。\n\n定期进行离线备份是防止数据丢失的关键措施。"
                ],
                "avg_score": 347.1428629557292
              },
              {
                "query": "常见向量数据库（如Milvus、Pinecone、Weaviate）如何配置自动删除策略",
                "retrieved_docs": 3,
                "top_docs": [
                  "14159081108478730 53",
                  "14159082057897410 90",
                  "14159082057897410 91"
                ],
```

```
            "top_doc_texts": [
              "综合体验提升特性 | - 索引参数修改 modifyVectorIndex，可修改原数据表的向量索引参数，降低重新建表并导入数据的成本。注意：这里修改索引参数后会自动发生一次 Rebuild Index。 | - 支持 Count，返回 Collection 所有的文档数量或满足 Filter条件的文档数量。 | - Delete 支持 Limit，单次删除的速度非常快，同时很好地控制对数据库业务影响。如果您的业务中需要删除海量数据，可以循环通过 Limit 删除配合 affectedCount 返回值来达到效果。",
              "# 向量数据库备份&恢复技术方案\n\n## 使用场景\n\n主要有以下两个场景：\n\n数据安全是云上数据库产品的生命线，备份回档作为保障数据安全的核心能力，是云上数据库产品的基本功能，提供后也能够为客户误操作导1.\n\n致的数据丢失、数据错乱等现象提供兜底。\n\n部分客户存在克隆数据到新实例的需求。2.\n\n具体到某个场景\n\n在数据库出现性能问题或故障时，离线备份提供了一种安全且可控的方式来排查问题。通过将数据库实例迁移到新的节点，可问题排查与性能验证：1.\n\n以在不影响生产环境的情况下进行详细的性能分析和故障排查。\n\n定期进行离线备份是防止数据丢失的关键措施。",
              "通过在非工作时间进行备份，可以最大限度地减少对业务的影响，同时确保数据的定期备份与恢复：2.\n\n一致性和完整性。定期备份不仅提供了数据恢复的保障，还支持在发生灾难性事件时快速恢复业务运行，确保业务的连续性和数据的完整性。\n\n数据库归档：对于不再频繁使用的数据库，或用户欠费的实例，离线备份可以作为一种有效的归档手段。通过备份某个时间点的数据，可以在用户再3.\n\n次启用时快速恢复数据库，确保数据的完整性和一致性。\n\n"
            ],
            "avg_score": 340.3140869140625
          },
          {
            "query": "配置向量数据库自动删除策略时的注意事项与最佳实践",
            "retrieved_docs": 3,
            "top_docs": [
              "14159082057897410 90",
              "14159082057897410 91",
              "14159082057939354 04"
            ],
            "top_doc_texts": [
              "# 向量数据库备份&恢复技术方案\n\n## 使用场景\n\n主要有以下两个场景：\n\n数据安全是云上数据库产品的生命线，备份回档作为保障数据安全的核心能力，是云上数据库产品的基本功能，提供后也能够为客户误操作导1.\n\n致的数据丢失、数据错乱等现象提供兜底。\n\n部分客户存在克隆数据到新实例的需求。2.\n\n具体到某个场景\n\n在数据库出现性能问题或故障时，离线备份提供了一种安全且可控的方式来排查问题。通过将数据库实例迁移到新的节点，可问题排查与性能验证：1.\n\n以在不影响生产环境的情况下进行详细的性能分析和故障排查。\n\n定期进行离线备份是防止数据丢失的关键措施。",
              "通过在非工作时间进行备份，可以最大限度地减少对业务的影响，同时确保数据的定期备份与恢复：2.\n\n一致性和完整性。定期备份不仅提供了数据恢复的保障，还支持在发生灾难性事件时快速恢复业务运行，确保业务的连续性和数据的完整性。\n\n数据库归档：对于不再频繁使用的数据库，或用户欠费的实例，离线备份可以作为一种有效的归档手段。通过备份某
```

个时间点的数据，可以在用户再3.\n\n次启用时快速恢复数据库，确保数据的完整性和一致性。\n\n",
"\"4 \n<\n<\n \"logid.index: \" \n<\n<\n logid.index 5 \n<\n<\n \", backup_index: \" \n<\n<\n backup_index;6 node_->leave_recover_mode();7 do_recover_events();8 }9 10 // 备份点之后且时间戳在新实例创建前的日志，需要跳过11 if (logid.index > backup_index && timestamp < FLAGS_recover_create_timestamp) {12 continue;"
],
"avg_score": 343.1875406901042
}
],
"total_retrieved_docs": 9,
"token_usage": 73
},
"rerank": {
"rerank_model": "bge-reranker-large",
"input_docs": 6,
"output_docs": 3,
"top_docs": [
"1415908110847873053",
"1415908107303686148",
"1415908205789741090"
],
"top_doc_texts": [
"综合体验提升特性 | - 索引参数修改 modifyVectorIndex，可修改原数据表的向量索引参数，降低重新建表并导入数据的成本。注意：这里修改索引参数后会自动发生一次 Rebuild Index。 | - 支持 Count，返回 Collection 所有的文档数量或满足 Filter条件的文档数量。 | - Delete 支持 Limit，单次删除的速度非常快，同时很好地控制对数据库业务影响。如果您的业务中需要删除海量数据，可以循环通过 Limit 删除配合 affectedCount 返回值来达到效果。",
"腾讯云向量数据库提供数据备份与回档服务，支持预设备份时间点并周期性执行主动备份，同时允许随时手动触发即时备份操作；所有备份均以高可靠快照形式存储，通过实例克隆功能可实现数据回档，恢复业务状态。\n\n> **说明：**\n> \n\n> V2.4 版本起，支持产品化的备份克隆功能，若需要使用该功能，请 [提交工单](https://console.cloud.tencent.com/workorder/*****) 升级实例版本。\n> \n\n\n",
"# 向量数据库备份&恢复技术方案\n\n## 使用场景\n\n主要有以下两个场景：\n\n数据安全是云上数据库产品的生命线，备份回档作为保障数据安全的核心能力，是云上数据库产品的基本功能，提供后也能够为客户误操作导1.\n\n致的数据丢失、数据错乱等现象提供兜底。\n\n部分客户存在克隆数据到新实例的需求。2.\n\n具体到某个场景\n\n在数据库出现性能问题或故障时，离线备份提供了一种安全且可控的方式来排查问题。通过将数据库实例迁移到新的节

