

算力网分中心API接入要求

一、训练作业 (TrainJob)

1.1 创建训练作业

URL路径: `/v1/trainjob` **方法:** POST **描述:** 创建训练作业 **请求体:**

- `body`: 包含创建训练作业所需的详细信息
- 请求参数**

参数名	类型	描述	必需
name	string	训练任务名称	是
description	string	训练任务描述	否
config	object	训练任务配置	是

请求示例:

```
POST /v1/trainjob HTTP/1.1
Content-Type: application/json

{
  "name": "train_job_1",
  "description": "Example training job",
  "config": {
    "framework": "TensorFlow",
    "command": "python train.py",
    "resources": {
      "cpu": 4,
      "memory": "16GB",
      "gpu": 1
      // ... 更多作业资源配置信息, key-value形式
    },
    "nodeCount": 2,
    "runParams": "your_run_params",
    "bootFile": "your_boot_file"
    // ... 更多作业配置信息
  }
}
```

响应参数

参数名	类型	描述
success	boolean	操作成功标志
data	object	包含创建的训练任务信息
message	string	操作结果描述

响应示例:

```
{
  "success": true,
  "data": {
    "id": "tj123",
    "name": "train_job_1",
    "description": "Example training job",
    "createdAt": 1618003284
  },
  "message": "create successfully!"
}
```

1.2 删除训练作业

URL路径: `/v1/trainjob/{id}` **方法:** DELETE **描述:** 根据训练作业ID删除训练作业 **参数:**

- 路径参数

参数名	类型	描述	必需
id	string	训练任务ID	是

- 响应参数

参数名	类型	描述
success	boolean	操作成功标志
message	string	操作结果描述

请求示例:

```
DELETE /v1/trainjob/123456 HTTP/1.1
```

响应示例:

```
{
  "success": true,
  "message": "Training job deleted successfully."
}
```

1.3 查询训练作业详情

URL路径: `/v1/trainjob/{id}` **方法:** GET **描述:** 根据训练作业ID获取训练作业详情 **参数:**

- 路径参数

参数名	类型	描述	必需
id	string	训练任务ID	是

• 响应参数

参数名	类型	描述
success	boolean	操作成功标志
data	object	包含训练任务详细信息
message	string	操作结果描述

请求示例:

```
GET /v1/trainjob/123456?cluster=cluster1&jobName=trainjob1 HTTP/1.1
```

响应示例: (需要补充!)

```
{
  "success": true,
  "data": {
    "id": "tj123",
    "name": "train_job_1",
    "status": "running",
    "progress": 0.75,
    "createdAt": 1618003284,
    "startedAt": 1618003298,
    "updatedAt": 1618003300
  }
}
```

1.4 查询训练作业事件信息

URL路径: `/v1/trainjob/{id}/events` **方法:** GET **描述:** 根据训练作业ID获取训练作业事件

参数:

• 路径参数:

参数名	类型	描述	必需
id	string	训练任务ID	是

• 响应参数

参数名	类型	描述
success	boolean	操作成功标志
data	object	包含训练任务详细信息
message	string	操作结果描述

请求示例:

```
GET /v1/trainjob/123456/events?startTime=1609459200&endTime=1609545600 HTTP/1.1
```

响应示例:

```
{
  "totalSize": 10,
  "jobEvents": [
    {
      "name": "example_event",
      "message": "This is an example event message.",
      "reason": "Test event for demonstration.",
      "timestamp": "2023-04-01T12:00:00Z"
    },
    // ... 更多事件信息
  ]
}
```

1.5 查询训练任务列表

URL路径: /v1/trainjobs 方法: Get 描述: 获取训练作业列表

参数:

- 查询参数

参数名	类型	描述	必需
pageIndex	number	页码	否
pageSize	number	每页显示数量	否
status	string	状态过滤	否
searchKey	string	搜索关键词	否

