**标准化作业任务核心调度流程方案**

1. **调度面统一实体定义**

课题研究需对隐私计算作业调度中各实体的定义形成共识。

* 1. **组件（component）**

独立执行隐私计算任务的功能模块单元，其经过封装、符合开放接口规范、可以完成某个隐私计算功能并被使用在隐私计算流程DAG中，用顶点(vertex)表示。

* 1. **流程（flow）**

采用DAG结构定义的、可编排的隐私计算作业运行模板，用于描述一组组件的上下游及依赖关系。

* 1. **作业（job）**

一个隐私计算流程在经过运行参数配置后的运行实例，通常包含参与方节点信息、参与方资源、若干任务排列的集合等。

* 1. **任务（task）**

每个组件运行的实例，每个任务通常包含有组件的输入、输出、运行的参数等。

* 1. **资源（resource）(可选)**

一个隐私计算流程的作业在运行时可使用的硬件资源信息，调度层会根据此配置信息来为每个组件的任务实例分配运行时的配置。

1. **相关实体参考属性**

课题研究需对隐私计算作业调度中各实体应具备的通用属性形成共识，并对各实体状态及转换条件进行明确。

* 1. **作业(job)实体**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **数据类型** | **可选/必选** | **描述** |
| job\_id | String | 必选 | 作业ID，全局唯一，由调度方生成 |
| flow\_id | String | 可选 | 所属流程ID |
| status | String | 必选 | 作业状态，参考值为PENDING、 READY、REJECTED、RUNNING、 FINISHED |
| start\_time | Long | 必选 | 作业开始时间 |
| finish\_time | Long | 必选 | 作业结束时间 |
| update\_time | Long | 必选 | 作业信息更新时间 |
| dag | DAG | 必选 | 作业的组件之间组合的配置 |
| config | Config | 必选 | 作业运行时的参数配置 |
| priority | String | 可选 | 作业优先级。用于排队场景 |

* 1. **任务(task)实体**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **可选/必选** | **描述** |
| task\_id | String | 必选 | 任务ID，全局唯一，由调度方生成 |
| job\_id | String | 必选 | 所属作业ID |
| status | String | 必选 | 任务状态，参考值为READY、 RUNNING、SUCCESS、FAILED |
| start\_time | Long | 必选 | 任务开始时间 |
| finish\_time | Long | 必选 | 任务结束时间 |
| update\_time | Long | 必选 | 任务信息更新时间 |
| config | Config | 必选 | 任务运行时参数，包括节点标识、组件标识、角色标识等，在管里面明确任务参数定义 |

* 1. **资源(resource)实体属性表（可选）**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **类型** | **描述** |
| nodes | Integer | 计算节点数量 |
| cores\_per\_node | Long | 计算节点CPU数量 |
| bandwidth | Long | 计算节点带宽 |

* 1. **作业(job)实体状态表**

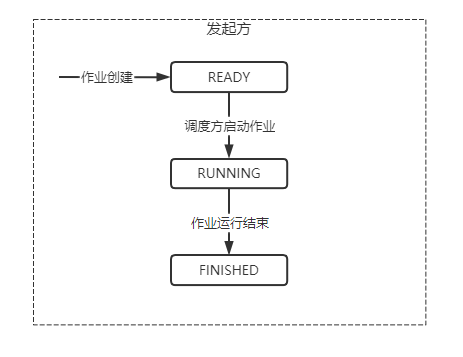
**发起方或参与方**

|  |  |
| --- | --- |
| 作业状态 | 条件 |
| PENDING | 当创建作业成功时，参与方的作业状态为PENDING |
| READY | 1、当创建作业成功时，发起方的作业默认不需要审批，发起方的作业状态为READY  2、当创建作业成功，并且通过审批后，参与方的作业状态为READY |
| REJECTED | 当创建作业成功，并且审批被拒绝后，作业状态为REJECTED，当前状态的作业无法开始运行 |
| RUNNING | 当启动作业成功并且该作业下至少有一个任务处于RUNNING时，作业状态为RUNNING |
| FINISHED | 当作业下的所有任务都运行结束时，作业状态为FINISHED |

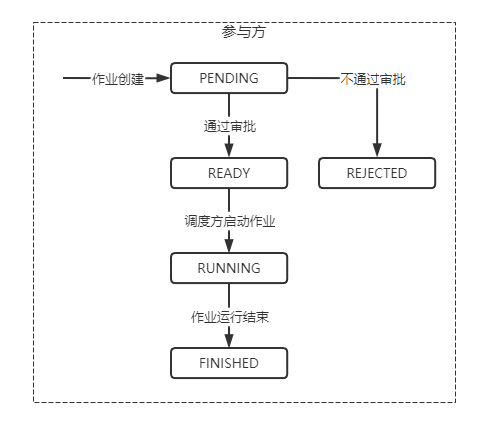
**调度方**

|  |  |
| --- | --- |
| 作业状态 | 条件 |
| PENDING | 存在任意参与方的作业状态为PENDING时，调度方的作业状态为PENDING |
| READY | 当所有参与方作业状态为READY时，调度方的作业状态为READY |
| REJECTED | 当存在任意参与方作业状态为REJECTED时，调度方的作业状态为REJECTED，当前状态的作业无法开始运行 |
| RUNNING | 当存在任意参与方作业状态为RUNNING时，调度方的作业状态为RUNNING |
| FINISHED | 当所有参与方作业状态都为FINISHED时，调度方的作业状态为FINISHED  **注：**如需要展示更细粒度状态，如成功（SUCCESS）或失败（FAILED）状态，可根据东西向查询作业状态返回的任务状态列表，汇总出己方需要的状态结果 |

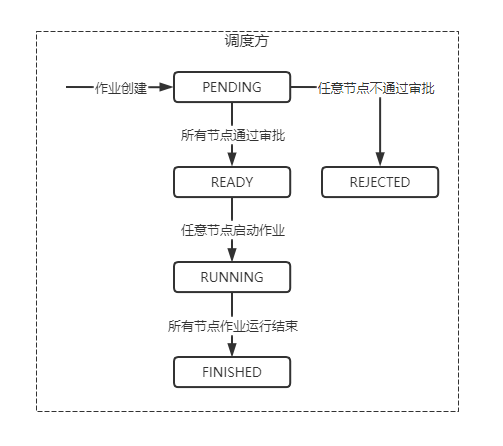
发起方作业状态



参与方作业状态



调度方作业状态



* 1. **任务(task)实体状态表**

**发起方或参与方**

|  |  |
| --- | --- |
| 任务状态 | 条件 |
| READY | 当创建作业成功，并且审批通过时，作业下所有任务状态为READY |
| RUNNING | 当任务对应的算法容器被正确加载并且容器正常运行时，任务状态为RUNNING |
| SUCCESS | 当任务对应的算法容器正常退出时，任务状态为SUCCESS |
| FAILED | 当任务对应的算法容器非正常退出，或算法容器加载失败，或算法容器超时运行时，任务状态为FAILED |

**调度方**

|  |  |
| --- | --- |
| 任务状态 | 条件 |
| READY | 当所有参与方任务状态都为READY时，调度方的任务状态为READY |
| RUNNING | 当存在任意参与方任务状态为RUNNING，并且所有参与方任务状态都不为FAILED时，调度方的任务状态为RUNNING |
| SUCCESS | 当所有参与方任务状态都为SUCCESS时，调度方的任务状态为SUCCESS |
| FAILED | 当存在任意参与方任务状态为FAILED时，调度方的任务状态为FAILED |

