工作流 - OpenAPI 文档

目录

简介

在使用工作流管理 V2 的时候，有些任务节点需要用到其他模块或外部系统的功能，需要用 API 调用的方式来运行这些功能，本文档整理了 AIHub 一些常用功能模块的 API 文档，并提供一些示例和说明。

API 文档

用户系统

登录

通过登录接口可以获取 API 调用所需的 Token。

若处于工作流节点中，可直接读取环境变量 AI_HUB_TOKEN 以获取 Token，该 Token 由系统自动注入至工作流节点的容器中。

Method	POST	
URL	http://<IP>:<PORT>/api/v1/auth/login	
Headers		
Path Params		
Query Params		

Request	JSON
	{
	  "username": "",
	  "password": ""
	}
		• username：用户名
		• password：密码

Response	JSON
	{
	  "code": 0,
		• code：错误码
		  ◦ 0：成功

	<pre>"msg": "",
	"data": {
	"id": 1,
	"token": ""
	}
	}</pre>
		◦ 非 0：失败
		• msg：错误消息，code 非 0 时可能有。
		• data
		◦ id：用户 ID。
		◦ token：JWT Token。

数据仓库

创建数据查询任务

Method	POST	
URL	http://<IP>:<PORT>/data-warehouse/api/v1/searches	
Headers	Authorization: Bearer <JWT>	
Path Params		
Query Params		

Request	<pre>JSON
	{
	"type": 1,
	"name": "",
	"description": "",
	"sql": "",
	"feature_lib_id": 0,
	"images": [
	{
	"url": "",
	"box":[1,1,1,1]
	}
	],</pre>
		• type：任务类型
		◦ 1：SQL 搜索
		◦ 2：图搜图
		◦ 3：文搜图
		• name：任务名称
		• description：任务描述，可选。
		• sql：SQL 查询语句，type 为 1 时需要

	<pre>"keywords": "",
	"top_k": 0
	}</pre>
		• feature_lib_id：特征库 ID，type 为 2 或 3 时需要。
		• images：图片列表，type 为 2 时需要
		◦ url：图片 HTTP 链接
		◦ box：框，格式为[x1,y1,x2,y2]
		• keywords：关键词，type 为 3 时需要
		• top_k：type 为 2 或 3 时需要。

Response	<pre>JSON
	{
	"code": 0,
	"msg": "",
	"data": {
	"id": 1
	}
	}</pre>
		• code：错误码
		◦ 0：成功
		◦ 非 0：失败
		• msg：错误消息，code 非 0 时可能有。
		• data
		◦ id：新创建的查询任务 ID。

获取数据查询任务信息

Method	GET	
URL	http://<IP>:<PORT>/data-warehouse/api/v1/searches/{id}	
Headers	Authorization: Bearer <JWT>	
Path	• id, 查询任务 ID	

Params		
Query Params		
Request		

Response	JSON
	{
	"code": 0,
	"msg": "",
	"data": {
	"id": 1,
	"type": 1,
	"name": "",
	"description": "",
	"sql": "",
	"feature_lib_id": 1,
	"feature_lib_name": "",
	"images": [
	{
	"url": "",
	"box": [1,1,1,1]
	}
	],
	"keywords": "",
	"top_k": 1,
	"status": 1,
	"message": "",
	"result_url": "",
	"results": [
	{
	"image_url": "",
	"label": "",
	"score": 1,
	"box": [1,1,1,1]
	}
	],
	"created_at": 1,
	"username": ""
	}
	}
		• code：错误码
		◦ 0：成功
		◦ 非 0：失败
		• msg：错误消息，code 非 0 时可能有。
		• data
		◦ id：查询任务 ID。
		◦ type：任务类型
		. 1：SQL 搜索
		. 2：图搜图
		. 3：文搜图
		◦ name：任务名称
		◦ description：任务描述，可选。
		◦ sql：SQL 查询语句，type 为 1 时有
		◦ feature_lib_id：特征库 ID
		◦ feature_lib_name：特征库名称
		◦ images：图片列表，type 为 2 时有
		. url：图片 HTTP 链接
		. box：框，格式

为[x1,y1,x2,y2]

