

本项目API文档

应用 API 接口统一以 JSON 格式返回，并且包含 3 个字段：`code`、`message`、`data`，分别代表业务状态码、提示信息、业务数据。

业务状态码共6种，其中只有 `success` (成功) 代表业务成功，其余均代表业务失败，并且失败会附加相关的信息：`fail` (通用失败)、`not_found` (未找到)、`unauthorized` (未授权)、`forbidden` (无权限) 和 `validate_error` (数据验证失败)。

接口示例：

```
{
  "code": "success",
  "data": {
    "redirect_url": "https://github.com/login/oauth/authorize?
client_id=f69102c6b97d90d69768&redirect_uri=http%3A%2F%2Flocalhost%3A5001%2Foauth%2Fauthorize%2Fgithub&scope=user%3Aemail"
  },
  "message": ""
}
```

带有分页数据的接口会在 `data` 内固定传递 `list` 和 `paginator` 字段，其中 `list` 代表分页后的列表数据，`paginator` 代表分页的数据。

`paginator` 内存在 4 个字段：`current_page` (当前页数)、`page_size` (每页数据条数)、`total_page` (总页数)、`total_record` (总记录条数)，示例数据如下：

```
{
  "code": "success",
  "data": {
    "list": [
      {
        "app_count": 0,
        "created_at": 1713105994,
        "description": "这是专门用来存LLM0ps的知识库",
        "document_count": 13,
        "icon": "https://a.com/111.png",
        "id": "c0759ca8-2d35-4480-83a8-1f41f29d1401",
        "name": "LLM0ps知识库",
        "updated_at": 1713106758,
        "word_count": 8850
      }
    ],
    "paginator": {
      "current_page": 1,
      "page_size": 20,
      "total_page": 1,
      "total_record": 2
    }
  }
}
```

```
    },
    "message": ""
  }
```

如果接口需要授权，需要在 **headers** 中添加 **Authorization**，并附加 **access_token** 即可完成授权登录，示例：

```
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE3MTY0NTY3OTgsImZcyI6ImxsbW9wcyIsInN1YiI6ImM5MDljMWRiLWIyMmUtNGZlNi04OGIyLWIyZTkxZWFiMWE3YiJ9.JDAtdWBBGjXa_XFihfopRe4Cz-RQ9_TAcno9w81tNbE
```

1. 应用模块

1.1 [todo]获取应用基础信息

- **接口说明：** 传递对应的应用 id，获取当前应用的基础信息+配置信息等内容。
- **接口信息：** 授权 + GET:/apps/:app_id
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
app_id	uuid	是	需要获取的应用 id

- **响应参数：**

参数名称	参数类型	参数说明
id	uuid	应用 id
name	string	应用名称
icon	string	应用图标
description	string	应用描述
published_app_config_id	uuid	已发布应用配置 id，如果不存在则为 null
drafted_app_config_id	uuid	草稿应用配置 id，如果不存在则为 null
debug_conversation_id	uuid	调试会话 id，如果不存在则为 null

参数名称	参数类型	参数说明	
published_app_config/drafted_app_config	json	应用配置信息,涵盖草稿配置、已发布配置，如果没有则为 null，两个配置的变量信息一致	
		字段名称	字段说明
		id	配置 id
		model_config	模型配置
		dialog_round	携带上下文轮数，类型为非负整型
		memory_mode	记忆模式，涵盖 long_term_memory(长记忆) 和 none(无记忆) 两种
		status	应用配置配置状态，涵盖 drafted(草稿) 和 published(已发布) 两种
		created_at	应用配置创建时间
		updated_at	应用配置更新时间
updated_at	int	更新时间	
created_at	int	创建时间	

• 响应示例：

```
{
  "code": "success",
  "data": {
    "icon": "https://a.com/111.png",
    "id": "c0759ca8-2d35-4480-83a8-1f41f29d1401",
    "name": "LLM0ps知识库",
    "description": "这是专门用来存LLM0ps的知识库",
    "published_app_config_id": "c0759ca8-2d35-4480-83a8-1f41f29d1401",
    "drafted_app_config_id": "c0759ca8-2d35-4480-83a8-1f41f29d1401",
  }
}
```

```
    "debug_conversation_id": "c0759ca8-2d35-4480-83a8-1f41f29d1401",
    "published_app_config": {
      "id": "c0759ca8-2d35-4480-83a8-1f41f29d1401",
      "model_config": {
        "dialog_round": 3
      },
      "memory_mode": "long_term_memory",
      "status": "published",
      "created_at": 1713105994,
      "updated_at": 1713106758
    },
    "drafted_app_config": {
      "id": "c0759ca8-2d35-4480-83a8-1f41f29d1401",
      "model_config": {
        "dialog_round": 3
      },
      "memory_mode": "long_term_memory",
      "status": "drafted",
      "created_at": 1713105994,
      "updated_at": 1713106758
    },
    "updated_at": 1713106758,
    "created_at": 1713105994
  },
  "message": ""
}
```

1.2 [todo]更新应用草稿配置信息

- **接口说明：**更新应用的草稿配置信息，涵盖：模型配置、长记忆模式等，该接口会查找该应用原始的草稿配置并进行更新，如果没有原始草稿配置，则创建一个新配置作为草稿配置。
- **接口信息:** 授权 + PUT:/apps/:app_id/config
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
app_id	uuid	是	需要更新的应用 id
model_config	json	是	模型配置信息
		字段名称	字段类型
		dialog_round	int
		携带上下文轮数，类型为非负整型	
memory_mode	string	是	记忆模式，涵盖 long_term_memory(长记忆) 和 none(无记忆) 两种

- **请求示例：**

```
{
  "model_config": {
    "dialog_round": 3
  },
  "memory_mode": "long_term_memory"
}
```

• 响应示例:

```
{
  "code": "success",
  "data": {},
  "message": "更新AI应用配置成功"
}
```

1.3 [todo]获取应用调试长记忆

- **接口说明:** 获取应用的调试长记忆, 该接口会返回当前应用的长记忆信息, 如果该应用并没有开启长记忆, 则会抛出错误信息。
- **接口信息:** 授权 + GET:/api/apps/:app_id/long-term-memory
- **请求参数:**

参数名称	参数类型	是否必须	参数说明
app_id	uuid	是	需要获取的应用 id

• 响应参数:

参数名称	参数类型	参数说明
id	uuid	应用 id
name	string	应用名称
icon	string	应用图标
description	string	应用描述
published_app_config_id	uuid	已发布应用配置 id, 如果不存在则为 null
drafted_app_config_id	uuid	草稿应用配置 id, 如果不存在则为 null
debug_conversation_id	uuid	调试会话 id, 如果不存在则为 null

参数名称	参数类型	参数说明	
published_app_config/drafted_app_config	json	应用配置信息,涵盖草稿配置、已发布配置，如果没有则为 null，两个配置的变量信息一致	
		字段名称	字段说明
		id	配置 id
		model_config	模型配置
		dialog_round	携带上下文轮数，类型为非负整型
		memory_mode	记忆模式，涵盖 long_term_memory(长记忆) 和 none(无记忆) 两种
		status	应用配置配置状态，涵盖 drafted(草稿) 和 published(已发布) 两种
		created_at	应用配置创建时间
		updated_at	应用配置更新时间
updated_at	int	更新时间	
created_at	int	创建时间	

• 响应示例：

```
{
  "code": "success",
  "data": {
    "icon": "https://a.com/111.png",
    "id": "c0759ca8-2d35-4480-83a8-1f41f29d1401",
    "name": "LLM0ps知识库",
    "description": "这是专门用来存LLM0ps的知识库",
    "published_app_config_id": "c0759ca8-2d35-4480-83a8-1f41f29d1401",
    "drafted_app_config_id": "c0759ca8-2d35-4480-83a8-1f41f29d1401",
  }
}
```

```
    "debug_conversation_id": "c0759ca8-2d35-4480-83a8-1f41f29d1401",
    "published_app_config": {
      "id": "c0759ca8-2d35-4480-83a8-1f41f29d1401",
      "model_config": {
        "dialog_round": 3
      },
      "memory_mode": "long_term_memory",
      "status": "published",
      "created_at": 1713105994,
      "updated_at": 1713106758
    },
    "drafted_app_config": {
      "id": "c0759ca8-2d35-4480-83a8-1f41f29d1401",
      "model_config": {
        "dialog_round": 3
      },
      "memory_mode": "long_term_memory",
      "status": "drafted",
      "created_at": 1713105994,
      "updated_at": 1713106758
    },
    "updated_at": 1713106758,
    "created_at": 1713105994
  },
  "message": ""
}
```

1.4 [todo]更新应用调试长记忆

- **接口说明：**更新应用的调试长记忆，该接口会更新当前应用的长记忆信息, 如果该应用并没有开启长记忆，则会抛出错误信息。
- **接口信息:** 授权 + `POST:/apps/:app_id/long-term-memory`
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
app_id	uuid	是	需要更新的应用 id (路由参数)
summary	string	是	长记忆的总结信息

- **请求示例：**

```
{
  "summary": "这是一个长记忆的总结信息"
}
```

- **响应示例:**

```
{
  "code": "success",
}
```

```
    "data": {},
    "message": "更新AI应用长记忆成功"
  }
```

1.5 [todo]应用调试对话

- **接口说明：**用于在编排 AI 应用时进行 debug 调试，如果当前应用没有草稿配置，则使用发布配置进行调试，如果有草稿配置则以草稿配置信息进行调试。
- **接口信息:** 授权 + **POST**: /apps/:app_id/debug
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
app_id	uuid	是	需要调试的应用 id
query	string	是	用户输入的 query 信息

- **请求示例：**

```
{
  "query": "你好"
}
```

- **响应参数：**

参数名称	参数类型	参数说明
id	uuid	响应消息的 id
conversation_id	uuid	消息关联会话的 id
query	string	用户输入的 query 信息
answer	string	AI 应用返回的回答信息
answer_tokens	int	生成消耗的token数
response_latency	float	响应延迟时间, 单位为毫秒
created_at	int	消息创建时间
updated_at	int	消息更新时间