```
点，可问题排查与性能验证：1.\n\n以在不影响生产环境的情况下进行详细的性能分析和故障排
查。\n\n定期进行离线备份是防止数据丢失的关键措施。"
          ],
          "token_usage": 744
        },
        "evaluation": {
          "evaluation_result": {
            "is_relevant": false,
            "reason": "结果仅涉及索引修改、删除功能与备份恢复机制，未提及自动删除
配置方法",
            "suggestions": {
              "query_refinement": "需明确向量数据库品牌或添加'自动过期/TTL'等关
键词",
              "recommended_queries": [
                {
                  "query": "向量数据库TTL自动删除配置",
                  "purpose": "查找基于时间的数据自动删除功能"
                },
                {
                  "query": "Milvus/Weaviate自动数据清理配置",
                  "purpose": "针对具体数据库系统查找自动删除方案"
                },
                {
                  "query": "向量数据库数据生命周期管理API",
                  "purpose": "寻找通过编程接口设置自动删除的方法"
                }
              ]
            }
          },
          "token_usage": 894
        }
      },
      {
        "iteration_no": 2,
        "query_processing": {
          "method": "sub_query_split",
          "input_query": "怎么配置向量数据库的数据自动删除",
          "sub_queries": [
            {
              "query": "向量数据库TTL自动删除配置",
              "purpose": "查找基于时间的数据自动删除功能"
            },
            {
```

```
            "query": "Milvus/Weaviate自动数据清理配置",
            "purpose": "针对具体数据库系统查找自动删除方案"
          },
          {
            "query": "向量数据库数据生命周期管理API",
            "purpose": "寻找通过编程接口设置自动删除的方法"
          }
        ],
        "token_usage": 0
      },
      "retrieval": {
        "mode": "dense",
        "sub_query_results": [
          {
            "query": "向量数据库TTL自动删除配置",
            "retrieved_docs": 3,
            "top_docs": [
              "14159081060705607 73",
              "14159081060705607 72",
              "14159081108478730 59"
            ],
            "top_doc_texts": [
              "## 适用场景\n\n某些业务场景下，业务数据的增长很快，并且业务数据的热度随着时间推移会有明显的降低，可以使用 TTL 将热度降低的数据在特定的过期时间时自动清理。例如：在电子商务平台中，用户搜索和浏览行为产生的向量数据，如点击流和用户偏好向量，通常具有很高的时效性。这些数据在短期内对于个性化推荐至关重要，但随着时间的推移，其相关性会迅速降低。在向量数据库集合中，设置TTL，便会自动删除过时的用户行为数据，确保推荐系统能够基于最新的用户互动提供实时、相关的推荐，从而提升用户体验和业务效率。\n\n",
              "## 功能介绍\n\nTTL（Time to Live，生存周期）功能，指数据项从创建或更新后开始，到自动被系统删除的时间期限。在创建数据库集合时，通过为文档设置TTL，数据库能够自动清理过期的数据，从而优化存储空间的使用，保持数据的时效性，并确保查询结果的准确性，特别是在数据具有时效性要求的场景中，如实时推荐系统等。\n\n",
              " | [TTL](https://cloud.tencent.com/document/product/*******) | \nSDK 功能更新 | Python、GO、Java SDK 支持对 Database、Collection 的存在判断。 | - | \n\n\n\n"
            ],
            "avg_score": 353.4836018880208
          },
          {
            "query": "Milvus/Weaviate自动数据清理配置",
            "retrieved_docs": 3,
            "top_docs": [
```