- keywords：关键词，type 为 3 时有
- top_k：type 为 2 或 3 时有。
- status：任务状态
  - 1：等待中
  - 2：运行中
  - 3：成功
  - 4：失败
- message：错误消息
- result_url：搜索结果 csv 文件的 HTTP URL，任务状态为成功时才有。
- results：文搜图或图搜图的结果。
  - image_url：图片 URL
  - label：目标标签
  - score：相似度
  - box：目标框
- created_at：创建时间毫秒时间戳
- username：创建人

数据集管理

创建数据集

Method	POST	
URL	http://<IP>:<PORT>/dataset-mng/api/v2/datasets	
Headers	Authorization: Bearer <JWT>	
Path Params		
Query Params		

Request	JSON
	{
	"name": "",
	"description": "",
	"tags": [1,2,3],
	"cover_img": "",
	"create_by": 1
	}
		• name：数据集名称
		• description：数据集描述
		• tags：标签 ID 列表
		• cover_img：封面图片 s3 路径
		• create_by：创建人用户 ID

Response	JSON
	{
	"code": 0,
	"msg": "",
	"data": {
	"id": 1
	}
	}
		• code：错误码
		◦ 0：成功
		◦ 非 0：失败
		• msg：错误消息，code 非 0 时可能有。
		• data
		◦ id：新创建的数据集 ID。

上传并创建数据集版本

Method	POST	
URL	http://<IP>:<PORT>/dataset-mng/api/v2/dataset-versions-upload	
Headers	Authorization: Bearer <JWT>	
Path Params		
Query Params		

Request	JSON
	{
	"upload_type": 1,
	"upload_path": "",
	"description": "",
	"parent_version_id": 1,
	"dataset_id": 1
	}
		• upload_type：上传类型
		◦ 1：zip 文件
		◦ 2：s3 目录
		◦ 3：主机目录
		• upload_path：上传路径
		• description：描述，可选
		• parent_version_id：父版本 ID，可选
		• dataset_id：所属的数据集 ID

Response	JSON
	{
	"code": 0,
	"msg": "",
	"data": {
	"id": 1
	}
	}
		• code：错误码
		◦ 0：成功
		◦ 非 0：失败
		• msg：错误消息，code 非 0 时可能有。
		• data
		◦ id：新创建的数

		据集版本 ID。

获取数据集版本信息

Method	GET	
URL	http://<IP>:<PORT>/dataset-mng/api/v2/dataset-versions-detail	
Headers		
Path Params		
Query Params	• name	数据集版本名称，格式为<数据集名称>/V<版本号>，例如 mnist/V1
Request		

Response	JSON
	{
	"code": 0,
	"msg": "",
	"data": {
	"id": 1,
	"version": 1,
	"dataset_id": 1,
	"upload_path": "",
	"upload_type": 1,
	"parent_version_id": 0,
	"description": "",
	"status": 1,
	"message": "",
	"created_at": 1,
	"user_id": 1,
	"data_size": 1,
	"data_count": 1,
	"parquet_index_path": "1"
	}
		• code：错误码
		◦ 0：成功
		◦ 非 0：失败
		• msg：错误消息，code 非 0 时可能有。
		• data
		◦ id：数据集版本 ID
		◦ version：数据集版本号
		◦ dataset_id：数据集 ID
		◦ upload_path：上传路径
		◦ upload_type：

	} 	上传类型 . 1：zip 文件 . 2：s3 目录 . 3：主机目录 ∘ parent_version_id：父版本 ID ∘ description：描述 ∘ status：状态 . 1：等待中 . 2：创建中 . 3：成功 . 4：失败 ∘ message：错误消息 ∘ created_at：创建时间，毫秒时间戳 ∘ user_id：创建人用户 ID ∘ data_size：数据大小 ∘ data_count：文件数量 ∘ parquet_index_path：parquet 索引文件 URL

工作流管理 V2

创建运行实例

Method	POST	
URL	http://<IP>:<PORT>/workflow-center/api/v1/runs	
Headers	Authorization: Bearer <JWT>	
Path Params		
Query Params		

Request	JSON
	{
	"name": "",
	"description": "",
	"pipeline_id": 1,
	"pipeline_version_id": 1,
	"params": [
	{"key": "x", "value": "y"}
	]
	}
		• name：名称
		• description：描述，可选
		• pipeline_id：工作流 ID
		• pipeline_version_id：工作流版本 ID
		• params：运行参数，工作流版本第一个节点的输入参数。