- 响应参数

参数名	类型	描述
success	boolean	操作成功标志
data	array	包含训练任务列表
totalSize	number	总任务数

- 响应示例

```
{
  "success": true,
  "data": [
    {
      "id": "tj123",
      "name": "train_job_1",
      "status": "running",
      "createdAt": 1618003284
    },
    // ... 更多任务
  ],
  "totalSize": 10
}
```

1.6 停止训练作业

URL路径: `/v1/trainjob/{id}/stop` **方法:** POST **描述:** 根据训练作业ID停止训练作业 **参数:**

- 路径参数

参数名	类型	描述	必需
id	string	训练任务ID	是

- 响应参数

参数名	类型	描述
success	boolean	操作成功标志
message	string	操作结果描述

请求示例:

```
POST /v1/trainjob/123456/stop HTTP/1.1
Content-Type: application/json

{
  "id": "id1233",
  "cluster": "cluster1",
}
```

响应示例:

```
{
  "success": true,
  "message": "Training job stopped successfully."
}
```

1.7 训练任务指标信息

URL路径: `/v1/trainjob/{id}/metric` **方法:** GET **描述:** 获取训练作业在分中心的运行指标信息 **参数:**

- 路径参数

参数名	类型	描述	必需
id	string	训练任务ID	是

请求示例:

```
GET /v1/trainjob/12345 HTTP/1.1
```

响应示例:

• 响应参数

参数名	类型	描述
success	boolean	操作成功标志
data	object	包含训练任务运行指标信息
message	string	操作结果描述

• 响应示例

```
{
  "success": true,
  "data": {
    "cpuUsage": 0.85,
    "memoryUsage": 11.2,
    "diskUsage": 4.5,
    "networkSendRate": 0.03,
    "networkReceiveRate": 0.05
    // ... 更多运行指标信息
  }
}
```

二、调试任务 (Notebook)

2.1 创建NoteBook

URL路径: /v1/notebook 方法: POST 描述: 创建NoteBook 参数:

• 请求参数

参数名	类型	描述	必需
name	string	训练任务名称	是
description	string	训练任务描述	否
config	object	训练任务配置	是

请求示例:

```
POST /v1/notebook HTTP/1.1
Content-Type: application/json
```

```
{
  "name": "your_train_job_name",
  "command": "your_command",
  "resourceSpec": {
    "cpu": "2",
    "memory": "4Gi",
    "nvidia.com/gpu": "1"
    //...更多作业资源配置信息
  },
  "autoStopDuration": 1200,
  "nodeCount": 2,
  "runParams": "your_run_params",
  "bootFile": "your_boot_file"
  //...更多作业配置信息
}
```

响应参数

参数名	类型	描述
success	boolean	操作成功标志
data	object	包含创建的调试任务信息
message	string	操作结果描述

响应示例:

```
{
  "success": true,
  "data": {
    "id": "nb456",
    "name": "my_notebook",
    "description": "Example notebook instance",
    "createdAt": 1618003284
  }
}
```

2.2 删除NoteBook

URL路径: `/v1/notebook/{id}` **方法:** DELETE **描述:** 根据NoteBook ID删除NoteBook **参数:**

- 路径参数

参数名	类型	描述	必需
id	string	NoteBook实例ID	是

- 响应参数

参数名	类型	描述
success	boolean	操作成功标志
message	string	操作结果描述

请求示例:

```
DELETE /v1/notebook/123456?cluster=cluster1 HTTP/1.1
```

响应示例:

```
{
  "success": true,
  "message": "Notebook instance deleted successfully."
}
```

2.3 查询NoteBook详情

URL路径: /v1/notebook/{id} 方法: GET 描述: 根据NoteBook ID获取NoteBook详情 参数:

- 路径参数

参数名	类型	描述	必需
id	string	NoteBook实例ID	是

- 响应参数

参数名	类型	描述
success	boolean	操作成功标志
data	object	包含NoteBook详细信息
message	string	操作结果描述