```
                "1415908110847873042",
                "1415908110847873059",
                "1415908205793935427"
              ],
              "top_doc_texts": [
                "## 2025年4月\n\n**版本信息** | **动态名称** | **动态描述** | **相关文档** \n:-: | :-: | :-: | :-: \nV2.4 | 备份回档能力产品化（原后台功能升级） | 用户可通过控制台直观查看实例备份记录及详情。系统在保留自动备份机制的同时，新增手动即时备份触发功能，为关键业务操作提供双重保护。同步推出的实例克隆功能支持通过创建新实例实现数据快速恢复，有效提升数据恢复效率与操作准确性。 | - [备份概述](https://write.woa.com/document/174646208485576704) | - [自动备份](https://write.woa.com/document/**********) | - [手动备份](https://write.woa.com/document/**********) | - [克隆实例](https://write.woa.com/document/**********) | - [查看备份记录](https://write.woa.com/document/**********) | \n",
                " | [TTL](https://cloud.tencent.com/document/product/**********) | \nSDK 功能更新 | Python、GO、Java SDK 支持对 Database、Collection 的存在判断。 | - | \n\n\n\n",
                "|recover_from|-recover_from=172.17.**.**:****,172.17.0.**.**:****,172.17.0.**:****|from \\to 为 Worker 节点的 ip 值，需一一对应；Worker 节点也需要填写，因为有可能从 localdb 启动，此时需要修改 ip 为对应的新值|\n||recover_timestamp|-recover_timestamp=1734593270132|从某个备份集恢复||\n\n"
              ],
              "avg_score": 332.4210611979167
            },
            {
              "query": "向量数据库数据生命周期管理API",
              "retrieved_docs": 3,
              "top_docs": [
                "1415908106506768384",
                "1415908108004134918",
                "1415908205789741090"
              ],
              "top_doc_texts": [
                "腾讯云向量数据库（Tencent Cloud VectorDB）作为一种专门存储和检索向量数据的服务提供给用户， 在高性能、高可用、大规模、低成本、简单易用、稳定可靠等方面体现出显著优势。 \n\n",
                "腾讯云向量数据库（Tencent Cloud VectorDB）在建表时，需指定不同字段的索引类型，数据存储时将会按照指定的索引类型进行索引；在检索时，便会依据索引并使用已选择的相似性计算方法进行匹配，快速高效地获取目标数据。腾讯云向量数据库（Tencent
```

Cloud VectorDB）支持主键索引、向量索引和 Filter 索引，各自适用于不同的查询场景。\n\n",
"# 向量数据库备份&恢复技术方案\n\n## 使用场景\n\n主要有以下两个场景：\n\n数据安全是云上数据库产品的生命线，备份回档作为保障数据安全的核心能力，是云上数据库产品的基本功能，提供后也能够为客户误操作导1.\n\n致的数据丢失、数据错乱等现象提供兜底。\n\n部分客户存在克隆数据到新实例的需求。2.\n\n具体到某个场景\n\n在数据库出现性能问题或故障时，离线备份提供了一种安全且可控的方式来排查问题。通过将数据库实例迁移到新的节点，可问题排查与性能验证：1.\n\n以在不影响生产环境的情况下进行详细的性能分析和故障排查。\n\n定期进行离线备份是防止数据丢失的关键措施。"
],
"avg_score": 348.5239664713542
}
],
"total_retrieved_docs": 9,
"token_usage": 43
},
"rerank": {
"rerank_model": "bge-reranker-large",
"input_docs": 8,
"output_docs": 3,
"top_docs": [
"1415908106070560773",
"1415908106070560772",
"1415908205789741090"
],
"top_doc_texts": [
"## 适用场景\n\n某些业务场景下，业务数据的增长很快，并且业务数据的热度随着时间推移会有明显的降低，可以使用 TTL 将热度降低的数据在特定的过期时间时自动清理。例如：在电子商务平台中，用户搜索和浏览行为产生的向量数据，如点击流和用户偏好向量，通常具有很高的时效性。这些数据在短期内对于个性化推荐至关重要，但随着时间的推移，其相关性会迅速降低。在向量数据库集合中，设置TTL，便会自动删除过时的用户行为数据，确保推荐系统能够基于最新的用户互动提供实时、相关的推荐，从而提升用户体验和业务效率。\n\n",
"## 功能介绍\n\nTTL（Time to Live，生存周期）功能，指数据项从创建或更新后开始，到自动被系统删除的时间期限。在创建数据库集合时，通过为文档设置TTL，数据库能够自动清理过期的数据，从而优化存储空间的使用，保持数据的时效性，并确保查询结果的准确性，特别是在数据具有时效性要求的场景中，如实时推荐系统等。\n\n",
"# 向量数据库备份&恢复技术方案\n\n## 使用场景\n\n主要有以下两个场景：\n\n数据安全是云上数据库产品的生命线，备份回档作为保障数据安全的核心能力，是云上数据库产品的基本功能，提供后也能够为客户误操作导1.\n\n致的数据丢失、数据错乱等现象提供兜底。\n\n部分客户存在克隆数据到新实例的需求。2.\n\n具体到某个场景\n\n在数据库出现性能问题或故障时，离线备份提供了一种安全且可控的方式来排查问题。通过将数据库实例迁移到新的节点，可问题排查与性能验证：1.\n\n以在不影响生产环境的情况下进行详细的性能分析和故障排查。\n\n定期进行离线备份是防止数据丢失的关键措施。"

```
          ],
          "token_usage": 1039
        },
        "evaluation": {
          "evaluation_result": {
            "is_relevant": true,
            "reason": "提到了TTL配置与自动删除相关机制，但未给出具体配置步骤或代码
示例",
            "suggestions": {
              "query_refinement": "需补充具体数据库类型和配置语法",
              "recommended_queries": [
                {
                  "query": "Milvus向量数据库TTL配置步骤",
                  "purpose": "获取该数据库具体操作指南"
                },
                {
                  "query": "Weaviate自动删除数据参数示例",
                  "purpose": "补充不同数据库的实现方式"
                },
                {
                  "query": "elasticsearch向量索引TTL设置",
                  "purpose": "覆盖常见向量数据库方案"
                }
              ]
            },
            "token_usage": 873
          }
        }
      ]
    },
    "token_used": {
      "query_preprocessing_tokens": 351,
      "embedding_tokens": 116,
      "rerank_tokens": 1783,
      "evaluation_tokens": 1767
    }
}
```


参数（一级）	参数（二级）	参数（三级）	参数含义

documents	id	–	检索到的文档主键 ID。
	score	–	相似性得分，表示该文档与（最终优化后的）查询的匹配程度。
	author	–	文档的其他属性字段。
execution_details	original_query	–	原始查询的问题。
	total_iterations	–	多轮迭代总循环次数。
	max_iterations	–	系统配置的最大允许轮数，防止无限循环。 默认为3轮，最大为5轮。
	iterations（迭代详情列表）	iteration_no	本轮的轮次。
		query_processing (查询处理)	○ method：预处理方式 ○ input_query：本轮迭代输入的查询内容。 ○ sub_queries：拆分后的子查询列表。每个子查询对象包含 query (问题文本) 和 purpose (拆分目的)。 ○ token_usage：本阶段消耗的 Token 数量。

		retrieval（检索）	○ mode：检索模式。
			  ○ dense：单路稠密型向量检索。默认为：dense。
			  ○ sparse：单路稀疏型向量检索。
			  ○ hybrid：稠密向量与稀疏向量两路混合检索
			○ sub_query_results：每个子查询的独立检索结果。
			  ○ query：子查询的输入问题。
			  ○ retrieved_docs：检索的文档数量。
			  ○ top_docs：最相关的文档 ID 列表。
			  ○ avg_score：检索到的文档的平均相关性分数。
			○ total_retrieved_docs：所有子查询检索到的文档总数。

			○ token_usage：本阶段消耗的 Token 数量。
		rerank（重排）	○ rerank_model：使用的重排模型名称，"bge-reranker-large" 是一个专门用于对检索结果进行精细排序的模型。 ○ input_docs：送入重排模型的文档数量。 ○ output_docs：重排后输出的文档数量。 ○ top_docs：经过重排模型计算后，全局最相关的文档 ID 列表（按相关性从高到低排序）。 ○ token_usage：本阶段消耗的 Token 数量。
		evaluation（评估）	• evaluation_result：对本轮迭代结果的评估结论。 ○ is_relevant：评估是否通过。false 表示系统自动判断当前结果不足以很好地回答原始问题。 ○ reason：评估不通过的原因。 ○ suggestions：改进建议。系统根据失败原因生成的优化策略。 • token_usage：本阶段消耗的 Token 数量。
token_used	query_preprocessing_tokens	-	在查询处理阶段（如查询改写、拆分）所消耗的 Token。
	embedding_tokens	-	将处理后的查询（Processed Queries）转换为向量表示（Embedding）所消耗的 Token。
	rerank_tokens	-	将检索到的文档输入重排模型进行精细排序所消耗的 Token。通常与输入的文档数量成正比。
	evaluation_tokens	-	检索结果和原始查询输入评估模型进行判断所消耗的 Token。

HTTP API
deep_search

最近更新时间：2025-09-17 16:17:35

接口介绍

本接口（/deepsearch/v1/deep_search）提供端到端的 DeepSearch 检索服务，其完整流程包括：查询预处理、相似性检索、模型重排序（Rerank）以及基于 LLM 的评估与迭代。

> **说明：**
> DeepSearch 每秒最多能处理10,000个 token，涵盖了 Embedding（向量化）、LLM（大语言模型生成）和 Rerank（重排序）三个核心组件的消耗。

Method 与 URL

`POST https://{服务访问地址}/deepsearch/v1/deep_search`

使用示例

连接 DeepSearch，需要配置 DeepSearch ID与访问密钥。获取方式，请参见 查看 DeepSearch 。

- Authorization：配置 DeepSearch 访问密钥。
- x-tdai-service-id：配置 DeepSearch ID。