- **响应示例:**

```
{
  "code": "success",
  "data": {
    "id": "c0759ca8-2d35-4480-83a8-1f41f29d1401",
    "conversation_id": "c0759ca8-2d35-4480-83a8-1f41f29d1401",
    "query": "你好",
    "answer": "你好, 我是AI助手",
  }
}
```



```
    "answer_tokens": 3,
    "response_latency": 0.1,
    "created_at": 1713105994,
    "updated_at": 1713106758
  },
  "message": ""
}
```

1.6 [todo]获取应用调试历史对话列表

- **接口说明：**用于获取应用调试历史对话列表信息，该接口支持分页，单次最多返回 20 组对话消息，并且分页以时间字段进行降序，接口不会返回软删除对应的数据。
- **接口信息：** 授权 + GET: /apps/:app_id/messages
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
app_id	uuid	是	需要获取的应用 id

- **响应参数：**

参数名称	参数类型	参数说明
id	uuid	响应消息的 id
conversation_id	uuid	消息关联会话的 id
query	string	用户输入的 query 信息
answer	string	AI 应用返回的回答信息
answer_tokens	int	生成消耗的token数
response_latency	float	响应延迟时间, 单位为毫秒
created_at	int	消息创建时间
updated_at	int	消息更新时间

- **响应示例：**

```
{
  "code": "success",
  "data": {
    "list": [
      {
        "id": "c0759ca8-2d35-4480-83a8-1f41f29d1401",
        "conversation_id": "c0759ca8-2d35-4480-83a8-1f41f29d1401",
        "query": "你好",
        "answer": "你好, 我是AI助手",
        "answer_tokens": 3,
        "response_latency": 0.1,
        "created_at": 1713105994,
        "updated_at": 1713106758
      }
    ]
  }
}
```

```
    }
  ],
  "paginator": {
    "current_page": 1,
    "page_size": 20,
    "total_page": 1,
    "total_record": 2
  }
},
"message": ""
}
```

1.7 [todo]删除特定的调试消息

- **接口说明：**用于删除 AI 应用调试对话过程中指定的消息，该删除会在后端执行软删除操作，并且只有当会话 id 和消息 id 都匹配上时，才会删除对应的调试消息。
- **接口信息：** 授权 + `POST:/apps/:app_id/messages/:message_id/delete`
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
app_id	uuid	是	需要删除的应用 id
message_id	uuid	是	需要删除的消息 id

- **请求示例：**

```
{
  "app_id": "c0759ca8-2d35-4480-83a8-1f41f29d1401",
  "message_id": "c0759ca8-2d35-4480-83a8-1f41f29d1401"
}
```

- **响应示例：**

```
{
  "code": "success",
  "data": {},
  "message": "删除AI应用调试消息成功"
}
```

插件模块

2.1 获取内置插件分类表

- **接口说明：**用于获取插件广场页面中所有插件的分类信息，该接口不支持分页，会一次性返回所有信息。
- **接口信息：** 授权 + `GET:/builtin-tools/categories`
- **请求参数：** 无
- **响应参数：**

参数名称	参数类型	参数说明
icon	string	分类图标
category	string	分类英文唯一标识名，例如 search、image、weather 等，用于在前端进行唯一标识判断
name	string	分类名称

• 响应示例：

```
{
  "code": "success",
  "data": [
    {"category": "search", "name": "搜索", "icon": "xxx"},
    {"category": "image", "name": "图片", "icon": "xxx"},
    {"category": "videos", "name": "视频", "icon": "xxx"}
  ],
  "message": ""
}
```

2.2 获取所有内置插件列表信息

- 接口说明：获取 LLMOps 项目中所有内置插件列表信息，该接口会一次性获取所有提供商/工具，无分页，适用于 插件广场 与 AI应用编排 页面。
- 接口信息: 授权 + GET:/builtin-tools
- 请求参数：无
- 响应参数：

参数名称	参数类型	参数说明	
name	string	提供商的名称。	
label	string	提供商对应的标签。	
category	string	提供商对应的分类。	
background	string	提供商 icon 图标的背景。	
tools	list	提供商下的工具列表。	
	name	string	工具名称。
	label	string	工具对应的标签。
	inputs	list	大语言模型调用对应的参数，如果没有则为空列表，该参数信息对应 LLM 工具调用的信息完全一致，不做任何转换。

参数名称	参数类型	参数说明		
	params	name	string	参数名称。
		description	string	参数描述。
		required	bool	是否必填。
		type	string	参数类型。
		list	工具设置对应的参数列表信息，如果没有则为空。	
		name	string	参数名称。
		label	string	参数对应的标签。
		type	string	参数类型,涵盖了 string、number、boolean、select。
		default	Any	参数默认值。如果没有默认值则为 None。
		required	bool	是否必填。
		min	float	参数最小值。如果不需要则为 None
		max	float	参数最大值。如果不需要则为 None
		help	string	参数帮助信息, 如果没有则为 None 或者空字符串。
		options	list	类型为下拉列表时需要配置的选项。
			label	string 选项名称。
			value	string 下拉菜单对应的值
created_at	int	创建/发布该服务商插件的时间戳。		

- 响应示例：

```
{
  "code": "success",
  "data": [
    {
      "name": "google",
      "label": "Google",
      "description": "谷歌服务提供商，涵盖了谷歌搜索等工具。",
      "background": "#E5E7EB",
      "category": "search",
      "tools": [
        {
          "name": "google_serper",
          "label": "谷歌Serper搜索",
          "description": "一个低成本的谷歌搜索API。",
          "inputs": [
            {
              "name": "query",
              "description": "输入应该是搜索查询语句",
              "required": true,
              "type": "string"
            }
          ],
          "params": [],
        }
      ],
      "created_at": 1721460914
    },
    {
      "name": "dalle",
      "label": "DALLE",
      "description": "DALLE是一个文生图工具。",
      "background": "#E5E7EB",
      "category": "image",
      "tools": [
        {
          "name": "dalle3",
          "label": "DALLE-3绘图工具",
          "description": "DALLE-3是一个将文本转换成图片的绘图工具",
          "inputs": [
            {
              "name": "query",
              "description": "输入应该是生成图像的文本提示(prompt)",
              "required": true,
              "type": "string"
            }
          ],
          "params": [
            {
              "name": "size",
              "label": "图片尺寸",
              "type": "select",
              "required": true,
              "help": "",
              "min": null,
            }
          ]
        }
      ]
    }
  ]
}
```

```
        "max": null,
        "options": [
            {"value": "1024x1024", "label": "(方)1024x1024"},
            {"value": "1792x1024", "label": "(横屏)1792x1024"},
            {"value": "1024x1792", "label": "(竖屏)1024x1792"}
        ],
        {
            "name": "style",
            "label": "图片风格",
            "type": "select",
            "required": true,
            "help": "",
            "min": null,
            "max": null,
            "options": [
                {"value": "vivid", "label": "生动"},
                {"value": "natural", "label": "自然"}
            ]
        }
    ],
    "created_at": 1721460914
},
{
    "message": ""
}
```

2.3 获取指定工具的信息

- **接口说明：**根据传递的 提供商名称 + 工具名称 获取对应工具信息详情，该接口用于在 AI 应用编排页面，点击工具设置时进行相应的展示
- **接口信息:** 授权 + GET:/builtin-tools/:provider/tools/:tool
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
provider	string	是	路由参数，服务提供商对应的名字，例如 google、dalle 等。
tool	string	是	路由参数，工具对应的名字，例如 google_serper、dalle3 等。

- **响应参数：**

参数名称	参数类型	参数说明
provider	dict	该工具所属的提供商对应的信息字典。
	name	string 提供商的名称。
	label	string 提供商对应的标签。

参数名称	参数类型	参数说明	
	category	string	提供商对应的分类。
	background	string	提供商 icon 图标的背景。
	description	string	提供商的描述信息。
name	string	工具名称。	
label	string	工具对应的标签。	
description	string	工具的描述信息。	
inputs	list	大语言模型调用对应的参数，如果没有则为空列表，该参数信息对应 LLM 工具调用的信息完全一致，不做任何转换。	
	name	string	参数名称。
	description	string	参数描述。
	required	bool	是否必填。
	type	string	参数类型。
params	list	工具设置对应的参数列表信息，如果没有则为空。	
	name	string	参数名称。
	label	string	参数对应的标签。
	type	string	参数类型,涵盖了 string、number、boolean、select。
	default	Any	参数默认值。如果没有默认值则为 None。
	required	bool	是否必填。
	min	float	参数最小值。如果不需要则为 None
	max	float	参数最大值。如果不需要则为 None
	help	string	参数帮助信息, 如果没有则为 None 或者空字符串。
	options	list	类型为下拉列表时需要配置的选项。

参数名称	参数类型	参数说明
	label	string 选项名称。
	value	string 下拉菜单对应的值
created_at	int	创建工具的创建时间。

• 请求示例：

```
GET: /buildin-tools/google/tools/google_serper
```

• 响应示例：

```
{
  "code": "success",
  "data": {
    "provider": {
      "name": "dalle",
      "label": "DALLE",
      "description": "DALLE是一个文生图工具。",
      "background": "#E5E7EB",
      "category": "image"
    },
    "name": "dalle3",
    "label": "DALLE-3绘图工具",
    "description": "DALLE-3是一个将文本转换成图片的绘图工具",
    "inputs": [
      {
        "name": "query",
        "description": "输入应该是生成图像的文本提示(prompt)",
        "required": true,
        "type": "string"
      }
    ],
    "params": [
      {
        "name": "size",
        "label": "图片尺寸",
        "type": "select",
        "required": true,
        "help": "",
        "min": null,
        "max": null,
        "options": [
          {"value": "1024x1024", "label": "(方)1024x1024"},
          {"value": "1792x1024", "label": "(横屏)1792x1024"},