Response	JSON
	{
	"code": 0,
	"msg": "",
	"data": {
	"id": 1
	}
	}
		• code：错误码
		◦ 0：成功
		◦ 非 0：失败
		• msg：错误消息，code 非 0 时可能有。
		• data
		◦ id：新创建的运行实例 ID。

获取运行实例信息

Method	POST	
URL	http://<IP>:<PORT>/workflow-center/api/v1/runs/{id}	
Headers	Authorization: Bearer <JWT>	
Path Params	• id	运行实例 ID
Query Params		
Request		

Response	JSON
	{
	"code": 0,
	"msg": "",
	"data": {
	"id": 1,
	"name": "",
	"description": "",
	"duration": 1,
	"pipeline_id": 1,
	"pipeline_name": "",
	"pipeline_version_id": 1,
	"pipeline_version_name": "",
	"started_at": 1,
	"finished_at": 1,
	"created_at": 1,
	"status": 1,
	"task_nodes": [
	{
	"name": "",
	"params": {
	"input": "null",
	"output": "null"
		• code：错误码
		◦ 0：成功
		◦ 非 0：失败
		• msg：错误消息，code 非 0 时可能有。
		• data
		◦ id：运行实例 ID。
		◦ name：名称
		◦ description：描述
		◦ duration：运行时长，单位秒
		◦ pipeline_id：工作流 ID
		◦ pipeline_name：工作流名称
		◦ pipeline_version_id：工作流版本 ID
		◦ pipeline_versio

	<pre>},
	"message": "",
	"status": 1,
	"pod_name": "",
	"started_at": 1,
	"finished_at": 1,
	"dependencies": []
	}
	],
	"params": []
	}
	}
		n_name：工作流版本名称
		◦ started_at：开始时间，毫秒时间戳
		◦ finished_at：结束时间，毫秒时间戳
		◦ created_at：创建时间，毫秒时间戳
		◦ status：状态
		. 1：等待中
		. 2：运行中
		. 3：成功
		. 4：失败
		. 5：被终止
		◦ task_nodes：任务节点树
		◦ params：运行参数。

模型训练

创建训练任务

Method	POST	
URL	http://<IP>:<PORT>/model-training-platform/api/v1/trainings	
Headers	Authorization: Bearer <JWT>	
Path Params		

Query Params		

Request	JSON
	{
	"framework": "",
	"name": "",
	"description": "",
	"command": "",
	"image": "",
	"virtual_cluster_id": 1,
	"sku_cnt": 1,
	"enable_ssh": false,
	"envs": [
	{"key": "x", "value": "y"}
	],
	"storage_ids": [1,2],
	"instances": 1,
	"use_ib_network": false,
	"always_pull_image": false,
	"shm": 1,
	"category_id": 1,
	"project_id": 1,
	"estimate_run_time": 1,
	"is_vip": false,
	"preempt_policy": 1,
	"vip_node_names": [""]
	}
		• framework：训练框架
		◦ MpiJob
		◦ PyTorchJob
		• name：名称
		• description：描述，可选
		• image：镜像
		• virtual_cluster_id：虚拟集群 ID，可以用接口工作流 - OpenAPI 文档获取。
		• sku_cnt：sku 数量
		• enable_ssh：是否启用 SSH 登录
		• envs：环境变量
		• storage_ids：存储 ID 列表，可以通过接口工作流 - OpenAPI 文档获取。
		• instances：实例数量
		• use_ib_network：是否启用 IB 网络
		• always_pull_image：是否总是拉取镜像
		• shm：共享内存大小，单位 GiB
		• category_id：分类 ID
		◦ 1：训练

		◦ 2：推理 ◦ 3：调试 ◦ 4：特别 • project_id：研发项目 ID，可以在 AIHub 左侧菜单【研发项目】查看 • estimate_run_time：预估运行时长，单位秒 • is_vip：是否是 VIP 任务 • preempt_policy：VIP 任务抢占策略 ◦ 1：等待现有低优先级任务完成 ◦ 2：杀死现有低优先级任务 • vip_node_names：为 VIP 任务预留的节点。

Response	JSON
	{
	"code": 0,
	"msg": "",
	"data": {
	"id": 1
	}
	}
		• code：错误码
		◦ 0：成功
		◦ 非 0：失败
		• msg：错误消息，code 非 0 时可能有。
		• data
		◦ id：新创建的训练任务 ID。