```
curl -i -k -X POST \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer ****************************************' \
  -H "x-tdai-service-id: tdai-dps-9f3w****" \
  https://deepsearch.tdai.tencentyun.com/deepsearch/v1/deep_search \
  -d '{
    "database": "dps_test_db",
    "collection": "dps_test_coll",
    "search": {
      "query": "怎么配置向量数据库的数据自动删除",
      "query_preprocessing": {
        "method": "sub_query_split"
      },
      "retrieval_config": {
        "mode": "dense",
        "limit": 3,
        "retrieval_params": {
```

```
            "ef": 200
          }
        },
        "rerank": {
          "rerank_model": "bge-reranker-large"
        },
        "evaluate_config": {
          "max_iterations": 3
        },
        "limit": 3,
        "return_execution_details": true
      }
    }'
```

请求参数

参数（一级）	参数（二级）	参数（三级）	是否必选	参数含义	配置方法及要求
database	-	-	是	指定 DeepSearch 检索所关联的向量数据库名称。	在控制台关联的向量数据库。
collection	-	-	是	指定 DeepSearch 检索的向量数据库的集合名称。	所关联向量数据库中已经创建好的向量集合。
search	query	-	是	用户传入的自然语言问题。	数据类型：String。
	query_preprocessing	method	否	指定输入问题的预处理方法。	当前仅支持指定为 sub_query_split，表示系统会使用 LLM 将复杂的主查询语句拆解为多个更易回答的子问题，并行检索后再合并结果，以提高复杂问题的检索效果。
	retrieval_config	embedding_model	否	指定用于检索的向量化模型。 **说明：**	您需根据业务的语言类型、数据维度要求等综合选择合适的模型。取值如下所示： • bge-large-zh-v1.5：适用中文，1024维，推

				如果创建集合时已经绑定了模型，则以集合配置为准，此处的指定会被忽略。	荐使用。 • **bge-base-zh-v1.5**：适用中文，768维。 • **bge-large-zh**：适用中文，1024维。 • **bge-base-zh**：适用中文，768维。 • **m3e-base**：适用中文，768维。 • **e5-large-v2**：适用英文，1024维。 • **text2vec-large-chinese**：适用中文，1024维。 • **multilingual-e5-base**：适用于多种语言类型，768维。 • **BAAI/bge-m3**：适用于多种语言类型，1024维。
		text_field	否	指定向量数据库集合中所检索的文本字段。	以向量数据库集合中配置的文本字段为准。
		mode	否	指定检索模式。	支持如下检索模式。 • dense：单路稠密型向量检索。默认为：dense。 • sparse：单路稀疏型向量检索。 • hybrid：稠密向量与稀疏向量两路混合检索。

		fusion_mode	否	若 **mode** 为 **hybrid**，则需指定混合检索的重排序（rerank）的方式。	支持的重排序（rerank）方式如下所示。
					• **weighted**：基于稠密向量与稀疏向量的加权组合来进行排序。
					  ○ **dense_weight**：指定稠密向量的权重。例如，0.9。
					  ○ **sparse_vector**：指定稀疏向量的权

					重。例如，0.1。 • rrf：一种融合稠密向量与稀疏向量各自相关性评分体系的检索结果排序算法。参数 k 是一个平滑因子，主要用于调节低排名文档对最终结果的影响程度。默认值为60。
		params	否	指定向量数据库集合索引类型所对应的检索参数。	• FLAT：无需指定参数。 • HNSW 类型：需配置参数 ef，指定需要访问向量的数目。取值范围[1,32768]，默认为10。 • IVF 系列：需设置参数 nprobe，指定所需查询的单位数量。取值范围[1,nlist]，其中 nlist 在创建 Collection 时已设置。
		limit	否	指定检索向量数据库返回的最大数量。	• 默认值：最终返回数量 Limit 的三倍。 • 最大值：100。 • 取值：向量数据库检索返回的数量与最终 DeepSearch 检索返回的数量，二者取较大者作为实际取值。
	rerank	rerank_model	否	指定重排序配置模型。	当前仅支持 bge-reranker-large 模型，评估检索出来的结果和原始问题的关联程度去做重排序。
	evaluate_config	max_iterations	否	指定多轮计算的次数。	正整数，最小值为1，最大值为5。
	limit	-	是	最终返回的文档数量。	正整数，取值范围[1,10]。
	filter	-	否	设置检索的过滤条件。	对向量数据库集合的标量字段设置 FIlter 表达式。具体

					信息，请参见 Filter 表达式。
	output_fields	-	否	指定返回结果中应包含哪些字段。	返回字段为所检索的向量数据库集合中的的字段。
	return_execution_details	-	否	指定是否返回检索过程信息。	• false：不返回检索过程。默认 false。 • true：返回检索过程。

响应示例