```



```
        {"value": "1024x1792", "label": "(竖屏)1024x1792"}
      ],
    },
    {
      "name": "style",
      "label": "图片风格",
      "type": "select",
      "required": true,
      "help": "",
      "min": null,
      "max": null,
      "options": [
        {"value": "vivid", "label": "生动"},
        {"value": "natural", "label": "自然"}
      ]
    }
  ],
  "created_at": 15121213465,
},
"message": ""
}
```

2.4 获取内置插件提供商icon

- **接口说明：**根据传递的 服务提供商名称 对应的 icon 信息，返回的是 icon 图片流，例如 svg 图片就是对应的源码，png/jpeg 等就是图片流信息。
- **接口信息：**授权 + GET:/builtin-tools/:provider/icon
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
provider	string	是	路由参数，服务提供商对应的名字，例如 google、dalle 等。

- **请求示例：**

```
GET: /builtin-tools/google/icon
```

- **响应示例：**

```
<svg xmlns="http://www.w3.org/2000/svg" width="24" height="25" viewBox="0 0 24 25" fill="none">
  <path d="M22.501 12.7332C22.501 11.8699 22.4296 11.2399 22.2748 10.5865H12.2153V14.4832H18.12C18.001 15.4515 17.3582 16.9099 15.9296 17.8898L15.9096 18.0203L19.0902 20.435L19.3106 20.4565C21.3343 18.6249 22.501 15.9298 22.501 12.7332" fill="#4285F4"/>
  <path d="M12.214 23C15.1068 23 17.5353 22.0666 19.3092 20.4567L15.9282 17.8899C15.0235 18.5083 13.8092 18.9399 12.214 18.9399C9.38069 18.9399 6.97596 17.1083 6.11874 14.5766L5.99309 14.5871L2.68583 17.0954L2.64258 17.2132C4.40446 20.6433 8.0235 23 12.214 23" fill="#34A853"/>
  <path d="M6.12046 14.5766C5.89428 13.9233 5.76337 13.2233 5.76337 12.5C5.76337 11.7766 5.89428 11.0766 6.10856 10.4233L6.10257 10.2841L2.75386
```

```
7.7355L2.64429 7.78658C1.91814 9.20993 1.50146 10.8083 1.50146 12.5C1.50146
14.1916 1.91814 15.7899 2.64429 17.2132L6.12046 14.5766Z" fill="#FBBC05"/>
  <path d="M12.2141 6.05997C14.2259 6.05997 15.583 6.91163 16.3569
7.62335L19.3807 4.73C17.5236 3.03834 15.1069 2 12.2141 2C8.02353 2 4.40447
4.35665 2.64258 7.78662L6.10686 10.4233C6.97598 7.89166 9.38073 6.05997 12.2141
6.05997Z" fill="#EB4335"/>
</svg>
```

2.5 获取自定义 API 工具提供者列表

- **接口说明：** 获取特定账号创建的 API 插件/自定义工具信息，该接口携带分页，并支持搜索
- **接口信息：** 授权 + GET:/api-tools
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
search_word	string	否	搜索词，用于搜索自定义 API 工具，默认为空代表不搜索任何内容
current_page	int	否	当前页码，默认为 1
page_size	int	否	每页返回的数据量，默认为 20，范围从 1~50

- **响应参数：**

参数名称	参数类型	参数说明
list	list	分页后的 API 插件列表，列表里的每一个元素都是一个字典。
id	uuid	当前工具提供者对应的 id。
name	string	工具提供者的名称。
icon	string	工具提供者的 icon 图标。
tools	list	该工具提供商的所有工具列表信息。
id	uuid	工具的 id。
description	string	工具的描述信息。
name	string	工具的名称,同一个提供者下的工具名称不能重复。
inputs	list	工具的大语言模型输入列表，列表里的每一个元素都是一个字典。

参数名称	参数类型	参数说明	
		name	string 参数名称。
		description	string 参数描述。
		required	bool 是否必填。
		type	string 参数类型。
	description	string	工具提供者的描述信息。
	headers	list	工具提供者的请求头信息。
		key	string 请求头的 key。
		value	string 请求头的 value。
	created_at	int	工具提供者的创建时间。
paginator	dict	分页信息字典。	
	current_page	int	当前页码。
	page_size	int	每页返回的数据量。
	total_page	int	总页数。
	total_record	int	总记录数。

• 请求示例：

```
/api-tools?search_word=&current_page=1&page_size=21
```

• 响应示例：

```
{
  "code": "success",
  "data": {
    "list": [
      {
        "id": "46db30d1-3199-4e79-a0cd-abf12fa6858f",
```

```

    "name": "高德工具包",
    "icon": "https://cdn.imooc.com/gaode.png",
    "description": "查询ip所在地、天气预报、路线规划等高德工具包",
    "tools": [
      {
        "id": "d400cec0-892f-49ab-8f72-821b88c1aaa9",
        "description": "根据传递的城市名获取指定城市的天气预报，例如：
广州。",
        "name": "GetCurrentWeather",
        "inputs": [
          {
            "type": "str",
            "required": true,
            "name": "query",
            "description": "需要搜索的查询语句"
          }
        ]
      }
    ],
    "headers": [
      { "key": "Authorization", "value": "Bearer
QQYnRFerJTSEcrfB89fw8pr0a0bmrch8" }
    ],
    "created_at": 1721460914
  }
],
"paginator": {
  "current_page": 1,
  "page_size": 21,
  "total_page": 1,
  "total_record": 2
},
"message": ""
}
```

2.6 创建自定义 API 工具提供者

- **接口说明：**用于将企业现有的 API 服务接入到 LLMOps 项目创建自定义 API 工具，对于该自定义工具，支持 GET+POST 两种 HTTP 方法的 URL 链接，并且对 OpenAPI-Schema 规范进行简化+调整，以让其更适配 LLMOps 项目
- **接口信息：**授权 + **POST:/api-tools**
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
name	string	是	工具提供者的名称,同一个账号下的工具提供商名字必须唯一，否则容易识别错误，名字的长度范围是 0-30 个字符
icon	string	是	工具提供者的 icon 图标，用于在工具列表中进行展示，icon,类型为图片 URL 字符串

参数名称	参数类型	是否必须	参数说明
openapi_schema	str	是	OpenAPI-Schema 规范的 JSON 字符串,在字符串中涵盖基础信息、服务, 该数据在后端会进行校验, 如果缺少了对应的数据会抛出数据校验错误
headers	list	是	接口附加的请求头信息, 类型为列表, 列表的每个元素都是一个字典, 如果没有请求头信息则传递空列表即可。
	key	string	请求头的 key。
	value	string	请求头的 value。

• 请求示例:

```
{
  "name": "谷歌搜索",
  "icon": "https://cdn.imooc.com/google.png",
  "openapi_schema": "{\n  \"description\": \"这是一个查询对应英文单词字典的工具\",\n  \"server\": \"https://dict.youdao.com\",\n  \"paths\": {\n    \"/suggest\": {\n      \"get\": {\n        \"description\": \"根据传递的单词查询其字典信息\",\n        \"operationId\": \"YoudaoSuggest\",\n        \"parameters\": [\n          {\n            \"name\": \"q\",\n            \"in\": \"query\",\n            \"description\": \"要检索查询的单词, 例如 love/computer\",\n            \"required\": true,\n            \"type\": \"str\"\n          },\n          {\n            \"name\": \"doctype\",\n            \"in\": \"query\",\n            \"description\": \"返回的数据类型, 支持json和xml两种格式, 默认情况下json数据\",\n            \"required\": false,\n            \"type\": \"str\"\n          }\n        ]\n      }\n    }\n  },\n  \"headers\": [\n    {\n      \"key\": \"Authorization\",\n      \"value\": \"Bearer QQYnRFerJTSEcrfB89fw8pr0a0bmrch8\"\n    }\n  ]\n}"
```

• 响应示例:

```
{
  "code": "success",
  "data": {},
  "message": "创建自定义 API 工具提供者成功"
}
```

2.7 删除自定义 API 工具提供者

- **接口说明:** 用于删除特定的自定义 API 插件, 删除对应的 API 插件后, 关联的应用、工作流也无法使用该插件/工具 (在应用对话交流、工作流运行之前都会检测对应的 API 插件是否存在, 如果被删除了, 均会剔除并无法使用)。
- **接口信息:** 授权 + `POST:/api-tools/:provider_id/delete`
- **请求参数:**

参数名称	参数类型	是否必须	参数说明
provider_id	uuid	是	路由参数, 需要删除的 API 工具提供商 id, 类型为 uuid。

- 请求示例:

```
POST:/api-tools/e1baf52a-1be2-4b93-ad62-6fad72f1ec37/delete
```

- 响应示例:

```
{
  "code": "success",
  "data": {},
  "message": "删除自定义 API 工具提供者成功"
}
```

2.8 更新自定义 API 工具提供者

- 接口说明:** 用于更新自定义 API 工具信息, 每次更新的时候, 在后端都会删除原有工具信息, 并记录创建新的工具数据, 在后端使用 provider_id+tool_name 唯一标识进行判断, 更新时如果同个账号出现重名, 则会抛出错误。
- 接口信息:** 授权 + `POST:/api-tools/:provider_id`
- 请求参数:**

参数名称	参数类型	是否必须	参数说明
provider_id	uuid	是	路由参数, 需要更新的 API 工具提供商 id, 类型为 uuid。
name	string	是	工具提供者的名称,同一个账号下的工具提供商名字必须唯一, 否则容易识别错误, 名字的长度范围是 0-30 个字符
icon	string	是	工具提供者的 icon 图标, 用于在工具列表中进行展示, icon,类型为图片 URL 字符串
openapi_schema	str	是	OpenAPI-Schema 规范的 JSON 字符串,在字符串中涵盖基础信息、服务, 该数据在后端会进行校验, 如果缺少了对应的数据会抛出数据校验错误
headers	list	是	接口附加的请求头信息, 类型为列表, 列表的每个元素都是一个字典, 如果没有请求头信息则传递空列表即可。
	key	string	请求头的 key。
	value	string	请求头的 value。