请求示例:

```
GET /v1/notebook/123456 HTTP/1.1
```

响应示例: (需要补充)

```
{
  "success": true,
  "data": {
    "id": "nb456",
    "name": "my_notebook",
    "status": "running",
    "progress": 0.65,
    "createdAt": 1618003284,
    "updatedAt": 1618003300
    // ... 更多作业信息
  }
}
```

2.4 获取NoteBook事件

URL路径: /v1/notebook/{id}/events 方法: GET 描述: 根据NoteBook ID获取NoteBook事件

参数:

• 路径参数

参数名	类型	描述	必需
id	string	notebook任务ID	是

• 响应参数

参数名	类型	描述
success	boolean	操作成功标志
data	object	包含notebook任务详细信息
message	string	操作结果描述

请求示例:

```
GET /v1/notebook/123456/events?startTime=1609459200&endTime=1609545600 HTTP/1.1
```

响应示例:

```
{
  "totalSize": 10,
  "notebookEvents": [
    {
      "id": "idxxx",
      "startTime": "2023-04-01T12:00:00Z",
      "endTime": "2023-04-01T12:00:00Z"
    },
    // ... 更多事件信息
  ]
}
```

2.5 查询Notebook任务列表

URL路径: /v1/notebooks 方法: GET 描述: 获取训练作业列表

参数:

• 查询参数

参数名	类型	描述	必需
pageIndex	number	页码	否
pageSize	number	每页显示数量	否
status	string	状态过滤	否
searchKey	string	搜索关键词	否

• 响应参数

参数名	类型	描述
success	boolean	操作成功标志
data	array	包含训练任务列表
totalSize	number	总任务数

- 响应示例

```
{
  "success": true,
  "data": [
    {
      "id": "tj123",
      "name": "train_job_1",
      "status": "running",
      "createdAt": 1618003284
    },
    // ... 更多其他任务
  ],
  "totalSize": 10
}
```

2.6 保存NoteBook

URL路径: `/v1/notebook/{id}/save` **方法:** POST **描述:** 保存NoteBook **参数:**

- `body`: 包含保存NoteBook所需的详细信息

请求示例:

```
POST /v1/notebook/123456/save HTTP/1.1
Content-Type: application/json

{
  "body": {
    // ... 保存NoteBook所需的详细信息
  }
}
```

响应示例:

```
{
  "imageId": "string",
  "errorCode": 0,
  "errorMsg": "string"
}
```

2.7 启动NoteBook

URL路径: `/v1/notebook/{id}/start` **方法:** POST **描述:** 启动NoteBook

参数:

- `body`: 包含启动NoteBook所需的详细信息

请求示例:

```
POST /v1/notebook/123456/start HTTP/1.1
Content-Type: application/json

{
  "body": {
    // ... 启动NoteBook所需的详细信息
  }
}
```

响应示例:

```
{
  "status": "string",
  "failReason": "string",
  "errorCode": 0,
  "errorMsg": "string"
}
```

2.8 停止NoteBook

URL路径: `/v1/notebook/{id}/stop` 方法: POST 描述: 停止NoteBook

请求体:

- `body`: 包含停止NoteBook所需的详细信息

请求示例:

```
POST /v1/notebook/123456/stop HTTP/1.1
Content-Type: application/json

{
  "body": {
    // ... 停止NoteBook所需的详细信息
  }
}
```

响应示例:

```
{
  "status": "string",
  "failReason": "string",
  "errorCode": 0,
  "errorMsg": "string"
}
```

2.9 Notebook任务指标信息

URL路径: `/v1/notebook/{id}/metric` **方法:** GET **描述:** 获取notebook作业在分中心的运行指标信息

请求体:

• 路径参数

参数名	类型	描述	必需
id	string	notebook任务ID	是

• 响应参数

参数名	类型	描述
success	boolean	操作成功标志
data	object	包含Notebook任务运行指标信息
message	string	操作结果描述