* 1. **组件注册信息（ComponentInfo）实体**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **可选/必选** | **描述** |
| componentName | String | 必选 | 算法组件名称，跟算法容器的环境变量相对应 |
| title | String | 必选 | 页面显示名称 |
| provider | String | 必选 | 算法提供商 |
| version | String | 必选 | 算法版本 |
| description | String | 可选 | 算法描述 |
| roleList | Array<String> | 必选 | 算法支持的角色定义,数组类型。例如：["guest", "host","arbiter"] |
| desVersion | String | 可选 | 描述文件采用的版本 |
| storageEngine | Array<String> | 必选 | 说明算法组件所支持的存储引擎，数组类型。如：["s3", "hdfs","eggroll"] |
| inputParam | Array<InputParam> | 必选 | 描述算法输入参数 |
| inputData | Array<InputData> | 必选 | 描述算法组件在运行时的输入数据 |
| outputData | Array<OutputData> | 必选 | 描述算法组件在运行时的输出数据 |
| result | Array<Result> | 必选 | 描述算法组件的异常 |

* 1. **组件注册算法输入参数（InputParam）实体**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **可选/必选** | **描述** |
| name | String | 必选 | 输入字段的形式参数名称，跟算法容器的环境变量相对应 |
| title | String | 必选 | 页面显示字段标题 |
| description | String | 可选 | 参数描述 |
| type | String | 必选 | 描述输入字段的类型  **string：**字符串类型  **int:** 整数类型  **float:** 浮点数类型  **boolean:** 布尔类型  **integer:** 整型对象，可以为null  **number:** 数值型，包括整数和小数（当不具体区分整型和浮点型时采用该种类型）  **object:** 复杂类型,可以使用json表示,前端就具体的json数据结构可扩展相关控件。  ...等其他类型 |
| bindingData | List<Object> | 可选 | 可枚举的静态绑定数据,可以引用外部其它函数获取控件绑定的数据,如可渲染下拉菜单、单选框、多选框等的数据获取方式  例如1:静态绑定数据  bindingData:"[{label: "${label}",value: "${value}"}]",其中是可以执行的 kv Json，格式参考为kv数组。  例如2:动态绑定数据  bindingData:"call\_function getKV()"。//定义一个调用的回调函数获取数据，  这里的call\_function作为一个函数调用的标识符使用，解析执行器使用call\_function作为标识符判断，用于与静态值区别，  后面的getKV()，解析执行器需要保障执行上下文中存在可调用的getKV()函数，getKV()的函数命名，在实际使用中，可自定义。 |
| optional | Boolean | 必选 | 表示该参数是必须的还是可选的,默认为true表示可选,false表示必填 |
| defaultValue | String | 可选 | 输入的默认值 |
| validator | String | 可选 | 选择一个字段校验器,如可以写成 regular:'正则表达式',来做正则验证；使用（inf,sup）来对数值的上下界限进行限定 |
| dependsOn | Array<String> | 可选 | 只用于bindingData,表示算法的条件参数，用于组件内部依赖关系展示,默认为空值。若依赖于上游参数，则提供依赖列表，如：["param.value",""] |
| UIPattern | String | 可选 | 表示字段的UI模式,默认为editeable,此外有readOnly表示只读,hidden,表示隐藏控件 |
| groupTag | Array<String> | 可选 | 分组标签,用于分组显示,如[默认分组-显示、高级分组-默认不显示] |
| UIType | String | 可选 | 表示前端采用的控件类型,默认为input,其它的比如textArea,numberPicker、checkbox、redio、switch、select、selectTree等 |

* 1. **组件注册输入数据（InputData）实体**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **可选/必选** | **描述** |
| name | String | 必选 | 输入数据的形式参数名称，跟算法容器的环境变量相对应 |
| description | String | 可选 | 输入数据描述信息 |
| category | String | 必选 | 输入数据的类型 model、dataset、training\_set、test\_set、datasets（多个数据文件） |
| dataFormat | Array<String> | 可选 | 自持的输入数据文件格式，数组类型。如：["csv","pmml","json","yaml","zip"] |

* 1. **组件注册输出数据（OutputData）实体**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **可选/必选** | **描述** |
| name | String | 必选 | 输出数据的形式参数名称，跟算法容器的环境变量相对应 |
| description | String | 可选 | 输出数据描述信息 |
| category | String | 必选 | 输出数据的类型 model、dataset、training\_set、test\_set、report、metric |
| dataFormat | Array<String> | 可选 | 支持的输出数据文件格式，数组类型。如：["csv","pmml","json","yaml","zip"] |

* 1. **组件注册结果信息（Result）实体**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **可选/必选** | **描述** |
| resultCode | String | 必选 | 结果编码 |
| resultMessage | String | 可选 | 结果提示信息 |

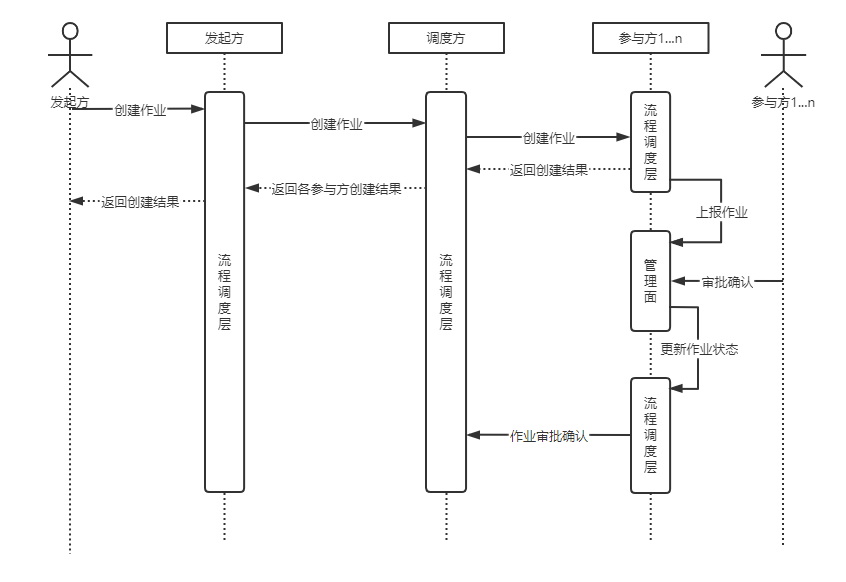
1. **流程调度时序图**

**说明：**

1. 流程调度时序图将调度方隔离，作业调度过程中的各参与方分别为发起方、调度方及合作方。调度方与发起方也可为同一方，此时下图中调度方与发起方之间的东西向调度关系转变为南北向调度关系，下图不再赘述。
2. 作业创建前，作业的参数配置可在管理面进行可选的校验审核，该接口将在管理面子课题中定义，该过程不在下图中进行展示。
3. 作业创建前，发起方可向合作方查询计算资源信息，该过程作为可选项不在下图中进行展示。
4. 本方案在一阶段中不对作业重跑与续跑进行流程定义，将在后续阶段进行讨论与进一步定义。
   1. **作业与任务创建与运行时序图**
      1. **作业审批**