```
{
  "code": 0,
  "message": "operation success",
  "documents": [
    {
      "id": "1415908106070560773",
      "score": 0.98535156,
      "chunk_num": 2,
      "file_name": "VDB TTL.md",
      "section_num": 2,
      "text": "## 适用场景\n\n某些业务场景下，业务数据的增长很快，并且业务数据的热度随着时间推移会有明显的降低，可以使用 TTL 将热度降低的数据在特定的过期时间时自动清理。例如：在电子商务平台中，用户搜索和浏览行为产生的向量数据，如点击流和用户偏好向量，通常具有很高的时效性。这些数据在短期内对于个性化推荐至关重要，但随着时间的推移，其相关性会迅速降低。在向量数据库集合中，设置TTL，便会自动删除过时的用户行为数据，确保推荐系统能够基于最新的用户互动提供实时、相关的推荐，从而提升用户体验和业务效率。\n\n"
    },
    {
      "id": "1415908106070560772",
      "score": 0.46704102,
      "chunk_num": 1,
      "file_name": "VDB TTL.md",
      "section_num": 1,
      "text": "## 功能介绍\n\nTTL（Time to Live，生存周期）功能，指数据项从创建或更新后开始，到自动被系统删除的时间期限。在创建数据库集合时，通过为文档设置TTL，数据库能够自动清理过期的数据，从而优化存储空间的使用，保持数据的时效性，并确保查询结果的准确性，特别是在数据具有时效性要求的场景中，如实时推荐系统等。\n\n"
    },
    {
      "id": "1415908205789741090",
      "score": 0.07116699,
```

```
      "chunk_num": 1,
      "file_name": "VDB备份&恢复技术方案.pdf",
      "section_num": 1,
      "text": "# 向量数据库备份&恢复技术方案\n\n## 使用场景\n\n主要有以下两个场景：\n\n数据安全是云上数据库产品的生命线，备份回档作为保障数据安全的核心能力，是云上数据库产品的基本功能，提供后也能够为客户误操作导1.\n\n致的数据丢失、数据错乱等现象提供兜底。\n\n部分客户存在克隆数据到新实例的需求。2.\n\n具体到某个场景\n\n在数据库出现性能问题或故障时，离线备份提供了一种安全且可控的方式来排查问题。通过将数据库实例迁移到新的节点，可问题排查与性能验证：1.\n\n以在不影响生产环境的情况下进行详细的性能分析和故障排查。\n\n定期进行离线备份是防止数据丢失的关键措施。"
    }
  ],
  "execution_details": {
    "original_query": "怎么配置向量数据库的数据自动删除",
    "total_iterations": 2,
    "max_iterations": 3,
    "iterations": [
      {
        "iteration_no": 1,
        "query_processing": {
          "method": "sub_query_split",
          "input_query": "怎么配置向量数据库的数据自动删除",
          "sub_queries": [
            {
              "query": "向量数据库支持哪些数据自动删除机制",
              "purpose": "识别实体与核心功能"
            },
            {
              "query": "常见向量数据库（如Milvus、Pinecone、Weaviate）如何配置自动删除策略",
              "purpose": "获取具体产品的配置方法"
            },
            {
              "query": "配置向量数据库自动删除策略时的注意事项与最佳实践",
              "purpose": "补充操作细节与限制条件"
            }
          ],
          "token_usage": 351
        },
        "retrieval": {
          "mode": "dense",
          "sub_query_results": [
            {
```