获取训练任务信息

Method	GET	

URL	http://<IP>:<PORT>/model-training-platform/api/v1/trainings/{id}	
Headers	Authorization: Bearer <JWT>	
Path Params	• id	训练任务 ID
Query Params		
Request		

Response	JSON
	{
	"code": 0,
	"msg": "",
	"data": {
	"id": 1,
	"framework": "1",
	"name": "",
	"description": "",
	"command": "",
	"image": "",
	"virtual_cluster": {
	"id": 1,
	"name": "",
	"gpu_type": "",
	"label": "",
	"sku": {
	"cpu": 1,
	"gpu": 1,
	"memory": 1
	}
	},
	"sku_cnt": 1,
	"enable_ssh": false,
	"envs": null,
	"storages": null,
	"instances": 1,
	"created_at": 1,
	"username": "",
	"user_id": 1,
		• code：错误码
		◦ 0：成功
		◦ 非 0：失败
		• msg：错误消息，code 非 0 时可能有。
		• data
		◦ id：新创建的运行实例 ID。
		◦ framework：训练框架
		▪ MpiJob
		▪ PyTorchJob
		◦ name：名称
		◦ description：描述
		◦ image：镜像
		◦ virtual_cluster：虚拟集群
		◦ sku_cnt：sku 数量
		◦ enable_ssh：是否启用 SSH 登录

```
        "namespace": "",
        "res_name": "",
        "status": 1,
        "use_ib_network":
false,
        "always_pull_image":
true,
        "shm": 0,
        "category": {
            "id": 1,
            "name": ""
        },
        "project": {
            "id": 1,
            "name": "",
            "description": ""
        },
        "avg_gpu_util": 0,
        "finished_at": 1,
        "started_at": 1,
        "estimate_run_time":
1,
        "is_vip": false,
        "cluster_partition":
"",
        "preempt_policy": 0,
        "vip_node_names":
null,
        "stop_op_user": {
            "id": 1,
            "name": ""
        },
        "use_new_log": true
    }
}
```


- envs：环境变量
- storages：存储列表
- instances：实例数量
- created_at：创建时间
- username：创建人
- user_id：创建人用户 ID
- namespace：k8s 命名空间
- res_name：k8s 任务名称
- status：状态
  - 1：等待中
  - 2：运行中
  - 3：成功
  - 4：失败
  - 5：停止
- use_ib_network：是否启用 IB 网络
- always_pull_image：是否总是拉取镜像
- shm：共享内存大小，单位 GiB
- category：分类
- project：研发项目

		◦ avg_gpu_util：平均 GPU 利用率 ◦ finished_at：完成时间 ◦ started_at：开始时间 ◦ estimate_run_time：预估运行时长 ◦ is_vip：是否是 VIP 任务 ◦ cluster_partition：VIP 集群分区 ◦ preempt_policy：VIP 任务抢占策略 . 1：等待现有低优先级任务完成 . 2：杀死现有低优先级任务 ◦ vip_node_names：为 VIP 任务预留的节点。 ◦ stop_op_user：停止任务操作人 ◦ use_new_log：是否使用新版本日志存储方式

获取训练任务 pod 列表

Method	GET	
URL	http://<IP>:<PORT>/model-training-platform/api/v1/trainings/{id}/pods	

Headers	Authorization: Bearer <JWT>	
Path Params	• id	训练任务 ID
Query Params	• page_size: 9999 • page_num: 1	
Request		

Response	JSON
	{
	"code": 0,
	"data": {
	"total": 1,
	"page_size": 9999,
	"page_num": 1,
	"data": [
	{
	"id": 395,
	"namespace": "live-training-1",
	"name": "u-1-84-kfqblmgwg2s0957dqmo4bybb-hrkpg",
	"status": "Deleted",
	"created_at": 1729063122000,
	"started_at": 1729063124000,
	"finished_at": 1729063223227,
	"host_ip": "192.168.13.108",
	"node_name": "ai-13-108",
	"ssh_port": 0,
	"ssh_info": "",
	"use_new_log": true
	}
	]
	}
	}
		• code：错误码
		◦ 0：成功
		◦ 非 0：失败
		• msg：错误消息，code 非 0 时可能有。
		• data
		◦ total：总的条目数
		◦ page_size：分页大小
		◦ page_num：第几页
		◦ id：pod 的 ID
		◦ namespace：pod 的命名空间
		◦ name：pod 的名称
		◦ status：pod 的状态
		. Pending：等待中
		. Running：运行中
		. Succeeded：成