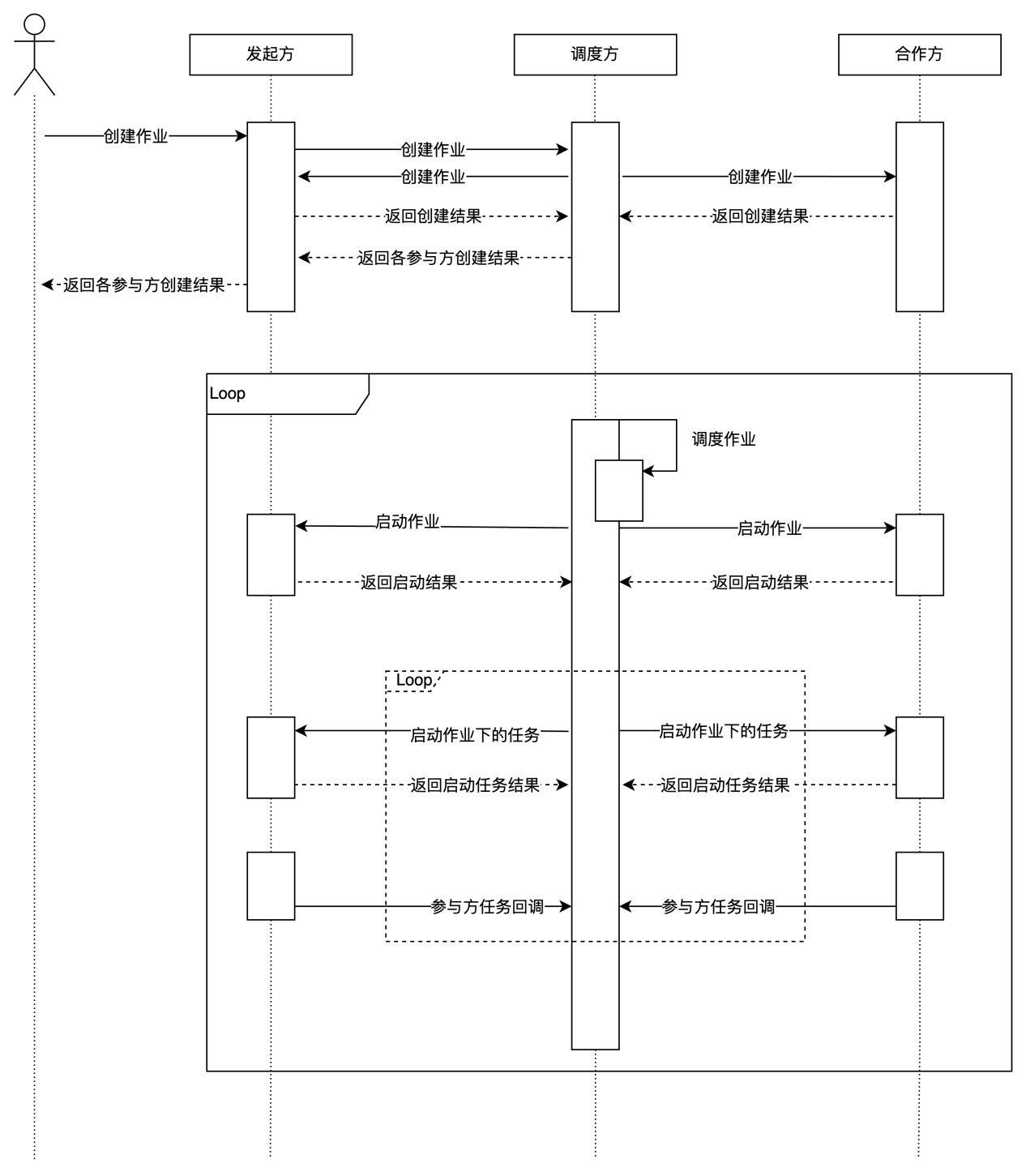
**注：**

1. 当节点不需要作业审批时，功能流程与作业审批一致，只是在确认审批需要实现自动确认即可
2. 当发起方或参与方需要查看作业审批进度时，可通过东西向查询作业状态接口实现