• 响应示例

```
{
  "success": true,
  "data": {
    "cpuUsage": 0.85,
    "memoryUsage": 11.2,
    "diskUsage": 4.5,
    "networkSendRate": 0.03,
    "networkReceiveRate": 0.05
    // ... 更多任务运行指标信息
  }
}
```

三、镜像 (Image)

3.1 创建镜像

URL路径: `/v1/image` **方法:** POST **描述:** 创建镜像

参数:

• 请求参数

参数名	类型	描述	必需
name	string	镜像名称	是
description	string	镜像描述	否
config	object	镜像配置	是

• 请求示例

```
{
  "name": "my_image",
  "description": "Example image for notebooks",
  "config": {
    "addr": "registry.example.com/my_image:latest",
    "version": "1.0",
    "source": "dockerhub",
    "tags": ["latest", "1.0"]
  }
}
```

• 响应参数

参数名	类型	描述
success	boolean	操作成功标志
data	object	包含创建的镜像信息
message	string	操作结果描述

• 响应示例

```
{
  "success": true,
  "data": {
    "id": "img789",
    "name": "my_image",
    "description": "Example image for notebooks",
    "createdAt": 1618003284
  }
}
```

3.2 删除镜像

URL路径: /v1/image/{id} 方法: DELETE 描述: 根据镜像ID删除镜像 参数:

• 路径参数

参数名	类型	描述	必需
id	string	镜像实例ID	是

• 响应参数

参数名	类型	描述
success	boolean	操作成功标志
message	string	操作结果描述

请求示例:

```
DELETE /v1/image/123456 HTTP/1.1
```

响应示例:

```
{
  "success": true,
  "message": "Image deleted successfully."
}
```

3.3查询镜像详情

URL路径: /v1/image/{id} 方法: GET 描述: 根据镜像ID获取镜像详情 参数:

- id (必需): 镜像ID

请求示例:

```
GET /v1/image/123456 HTTP/1.1
```

响应示例:

```
{
  "id": "string",
  "imageName": "string",
  "imageVersion": "string",
  "imageAddr": "string",
  // ... 更多镜像属性
}
```

3.4 获取镜像列表

URL路径: /v1/images 方法: GET 描述: 获取镜像列表 参数:

- 查询参数

参数名	类型	描述	必需
pageIndex	number	页码	否
pageSize	number	每页显示数量	否
sourceType	number	来源类型过滤	否
imageStatus	number	镜像状态过滤	否
searchKey	string	搜索关键词	否

- 响应参数

参数名	类型	描述
success	boolean	操作成功标志
data	array	包含镜像列表
totalSize	number	总镜像数

请求示例:

```
GET /v1/imageList?
pageIndex=1&pageSize=10&orderBy=desc&imageNameLike=notebook&imageStatus=1
HTTP/1.1
```

响应示例:

```
{
  "success": true,
  "data": [
    {
      "id": "img789",
      "name": "my_image",
      "status": "active",
      "createdAt": 1618003284
    },
    // ... 更多镜像
  ],
  "totalSize": 3
}
```

3.5 更新镜像

HTTP 方法: PUT

描述: 用于镜像更新操作

路径: `/v1/image/{imageId}`

请求体:

```
1{
2  "imageId": "existing-image-id-to-update",
3  "imageName": "updated-image-name",
4  "imageVersion": "2.0",
5  "imageType": 2,
6  "imageDesc": "更新后的镜像描述",
7  "imageAddr": "http://new-location.example.com/new-image.iso"
8}
```

响应:

```
1{
2  "updatedAt": 1609459200
3}
```

3.6 上传镜像

HTTP 方法: POST 路径: `/v1/image/{imageId}/upload`

描述: 将制作的镜像上传

请求体:

```
1{
2  "imageId": "existing-image-id",
3  "fileName": "example.iso",
4  "domain": "https://your-storage.example.com/upload/path"
5}
```

响应:

```
1{
2  "uploadUrl": "https://temp-upload-url.example.com/target",
3  "headers": {
4    "Content-Type": "application/octet-stream",
5    "Authorization": "Bearer your-access-token"
6  }
7}
```