- 请求示例:

```
POST:/api-tools/e1baf52a-1be2-4b93-ad62-6fad72f1ec37
```

```
{
    "name": "",
    "icon": "",
    "openapi_schema": "{\n\"description\":\"\n这是一个查询对应英文单词字典的工具\n\", \"server\": \"https://dict.youdao.com\", \"paths\": {\n\"/suggest\": {\n\"get\": {\n\"description\": \"根据传递的单词查询其字典信息\n\", \"operationId\": \"YoudaoSuggest\", \"parameters\":\n[{\n\"name\": \"q\", \"in\": \"query\", \"description\": \"要检索查询的单词，例如love/computer\", \"required\": true, \"type\": \"str\"},\n{\n\"name\": \"doctype\", \"in\": \"query\", \"description\": \"返回的数据类型，支持json和xml两种格式，默认情况下json数据\", \"required\": false, \"type\": \"str\"}]]}}}\",\n    \"headers\": [\n        {\n            \"key\": \"Authorization\", \"value\": \"Bearer QQYnRFerJTSEcrfB89fw8pr0a0bmrch8\"\n        }\n    ]\n}"
```

- **响应示例：**

```
{
  "code": "success",
  "data": {},
  "message": "更新自定义 API 工具提供者成功"
}
```

2.9 获取指定 API 工具提供者信息

- 接口说明: 根据传递的工具提供者 id 获取对应的工具提供者详细信息。
- 接口信息: 授权 + GET:/api-tools/:provider_id
- 请求参数:

参数名称	参数类型	是否必须	参数说明
provider_id	uuid	是	路由参数，需要获取的 API 工具提供商 id，类型为 uuid。

- **响应参数:**

参数名称	参数类型	参数说明
id	uuid	当前工具提供者对应的 id。
name	string	工具提供者的名称。
icon	string	工具提供者的 icon 图标。
openapi_schema	str	OpenAPI-Schema 规范的 JSON 字符串。
headers	list	工具提供者的请求头信息。
	key	string 请求头的 key。
	value	string 请求头的 value。

参数名称	参数类型	参数说明
created_at	int	工具提供者的创建时间。

- 请求示例：

```
GET:/api-tools/e1baf52a-1be2-4b93-ad62-6fad72f1ec37
```

- 响应示例：

```
{
  "code": "success",
  "data": {
    "id": "46db30d1-3199-4e79-a0cd-abf12fa6858f",
    "name": "高德工具包",
    "icon": "https://cdn.imooc.com/google.png",
    "openapi_schema": "{\n  \"description\": \"这是一个查询对应英文单词字典的工具\",\n  \"server\": \"https://dict.youdao.com\",\n  \"paths\": {\n    \"/suggest\": {\n      \"get\": {\n        \"description\": \"根据传递的单词查询其字典信息\",\n        \"operationId\": \"YoudaoSuggest\",\n        \"parameters\": [\n          {\n            \"name\": \"q\",\n            \"in\": \"query\",\n            \"description\": \"要检索查询的单词，例如love/computer\",\n            \"required\": true,\n            \"type\": \"str\"},\n          {\n            \"name\": \"doctype\",\n            \"in\": \"query\",\n            \"description\": \"返回的数据类型，支持json和xml两种格式，默认情况下json数据\",\n            \"required\": false,\n            \"type\": \"str\"}\n        ]\n      }\n    }\n  },\n  \"created_at\": 1721460914\n},\n  \"message\": \"\"
}
```

2.10 获取指定 API 工具信息

- **接口说明：**根据传递的工具提供者 id + 工具的名称查看自定义 API 插件的相关信息，如果没有找到则返回 404 信息。
- **接口信息：**授权 + GET: /api-tools/:provider_id/tools/:tool_name
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
provider_id	uuid	是	路由参数，需要获取的 API 工具提供商 id，类型为 uuid。
tool_name	string	是	路由参数，需要获取的 API 工具名称。

- **响应参数:**

参数名称	参数类型	参数说明
------	------	------

参数名称	参数类型	参数说明	
id	uuid	工具的 id。	
description	string	工具的描述信息。	
name	string	对应工具的名称（OperationId，操作 id）。	
inputs	list	工具的输入信息，类型为列表。	
	name	string	参数名称。
	description	string	参数描述。
	required	bool	是否必填。
	type	string	参数类型。
provider	dict	该工具所属的提供商对应的信息字典。	
	name	string	提供商的名称。
	id	uuid	提供商的 id。
	icon	string	提供商的 icon 图标。
	description	string	提供商的描述信息。
	headers	list	提供商的请求头信息。
		key	string 请求头的 key。
		value	string 请求头的 value。

• 请求示例：

```
GET:/api-tools/46db30d1-3199-4e79-a0cd-abf12fa6858f/tools/GetCurrentName
```

• 响应示例：

```
{
  "code": "success",
  "data": {
    "id": "d400cec0-892f-49ab-8f72-821b88c1aaa9",
    "name": "GetCurrentWeather",
    "description": "根据传递的城市名获取指定城市的天气预报，例如：广州。",
    "inputs": [
      {
        "type": "str",
        "required": true,
```

```
        "name": "query",
        "description": "需要搜索的查询语句"
    },
    ],
    "provider": {
        "id": "46db30d1-3199-4e79-a0cd-abf12fa6858f",
        "name": "高德工具包",
        "icon": "https://cdn.imooc.com/gaode.png",
        "description": "查询ip所在地、天气预报、路线规划等高德工具包",
        "headers": [
            {
                "key": "Authorization",
                "value": "Bearer QQYnRFerJTSEcrfB89fw8pr0a0bmrch8"
            }
        ],
        "created_at": 1721460914
    },
    },
    "message": ""
}
```

2.11 校验 OpenAPI 字符串是否正确

- **接口说明**: 用于校验传递的 OpenAPI-Schema 规范的 JSON 字符串是否正确，如果正确则返回 success，否则返回错误信息。
- **接口信息**: 授权 + POST:/api-tools/validate-openapi-schema
- **请求参数**:

参数名称	参数类型	是否必须	参数说明
openapi_schema	str	是	需要校验的 openapi-schema 字符串，该字符串的规则符合项目 OpenAPI-Schema 规范，该接口只校验数据是否符合规则，不校验对应的提供商名字、工具名字等是否唯一。

- **请求示例**:

```
{
  "openapi_schema": "{\n  \"description\": \"这是一个查询对应英文单词字典的工具\",\n  \"server\": \"https://dict.youdao.com\",\n  \"paths\": {\n    \"/suggest\": {\n      \"get\": {\n        \"description\": \"根据传递的单词查询其字典信息\",\n        \"operationId\": \"YoudaoSuggest\",\n        \"parameters\": [\n          {\n            \"name\": \"q\",\n            \"in\": \"query\",\n            \"description\": \"要检索查询的单词，例如 love/computer\",\n            \"required\": true,\n            \"type\": \"str\"\n          },\n          {\n            \"name\": \"doctype\",\n            \"in\": \"query\",\n            \"description\": \"返回的数据类型，支持 json 和 xml 两种格式，默认情况下 json 数据\",\n            \"required\": false,\n            \"type\": \"str\"\n          }\n        ]\n      }\n    }\n  }\n}"
```

- **响应示例**:

```
{
  "code": "success",
  "data": {},
  "message": "OpenAPI-Schema 校验成功"
}
```

```
{
  "code": "validate_error",
  "data": {},
  "message": "openapi-schema校验失败, info不能为空"
}
```

文件模块

3.1 将文件上传到腾讯云 COS

- **接口说明：**将文件上传到腾讯云对象存储中，该接口主要用于上传文件，调用接口后返回对应的文件 id、名字、云端位置等信息，主要用于知识库、工具、多模态应用对话。
- **接口信息：**授权 + `POST:/upload-files/file`
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
file	file	是	需要上传的文件，最多支持上传一个文件，最大支持的文件不能超过 15 MB

- **响应参数：**

参数名称	参数类型	参数说明
id	uuid	上传文件的引用 id，类型为 uuid，在知识库模块、应用对话模块会使用该引用文件 id。
account_id	uuid	该文件所归属的账号 id，用于标记是哪个账号上传了该文件。
name	string	原始文件名字。
key	string	云端文件对应的 key 或者路径。
size	int	文件大小，单位为字节。
extension	string	文件扩展名。例如 .md
mime_type	string	文件 mime-type 类型推断。
created_at	string	文件上传时间戳。

- **响应示例：**

```
{
  "code": "success",
```

```
    "data": {
      "id": "46db30d1-3199-4e79-a0cd-abf12fa6858f",
      "account_id": "e1baf52a-1be2-4b93-ad62-6fad72f1ec37",
      "name": "项目API文档.md",
      "key": "2024/05/14/218e5217-ab10-4634-9681-022867955f1b.md",
      "size": 30241,
      "extension": ".md",
      "mime_type": "txt",
      "created_at": 1721460914
    },
    "message": ""
  }
}
```

3.2 将图片上传到腾讯云cos

- **接口说明：** 将图片上传到腾讯云 cos 对象存储中，该接口用于需要上传图片的模块，接口会返回图片的 URL 地址。
- **接口信息：** 授权 + POST:/upload-files/image
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
file	File	是	需要上传的图片文件，支持上传 jpg、jpeg、png、gif，最大不能超过 15 MB。

- **响应参数：**

参数名称	参数类型	参数说明
image_url	string	上传图片的 URL 地址。

- **响应示例：**

```
{
  "code": "success",
  "data": {
    "image_url": "https://cdn.imoooc.com/2024/05/14/218e5217-ab10-4634-9681-022867955f1b.jpg"
  },
  "message": ""
}
```

知识库模块

4.1 获取知识库列表

- **接口说明：** 用于获取当前登录账号的知识库列表信息，该接口支持搜索+分页，传递搜索词为空时代表不搜索。
- **接口信息：** 授权 + GET:/datasets
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
search_word	string	否	搜索词，用于知识库名称模糊搜索，默认为空代表不搜索任何内容。
current_page	int	否	当前页码，默认为 1。
page_size	int	否	每页返回的数据量，默认为 20，范围从 1~50。