```
            "query": "向量数据库支持哪些数据自动删除机制",
            "retrieved_docs": 3,
            "top_docs": [
              "1415908110847873059",
              "1415908107303686148",
              "1415908205789741090"
            ],
            "top_doc_texts": [
              " | [TTL]
(https://cloud.tencent.com/document/product/****) | \nSDK 功能更新 |
Python、GO、Java SDK 支持对 Database、Collection 的存在判断。 | - |
\n\n\n\n",
              "腾讯云向量数据库提供数据备份与回档服务，支持预设备份时间点并周期性
执行主动备份，同时允许随时手动触发即时备份操作；所有备份均以高可靠快照形式存储，通过实
例克隆功能可实现数据回档，恢复业务状态。\n\n> **说明：**\n> \n\n>  V2.4 版本起，支
持产品化的备份克隆功能，若需要使用该功能，请［提交工单］
(https://console.cloud.tencent.com/workorder/category) 升级实例版本。\n>
\n\n\n",
              "# 向量数据库备份&恢复技术方案\n\n## 使用场景\n\n主要有以下两个场
景：\n\n数据安全是云上数据库产品的生命线，备份回档作为保障数据安全的核心能力，是云上数
据库产品的基本功能，提供后也能够为客户误操作导1.\n\n致的数据丢失、数据错乱等现象提供兜
底。\n\n部分客户存在克隆数据到新实例的需求。2.\n\n具体到某个场景\n\n在数据库出现性能
问题或故障时，离线备份提供了一种安全且可控的方式来排查问题。通过将数据库实例迁移到新的
节点，可问题排查与性能验证：1.\n\n以在不影响生产环境的情况下进行详细的性能分析和故障排
查。\n\n定期进行离线备份是防止数据丢失的关键措施。"
            ],
            "avg_score": 347.1428629557292
          },
          {
            "query": "常见向量数据库（如Milvus、Pinecone、Weaviate）如何配置
自动删除策略",
            "retrieved_docs": 3,
            "top_docs": [
              "1415908110847873053",
              "1415908205789741090",
              "1415908205789741091"
            ],
            "top_doc_texts": [
              "综合体验提升特性 | - 索引参数修改 modifyVectorIndex，可修改原
数据表的向量索引参数，降低重新建表并导入数据的成本。注意：这里修改索引参数后会自动发生
一次 Rebuild Index。 | - 支持 Count，返回 Collection 所有的文档数量或满足
Filter条件的文档数量。 | - Delete 支持 Limit，单次删除的速度非常快，同时很好地控制
```

对数据库业务影响。如果您的业务中需要删除海量数据，可以循环通过 Limit 删除配合 affectedCount 返回值来达到效果。",
"# 向量数据库备份&恢复技术方案\n\n## 使用场景\n\n主要有以下两个场景：\n\n数据安全是云上数据库产品的生命线，备份回档作为保障数据安全的核心能力，是云上数据库产品的基本功能，提供后也能够为客户误操作导1.\n\n致的数据丢失、数据错乱等现象提供兜底。\n\n部分客户存在克隆数据到新实例的需求。2.\n\n具体到某个场景\n\n在数据库出现性能问题或故障时，离线备份提供了一种安全且可控的方式来排查问题。通过将数据库实例迁移到新的节点，可问题排查与性能验证：1.\n\n以在不影响生产环境的情况下进行详细的性能分析和故障排查。\n\n定期进行离线备份是防止数据丢失的关键措施。",
"通过在非工作时间进行备份，可以最大限度地减少对业务的影响，同时确保数据的定期备份与恢复：2.\n\n一致性和完整性。定期备份不仅提供了数据恢复的保障，还支持在发生灾难性事件时快速恢复业务运行，确保业务的连续性和数据的完整性。\n\n数据库归档：对于不再频繁使用的数据库，或用户欠费的实例，离线备份可以作为一种有效的归档手段。通过备份某个时间点的数据，可以在用户再3.\n\n次启用时快速恢复数据库，确保数据的完整性和一致性。\n\n"
],
"avg_score": 340.3140869140625
},
{
"query": "配置向量数据库自动删除策略时的注意事项与最佳实践",
"retrieved_docs": 3,
"top_docs": [
"1415908205789741090",
"1415908205789741091",
"1415908205793935404"
],
"top_doc_texts": [
"# 向量数据库备份&恢复技术方案\n\n## 使用场景\n\n主要有以下两个场景：\n\n数据安全是云上数据库产品的生命线，备份回档作为保障数据安全的核心能力，是云上数据库产品的基本功能，提供后也能够为客户误操作导1.\n\n致的数据丢失、数据错乱等现象提供兜底。\n\n部分客户存在克隆数据到新实例的需求。2.\n\n具体到某个场景\n\n在数据库出现性能问题或故障时，离线备份提供了一种安全且可控的方式来排查问题。通过将数据库实例迁移到新的节点，可问题排查与性能验证：1.\n\n以在不影响生产环境的情况下进行详细的性能分析和故障排查。\n\n定期进行离线备份是防止数据丢失的关键措施。",
"通过在非工作时间进行备份，可以最大限度地减少对业务的影响，同时确保数据的定期备份与恢复：2.\n\n一致性和完整性。定期备份不仅提供了数据恢复的保障，还支持在发生灾难性事件时快速恢复业务运行，确保业务的连续性和数据的完整性。\n\n数据库归档：对于不再频繁使用的数据库，或用户欠费的实例，离线备份可以作为一种有效的归档手段。通过备份某个时间点的数据，可以在用户再3.\n\n次启用时快速恢复数据库，确保数据的完整性和一致性。\n\n",
"\"4 \n<\n<\n \"logid.index: \" \n<\n<\n logid.index 5 \n<\n<\n \", backup_index: \" \n<\n<\n backup_index;6 node_->leave_recover_mode();7

```
do_recover_events();8    }9 10    // 备份点之后且时间戳在新实例创建前的日志，需要跳过11    if (logid.index > backup_index && timestamp < FLAGS_recover_create_timestamp) {12      continue;"
              ],
              "avg_score": 343.1875406901042
            }
          ],
          "total_retrieved_docs": 9,
          "token_usage": 73
        },
        "rerank": {
          "rerank_model": "bge-reranker-large",
          "input_docs": 6,
          "output_docs": 3,
          "top_docs": [
            "1415908110847873053",
            "1415908107303686148",
            "1415908205789741090"
          ],
          "top_doc_texts": [
            "综合体验提升特性 | - 索引参数修改 modifyVectorIndex，可修改原数据表的向量索引参数，降低重新建表并导入数据的成本。注意：这里修改索引参数后会自动发生一次 Rebuild Index。 | - 支持 Count，返回 Collection 所有的文档数量或满足 Filter条件的文档数量。 | - Delete 支持 Limit，单次删除的速度非常快，同时很好地控制对数据库业务影响。如果您的业务中需要删除海量数据，可以循环通过 Limit 删除配合 affectedCount 返回值来达到效果。",
            "腾讯云向量数据库提供数据备份与回档服务，支持预设备份时间点并周期性执行主动备份，同时允许随时手动触发即时备份操作；所有备份均以高可靠快照形式存储，通过实例克隆功能可实现数据回档，恢复业务状态。\n\n> **说明：**\n> \n\n>  V2.4 版本起，支持产品化的备份克隆功能，若需要使用该功能，请［提交工单］(https://console.cloud.tencent.com/workorder/*****) 升级实例版本。\n> \n\n\n",
            "# 向量数据库备份&恢复技术方案\n\n## 使用场景\n\n主要有以下两个场景：\n\n数据安全是云上数据库产品的生命线，备份回档作为保障数据安全的核心能力，是云上数据库产品的基本功能，提供后也能够为客户误操作导1.\n\n致的数据丢失、数据错乱等现象提供兜底。\n\n部分客户存在克隆数据到新实例的需求。2.\n\n具体到某个场景\n\n在数据库出现性能问题或故障时，离线备份提供了一种安全且可控的方式来排查问题。通过将数据库实例迁移到新的节点，可问题排查与性能验证：1.\n\n以在不影响生产环境的情况下进行详细的性能分析和故障排查。\n\n定期进行离线备份是防止数据丢失的关键措施。"
          ],
          "token_usage": 744
        },
        "evaluation": {
```