		功 . Failed：失败 . Deleted：被删除 ◦ created_at：创建时间 ◦ started_at：开始时间 ◦ finished_at：结束时间 ◦ host_ip：主机IP（节点 IP） ◦ node_name：节点名称 ◦ ssh_port：ssh端口号 ◦ ssh_info：ssh信息 ◦ use_new_log：是否使用新版本日志存储方式

获取存储列表

Method	GET	
URL	http://<IP>:<PORT>/model-training-platform/api/v1/storages	
Headers	Authorization: Bearer <JWT>	
Path Params		

Query Params		
Request		

	JSON
	{
	"code": 0,
	"msg": "",
	"data": {
	"data": [
	{
	"id": 1,
	"name": "13-160-data1",
	"path": "/mnt/13-160-data1",
	"server_path": "/data1/model_training_platform",
	"server_host": "192.168.13.160",
	"server_type": "nfs",
	"permission": "rw",
	"description": ""
	}
	]
	}
	}
		• code：错误码
		◦ 0：成功
		◦ 非 0：失败
		• msg：错误消息，code 非 0 时可能有。
		• data.data
		◦ id：存储的 ID
		◦ name：pod 的名称
		◦ path：容器内路径
		◦ server_path：服务器路径
		◦ server_host：服务器 IP
		◦ server_type：服务器挂载类型
		▪ nfs
		▪ hostpath
		◦ permission：权限
		▪ r：只读
		▪ rw：读写
		◦ description：描述

停止训练任务

Method	POST	
URL	http://<IP>:<PORT>/model-training-platform/api/v1/trainings/{id}/stop	
Headers	Authorization: Bearer <JWT>	
Path Params	• id	训练任务 ID
Query Params		
Request		

Response	JSON
	{
	"code": 0,
	"msg": ""
	}
		• code：错误码
		◦ 0：成功
		◦ 非 0：失败
		• msg：错误消息，code 非 0 时可能有。

配额任务

创建配额任务

Method	POST	
URL	http://<IP>:<PORT>/quota-schedule-management/api/v1/tasks	
Headers	Authorization: Bearer <JWT>	
Path Params		
Query Params		

Request	JSON
	{
	"priority": 1,
	"framework":
	"MpiJobMpiRun",
	"name": "string",
	"description": "string",
	"command": "string",
	"image": "string",
	"virtual_cluster_id": 0,
	"sku_cnt": 0,
	"enable_ssh": true,
	"envs": [
	{
	"key": "string",
	"value": "string"
	}
	],
	"storage_ids": [
	0
	],
	"instances": 0,
	"use_ib_network": true,
	"always_pull_image": true,
	"shm": 0,
	"category_id": 0,
	"project_id": 0
	}
		• priority：优先级 1 低优先级 2 高优先级
		• framework：训练框架
		◦ MpiJob
		◦ PyTorchJob
		• name：名称
		• description：描述，可选
		• image：镜像
		• virtual_cluster_id：虚拟集群 ID，可以用接口工作流 - OpenAPI 文档获取。
		• sku_cnt：sku 数量
		• enable_ssh：是否启用 SSH 登录
		• envs：环境变量
		• storage_ids：存储 ID 列表，可以通过接口工作流 - OpenAPI 文档获取。
		• instances：实例数量
		• use_ib_network：是否启用 IB 网络
		• always_pull_image：是否总是拉取镜像
		• shm：共享内存大小，单位 GiB
		• category_id：分类 ID

		◦ 1：训练 ◦ 2：推理 ◦ 3：调试 ◦ 4：特别 • project_id：研发项目 ID，可以在 AIHub 左侧菜单【研发项目】查看 • estimate_run_time：预估运行时长，单位秒

Response	JSON
	{
	"code": 0,
	"msg": "",
	"data": {
	"id": 1
	}
	}
		• code：错误码
		◦ 0：成功
		◦ 非 0：失败
		• msg：错误消息，code 非 0 时可能有。
		• data
		◦ id：新创建的训练任务 ID。

