

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.eyJleHAiOjE3MTY0NTY3OTgsImZcyI6ImxsbW9wcyIsInN1YiI6ImM5MDljMWRiLWlyMmUtNGZlNi04OGIyLWlyZTkxZWFiMWE3YiJ9.JDAtdWBBGiXa_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	字段类型	字段说明	
	id	uuid	配置 id	
	model_config	json	模型配置	
		dialog_round	int	携带上下文轮数，类型为非负整型
	memory_mode	string	记忆模式，涵盖 long_term_memory(长记忆) 和 none(无记忆) 两种	
	status	string	应用配置配置状态，涵盖 drafted(草稿) 和 published(已发布) 两种	
	created_at	int	应用配置创建时间	
	updated_at	int	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	字段类型
		id	uuid
		model_config	json
		dialog_round	int
		memory_mode	string
		status	string
		created_at	int
		updated_at	int
updated_at	int	更新时间	
created_at	int	创建时间	

携带上下文轮数，类型为非负整型

记忆模式，涵盖 `long_term_memory`(长记忆) 和 `none`(无记忆) 两种

应用配置配置状态，涵盖 `drafted`(草稿) 和 `published`(已发布) 两种

• 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 [todo]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 [todo]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		
		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。
			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": "百度",
      "label": "baidu",
      "category": "search",
      "background": "xxx",
      "tools": [
        {
          "name": "百度搜索",
          "label": "baidu-search",
          "inputs": [
            {
              "name": "query",
              "description": "搜索关键词",
              "required": true,
              "type": "string"
            }
          ],
          "params": [
            {
              "name": "query",
              "label": "搜索关键词",
              "type": "string",
              "default": "",
              "required": true,
              "min": null,
              "max": null,
              "help": "请输入搜索关键词",
              "options": []
            }
          ]
        }
      ],
      "created_at": 1713105994
    }
  ],
  "message": ""
}
```