```
            "evaluation_result": {
              "is_relevant": false,
              "reason": "结果仅涉及索引修改、删除功能与备份恢复机制，未提及自动删除
配置方法",
              "suggestions": {
                "query_refinement": "需明确向量数据库品牌或添加'自动过期/TTL'等关
键词",
                "recommended_queries": [
                  {
                    "query": "向量数据库TTL自动删除配置",
                    "purpose": "查找基于时间的数据自动删除功能"
                  },
                  {
                    "query": "Milvus/Weaviate自动数据清理配置",
                    "purpose": "针对具体数据库系统查找自动删除方案"
                  },
                  {
                    "query": "向量数据库数据生命周期管理API",
                    "purpose": "寻找通过编程接口设置自动删除的方法"
                  }
                ]
              }
            },
            "token_usage": 894
          }
        },
        {
          "iteration_no": 2,
          "query_processing": {
            "method": "sub_query_split",
            "input_query": "怎么配置向量数据库的数据自动删除",
            "sub_queries": [
              {
                "query": "向量数据库TTL自动删除配置",
                "purpose": "查找基于时间的数据自动删除功能"
              },
              {
                "query": "Milvus/Weaviate自动数据清理配置",
                "purpose": "针对具体数据库系统查找自动删除方案"
              },
              {
                "query": "向量数据库数据生命周期管理API",
                "purpose": "寻找通过编程接口设置自动删除的方法"
```

```
            }
          ],
          "token_usage": 0
        },
        "retrieval": {
          "mode": "dense",
          "sub_query_results": [
            {
              "query": "向量数据库TTL自动删除配置",
              "retrieved_docs": 3,
              "top_docs": [
                "14159081060705607 73",
                "14159081060705607 72",
                "14159081108478730 59"
              ],
              "top_doc_texts": [
                "## 适用场景\n\n某些业务场景下，业务数据的增长很快，并且业务数据的热度随着时间推移会有明显的降低，可以使用 TTL 将热度降低的数据在特定的过期时间时自动清理。例如：在电子商务平台中，用户搜索和浏览行为产生的向量数据，如点击流和用户偏好向量，通常具有很高的时效性。这些数据在短期内对于个性化推荐至关重要，但随着时间的推移，其相关性会迅速降低。在向量数据库集合中，设置TTL，便会自动删除过时的用户行为数据，确保推荐系统能够基于最新的用户互动提供实时、相关的推荐，从而提升用户体验和业务效率。\n\n",
                "## 功能介绍\n\nTTL（Time to Live，生存周期）功能，指数据项从创建或更新后开始，到自动被系统删除的时间期限。在创建数据库集合时，通过为文档设置TTL，数据库能够自动清理过期的数据，从而优化存储空间的使用，保持数据的时效性，并确保查询结果的准确性，特别是在数据具有时效性要求的场景中，如实时推荐系统等。\n\n",
                " | [TTL](https://cloud.tencent.com/document/product/*******) | \nSDK 功能更新 | Python、GO、Java SDK 支持对 Database、Collection 的存在判断。 | - | \n\n\n\n"
              ],
              "avg_score": 353.4836018880208
            },
            {
              "query": "Milvus/Weaviate自动数据清理配置",
              "retrieved_docs": 3,
              "top_docs": [
                "14159081108478730 42",
                "14159081108478730 59",
                "14159082057939354 27"
              ],
              "top_doc_texts": [
```

```
                "## 2025年4月\n\n**版本信息** | **动态名称** | **动态描述** | **相关文档** \n:-: | :-: | :-: | :-: \nV2.4 | 备份回档能力产品化（原后台功能升级） | 用户可通过控制台直观查看实例备份记录及详情。系统在保留自动备份机制的同时，新增手动即时备份触发功能，为关键业务操作提供双重保护。同步推出的实例克隆功能支持通过创建新实例实现数据快速恢复，有效提升数据恢复效率与操作准确性。 | - [备份概述](https://write.woa.com/document/174646208485576704) | - [自动备份](https://write.woa.com/document/**********) | - [手动备份](https://write.woa.com/document/**********) | - [克隆实例](https://write.woa.com/document/**********) | - [查看备份记录](https://write.woa.com/document/**********) | \n",
                " | [TTL](https://cloud.tencent.com/document/product/**********) | \nSDK 功能更新 | Python、GO、Java SDK 支持对 Database、Collection 的存在判断。 | - | \n\n\n\n",
                "|recover_from|-recover_from=172.17.**.**:****,172.17.0.**.**:****,172.17.0.**:****|from \\to 为 Worker 节点的 ip 值，需一一对应；Worker 节点也需要填写，因为有可能从 localdb 启动，此时需要修改 ip 为对应的新值|\n||recover_timestamp|-recover_timestamp=1734593270132|从某个备份集恢复||\n\n"
              ],
              "avg_score": 332.4210611979167
            },
            {
              "query": "向量数据库数据生命周期管理API",
              "retrieved_docs": 3,
              "top_docs": [
                "1415908106506768384",
                "1415908108004134918",
                "1415908205789741090"
              ],
              "top_doc_texts": [
                "腾讯云向量数据库（Tencent Cloud VectorDB）作为一种专门存储和检索向量数据的服务提供给用户， 在高性能、高可用、大规模、低成本、简单易用、稳定可靠等方面体现出显著优势。 \n\n",
                "腾讯云向量数据库（Tencent Cloud VectorDB）在建表时，需指定不同字段的索引类型，数据存储时将会按照指定的索引类型进行索引；在检索时，便会依据索引并使用已选择的相似性计算方法进行匹配，快速高效地获取目标数据。腾讯云向量数据库（Tencent Cloud VectorDB）支持主键索引、向量索引和 Filter 索引，各自适用于不同的查询场景。\n\n",
                "# 向量数据库备份&恢复技术方案\n\n## 使用场景\n\n主要有以下两个场景：\n\n数据安全是云上数据库产品的生命线，备份回档作为保障数据安全的核心能力，是云上数据库产品的基本功能，提供后也能够为客户误操作导1.\n\n致的数据丢失、数据错乱等现象提供兜底。\n\n部分客户存在克隆数据到新实例的需求。2.\n\n具体到某个场景\n\n在数据库出现性能
```

问题或故障时，离线备份提供了一种安全且可控的方式来排查问题。通过将数据库实例迁移到新的节点，可问题排查与性能验证：1.\n\n以在不影响生产环境的情况下进行详细的性能分析和故障排查。\n\n定期进行离线备份是防止数据丢失的关键措施。"
            ],
            "avg_score": 348.5239664713542
          }
        ],
        "total_retrieved_docs": 9,
        "token_usage": 43
      },
      "rerank": {
        "rerank_model": "bge-reranker-large",
        "input_docs": 8,
        "output_docs": 3,
        "top_docs": [
          "1415908106070560773",
          "1415908106070560772",
          "1415908205789741090"
        ],
        "top_doc_texts": [
          "## 适用场景\n\n某些业务场景下，业务数据的增长很快，并且业务数据的热度随着时间推移会有明显的降低，可以使用 TTL 将热度降低的数据在特定的过期时间时自动清理。例如：在电子商务平台中，用户搜索和浏览行为产生的向量数据，如点击流和用户偏好向量，通常具有很高的时效性。这些数据在短期内对于个性化推荐至关重要，但随着时间的推移，其相关性会迅速降低。在向量数据库集合中，设置TTL，便会自动删除过时的用户行为数据，确保推荐系统能够基于最新的用户互动提供实时、相关的推荐，从而提升用户体验和业务效率。\n\n",
          "## 功能介绍\n\nTTL（Time to Live，生存周期）功能，指数据项从创建或更新后开始，到自动被系统删除的时间期限。在创建数据库集合时，通过为文档设置TTL，数据库能够自动清理过期的数据，从而优化存储空间的使用，保持数据的时效性，并确保查询结果的准确性，特别是在数据具有时效性要求的场景中，如实时推荐系统等。\n\n",
          "# 向量数据库备份&恢复技术方案\n\n## 使用场景\n\n主要有以下两个场景：\n\n数据安全是云上数据库产品的生命线，备份回档作为保障数据安全的核心能力，是云上数据库产品的基本功能，提供后也能够为客户误操作导1.\n\n致的数据丢失、数据错乱等现象提供兜底。\n\n部分客户存在克隆数据到新实例的需求。2.\n\n具体到某个场景\n\n在数据库出现性能问题或故障时，离线备份提供了一种安全且可控的方式来排查问题。通过将数据库实例迁移到新的节点，可问题排查与性能验证：1.\n\n以在不影响生产环境的情况下进行详细的性能分析和故障排查。\n\n定期进行离线备份是防止数据丢失的关键措施。"
        ],
        "token_usage": 1039
      },
      "evaluation": {
        "evaluation_result": {
          "is_relevant": true,