• 响应参数：

参数名称	参数类型	参数说明
list	list	分页后的知识库列表信息。
id	uuid	知识库的 id。
name	string	知识库的名称。
description	string	知识库的描述信息。
icon	string	知识库的 icon 图标。
document_count	int	知识库下的文档数量。
character_count	int	该知识库拥有的文档的总字符数
related_app_count	int	该知识库关联的 APP 数量
created_at	int	知识库的创建时间。
updated_at	int	知识库的更新时间。
paginator	dict	分页信息字典。
current_page	int	当前页码。
page_size	int	每页返回的数据量。
total_page	int	总页数。
total_record	int	总记录数。

• 响应示例：

```
{
  "code": "success",
  "data": {
    "list": [
      {
        "id": "46db30d1-3199-4e79-a0cd-abf12fa6858f",
        "name": "慕课LLM0ps知识库",
        "icon": "https://cdn.imooc.com/dataset.png",
        "description": "JavaScript 是一种高级编程语言，用于创建交互式网页和动态效果。JavaScript 在前端开发中扮演着非常重要的角色，因此学习 JavaScript 对于初级前端工程师来说非常必要。JavaScript...",
        "document_count": 10,
        "character_count": 14651,
        "related_app_count": 2,
        "updated_at": 1721460914,
      }
    ]
  }
}
```

```
        "created_at": 1721460914
      },
    ],
    "paginator": {
      "current_page": 1,
      "page_size": 21,
      "total_page": 1,
      "total_record": 2
    }
  },
  "message": ""
}
```

4.2 创建知识库

- **接口说明：**根据传递的信息创建知识库，在同一个账号下，只能创建一个同名的知识库，避免在引用的时候发生误解。
- **接口信息：**授权 + **POST:/datasets**
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
name	string	是	知识库的名称，在同一个账号下，只能创建一个同名的知识库，逻辑和 自定义API插件 一样，知识库的名称最长不能超过 100 个字符。
icon	string	是	知识库的 icon 图标，在前端可以调用 图片上传接口 获取 URL 链接后提交。
description	string	否	可选参数，知识库的描述信息，描述最大不能超过 2000 个字符，当该参数没有填写时，会自动生成类似 Useful for when you want to answer queries about the xxx 的描述，在后端确保该字段永远不会为空

- **请求示例：**

```
{
  "name": "慕课LLMOps知识库",
  "icon": "https://cdn.imoooc.com/dataset.png",
  "description": "JavaScript 是一种高级编程语言，用于创建交互式网页和动态效果。JavaScript 在前端开发中扮演着非常重要的角色，因此学习 JavaScript 对于初级前端工程师来说非常必要。JavaScript..."
}
```

- **响应示例：**

```
{
  "code": "success",
  "data": {},
  "message": "创建知识库成功"
}
```

4.3 更新指定知识库信息

- **接口说明：**该接口主要用于更新指定的知识库信息，涵盖：知识库名称、图标、描述等信息。
- **接口信息：**授权 + `POST:/datasets/:dataset_id`
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
dataset_id	uuid	是	需要更新的知识库 id，类型为 uuid，该参数为路由参数。
name	string	是	知识库的名称，在同一个账号下，只能创建一个同名的知识库，逻辑和 自定义API插件 一样，知识库的名称最长不能超过 100 个字符。
icon	string	是	知识库的 icon 图标，在前端可以调用 图片上传接口 获取 URL 链接后提交。
description	string	否	可选参数，知识库的描述信息，描述最大不能超过 2000 个字符，当该参数没有填写时，会自动生成类似 Useful for when you want to answer queries about the xxx 的描述，在后端确保该字段永远不会为空。

- **请求示例：**

```
POST:/dataset/e1baf52a-1be2-4b93-ad62-6fad72f1ec37

{
  "name": "慕课LLM0ps知识库",
  "icon": "https://cdn.imoooc.com/dataset.jpg",
  "description": "Useful for when you want to answer queries about the 慕课LLM0ps知识库"
}
```

- **响应示例：**

```
{
  "code": "success",
  "data": {},
  "message": "更新知识库成功"
}
```

4.4 删除指定知识库

- **接口说明：**用于删除指定的知识库，删除知识库后，在后端会将关联的应用配置、知识库下的所有文档/文档片段/查询语句也进行一并删除（该接口为耗时接口，将使用异步/消息队列的形式来实现），删除后以前关联的应用将无法引用该知识库。
- **接口信息：**授权 + `POST:/datasets/:dataset_id/delete`
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
dataset_id	uuid	是	需要删除的知识库 id，类型为 uuid，该参数为路由参数。

- 请求示例：

POST:/dataset/e1baf52a-1be2-4b93-ad62-6fad72f1ec37/delete

- 响应示例：

```
{
  "code": "success",
  "data": {},
  "message": "删除知识库成功"
}
```

4.5 获取指定知识库详情

- 接口说明：用于获取指定的知识库详情信息。
- 接口信息: 授权 + GET:/datasets/:dataset_id
- 请求参数：

参数名称	参数类型	是否必须	参数说明
dataset_id	uuid	是	需要获取的知识库 id，类型为 uuid，该参数为路由参数。

- 响应参数：

参数名称	参数类型	参数说明
id	uuid	知识库的 id。
name	string	知识库的名称。
icon	string	知识库的 icon 图标。
description	string	知识库的描述信息。
document_count	int	知识库下的文档数量。
hit_count	int	知识库的文档总命中次数，该知识库下的每一个文档被命中，则次数+1，如果一次查询命中多个同属于同一个文档的片段，该文档的命中次数也只+1，这样计算会更加均衡（计算逻辑会更简单）。
related_app_count	int	知识库关联的 AI/Agent 应用数，类型为整型。
character_count	int	该知识库拥有的文档的总字符数
updated_at	int	知识库的更新时间。
created_at	int	知识库的创建时间。

• 请求示例：

```
GET:/dataset/e1baf52a-1be2-4b93-ad62-6fad72f1ec37
```

• 响应示例

```
{
  "code": "success",
  "data": {
    "id": "46db30d1-3199-4e79-a0cd-abf12fa6858f",
    "name": "慕课LLMOps知识库",
    "icon": "https://cdn.imoooc.com/dataset.png",
    "description": "JavaScript 是一种高级编程语言，用于创建交互式网页和动态效果。JavaScript 在前端开发中扮演着非常重要的角色，因此学习 JavaScript 对于初级前端工程师来说非常必要。JavaScript...",
    "document_count": 10,
    "character_count": 14651,
    "hit_count": 10,
    "related_app_count": 2,
    "updated_at": 1721460914,
    "created_at": 1721460914
  },
  "message": ""
}
```

4.6 指定知识库进行召回测试

- 接口说明：使用指定的知识库进行召回测试，用于检测不同的查询 query 在数据库中的检索效果，每次执行召回测试的时候都会将记录存储到 最近查询列表 中，返回的数据为检索到的 文档片段 列表
- 接口信息: 授权 + POST:/datasets/:dataset_id/hit
- 请求参数：

参数名称	参数类型	是否必须	参数说明
dataset_id	uuid	是	需要召回测试的知识库 id，类型为 uuid，该参数为路由参数。
query	string	是	需要检索的查询语句，该参数最大长度不能超过 200 个字符。
retrieval_strategy	string	是	检索策略，类型为字符串，支持的值为 full_text(全文/关键词检索)、semantic(向量/相似性检索)、hybrid(混合检索)。
k	int	是	最大召回数量，类型为整型，数据范围为 0-10，必填参数。
score	float	是	最小匹配度，类型为浮点型，范围从 0-1，保留 2 位小数，数字越大表示相似度越高。

• 响应参数：

参数名称	参数类型	参数说明
------	------	------

参数名称	参数类型	参数说明	
id	uuid	文档片段的 id。	
document	dict	片段归属的文档信息	
	id	uuid	文档的 id。
	name	string	文档的名称。
	extension	string	文档的扩展名。
	mime_type	string	文档的 mime-type 类型。
dataset_id	uuid	文档片段所属的知识库 id。	
score	float	片段的召回得分，类型为浮点型，数值范围从 0-1，只有检索类型为 相似性检索 的时候才会返回得分，full_text 和 hybrid 这两种检索策略不会计算召回得分（返回结果为 0）。	
positon	int	片段在文档中的位置，类型为整型,数字越小越靠前（自然排序）。	
content	string	片段的内容。	
keywords	list[string]	片段的关键词列表。	
character_count	int	片段的字符串长度，类型为整型。	
token_count	int	片段的 token 数量，类型为整型。	
hit_count	int	文档片段的命中次数，类型为整型。	
enabled	bool	片段是否启用，true 表示启用，false 表示禁用（人为禁用或者程序处理异常、未处理完导致的禁用），只有当 status 为 completed(完成) 时，enabled 才有可能为 true。	
disabled_at	int	片段被人为禁用的时间，为 0 表示没有人为禁用，类型为整型。	
status	string	片段的 状态，涵盖 waiting(等待处理)、indexing(构建索引)、completed(构建完成)、error(错误) 等状态，不同的状态代表不同的处理程度。	
error	string	错误信息，类型为字符串，当后端程序处理出现错误的时候，会记录错误信息。	
created_at	int	片段的创建时间。	
updated_at	int	片段的更新时间。	

• 请求示例：

```
POST:/dataset/e1baf52a-1be2-4b93-ad62-6fad72f1ec37/hit

{
  "query": "LLM0ps",
  "retrieval_strategy": "hybrid",
  "k": 10,
  "score": 0.4
}
```