获取配额任务信息

Method	GET	
URL	http://<IP>:<PORT>/quota-schedule-management/api/v1/tasks/{id}	
Headers	Authorization: Bearer <JWT>	
Path Params	• id	任务 ID
Query Params		
Request		

Response	JSON
	{
	"code": 0,
	"msg": "string",
	"data": {
	"id": 0,
	"priority": 1,
	"mtp_id": 0,
	"framework": "string",
	"name": "string",
	"description": "string",
	"command": "string",
	"image": "string",
	"virtual_cluster": {
	"id": 0,
	"name": "string",
	"gpu_type": "string",
	"label": "string",
	"sku": {
	"cpu": 0,
	"gpu": 0,
	"memory": 0
	},
	"node_tasks": [
	{
	"node_name": "string",
	"node_ip": "string",
	"tasks": [
	{
	"id": "string",
	"name": "string",
	"sku_cnt": 0,
	"category_name": "string",
	"project_name": "string",
	"estimate_run_time": 0,
		• code：错误码
		◦ 0：成功
		◦ 非 0：失败
		• msg：错误消息，code 非 0 时可能有。
		• data
		◦ id：新创建的运行实例 ID。
		◦ priority：优先级 1 低优先级 2 高优先级
		◦ framework：训练框架
		▪ MpiJob
		▪ PyTorchJob
		◦ name：名称
		◦ description：描述
		◦ image：镜像
		◦ virtual_cluster：虚拟集群
		◦ sku_cnt：sku 数量
		◦ enable_ssh：是否启用 SSH 登录
		◦ envs：环境变量
		◦ storages：存储列表
		◦ instances：实例数量
		◦ created_at：

```
"actual_run_time": 0,

"username": "string"
                        }
                    ]
                }
            ]
        },
        "sku_cnt": 0,
        "enable_ssh": true,
        "envs": [
            {
                "key": "string",
                "value":
"string"
            }
        ],
        "storages": [
            {
                "id": 0,
                "name":
"string",
                "path":
"string",
                "server_path":
"string",
                "server_host":
"string",
                "server_type":
"string",
                "permission":
"string",
                "description":
"string"
            }
        ],
        "instances": 0,
        "created_at": 0,
        "username": "string",
        "user_id": 0,
        "namespace": "string",
        "res_name": "string",
        "status": 1,
```


创建时间

- username：创建人
- user_id：创建人用户 ID
- namespace：k8s 命名空间
- res_name：k8s 任务名称
- status：状态
  - 1：等待中
  - 2：运行中
  - 3：成功
  - 4：失败
  - 5：停止
  - 6：停止中
- use_ib_network：是否启用 IB 网络
- always_pull_image：是否总是拉取镜像
- shm：共享内存大小，单位 GiB
- category：分类
- project：研发项目
- avg_gpu_util：平均 GPU 利用率
- finished_at：完成时间

	<pre>"use_ib_network": true,
	"category": {
	"id": 0,
	"name": "string"
	},
	"project": {
	"id": 0,
	"name": "string",
	"description": "string"
	},
	"avg_gpu_util": 0,
	"estimate_run_time": 0,
	"is_vip": true,
	"cluster_partition": "string",
	"preempt_policy": 0,
	"vip_node_names": [
	"string"
	],
	"stop_op_user": {
	"id": 0,
	"name": "string"
	}
	}
	}</pre>
		◦ started_at：开始时间
		◦ estimate_run_time：预估运行时长
		◦ is_vip：是否是 VIP 任务
		◦ cluster_partition：VIP 集群分区
		◦ preempt_policy：VIP 任务抢占策略
		. 1：等待现有低优先级任务完成
		. 2：杀死现有低优先级任务
		◦ vip_node_names：为 VIP 任务预留的节点。
		◦ stop_op_user：停止任务操作人
		◦ use_new_log：是否使用新版本日志存储方式

获取配额任务 pod 列表

Method	GET	
URL	http://<IP>:<PORT>/quota-schedule-management/api/v1/tasks/{id}/pods	
Headers	Authorization: Bearer <JWT>	
Path	• id	训练任务 ID