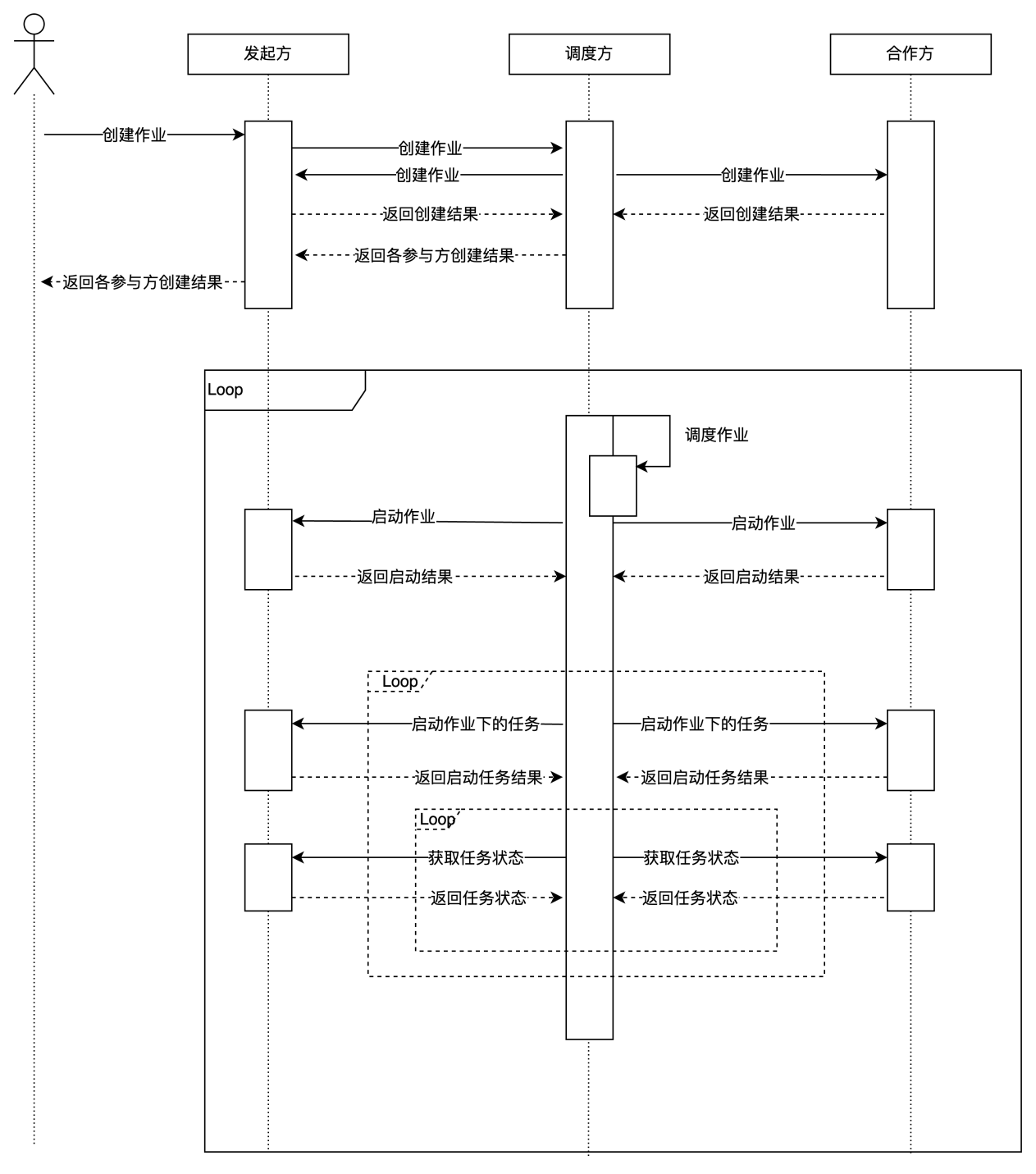
* + 1. **callback回调模式**

说明：调度器创建作业和任务，由各参与方主动回调作业或任务的状态

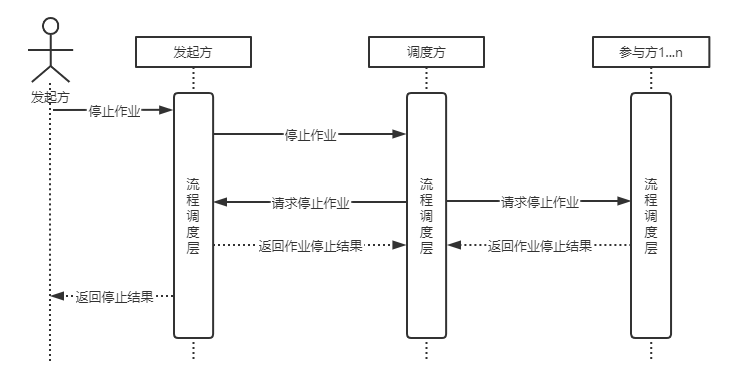


* + 1. **poll轮询模式**

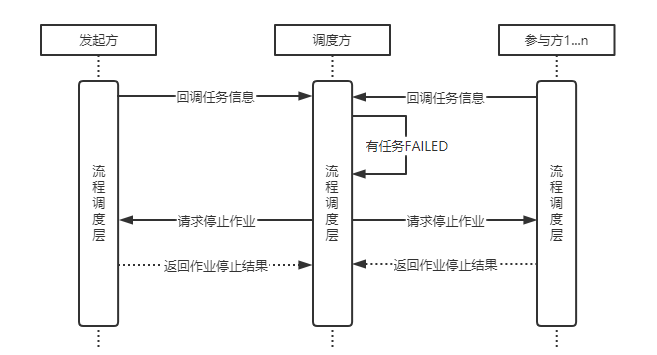
说明：调度器不仅需创建作业和任务，在调度过程中会轮询参与方的作业或任务的状态



* 1. **作业与任务停止**
     1. **发起方或合作方主动停止作业**



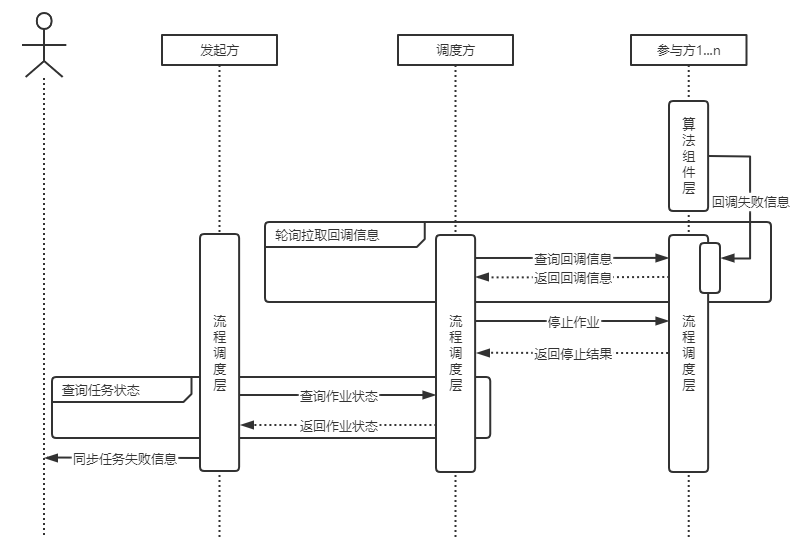
* + 1. **调度方根据任务状态停止作业**



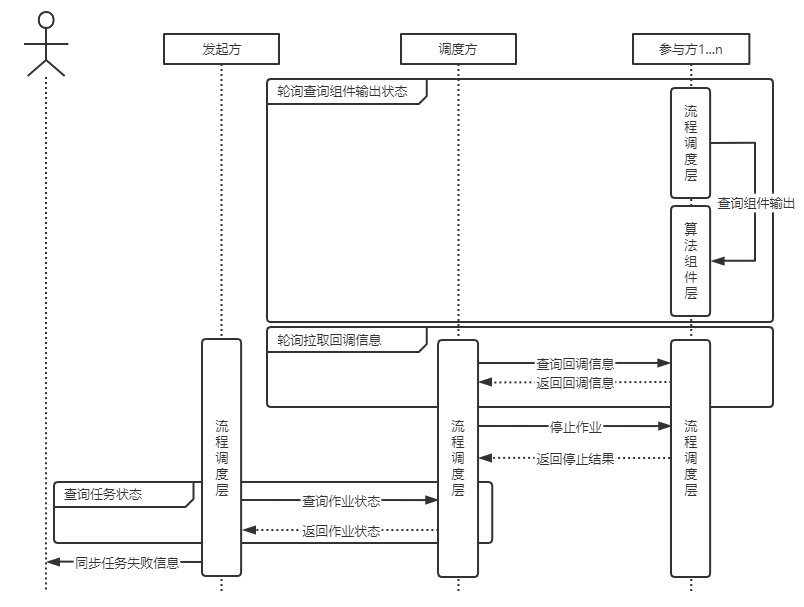
* 1. **异常情况**
     1. **任务异常回调**

**注：**发起方或某个参与方的任务异常会导致由发起方或调度方发起停止作业，但并非停止整个作业，只针对该异常任务所在的分支

组件回调任务异常状态



组件输出任务异常状态



1. **调度流程生命周期及状态说明**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **流程说明** | **调度方**  **（记录各方状态并汇总）** | **发起方** | **合作方** |
| **1** | 发起方创建作业，会将请求发给调度方。由调度方将作业分发给各参与方，并把创建结果返回给发起方。 | job:PENDING | job:READY | job:PENDING |
| **2** | 所有参与方节点对作业通过审批后，并通知调度方。 | job:PENDING->READY | job:READY | job:PENDING->READY |
| **3** | 任意参与方节点对作业拒绝审批后，并通知调度方。 | job:PENDING->REJECTED | job:READY | job:PENDING->REJECTED |
| **4** | 调度器获取READY状态的作业进行调度，尝试启动作业（需考虑资源情况） | job:READY->RUNNING  task:READY | job:READY->RUNNING  task:READY | job:READY->RUNNING  task:READY |
| **5** | 调度器对RUNNING状态的作业进行task维度的调度，启动作业的第一个task | job:RUNNING  task:READY->RUNNING | job:RUNNING  task:READY->RUNNING | job:RUNNING  task:READY->RUNNING |
| **6** | 调度器对RUNNING状态的作业进行task维度的调度   1. 若查询到各参与方当前运行的task都成功，则启动下一个task； 2. 若某方task失败，则调度器把此任务置为失败并触发stop操作 | job:RUNNING  task:RUNNING->SUCCESS  (2)  job:RUNNING->FINISHED  task:RUNNING->FAILED | (1)  job:RUNNING  task:RUNNING->SUCCESS  job:RUNNING->FINISHED  task:RUNNING->FAILED | (1)  job:RUNNING  task:RUNNING->SUCCESS  (2)  job:RUNNING->FINISHED  task:RUNNING->FAILED |
| **7** | task周期调度，直到所有的task运行成功 | job:RUNNING->FINISHED | job:RUNNING->FINISHED | job:RUNNING->FINISHED |

1. **流程调度接口定义**
   1. **接口通用规范**

**Content-Type：**application/json

**HTTP Method：**读GET，写POST

**Request Header：**

**X-Auth-Sign：**利用签约时指定的算法构造的签名值，用于安全校验

**X-Node-Id：**请求方节点id

**X-Nonce：**系统生成的防重放随机串，如 UUID

**X-Trace-Id：**业务自定义标识，用于全链路跟踪

**X-Timestamp：**调用方毫秒时间戳（Unix epoch time，具有时区无关性）

* 1. **公共出参**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 类型 | 描述 |
| code | Integer | 200正常，否则错误码 |
| msg | String | 错误码信息说明 |
| data | Object | 具体的接口数据 |

**下述接口详细设计中只定义data里的具体结构**

* 1. **南北向接口**
     1. **组件注册**

1. 接口名称：component\_register
2. 接口描述：组件信息注册
3. 调用方式：POST
4. 接口地址：/v1/platform/schedule/component/register
5. 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| imageUrl | String | 是 | 镜像地址 |
| component\_info | ComponentInfo | 是 | 注册的组件信息 |

1. 返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |

* + 1. **创建作业**

1. 接口名称：create\_job\_all
2. 接口描述：发起方发起创建作业配置，并发送给己方调度层时调用的接口
3. 调用方式：POST
4. 接口地址：/v1/platform/schedule/job/create\_all
5. 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| flow\_id | String | 否 | 流程ID，该参数调度层不会使用，只会用来存储flow和job的对应关系 |
| dag | DAG | 是 | 描述组件组合流程的配置信息 |
| config | Config | 是 | 作业参与方节点信息和每个组件参数配置等等 |
| old\_job\_id | String | 否 | 已经运行完的旧作业ID，用于作业重启功能，新作业DAG配置中的第一个组件输入关联到该旧job\_id的组件 |