• 响应示例：

```
{
  "code": "success",
  "data": [
    {
      "id": "b7087193-8e1b-4e88-8ae4-48a0f90a8ad5",
      "document": {
        "id": "6a266b4b-d03b-4066-a4bb-f64abfe23b9d",
        "name": "慕课LLM0ps项目API文档.md",
        "extension": "md",
        "mime_type": "md"
      },
      "dataset_id": "bde70d64-cbcc-47e7-a0f5-b51200b87c7c",
      "position": 1,
      "score": 0.54,
      "content": "为了借助社交产品的流量，让用户主动分享APP中的内容到社交平台来达到拉新和促活的目的，市场上绝大多数APP都有第三方分享的功能，它是内容分发的最有效途径，并且大大降低了企...",
      "keywords": ["社交", "App", "成本", "功能", "内容分发"],
      "character_count": 487,
      "token_count": 407,
      "hit_count": 1,
      "enabled": true,
      "disabled_at": 0,
      "status": "completed",
      "error": "",
      "updated_at": 1726858854,
      "created_at": 1726858854
    }
  ]
}
```

4.7 获取指定知识库最近的查询列表

- **接口说明：**用于获取指定知识库最近的查询列表，该接口会返回最近的 10 条记录，没有分页+搜索功能，返回的数据是按照 created_at 进行倒序，即数据越新越靠前。
- **接口信息：**授权 + GET:/datasets/:dataset_id/queries
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
------	------	------	------

参数名称	参数类型	是否必须	参数说明
dataset_id	uuid	是	需要获取的知识库 id，类型为 uuid，该参数为路由参数。

• 响应参数：

参数名称	参数类型	参数说明
id	uuid	查询的 id。
dataset_id	uuid	查询所属的知识库 id。
query	string	查询的内容。
source	string	查询的来源信息，支持 Hit Testing(召回测试)、App(AI/Agent应用调用)
created_at	int	查询的创建时间。

• 请求示例：

```
GET:/dataset/e1baf52a-1be2-4b93-ad62-6fad72f1ec37/queries
```

• 响应示例：

```
{
  "code": "success",
  "data": [
    {
      "id": "26834b62-8bb4-410b-a626-00aded4892b9",
      "dataset_id": "e1baf52a-1be2-4b93-ad62-6fad72f1ec37",
      "query": "慕课LLMOps是什么?",
      "source": "Hit Testing",
      "created_at": 1726858849
    }
  ],
  "message": ""
}
```

4.8 获取指定知识库的文档列表

- **接口说明：**用于获取指定知识库下的文档列表，该接口支持搜索+分页，如果传递的搜索词为空代表不搜索任何内容，这里的搜索词使用 文档名称 进行模糊匹配。
- **接口信息:** 授权 + GET:/datasets/:dataset_id/documents
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
dataset_id	uuid	是	需要获取的知识库 id，类型为 uuid，该参数为路由参数。
search_word	string	否	搜索词，用于文档名称模糊搜索，默认为空代表不搜索任何内容。
current_page	int	否	当前页码，默认为 1。

参数名称	参数类型	是否必须	参数说明
page_size	int	否	每页返回的数据量，默认为 20，范围从 1~50。

• 响应参数：

参数名称	参数类型	参数说明
list	list	分页后的文档列表信息。
	id	uuid文档的 id。
	name	string文档的名称。
	character_count	int文档的字符数。
	hit_count	int文档的召回命中次数
	position	int文档在知识库中的位置，数字越小越靠前。
	enabled	bool文档是否启用，true 表示启用，false 表示禁用（人为禁用或者程序处理异常、未处理完导致的禁用），只有当 status 为 completed(完成) 时，enabled 才有可能为 true。
	disabled_at	int文档被人为禁用的时间，为 0 表示没有人为禁用，类型为整型。
	status	string文档的状态，涵盖 waiting(等待中)、parsing(解析处理中)、splitting(分割中)、indexing(构建索引中)、completed(构建完成)、error(出错) 等，只有当构建完成时 enabled 才起作用
	error	string错误信息，类型为字符串，当后端程序处理出现错误的时候，会记录错误信息。
	created_at	int文档的创建时间。
	updated_at	int文档的更新时间。
paginator	dict	分页信息字典。
	current_page	int当前页码。
	page_size	int每页返回的数据量。
	total_page	int总页数。
	total_record	int总记录数。

• 请求示例：

```
GET:/datasets/46db30d1-3199-4e79-a0cd-abf12fa6858f/documents
```

• 响应示例：

```
{
  "code": "success",
  "data": {
    "list": [
      {
        "id": "bde70d64-cbcc-47e7-a0f5-b51200b87c7c",
        "name": "LLMOps项目提示词.md",
        "character_count": 4700,
        "hit_count": 0,
        "position": 21,
        "enabled": true,
        "disabled_at": 0,
        "status": "completed",
        "error": "",
        "updated_at": 1726949586,
        "created_at": 1726949586
      }
    ],
    "paginator": {
      "current_page": 1,
      "page_size": 21,
      "total_page": 1,
      "total_record": 2
    }
  },
  "message": ""
}
```

4.9 在指定知识库下新增文档

- **接口说明：**该接口用于在指定的知识库下添加新文档，该接口后端的服务会长时间进行处理，所以在后端服务中，创建好基础的 文档信息 后接口就会响应前端，在前端关闭页面/接口不影响后端逻辑的执行，该接口一次性最多可以上传 10 份文档。
- **接口信息：**授权 + `POST:/datasets/:dataset_id/documents`
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
dataset_id	uuid	是	需要新增文档的知识库 id，类型为 uuid，该参数为路由参数。
upload_file_ids	list[uuid]	是	必填参数，传递需要新增到知识库中的文件id列表，最多支持上传 10 份文件，要想获取 文件id，可以调用 文件上传接口。

参数名称	参数类型	是否必须	参数说明	
process_type	string	是	必填参数，文档处理类型，支持 automatic(自动模式) 和 custom(自定义)。	
rule	dict	否	可选参数，当处理类型为 custom 时为必填参数。	
	pre_process_rules	list	预处理规则列表，涵盖 id 和 enabled 两个属性。	
		id	uuid	预处理标识，支持 remove_extra_space(移除多余空格) 和 remove_url_and_email(移除链接和邮箱)。
		enabled	bool	是否启用该预处理规则，true 表示启用，false 表示禁用。
	segment	dict	片段的处理规则，包含分隔符、片段大小、片段之间的重叠	
		separators	list[str]	片段的分隔符列表，支持正则匹配。
		chunk_size	int	每个片段的最大 Token 数，类型为整型
		chunk_overlap	int	每个片段之间的重叠度，类型为整型

• 响应参数：

参数名称	参数类型	参数说明	
documents	list	返回处理的文档列表，包含文档的基础信息。	
	id	uuid	文档的 id。
	name	string	文档的名称。
	status	string	当前文档的状态，涵盖 waiting(等待中)、parsing(解析处理中)、splitting(分割中)、indexing(构建索引中)、completed(构建完成)、error(出错) 等。

参数名称	参数类型	参数说明
	created_at	int文档的创建时间。
batch	string	当前处理的批次标识，可以通过该批次来获取对应文档的处理信息，批次的格式为 %Y%m%d%H%M%S + 100000-999999随机字符串。

• 请求示例：

```
POST:/datasets/46db30d1-3199-4e79-a0cd-abf12fa6858f/documents

{
  "upload_file_ids": [
    "5537fc7d-22ef-416e-9535-e4faec532c54",
    "fbd81b3f-3d57-42c8-bfaa-c4b564b1306d",
    "c8bd1894-f64b-46d3-9928-54e452669f9e"
  ],
  "process_type": "custom",
  "rule": {
    "pre_proces_rules": [
      {"id": "remove_extra_space", "enabled": true},
      {"id": "remove_url_and_email", "enabled": false},
    ],
    "segment": {
      "separators": ["\n"],
      "chunk_size": 500,
      "chunk_overlap": 50,
    }
  }
}
```

• 响应示例：

```
{
  "code": "success",
  "data": {
    "documents": [
      {
        "id": "c8bd1894-f64b-46d3-9928-54e452669f9e",
        "name": "慕课LLM0ps项目API文档.md",
        "status": "waiting",
        "created_at": 1726858840
      },
      {
        "id": "f16fa6a3-3088-4b6c-9609-85827f45e9d5",
        "name": "慕课LLM0ps课程提示词.md",
        "status": "waiting",
        "created_at": 1726858837
      }
    ]
  }
}
```



```
    "batch": "20240516234156542163"
  },
  "message": ""
}
```

4.10 根据批处理标识获取处理进度

- **接口说明：**根据生成的批处理标识查询当前批次下文档的处理进度。
- **接口信息：**授权 + GET:/datasets/:dataset_id/documents/batch/:batch
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
dataset_id	uuid	是	路由参数，批处理标识关联的知识库id，类型为 uuid。
batch	string	是	批处理标识，类型为字符串，格式为 %Y%m%d%H%M%S + 100000-999999随机字符串。

- **响应参数：**

参数名称	参数类型	参数说明
id	uuid	处理文档的 id
name	string	文档的名称
size	int	文档关联的文件大小，类型为整型，单位为字节。
extension	string	文档的扩展名。
mime_type	string	文档的 mimetype 类型推断
position	int	文档在知识库中的位置，数字越小越靠前。
segment_count	int	文档的片段数量，类型为整型。
completed_segment_count	int	该文档下已经处理完成的文档片段数，类型为整型。
error	string	文档片段如果处理出错，会使用该字段记录，类型为整型。
status	string	文档的状态，涵盖 waiting(等待中)、parsing(解析处理中)、splitting(分割中)、indexing(构建索引中)、completed(构建完成)、error(出错) 等。
processing_started_at	int	开始处理时间，当程序开始处理当前的文档时，会记录该时间，类型为时间戳，下一步为 解析，如果没有完成则值为 0，当前的状态为 parsing，一开始的状态为 waiting。
parsing_completed_at	int	解析完成时间，当程序加载完当前文档的时候记录的时间，类型为时间戳，下一步为 分割，如果没有完成，则值为 0，当前的状态为 splitting，代表下一步需要分割，因为解析已经结束。