Params		
Query Params	• page_size: 9999 • page_num: 1	
Request		

Response	JSON
	{
	"code": 0,
	"msg": "string",
	"data": {
	"total": 0,
	"page_size": 0,
	"page_num": 0,
	"data": [
	{
	"id": 0,
	"namespace": "string",
	"name": "string",
	"status": "string",
	"created_at": 0,
	"started_at": 0,
	"host_ip": "string",
	"node_name": "string",
	"ssh_port": 0,
	"ssh_info": "string",
	"duration": 0
	}
	]
	}
	}
		• code：错误码
		◦ 0：成功
		◦ 非 0：失败
		• msg：错误消息，code 非 0 时可能有。
		• data
		◦ total：总的条目数
		◦ page_size：分页大小
		◦ page_num：第几页
		◦ id：pod 的 ID
		◦ namespace：pod 的命名空间
		◦ name：pod 的名称
		◦ status：pod 的状态
		. Pending：等待中
		. Running：运行中
		. Succeeded：成功
		. Failed：失败
		. Deleted：被

		删除 ◦ created_at：创建时间 ◦ started_at：开始时间 ◦ finished_at：结束时间 ◦ host_ip：主机 IP（节点 IP） ◦ node_name：节点名称 ◦ ssh_port：ssh 端口号 ◦ ssh_info：ssh 信息 ◦ use_new_log：是否使用新版本日志存储方式

停止配额任务

Method	POST	
URL	http://<IP>:<PORT>/quota-schedule-management/api/v1/tasks/{id}/stop	
Headers	Authorization: Bearer <JWT>	
Path Params	• id	训练任务 ID
Query Params		
Request		

Response	JSON
	{
	"code": 0,
	"msg": ""
	}
		• code：错误码
		◦ 0：成功
		◦ 非 0：失败
		• msg：错误消息，code 非 0 时可能有。

评测平台

创建评测任务

Method	POST	
URL	http://<IP>:<PORT>/api/v2/tasks/import_result	
Headers	Authorization: Bearer <JWT>	
Path Params		
Query Params		

Request	form:
	JSON
	{
	'model_name': str,
	'model_type': str,
	'epoch': str,
	'dataset_name': str,
	'id_dataset_name': str,
	'gallery_dataset_name': str,
	'user_id': str
	}
	files:
		• dataset_name, id_dataset_name, gallery_dataset_name 分别是 gallery 数据集，id 数据集和 query 数据集的名称
		• model_name 模型名称
		• model_type: 3 人脸 4 人体 5 车牌

	JSON
	{
	'model_file': (filename, file),
	'pred_file': (filename, file)
	}
		• model_file 模型文件（系统会对上传的文件进行备份）
		• pred_file 推理结果文件（jsonl 格式）

Response	JSON
	{
	"code": 0,
	"msg": "",
	"data": {
	"id": 1
	}
	}
		• code：错误码
		◦ 0：成功
		◦ 非 0：失败
		• msg：错误消息，code 非 0 时可能有。
		• data
		◦ id：新创建的评测任务 ID。

标签系统

获取用户虚拟集群列表

Method	GET	
URL	http://<IP>:<PORT>/tag-resource-management/api/v1/select-clusters	
Headers	Authorization: Bearer <JWT>	
Path Params		
Query Params	• user_id • module_type	• user_id：用户 ID，可以通过登录接口获取。 • module_type：模块类型

		◦ 1：模型训练 ◦ 2：工作流
Request		

Response	JSON
	{
	"code": 0,
	"msg": "",
	"data": {
	"data": [
	{
	"id": 29,
	"name": "vc-a800-train",
	"uuid": "5-0-14",
	"sku": {
	"id": 2,
	"description": "",
	"cpu": 67,
	"memory": 34,
	"gpu_type": 2,
	"gpu_memory": 12,
	"network": 0,
	"created_at": 1721382073775
	},
	"created_at": 1721396303983
	}
	]
	}
	}
		• code：错误码
		◦ 0：成功
		◦ 非 0：失败
		• msg：错误消息，code 非 0 时可能有。
		• data
		◦ id：虚拟集群 ID
		◦ name：名称
		◦ uuid：uuid
		◦ sku：sku 信息
		· id：sku ID
		· description：描述
		· cpu：cpu 的数量
		· memory：内存数量，单位 GiB
		· gpu_type：GPU 类型
		• 1：A800
		• 2：4090
		• 3：3090
		• 4：2080
		• 5：未知
		· gpu_memory：显存大小，单位