1. 返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| job\_id | String | 是 | 作业ID |

* + 1. **停止作业**

1. 接口名称：stop\_job\_all
2. 接口描述：发起方向己方调度层发起停止作业时调用的接口
3. 调用方式：POST
4. 接口地址：/v1/platform/schedule/job/stop\_all
5. 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| job\_id | String | 是 | 作业ID |

1. 返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |

* + 1. **停止任务**

1. 接口名称：stop\_task
2. 接口描述：发起方向己方调度层发起停止任务时调用的接口
3. 调用方式：POST
4. 接口地址：/v1/platform/schedule/job/stop\_task
5. 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| job\_id | String | 是 | 作业ID |
| task\_name | String | 是 | 组件名称，对应作业DAG中起点的component\_name |

1. 返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |

* + 1. **查询作业列表【可选】**

1. 接口名称：query\_job\_list
2. 接口描述：某个参与方向己方调度层发起查询作业列表时调用的接口
3. 调用方式：GET
4. 接口地址：/v1/platform/schedule/job/query\_job\_list
5. 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| flow\_id | String | 是 | 流程ID |

1. 返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| job\_list | Map<Key, Value> | 是 | 作业列表，key为job\_id，value为作业的状态，作业状态有PENDING、READY、REJECTED、RUNNING、FINISHED |

* + 1. **查询作业状态**

1. 接口名称：job\_status\_all
2. 接口描述：某个参与方向己方调度层发起查询作业状态时调用的接口
3. 调用方式：GET
4. 接口地址：/v1/platform/schedule/job/status\_all
5. 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| job\_id | String | 是 | 作业ID |

1. 返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| job\_status | String | 是 | 作业状态，状态有PENDING、READY、REJECTED、RUNNING、FINISHED |
| status | Map<Key, Value> | 否 | 作业中所有任务的状态，key为task\_id，value为任务的状态，任务状态有READY、RUNNING、SUCCESS、FAILED |

* + 1. **查询任务日志行数**

1. 接口名称：get\_task\_log\_line
2. 接口描述：某个参与方节点上层服务向调度层获取任务运行日志行数时调用的接口
3. 调用方式：GET
4. 接口地址：/v1/platform/schedule/task/get\_log\_line
5. 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| task\_id | String | 是 | 任务ID |
| log\_level | String | 是 | 日志类型，info/debug/error |

1. 返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| num | Integer | 是 | 日志行数 |

* + 1. **查询任务日志**

1. 接口名称：get\_task\_log
2. 接口描述：某个参与方节点上层服务向调度层获取任务运行日志内容时调用的接口
3. 调用方式：GET
4. 接口地址：/v1/platform/schedule/task/get\_log
5. 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| task\_id | String | 是 | 任务ID |
| log\_level | String | 是 | 日志类型，info/debug/error |
| start | Integer | 是 | 日志起始行 |
| length | Integer | 否 | 查询的日志行数，为空则返回起始行后的所有 |

1. 返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| data | String | 是 | 日志内容 |

* + 1. **回调任务信息【必选】**

1. 接口名称：callback
2. 接口描述：某个参与方的算法组件层将任务运行的状态等回调信息推送给调度层时调用的接口
3. 调用方式：POST
4. 接口地址：/v1/platform/schedule/task/callback
5. 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| task\_id | String | 是 | 任务ID |
| status | String | 是 | 任务状态，RUNNING、SUCCESS、FAILED |
| role | String | 是 | 回调任务所属角色 |

1. 返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |

* 1. **东西向接口**
     1. **创建作业（调度方）**

1. 接口名称：create\_job\_all
2. 接口描述：发起方向调度方发送创建作业请求时调用的接口
3. 调用方式：POST
4. 接口地址：/v1/interconn/schedule/job/create\_all
5. 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| job\_id | String | 是 | 作业ID |
| flow\_id | String | 否 | 流程ID |
| dag | DAG | 是 | 描述组件组合流程的配置信息 |
| config | Config | 是 | 作业参与方节点信息和每个组件参数配置等等 |
| old\_job\_id | String | 否 | 已经运行完的旧作业ID，用于作业重启功能，新作业DAG配置中的第一个组件输入关联到该旧job\_id的组件 |

1. 返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |

* + 1. **创建作业**

1. 接口名称：create\_job
2. 接口描述：发起方或调度方向所有参与方发送创建作业请求时调用的接口
3. 调用方式：POST
4. 接口地址：/v1/interconn/schedule/job/create
5. 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| job\_id | String | 是 | 作业ID |
| flow\_id | String | 否 | 流程ID |
| dag | DAG | 是 | 描述组件组合流程的配置信息 |
| config | Config | 是 | 作业参与方节点信息和每个组件参数配置等等 |
| old\_job\_id | String | 否 | 已经运行完的旧作业ID，用于作业重启功能，新作业DAG配置中的第一个组件输入关联到该旧job\_id的组件 |

1. 返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |

* + 1. **启动作业**

1. 接口名称：start\_job
2. 接口描述：发起方或调度方向所有参与方发送启动作业请求时调用的接口，每个参与方只会被请求一次，当某个参与方包含多个角色时，有参与方自己根据作业配置将不同角色的任务启动
3. 调用方式：POST
4. 接口地址：/v1/interconn/schedule/job/start
5. 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| job\_id | String | 是 | 作业ID |

1. 返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |

* + 1. **启动任务**

1. 接口名称：start\_task
2. 接口描述：发起方或调度方向所有参与方发送启动任务请求时调用的接口
3. 调用方式：POST
4. 接口地址：/v1/interconn/schedule/task/start
5. 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| task\_id | String | 是 | 任务ID，由发起方生成 |
| job\_id | String | 是 | 作业ID |
| task\_name | String | 是 | 组件名称，对应DAG中的component\_name |

1. 返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |

* + 1. **停止作业（调度方）**

1. 接口名称：stop\_job\_all
2. 接口描述：发起方向调度方发送停止作业请求时调用的接口
3. 调用方式：POST
4. 接口地址：/v1/interconn/schedule/job/stop\_all
5. 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| job\_id | String | 是 | 作业ID |
| task\_name | String | 否 | 组件名称，对应作业DAG中起点的component\_name  当该值不为空时，该任务被停止后，其后的所有任务都停止  当该值为空时，停止作业下的所有任务 |

1. 返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |

* + 1. **停止作业**

1. 接口名称：stop\_job
2. 接口描述：发起方或调度方向所有参与方发送停止作业请求时调用的接口
3. 调用方式：POST
4. 接口地址：/v1/interconn/schedule/job/stop
5. 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| job\_id | String | 是 | 作业ID |
| task\_name | String | 否 | 组件名称，对应作业DAG中起点的component\_name  当该值不为空时，该任务被停止后，其后的所有任务都停止  当该值为空时，停止作业下的所有任务 |

1. 返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |

* + 1. **查询作业状态**

1. 接口名称：job\_status\_all
2. 接口描述：某个参与方向发起方发起查询作业状态时调用的接口
3. 调用方式：GET
4. 接口地址：/v1/interconn/schedule/job/status\_all
5. 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| job\_id | String | 是 | 作业ID |

1. 返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| job\_status | String | 是 | 作业状态，状态有PENDING、READY、REJECTED、RUNNING、FINISHED |
| status | Map<Key, Value> | 否 | 作业中所有任务的状态，key为task\_id，value为任务的状态，任务状态有READY、RUNNING、SUCCESS、FAILED |
| reason | String | 否 | 用于说明作业状态为REJECTED时的原因，其他作业状态时可为空 |

* + 1. **查询作业状态（调度方）**

1. 接口名称：job\_status\_all
2. 接口描述：某个参与方向调度方发起查询作业状态时调用的接口
3. 调用方式：GET
4. 接口地址：/v1/interconn/schedule/job/status\_all
5. 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| job\_id | String | 是 | 作业ID |