参数名称	参数类型	参数说明
splitting_completed_at	int	分割完成时间，当程序使用分割器处理完该文档时记录的时间，类型为时间戳，下一步为 构建(索引构建+关键词构建)，如果没有完成，则值为 0，当前的状态为 indexing，代表下一步需要构建索引，当前分割已结束。
completed_at	int	构建完成时间，当程序使用 Embeddings 文本嵌入模型以及分词器完成向量转换+关键词提取动作的时候记录的时间，类型为时间戳，该阶段为最后一个阶段，如果没有完成，则值为 0，状态为 completed 代表处理完成
stopped_at	int	停止时间，类型为时间戳，文档没有正常处理完成的时候，记录的时间，如果没有停止，则值为 0，当前状态为 error，代表出错了。
created_at	int	文档的创建时间。

• 请求示例：

```
GET://datasets/46db30d1-3199-4e79-a0cd-abf12fa6858f/documents/batch/20240516234156542163
```

• 响应示例：

```
{
  "code": "success",
  "data": [
    {
      "completed_at": 1728227098,
      "completed_segment_count": 60,
      "created_at": 1728198253,
      "error": "",
      "extension": "md",
      "id": "6ddd7c75-a379-41ed-8f93-3e9bd766c850",
      "mime_type": "text/markdown",
      "name": "项目API文档-完整.md",
      "parsing_completed_at": 1728227070,
      "position": 2,
      "processing_started_at": 1728227065,
      "segment_count": 60,
      "size": 94003,
      "splitting_completed_at": 1728227072,
      "status": "completed",
      "stopped_at": 0
    }
  ],
  "message": ""
}
```

4.11 更新指定文档基础名称

- **接口说明：**该接口用于更定特定的文档基础信息（文档的名称），在同一个 知识库 中，文档是可以出现重名的，并且文档更新后的名称长度不能超过100个字符。
- **接口信息:** 授权 + `POST:/datasets/:dataset_id/documents/:document_id/name`
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
dataset_id	uuid	是	需要更新的文档所属的知识库 id，类型为 uuid，该参数为路由参数。
document_id	uuid	是	需要更新的文档 id，类型为 uuid，该参数为路由参数。
name	string	是	需要更新的文档名称，长度不能超过 100 个字符，必填，不能为空，更新的文档名字不必使用对应的扩展，可以任意起名。

- **请求示例：**

```
POST:/datasets/bde70d64-cbcc-47e7-a0f5-b51200b87c7c/documents/6a266b4b-d03b-4066-a4bb-f64abfe23b9d

{
  "name": "基于工具调用的智能体设计与实现.md"
}
```

- **响应示例：**

```
{
  "code": "success",
  "data": {},
  "message": "更新文档名称成功"
}
```

4.12 更改指定文档的启用状态

- **接口说明：**该接口主要用于更改指定文档的启用状态，例如 开启 或 关闭，并且该接口只有在 文档 状态为 completed(完成) 时才可以做相应的更新调整，否则会抛出错误
- **接口信息:** 授权 + `POST:/datasets/:dataset_id/documents/:document_id/enabled`
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
dataset_id	uuid	是	需要更新的文档所属的知识库 id，类型为 uuid，该参数为路由参数。
document_id	uuid	是	需要更新的文档 id，类型为 uuid，该参数为路由参数。
enabled	bool	是	对应文档的状态，true 为开启，false 为关闭，只有当文档处理完成后，才可以修改，文档如果没有执行完毕，将 enabled 修改为 true，会抛出错误信息。

• 请求示例：

```
POST://datasets/bde70d64-cbcc-47e7-a0f5-b51200b87c7c/documents/6a266b4b-d03b-4066-a4bb-f64abfe23b9d/enabled

{
  "enabled": false
}
```

• 响应示例：

```
{
  "code": "success",
  "data": {},
  "message": "更新文档启用状态成功"
}
```

4.13 获取指定文档基础信息

- 接口说明：该接口用于获取指定文档的基础信息，主要用于展示文档片段信息+更新文档信息对应的页面。
- 接口信息: 授权 + GET:/datasets/:dataset_id/documents/:document_id
- 请求参数：

参数名称	参数类型	是否必须	参数说明
dataset_id	uuid	是	需要获取的文档所属的知识库 id，类型为 uuid，该参数为路由参数。
document_id	uuid	是	需要获取的文档 id，类型为 uuid，该参数为路由参数。

• 响应参数：

参数名称	参数类型	参数说明
id	uuid	文档的 id。
dataset_id	uuid	文档所属的知识库 id。
name	string	文档的名称。
segment_count	int	文档的片段数量。
character_count	int	文档的字符数。
hit_count	int	文档的命中次数
position	int	文档在知识库中的位置，数字越小越靠前。
enabled	bool	文档的启用状态，true 表示启用，false 表示已禁用（多种原因禁用）。
disabled_at	int	文档被禁用的时间，为 0 表示没有被禁用，类型为整型。

参数名称	参数类型	参数说明
status	string	文档的状态，涵盖 waiting(等待中)、parsing(解析处理中)、splitting(分割中)、indexing(构建索引中)、completed(构建完成)、error(出错) 等。只有当构建完成时 enabled 才起作用
error	string	错误信息，类型为字符串，当后端程序处理出现错误的时候，会记录错误信息。
created_at	int	文档的创建时间。
updated_at	int	文档的更新时间。

• 请求示例：

```
GET:/datasets/bde70d64-cbcc-47e7-a0f5-b51200b87c7c/documents/6a266b4b-d03b-4066-a4bb-f64abfe23b9d
```

• 响应示例：

```
{
  "code": "success",
  "data": {
    "id": "6196a3bc-2c81-40b8-83a5-25ad837f5a84",
    "dataset_id": "bde70d64-cbcc-47e7-a0f5-b51200b87c7c",
    "name": "基于工具调用的智能体设计与实现.md",
    "segment_count": 15,
    "character_count": 4700,
    "hit_count": 0,
    "position": 21,
    "enabled": true,
    "disabled_at": 0,
    "status": "completed",
    "error": "",
    "updated_at": 1726949586,
    "created_at": 1726949586
  },
  "message": ""
}
```

4.14 删除指定文档信息

- **接口说明：**该接口会根据传递的信息删除文档信息，并删除该文档下的片段信息，同时会将操作同步到向量数据库，在向量数据库中删除归属该文档的所有片段信息，该接口属于耗时接口，所以在后端使用异步任务队列的方式进行操作，完成基础信息的删除（例如文档记录）后，接口即会正常响应前端（删除文档、文档片段、关键词表数据、weaviate数据，同时在删除的时候需要上锁）。
- **接口信息：**授权 + **POST:/datasets/:dataset_id/documents/:document_id/delete**
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
dataset_id	uuid	是	需要删除的文档所属的知识库 id，类型为 uuid，该参数为路由参数。
document_id	uuid	是	需要删除的文档 id，类型为 uuid，该参数为路由参数。

- 请求示例：

```
POST:/datasets/bde70d64-cbcc-47e7-a0f5-b51200b87c7c/documents/6a266b4b-d03b-4066-a4bb-f64abfe23b9d/delete
```

- 响应示例：

```
{
  "code": "success",
  "data": {},
  "message": "删除文档成功"
}
```

4.15 获取指定文档的片段列表

- 接口说明：该接口用于获取指定文档的片段列表，该接口支持分页+搜索，搜索模糊匹配片段内容，当搜索词为空时代表不进行任何检索，该接口只要 dataset_id、document_id 有任意一个不匹配就会抛出对应的错误。
- 接口信息: 授权 + GET:/datasets/:dataset_id/documents/:document_id/segments
- 请求参数：

参数名称	参数类型	是否必须	参数说明
dataset_id	uuid	是	需要获取的文档所属的知识库 id，类型为 uuid，该参数为路由参数。
document_id	uuid	是	需要获取的文档 id，类型为 uuid，该参数为路由参数。
search_word	string	否	搜索词，用于片段内容模糊搜索，默认为空代表不搜索任何内容。
current_page	int	否	当前页码，默认为 1。
page_size	int	否	每页返回的数据量，默认为 20，范围从 1~50。

- 响应参数：

参数名称	参数类型	参数说明
list	list	分页后的文档片段列表信息。
id	uuid	文档片段的 id。
document_id	uuid	文档片段所属的文档 id。
dataset_id	uuid	文档片段所属的知识库 id。

参数名称	参数类型	参数说明
position	int	文档片段在文档中的位置，数字越小越靠前。
content	string	文档片段的内容。
keywords	list[string]	文档片段的关键词列表。
character_count	int	文档片段的字符数。
token_count	int	文档片段的 token 数量。
hit_count	int	文档片段的命中次数。
enabled	bool	文档片段是否启用，true 表示启用，false 表示禁用（人为禁用或者程序处理异常、未处理完导致的禁用），只有当 status 为 completed(完成) 时，enabled 才有可能为 true。
disabled_at	int	文档片段被人为禁用的时间，为 0 表示没有人为禁用，类型为整型。
status	string	文档片段的狀態，涵盖 waiting(等待处理)、indexing(构建索引)、completed(构建完成)、error(错误) 等状态，不同的状态代表不同的处理程度。
error	string	错误信息，类型为字符串，当后端程序处理出现错误的时候，会记录错误信息。
created_at	int	文档片段的创建时间。
updated_at	int	文档片段的更新时间。
paginator	dict	分页信息字典。
current_page	int	当前页码。
page_size	int	每页返回的数据量。
total_page	int	总页数。
total_record	int	总记录数。

• 请求示例：

```
GET:/datasets/bde70d64-cbcc-47e7-a0f5-b51200b87c7c/documents/6a266b4b-d03b-4066-a4bb-f64abfe23b9d/segments
```