		GiB
		- network：网络类型
		  - 0：其他
		  - 1：roce
		  - 2：IB
		- created_at：创建时间
		  - created_at：创建时间

S3 中转服务

上传文件

Method	POST	
URL	https://<IP>:<PORT>/upload	• HTTPS 自签名证书联系管理员获取
Headers	• Token: <TOKEN>	• TOKEN 联系管理员获取
Path Params		
Query Params	• bucket • key	• bucket：存储桶 • key：存储路径
Request	Multipart-Form: • file: 文件	
Response	`JSON`	• code：错误码

	<pre>{
	  "code": 0,
	  "data":
	  {
	  "path": "s3://xxx",
	  "url": "http://xxx"
	  }
	}</pre>
		◦ 0：成功
		• data
		◦ path：s3 路径
		◦ url：HTTP URL

服务 IP 和 PORT 列表

服务	环境	IP	端口
用户系统	外网测试环境	192.168.13.160	31822
	内网测试环境	192.168.99.63	31031
	生产环境	192.168.99.63	30031
数据仓库	外网测试环境	192.168.13.160	30020
	内网测试环境	192.168.99.63	31020
	生产环境	192.168.99.63	30020
数据集管理	外网测试环境	192.168.13.160	30061
	内网测试环境	192.168.99.63	31061
	生产环境	192.168.99.63	30061
工作流管理 V2	外网测试环境	192.168.13.160	32538
	内网测试环境	192.168.99.63	32023
	生产环境	192.168.99.63	32731
模型训练	外网测试环境	192.168.13.160	30071

	内网测试环境	192.168.99.63	31032
	生产环境	192.168.99.63	30071
标签系统	外网测试环境	192.168.13.160	31043
	内网测试环境	192.168.99.63	31043
	生产环境	192.168.99.63	30043

API 调用示例

通过 API 调用，使用数据仓库的数据查询功能，输入 SQL 查询语句，获取查询结果 csv 文件的下载链接。

1. 登录获取 JWT token。
2. 创建数据仓库搜索任务。
3. 轮询搜索任务直至任务完成。
4. 得到搜索结果。

```Python
import time
import requests

USER_SYSTEM_HOST = "192.168.13.160:31822"
DATA_WAREHOUSE_HOST = "192.168.13.160:30020"

SEARCH_TYPE_SQL = 1
SEARCH_STATUS_SUCCESS = 3

class ResponseCodeNotOK(Exception):
    '''response code != 0'''

def raise_for_code(resp_json):
    if resp_json and resp_json["code"] != 0:
        raise ResponseCodeNotOK(f'code:{resp_json["code"]},
msg:{resp_json.get("msg", "")}')

def login(username, password):
    url = f'http://{USER_SYSTEM_HOST}/api/v1/auth/login'
```

```
    data = {
        "username": username,
        "password": password,
    }
    resp = requests.post(url, json=data)
    resp.raise_for_status()
    raise_for_code(resp.json())
    return resp.json()["data"]["token"]

def create_warehouse_search_task(token, name, sql):
    url = f'http://{DATA_WAREHOUSE_HOST}/data-
warehouse/api/v1/searches'
    data = {
        "type": SEARCH_TYPE_SQL,
        "name": name,
        "sql": sql,
    }
    headers = {
        "Authorization": f"Bearer {token}"
    }
    resp = requests.post(url, json=data, headers=headers)
    resp.raise_for_status()
    raise_for_code(resp.json())
    return resp.json()["data"]["id"]

def get_search_task_info(token, search_task_id):
    url = f'http://{DATA_WAREHOUSE_HOST}/data-
warehouse/api/v1/searches/{search_task_id}'
    headers = {
        "Authorization": f"Bearer {token}"
    }
    resp = requests.get(url, headers=headers)
    resp.raise_for_status()
    raise_for_code(resp.json())
    return resp.json()["data"]

def search_warehouse():
    token = login("admin", "123456")
    search_task_id = create_warehouse_search_task(token, "search-
001", "select object_name from meta where width>1000")
    while True:
        search_task = get_search_task_info(token, search_task_id)
        if search_task["status"] == SEARCH_STATUS_SUCCESS:
            break
```

```
        time.sleep(60)
    print(f'search result: {search_task["result_url"]}')

if __name__ == '__main__':
    search_warehouse()
```