四、推理服务 (Inference)

4.1 创建推理服务 (Inference)

URL路径: `/v1/inference` **方法:** POST **描述:** 创建推理服务 **请求体:**

- `body`: 包含创建推理服务所需的详细信息

请求示例:

```
POST /v1/inference HTTP/1.1
Content-Type: application/json

{
  "body": {
    // ... 创建推理服务所需的详细信息
  }
}
```

响应示例:

```
{
  "id": "string",
  "errorCode": 0,
  "errorMsg": "string",
  "blscToken": "string"
}
```

4.2 查询推理服务详情

URL路径: `/v1/inference/{id}` **方法:** GET **描述:** 根据推理任务ID获取推理服务详情 **参数:**

- `id` (必需): 推理服务ID

请求示例:

```
GET /v1/image/123456 HTTP/1.1
```


响应示例:

```
{
  "id": "string",
  "inferenceName": "string",
  "inferenceAddr": "string",
  // ... 更多镜像属性
}
```

4.3 删除推理服务

URL路径: /v1/inference/{id} 方法: DELETE 描述: 根据服务ID删除推理服务 参数:

- 路径参数

参数名	类型	描述	必需
id	string	推理服务ID	是

- 响应参数

参数名	类型	描述
success	boolean	操作成功标志
message	string	操作结果描述

请求示例:

```
DELETE /v1/inference/123456 HTTP/1.1
```

响应示例:

```
{
  "success": true,
  "message": "Inference service deleted successfully."
}
```

4.4 获取推理服务列表

URL路径: /v1/inferences 方法: GET 描述: 获取推理任务列表 参数:

- 查询参数

参数名	类型	描述	必需
pageIndex	number	页码	否
pageSize	number	每页显示数量	否
sourceType	number	来源类型过滤	否
imageStatus	number	镜像状态过滤	否
searchKey	string	搜索关键词	否

• 响应参数

参数名	类型	描述
success	boolean	操作成功标志
data	array	包含推理服务列表
totalSize	number	总推理服务数

请求示例:

```
GET /v1/inferenceList?
pageIndex=1&pageSize=10&orderBy=desc&inferenceNameLike=inference&inferenceStatus
=1 HTTP/1.1
```

响应示例:

```
{
  "success": true,
  "data": [
    {
      "id": "inf789",
      "name": "my_inference",
      "status": "active",
      "createdAt": 1618003284
    },
    // ... 更多推理服务
  ],
  "totalSize": 3
}
```

五、资源管理服务 (Resource)

5.1 查询分中心资源信息

URL路径: `/v1/resourcespec` 方法: GET 描述: 查询获取特定处理器类型的资源规格信息的各个分中心

参数:

• 查询参数

参数名	类型	描述
		指定处理器类型, 可以是 nvidia.com/gpu、

参数名	processorType	类型	描述
			huawei.com/NPU、cambricon.com/mlu、enflame-tech.com/gcu 或 ac.sugon.com/dcu

• 请求示例

```
GET /v1/resourcespec`?processorType=nvidia.com/gpu

HTTP/1.1
```

• 响应参数

```
{
  "resourceSpecs": [
    {
      "id": "string",
      "createdAt": "int64",
      "updatedAt": "int64",
      "name": "string",
      "processorType": "string",
      "centers": [
        {
          "id": "string", //分中心的ID
          "name": "string", //分中心的名称
          "spec": [
            {
              "id": "string",
              "poolType": "string",
              "name": "string",
              "isNetAccess": "string"
            }
          ]
        }
      ]
    },
    "specInfo": {
      "cpuCoreNum": "int32",
      "memorySize": "string",
      "accDeviceKind": "string",
      "accDeviceModel": "string",
      "accDeviceNum": "int32",
      "accDeviceMemory": "string",
      "vCpu": "int32",
      "cpuKind": "string",
      "cpuRate": "string",
      "innerNetBandwidth": "int32",
      "innerNetPackages": "int32",
      "diskSize": "int32",
      "bandwidth": "int32"
    }
  ]
}
```