1. 返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| job\_status | String | 是 | 作业状态，状态有PENDING、READY、REJECTED、RUNNING、FINISHED |
| status | Map<Key, Value> | 否 | 作业中所有任务的状态，key为task\_id，value为任务的状态，任务状态有READY、RUNNING、SUCCESS、FAILED |

* + 1. **作业审批确认**

1. 接口名称：audit\_confirm
2. 接口描述：某个参与方向调度方或发起方作业审批结果时调用的接口
3. 调用方式：POST
4. 接口地址：/v1/interconn/schedule/job/audit\_confirm
5. 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| job\_id | String | 是 | 作业ID |
| status | String | 是 | 作业状态，READY、REJECTED |

1. 返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |

* + 1. **查询任务回调信息**

1. 接口名称：poll\_callback
2. 接口描述：发起方或调度方向所有参与方查询任务回调信息时调用的接口
3. 调用方式：POST
4. 接口地址：/v1/interconn/schedule/task/poll
5. 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| task\_id | String | 是 | 任务ID |
| role | String | 是 | 当前回调的节点角色 |

1. 返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| status | String | 是 | 任务状态，PENDING、READY、RUNNING、SUCCESS、FAILED |

* + 1. **推送任务回调信息**

1. 接口名称：push\_callback
2. 接口描述：任意参与方向发起方或调度方向推送任务回调信息时调用的接口
3. 调用方式：POST
4. 接口地址：/v1/interconn/schedule/task/callback
5. 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |
| task\_id | String | 是 | 任务ID |
| status | String | 是 | 任务状态，PENDING、READY、RUNNING、SUCCESS、FAILED |
| role | String | 是 | 当前回调的节点角色 |

1. 返回参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必选 | 描述 |

1. **算法容器设计目标及约定**

* 包容关系：一个镜像可以包含一个或者多个组件
* 生命周期：算法运行的容器为即用即销毁形式
* 计算模式：离线模式（算法之间的计算数据传递使用静态的文件）
* 描述方式：完全无文件的方式为主

1. **算法容器相关名称定义**

* 算法包：package
* 算法操作：component
* 组件自描述部分: component self-description part
* 运行环境部分： running environment part
* 动态运行部分: dynamic runtime information part

1. **算法镜像定义**
   1. **镜像总体描述**

|  |  |  |  |
| --- | --- | --- | --- |
| **分类** | **子项目** | **获取位置** | **数据内容** |
| 组件自描述部分 | 算法包自描述 | LABEL | k=v, 算法包的信息描述 |
| 算法功能自描述 | LABEL | k=v, 镜像所包含组件 |
| 算法参数自描述 | LABEL | k=v, 各算法输入参数 |
| 算法输入数据自描述 | LABEL | k=v, 各算法输入数据描述 |
| 算法输出数据自描述 | LABEL | k=v, 各算法输出数据描述 |
|  |  |  |  |
| 运行环境部分 | 支撑系统接口 | ENV | k=v, 系统接口地址 |
| 配置信息 | ENV | k=v, 配置选项内容 |
|  |  |  |  |
| 动态运行部分 | 算法名称 | ENV | k=v, 算法子标识字符串 |
| 算法参数 | ENV | k=v, 算法参数 |
| 算法输入数据 | ENV | k=v, 输入数据文件标识 |
| 算法输出数据 | ENV | k=v, 系统分配的输出数据文件标识 |
|  |  |  |  |

示例

* 1. **键名定义规范**

**名称定义：**<部分名>.<部分序号>.<子项目>.<子项目序号>.<条目名称>.<条目序号>

**示例：**

* component.1.name： 第1个算法组件的名称
* component.1.parameter.2.name： 第1个算法组件的第2个参数名称
  1. **组件运行状态返回规范**

运行状态获取：再容器结束后，通过容器内的主进程的运行结果作为当前组件的状态。取值范围为含零非负整数。

* 状态码等于0：表示运行成功
* 状态码大于0: 表示运行失败
  1. **组件日志**

组件运行的需要上报的日志，通过容器的标准输出stdout的内容，作为组件运行的日志；日志内容可以包括来自组件运行期间不同模块，等级，线程等产生的日志输出，以行作为最小日志记录单元。

1. **组件自描述部分定义**
   1. **算法功能自描述**

|  |  |  |  |
| --- | --- | --- | --- |
| **标识名称** | **可选性** | **标识含义** | **描述** |
| specification.version | 必选 | 遵循的规范版本号 | 1.0.0 |
| package.name | 必选 | 组件名称 | xgboost |
| package.description | 可选 | 组件描述 |  |
| package.version | 必选 | 组件版本 | 1.0.0 |
| component.<comp\_n>.name | 必选 | 第n算法组件名称 | 例如xgboost\_infer |
| component.<comp\_n>.version | 可选 | 第n算法组件版本 | 继承component的version |
| component.<comp\_n>.description | 可选 | 第n算法组件描述 |  |
| component.<comp\_n>.role\_list | 必选 | 算法组件实现角色列表 | guest/host/arbiter |

示例：

LABEL：

specification.version=1.0.0

package.name=\*\*\*算法组件

package.description=\*\*\*的算法组件

package.version=1.0.0

component.1.name=hetero\_logistic\_regression

component.1.version=1.0.0

component.1.description=纵向逻辑回归算法

component.1.role\_list=guest,host,arbiter

component.2.name=xgboost

component.2.version=1.0.0

component.2.description=XGBoost算法

component.2.role\_list=guest,host

* 1. **算法参数自描述**

|  |  |  |  |
| --- | --- | --- | --- |
| **标识名称** | **可选性** | **标识含义** | **描述** |
| component.<comp\_n>.parameter.<param\_n>.name | 必选 | 第n参数名称 |  |
| component.<comp\_n>.parameter.<param\_n>.description | 可选 | 第n参数描述 |  |
| component.<comp\_n>.parameter.<param\_n>.type | 必选 | 第n参数类型 | integer/string/boolean/float |

示例：

LABEL：

component.1.parameter.1.name=max\_iter

component.1.parameter.1.description=最大迭代次数

component.1.parameter.1.type=integer

component.1.parameter.2.name=learning\_rate

component.1.parameter.2.description=学习率

component.1.parameter.2.type=float

component.1.parameter.3.name=init\_method

component.1.parameter.3.description=系数初始化方法

component.1.parameter.3.type=string

* 1. **算法输入数据自描述**

|  |  |  |  |
| --- | --- | --- | --- |
| **标识名称** | **可选性** | **标识含义** | **描述** |
| component.<comp\_n>.input.<in\_n>.name | 必选 | 第n个输入名称 |  |
| component.<comp\_n>.input.<in\_n>.description | 可选 | 第n个输入描述 |  |
| component.<comp\_n>.input.<in\_n>.categories | 必选 | 第n个输入类型 | model/dataset/training\_set/test\_set/report |

示例：

LABEL：

component.1.input.1.name=train\_data

component.1.input.1.description=训练集数据

component.1.input.1.categories=training\_set

component.1.input.2.name=test\_data

component.1.input.2.description=测试集数据

component.1.input.2.categories=test\_set

component.1.input.3.name=model

component.1.input.3.description=输入模型

component.1.input.3.categories=model

* 1. **算法输出数据自描述**

|  |  |  |  |
| --- | --- | --- | --- |
| **标识名称** | **可选性** | **标识含义** | **描述** |
| component.<comp\_n>.output.<out\_n>.name | 必选 | 第n个输出名称 |  |
| component.<comp\_n>.output.<out\_n>.description | 可选 | 第n个输出描述 |  |
| component.<comp\_n>.output.<out\_n>. categories | 必选 | 第n个输出类型 | model/dataset/training\_set/test\_set/report |