```
            "reason": "提到了TTL配置与自动删除相关机制，但未给出具体配置步骤或代码
示例",
            "suggestions": {
              "query_refinement": "需补充具体数据库类型和配置语法",
              "recommended_queries": [
                {
                  "query": "Milvus向量数据库TTL配置步骤",
                  "purpose": "获取该数据库具体操作指南"
                },
                {
                  "query": "Weaviate自动删除数据参数示例",
                  "purpose": "补充不同数据库的实现方式"
                },
                {
                  "query": "elasticsearch向量索引TTL设置",
                  "purpose": "覆盖常见向量数据库方案"
                }
              ]
            }
          },
          "token_usage": 873
        }
      }
    ]
  },
  "token_used": {
    "query_preprocessing_tokens": 351,
    "embedding_tokens": 116,
    "rerank_tokens": 1783,
    "evaluation_tokens": 1767
  }
}
```


参数（一级）	参数（二级）	参数（三级）	参数含义
documents	id	-	检索到的文档主键 ID。
	score	-	相似性得分，表示该文档与（最终优化后的）查询的匹配程度。
	author	-	文档的其他属性字段。

execution_details	original_query	_	原始查询的问题。
	total_iterations	_	多轮迭代总循环次数。
	max_iterations	_	系统配置的最大允许轮数，防止无限循环。 默认为3轮，最大为5轮。
	iterations（迭代详情列表）	iteration_no	本轮的轮次。
		query_processing (查询处理)	○ method：预处理方式 ○ input_query：本轮迭代输入的查询内容。 ○ sub_queries：拆分后的子查询列表。每个子查询对象包含 query (问题文本) 和 purpose (拆分目的)。 ○ token_usage：本阶段消耗的 Token 数量。

		retrieval（检索）	○ mode：检索模式。
			  ○ dense：单路稠密型向量检索。默认为：dense。
			  ○ sparse：单路稀疏型向量检索。
			  ○ hybrid：稠密向量与稀疏向量两路混合检索
			○ sub_query_results：每个子查询的独立检索结果。
			  ○ query：子查询的输入问题。
			  ○ retrieved_docs：检索的文档数量。
			  ○ top_docs：最相关的文档 ID 列表。
			  ○ avg_score：检索到的文档的平均相关性分数。
			○ total_retrieved_docs：所有子查询检索到的文档总数。
			○ token_usage：本阶段消耗的 Token 数量。

		rerank（重排）	○ rerank_model：使用的重排模型名称，"bge-reranker-large" 是一个专门用于对检索结果进行精细排序的模型。 ○ input_docs：送入重排模型的文档数量。 ○ output_docs：重排后输出的文档数量。

			○ top_docs：经过重排模型计算后，全局最相关的文档 ID 列表（按相关性从高到低排序）。 ○ token_usage：本阶段消耗的 Token 数量。
		evaluation（评估）	• evaluation_result：对本轮迭代结果的评估结论。 ○ is_relevant：评估是否通过。false 表示系统自动判断当前结果不足以很好地回答原始问题。 ○ reason：评估不通过的原因。 ○ suggestions：改进建议。系统根据失败原因生成的优化策略。 • token_usage：本阶段消耗的 Token 数量。
token_used	query_preprocessing_tokens	-	在查询处理阶段（如查询改写、拆分）所消耗的 Token。
	embedding_tokens	-	将处理后的查询（Processed Queries）转换为向量表示（Embedding）所消耗的 Token。
	rerank_tokens	-	将检索到的文档输入重排模型进行精细排序所消耗的 Token。通常与输入的文档数量成正比。
	evaluation_tokens	-	检索结果和原始查询输入评估模型进行判断所消耗的 Token。