响应参数详细说明

- resourceSpecs

: 一个包含多个资源规格详细信息的对象数组。

- `id`: 资源规格的唯一标识符。
- `createdAt` & `updatedAt`: 资源规格创建和更新的时间戳。
- `name`: 资源规格名称。
- `processorType`: 资源所使用的处理器类型。
- `centers`: 包含多个中心及其资源规格列表。

- `id` & `name`: 分中心的ID和名称。

- `resourceSpec`: 指的是每个中心的资源规格信息。

- `id`: 资源池ID。
- `poolType`: 资源池类型，例如 1:共享 2:专属。
- `name`: 资源池名称。
- `isNetAccess`: 队列是否联通外网。

- `specInfo`: 资源规格的具体配置信息。

- `cpuCoreNum`: CPU核心数。
- `memorySize`: 内存大小 (Mi/Gi单位) 。
- `accDeviceKind`: 加速器设备类型。
- `accDeviceModel`: 加速器设备型号。
- `accDeviceNum`: 加速器设备数量。
- `accDeviceMemory`: 加速器设备内存大小。
- `vCpu`: 虚拟CPU数量。
- `cpuKind`: 处理器类型。
- `cpuRate`: 处理器主频及睿频。
- `innerNetBandwidth`: 内网带宽。
- `innerNetPackages`: 内网收发包能力。
- `diskSize`: 系统盘大小。
- `bandwidth`: 带宽大小。

• 实际示例：

处理器类型 `processor_type` : `npu.huawei.com/NPU`

资源规格名称 : `Ascend: 1 * Ascend-910(32GB) | ARM: 24 核 | Memory: 96GB`

`specInfo`资源规格信息：

```
{
  "cpuCoreNum": 24,
  "memorySize": "96Gi",
  "accDeviceNum": 1,
  "accDeviceKind": "npu.huawei.com/NPU",
  "accDeviceModel": "ascend910",
  "accDeviceMemory": "32G"
  //...更多资源规格信息
}
```

拥有上述资源规格信息 (`specInfo`) 的分中心：

```
[
```

```
{
  "aiCenterId": "wuhan",
  "resourceSpec": [
    {
      "id": "0",
      "name": "共享资源池",
      "poolType": "1",
      "isNetAccess": "1"
    }
  ]
},
{
  "aiCenterId": "xian",
  "resourceSpec": [
    {
      "id": "0",
      "name": "共享资源池",
      "poolType": "1",
      "isNetAccess": "1"
    }
  ]
}
]
```

六、用户鉴权服务 (Auth)

6.1 获取用户令牌 (GetToken)

功能简要概述: 用户通过提供用户名和密码来获取访问令牌。

请求URL: `/v1/auth/token`

请求方式: `POST`

请求参数:

参数名	类型	必需	描述
username	string	是	用户名 (需为有效的电子邮件地址)
password	string	是	密码 (长度在8到30个字符之间)

响应参数:

参数名	类型	描述
token	string	访问令牌
expiration	int64	令牌过期时间 (时间戳)

请求示例:

```
POST /v1/auth/token HTTP/1.1
Content-Type: application/json

{
  "username": "user@example.com",
  "password": "strong_password"
}
```

响应示例:

```
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoiaXJ0e1MzYyNjUsImVtYWlsIjoiaXJ0e1MzYyNjUwMjB9.4WFXUVHVZY_lU6JzJ4tjK-KjKnMgp7Gd8cGHM0KHc",
  "expiration": 1641016400
}
```

6.2 删除用户令牌 (DeleteToken)

功能简要概述: 撤销当前用户的访问令牌，使其失效。

请求URL: `/v1/auth/token`

请求方式: `DELETE`

请求参数: 无

响应参数: 无

请求示例:

```
DELETE /v1/authmanage/token HTTP/1.1
Authorization: Bearer your_access_token
```

响应示例:

```
{}
```