示例：

LABEL：

component.1.ouput.1.name=train\_data

component.1.ouput.1.description=训练集数据

component.1.ouput.1.categories=training\_set

component.1.ouput.2.name=test\_data

component.1.ouput.2.description=测试集数据

component.1.ouput.2.categories=test\_set

component.1.ouput.3.name=model

component.1.ouput.3.description=输出模型

component.1.ouput.3.categories=model

component.1.ouput.3.name=train\_auc

component.1.ouput.3.description=训练集AUC

component.1.ouput.3.categories=report

1. **运行环境部分定义**
   1. **系统支撑接口**

|  |  |  |  |
| --- | --- | --- | --- |
| **标识名称** | **可选性** | **标识含义** | **描述** |
| system.storage | 必选 | 存储层服务地址 |  |
| system.callback | 可选 | 调度层回调地址 | 可通过接口回调，也可通过容器运行结果，或其他 |
| system.transport | 必选 | 传输服务地址 |  |
| system.compute | 可选 | 计算引擎服务地址 |  |

示例：

ENV：

system.storage=s3://192.168.1.1:9000?username=admin&password=123456

system.callback=http://127.0.0.1:8080/v1/schedule/task/callback

system.transport=127.0.0.1:8080

system.compute=spark://192.168.1.1:7077

* 1. **组件配置信息**

|  |  |  |  |
| --- | --- | --- | --- |
| **标识名称** | **可选性** | **标识含义** | **描述** |
| config.self\_role | 必选 | 当前组件运行任务的角色+index，多个是用英文逗号分割 | guest/host/arbiter |
| config.task\_id | 必选 | 组件运行的任务ID |  |
| config.node\_id.<role>.<role\_n> | 必选 | 角色对应的节点ID | 数据传输时需要 |
| config.inst\_id.<role>.<role\_n> | 必选 | 角色对应的机构ID |
| config.session\_id | 必选 | 传输SDK需要的会话ID |
| config.trace\_id | 必选 | 链路ID |
| config.token | 必选 | 对应管理面ResourcePermit实体中的token，用于传输接收方进行权限校验  **注：目前两个以上参与方时暂不确定传参方式，所以暂不建议对此进行字段进行校验** |
| config.log.path | 必选 | log日志路径（容器内） |  |
| config.n\_threads | 可选 | 存储SDK上传、下载数据的并行数 | 存储层需要 |
| config.chunk\_size | 可选 | 存储SDK上传、下载数据的块大小 |

示例：

ENV：

config.self\_role=guest.0

config.task\_id=task\_123456

config.node\_id.guest.0=node\_9999

config.node\_id.host.0=node\_10000

config.node\_id.host.1=node\_10001

config.inst\_id.guest.0=inst\_9999

config.inst\_id.host.0=inst\_10000

config.inst\_id.host.1=inst\_10001

config.session\_id=session\_123456

config.trace\_id=trace\_123456

config.token=token\_123456

config.log.path=/opt/task\_123456/

config.n\_threads=5

config.chunk\_size=1024

1. **动态运行部分定义**
   1. **算法组件名称**

|  |  |  |  |
| --- | --- | --- | --- |
| **标识名称** | **可选性** | **标识含义** | **描述** |
| runtime.component.name | 必选 | 当前运行的组件名称 |  |

示例：

ENV：

runtime.component.name=hetero\_logistic\_regression

* 1. **算法组件参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **标识名称** | **可选性** | **标识含义** | **描述** |
| runtime.component.parameter.<name> | 必选 | 组件运行期的参数<name>对应的值 |  |

示例：

ENV：

runtime.component.parameter.max\_iter=5

runtime.component.parameter.learning\_rate=0.01

runtime.component.parameter.init\_method=init\_zero

runtime.component.parameter.id=ID

runtime.component.parameter.label=Y

* 1. **算法输入数据**

|  |  |  |  |
| --- | --- | --- | --- |
| **标识名称** | **可选性** | **标识含义** | **描述** |
| runtime.component.input.<name> | 必选 | 组件运行期的输入<name>对应的文件标识 |  |

示例：

ENV：

runtime.component.input.train\_data={‘namespace’:’123’, ‘name’:’456’}

runtime.component.input.test\_data={‘namespace’:’123’, ‘name’:’456’}

* 1. **算法输出数据**

|  |  |  |  |
| --- | --- | --- | --- |
| **标识名称** | **可选性** | **标识含义** | **描述** |
| runtime.component.output.<name> | 必选 | 组件运行期的输出<name>对应的文件标识 |  |

示例：

ENV：

runtime.component.output.train\_data={‘namespace’:’123’, ‘name’:’456’}

runtime.component.output.test\_data={‘namespace’:’123’, ‘name’:’456’}

runtime.component.output.train\_auc={‘namespace’:’123’, ‘name’:’456’}

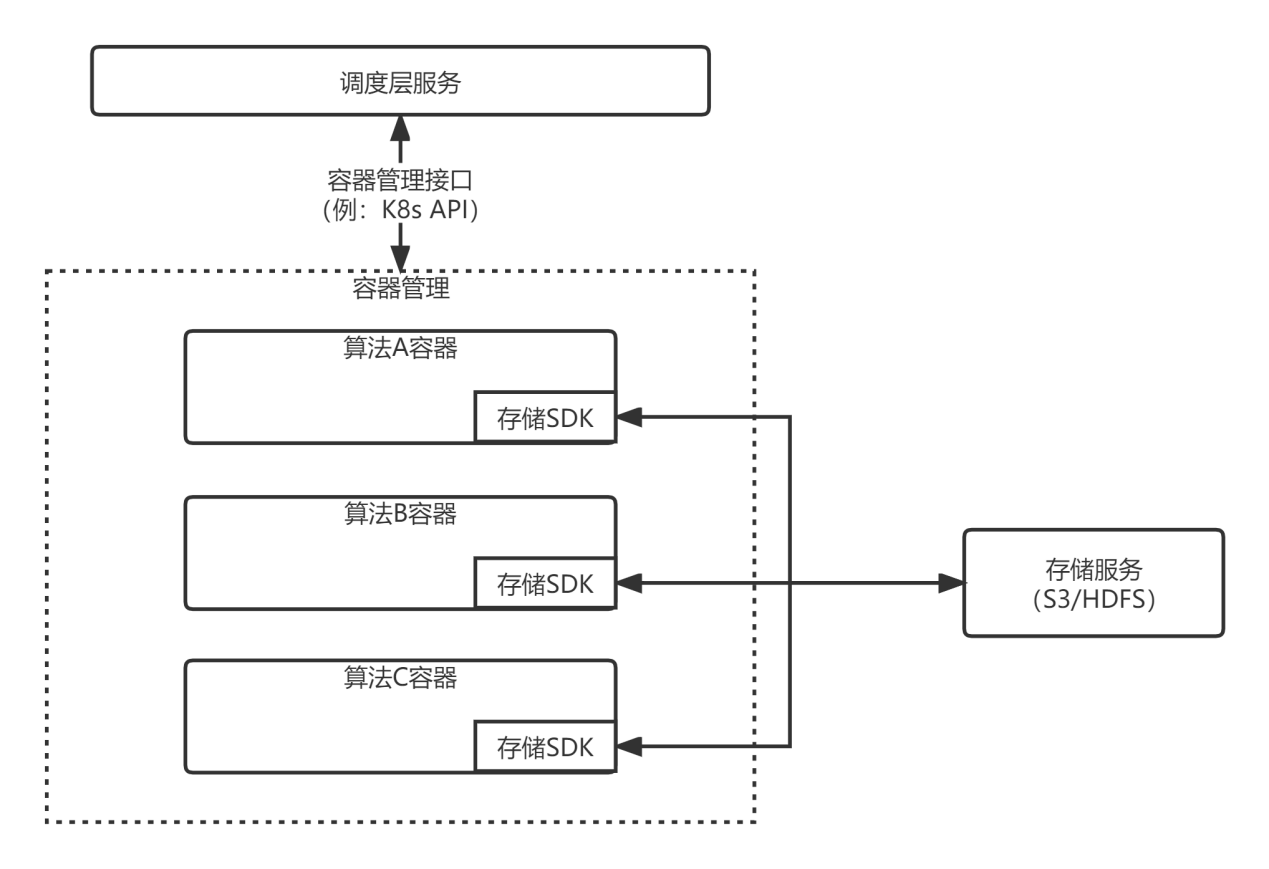
1. **数据存储层**
   1. **实体定义**
      1. **数据实体（Table）**

|  |  |  |
| --- | --- | --- |
| **属性名称** | **类型** | **说明** |
| name | String | 数据表名称 |
| namespace | String | 数据表命名空间 |
| count | Integer | 数据量 |
| partitions | Integer | 分区数量 |
| column\_info | List<ColumnInfo> | 特征的相关信息，比如特征的类型等 |
| description | String | 数据描述信息 |
| create\_time | Datetime | 数据创建时间 |
| expire\_time | Datetime | 数据有效截止时间 |
| file\_type | String | 数据集文件类型 |
| data\_type | String | 数据集类型，dense/sparse |