• 响应示例：

```
{
  "code": "success",
  "data": {
    "list": [
      {
        "id": "b7087193-8e1b-4e88-8ae4-48a0f90a8ad5",
```

```
        "document_id": "6a266b4b-d03b-4066-a4bb-f64abfe23b9d",
        "dataset_id": "bde70d64-cbcc-47e7-a0f5-b51200b87c7c",
        "position": 1,
        "content": "为了借助社交产品的流量，让用户主动分享APP中的内容到社交平台来达到拉新和促活的目的，市场上绝大多数APP都有第三方分享的功能，它是内容分发的最有效途径，并且大大降低了企...",
        "keywords": ["社交", "App", "成本", "功能", "内容分发"],
        "character_count": 487,
        "token_count": 407,
        "hit_count": 1,
        "enabled": true,
        "disabled_at": 0,
        "status": "completed",
        "error": "",
        "updated_at": 1726858854,
        "created_at": 1726858854
    },
    ],
    "paginator": {
        "current_page": 1,
        "page_size": 21,
        "total_page": 1,
        "total_record": 2
    },
    },
    "message": ""
}
```

4.16 新增文档片段信息

- **接口说明：**该接口主要用于在指定文档下新增 文档片段 信息，添加的片段位置会处于该文档的最后，并且由于每次只能新增一个文档片段，相对来说并不会这么耗时（无需加载分割，直接并行执行 关键词提取+文本转向量），所以该接口是同步的，接口会等待处理完毕后再返回，该接口如果任意一个 dataset_id 或 document_id 出错，都会抛出对应的错误。
- **接口信息：** [授权](#) + [POST:/datasets/:dataset_id/documents/:document_id/segment](#)
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
dataset_id	uuid	是	需要新增文档片段的知识库 id，类型为 uuid，该参数为路由参数。
document_id	uuid	是	需要新增文档片段的文档 id，类型为 uuid，该参数为路由参数。
content	string	是	片段内容，原则上长度不能超过 1000 个 token，类型为字符串
keywords	list[string]	是	片段对应的关键词列表，可选参数，如果该参数没有传，在后端会使用 分词服务 对片段内容进行分词，得到对应的关键词。

- **请求示例：**


```
POST:/datasets/bde70d64-cbcc-47e7-a0f5-b51200b87c7c/documents/6a266b4b-d03b-4066-a4bb-f64abfe23b9d/segments

{
  "content": "## 角色 你是一个拥有10年经验的资深Python工程师，精通Flask, Flask-SQLAlchemy, Postgres，以及其他Python开发工具，能够为用户提出的需求或者提供的代码段生成指定的",
  "keywords": ["Python", "Flask", "工程师"]
}
```

• 响应示例：

```
{
  "code": "success",
  "data": {},
  "message": "新增文档片段成功"
}
```

4.17 删除对应的文档片段信息

- **接口说明：**该接口用于删除对应的文档片段信息，并且该操作会同步到向量数据库中并行删除，并且由于该接口操作的数据比较少，没有耗时操作，所以无需在后端异步执行，执行完成后接口会正常响应。
- **接口信息：**授权 +
POST:/datasets/:dataset_id/documents/:document_id/segments/:segment_id/delete
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
dataset_id	uuid	是	需要删除文档片段的知识库 id，类型为 uuid，该参数为路由参数。
document_id	uuid	是	需要删除文档片段的文档 id，类型为 uuid，该参数为路由参数。
segment_id	uuid	是	需要删除的文档片段 id，类型为 uuid，该参数为路由参数。

• 请求示例：

```
POST:/datasets/bde70d64-cbcc-47e7-a0f5-b51200b87c7c/documents/6a266b4b-d03b-4066-a4bb-f64abfe23b9d/segments/26834b62-8bb4-410b-a626-00aded4892b9/delete
```

• 响应示例：

```
{
  "code": "success",
  "data": {},
  "message": "删除文档片段成功"
}
```

4.18 修改文档片段内容

- **接口说明：**该接口主要用于修改指定的文档片段信息，支持修改 内容、关键词，修改的数据会双向同步到 业务数据库 和 向量数据库，并且由于该接口修改的数据比较少，耗时相对较短，所以在后端无需异步处理，操作完成后接口进行响应。
- **接口信息：** 授权 +
 `POST:/datasets/:dataset_id/documents/:document_id/segments/:segment_id`
- **请求参数：**

参数名称	参数类型	是否必须	参数说明
dataset_id	uuid	是	需要修改文档片段的知识库 id，类型为 uuid，该参数为路由参数。
document_id	uuid	是	需要修改文档片段的文档 id，类型为 uuid，该参数为路由参数。
segment_id	uuid	是	需要修改的文档片段 id，类型为 uuid，该参数为路由参数。
content	string	是	片段内容，原则上长度不能超过 1000 个 token，类型为字符串
keywords	list[string]	是	片段对应的关键词列表，可选参数，如果该参数没有传，在后端会使用 分词服务 对片段内容进行分词，得到对应的关键词。传递了参数则不会调用 分词服务。

- **请求示例：**

```
POST:/datasets/bde70d64-cbcc-47e7-a0f5-b51200b87c7c/documents/6a266b4b-d03b-4066-a4bb-f64abfe23b9d/segments/26834b62-8bb4-410b-a626-00aded4892b9

{
  "content": "## 角色 你是一个拥有10年经验的资深Python工程师，精通Flask, Flask-SQLAlchemy, Postgres，以及其他Python开发工具，能够为用户提出的需求或者提供的代码段生成指定的",
  "keywords": ["Python", "Flask", "工程师"]
}
```

- **响应示例：**

```
{
  "code": "success",
  "data": {},
  "message": "更新文档片段成功"
}
```

4.19 更新文档片段的启用状态

- **接口说明：**该接口主要用于更新文档片段的启用状态，例如 启用 或 禁用，该接口会同步更新 业务数据库 和 向量数据库，并且耗时较短，所以无需执行异步任务

- 接口信息: 授权 +
POST:/datasets/:dataset_id/documents/:document_id/segments/:segment_id/enabled
- 请求参数:

参数名称	参数类型	是否必须	参数说明
dataset_id	uuid	是	需要更新文档片段的知识库 id，类型为 uuid，该参数为路由参数。
document_id	uuid	是	需要更新文档片段的文档 id，类型为 uuid，该参数为路由参数。
segment_id	uuid	是	需要更新的文档片段 id，类型为 uuid，该参数为路由参数。
enabled	bool	是	文档片段的启用状态，true 为开启，false 为关闭，只有当文档片段状态为 completed(完成) 时才可以修改，文档片段如果没有执行完毕，将 enabled 修改为 true，会抛出错误信息。

- 请求示例:

```
POST:/datasets/bde70d64-cbcc-47e7-a0f5-b51200b87c7c/documents/6a266b4b-d03b-4066-a4bb-f64abfe23b9d/segments/26834b62-8bb4-410b-a626-00aded4892b9

{
  "enabled": false
}
```

- 响应示例:

```
{
  "code": "success",
  "data": {},
  "message": "更新文档片段启用状态成功"
}
```

4.20 查询文档片段信息

- 接口说明: 该接口主要用于查询对应的文档片段信息，涵盖了片段内容、关键词、状态、字符数、召回次数、创建时间等内容，并且要求传递的 dataset_id、document_id、segment_id 保持一致，否则会抛出错误
- 接口信息: 授权 +
GET:/datasets/:dataset_id/documents/:document_id/segments/:segment_id
- 请求参数:

参数名称	参数类型	是否必须	参数说明
dataset_id	uuid	是	需要查询文档片段的知识库 id，类型为 uuid，该参数为路由参数。
document_id	uuid	是	需要查询文档片段的文档 id，类型为 uuid，该参数为路由参数。
segment_id	uuid	是	需要查询的文档片段 id，类型为 uuid，该参数为路由参数。

• 响应参数：

参数名称	参数类型	参数说明
id	uuid	文档片段的 id。
document_id	uuid	文档片段所属的文档 id。
dataset_id	uuid	文档片段所属的知识库 id。
position	int	文档片段在文档中的位置，数字越小越靠前。
content	string	文档片段的内容。
keywords	list[string]	文档片段的关键词列表。
character_count	int	文档片段的字符数。
token_count	int	文档片段的 token 数量。
hit_count	int	文档片段的命中次数。
hash	string	文档片段的 hash 值，用于确定唯一的片段内容。
enabled	bool	片段是否启用，true 表示启用，false 表示禁用（人为禁用或者程序处理异常、未处理完导致的禁用），只有当 status 为 completed(完成) 时，enabled 才有可能为 true。
disabled_at	int	片段被禁用的时间，为 0 表示没有被禁用，类型为整型。
status	string	片段的狀態，涵蓋 waiting(等待处理)、indexing(构建索引)、completed(构建完成)、error(错误) 等状态，不同的状态代表不同的处理程度。
error	string	错误信息，类型为字符串，当后端程序处理出现错误的时候，会记录错误信息。
created_at	int	片段的创建时间。
updated_at	int	片段的更新时间。

• 请求示例：

```
GET:/datasets/bde70d64-cbcc-47e7-a0f5-b51200b87c7c/documents/6a266b4b-d03b-4066-a4bb-f64abfe23b9d/segments/26834b62-8bb4-410b-a626-00aded4892b9
```

• 响应示例：

```
{
  "code": "success",
  "data": {
    "id": "b7087193-8e1b-4e88-8ae4-48a0f90a8ad5",
    "document_id": "6a266b4b-d03b-4066-a4bb-f64abfe23b9d",
    "dataset_id": "bde70d64-cbcc-47e7-a0f5-b51200b87c7c",
    "position": 1,
    "content": "为了借助社交产品的流量，让用户主动分享APP中的内容到社交平台来达到拉新和促活的目的，市场上绝大多数APP都有第三方分享的功能，它是内容分发的最有效途径，并且大大降低了
```

```
企...",
  "keywords": ["社交", "App", "成本", "功能", "内容分发"],
  "character_count": 487,
  "token_count": 407,
  "hit_count": 1,
  "hash":
"6d867db429d26ea426d6b67a88fce43e74760d039e9e2925f0083b7acb1f066a",
  "enabled": true,
  "disabled_at": 0,
  "status": "completed",
  "error": "",
  "updated_at": 1726858854,
  "created_at": 1726858854
},
"message": ""
}
```