* + 1. **特征信息（ColumnInfo）**

|  |  |  |
| --- | --- | --- |
| **属性名称** | **类型** | **说明** |
| name | String | 特征名称 |
| type | String | 特征类型：int、string、float、boolean等 |

* 1. **框架图**



* 1. **方法定义**
     1. **创建客户端对象**

创建S3、HDFS、Mysql的session连接，SDK会自动根据存储服务配置ENV连接存储服务

**方法名：**init\_session()

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **是否必填** | **说明** |

**返回参数：**

|  |  |  |
| --- | --- | --- |
| **属性名称** | **类型** | **说明** |
| session | Session | 客户端连接 |

**实现方式：**

**S3 API：**

boto3.resource(service\_name="s3",endpoint\_url="http://192.168.1.1:9200",use\_ssl=False,aws\_access\_key\_id="\*\*\*",aws\_secret\_access\_key="\*\*\*")

无

**HDFS API：**

* + 1. **创建新数据表**

**方法名：**session.create\_table(name, namespace, column\_info, partition, description)

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **是否必填** | **说明** |
| name | String | 是 | 数据表名称 |
| namespace | String | 是 | 数据表命名空间，不能有- |
| column\_info | ColumnInfo | 否 | 数据表的特征信息 |
| partition | Integer | 否 | 数据表的分区数 |
| description | String | 否 | 数据表的描述信息 |

**返回参数：**

|  |  |  |
| --- | --- | --- |
| **属性名称** | **类型** | **说明** |
| table | Table | 数据表对象 |

**实现方式：**

**S3 API：**

obj = s3.Object(bucket\_name="storage", key="{}-{}".format(namespace, name))

obj.load()

obj\_meta = s3.Object(bucket\_name="storage", key="{}-{}.metadata".format(namespace, name))

https://docs.aws.amazon.com/zh\_cn/AmazonS3/latest/API/API\_HeadObject.html

**HDFS API：**

* + 1. **获取已有数据表**

**方法名：**session.get\_table(name, namespace)

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **是否必填** | **说明** |
| name | String | 是 | 数据表名称 |
| namespace | String | 是 | 数据表命名空间 |

**返回参数：**

|  |  |  |
| --- | --- | --- |
| **属性名称** | **类型** | **说明** |
| table | Table | 数据表对象 |

**实现方式：**

**S3 API：**

obj = s3.Object(bucket\_name="storage", key="{}-{}".format(namespace, name))

obj\_meta = s3.Object(bucket\_name="storage", key="{}-{}.metadata".format(namespace, name))

obj\_meta.download\_fileobj()

https://docs.aws.amazon.com/zh\_cn/AmazonS3/latest/API/API\_GetObject.html

**HDFS API：**

* + 1. **上传本地文件**

**方法名：**table.upload(local\_path，overwrite, callback\_func)

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **是否必填** | **说明** |
| local\_path | String | 是 | 本地文件路径 |
| overwrite | Boolean | 否 | 是否覆盖，默认为True |
| callback\_func | String | 否 | 每块文件上传后调用的方法 |

**返回参数：**

|  |  |  |
| --- | --- | --- |
| **属性名称** | **类型** | **说明** |

**实现方式：**

**S3 API：**

obj.upload\_file(local\_path)

https://docs.aws.amazon.com/zh\_cn/AmazonS3/latest/API/API\_PutObject.html

**HDFS API：**

* + 1. **下载到本地**

**方法名：**table.download(local\_path, overwrite, callback\_func)

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **是否必填** | **说明** |
| local\_path | String | 是 | 本地文件路径 |
| overwrite | Boolean | 否 | 是否覆盖，默认为True |
| callback\_func | String | 否 | 每块文件下载后调用的方法 |

**返回参数：**

|  |  |  |
| --- | --- | --- |
| **属性名称** | **类型** | **说明** |

**实现方式：**

**S3 API：**

obj.download\_file(local\_path)

https://docs.aws.amazon.com/zh\_cn/AmazonS3/latest/API/API\_GetObject.html

**HDFS API：**

* + 1. **读取文件**

**方法名：**table.read(partition\_index, callback\_func)

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **是否必填** | **说明** |
| partition\_index | Integer | 否 | 分区序号，未进行分区则不需要传参 |
| callback\_func | String | 否 | 每块文件下载后调用的方法 |

**返回参数：**

|  |  |  |
| --- | --- | --- |
| **属性名称** | **类型** | **说明** |
| data | Bytes | 数据内容 |

**实现方式：**

**S3 API：**

obj.download\_fileobj(file\_stream)

https://docs.aws.amazon.com/zh\_cn/AmazonS3/latest/API/API\_GetObject.html

**HDFS API：**

* + 1. **写文件**

**方法名：**table.write(data, partition\_id, overwrite, callback\_func)

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **是否必填** | **说明** |
| data | Bytes | 是 | 数据内容 |
| partition\_id | Integer | 否 | 分区ID，未进行分区则不需要传参 |
| overwrite | Boolean | 否 | 是否覆盖，默认为True |
| callback\_func | String | 否 | 每块文件下载后调用的方法 |

**返回参数：**

|  |  |  |
| --- | --- | --- |
| **属性名称** | **类型** | **说明** |

**实现方式：**

**S3 API：**

obj.upload\_fileobj(file\_stream)

https://docs.aws.amazon.com/zh\_cn/AmazonS3/latest/API/API\_PutObject.html

**HDFS API：**

* + 1. **获取存储信息**

**方法名：**table.get\_path()

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **是否必填** | **说明** |

**返回参数：**

|  |  |  |
| --- | --- | --- |
| **属性名称** | **类型** | **说明** |
| path | String | S3存储路径、HDFS存储路径  例如：s3://{$namespace}/{name}  hdsf://{$namespace}/{name} |

**实现方式：**

**S3 API：**

"s3://storage/" + namespace + "-" + name

无

**HDFS API：**

* + 1. **设置特征信息**

**方法名：**table.set\_column\_info(column\_info)

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **是否必填** | **说明** |
| column\_info | List<ColumnInfo> | 是 | 特征的相关信息 |

**返回参数：**

|  |  |  |
| --- | --- | --- |
| **属性名称** | **类型** | **说明** |

**实现方式：**

**S3 API：**

obj\_meta.upload\_fileobj(file\_stream)

https://docs.aws.amazon.com/zh\_cn/AmazonS3/latest/API/API\_PutObject.html

**HDFS API：**

* + 1. **获取特征信息**

**方法名：**table.get\_column\_info()

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **是否必填** | **说明** |

**返回参数：**

|  |  |  |
| --- | --- | --- |
| **属性名称** | **类型** | **说明** |
| column\_info | List<ColumnInfo> | 特征的相关信息 |

**实现方式：**

**S3 API：**

无

**HDFS API：**

* + 1. **设置description**

**方法名：**table.set\_description(description)

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **是否必填** | **说明** |
| description | String | 是 | 数据描述信息 |

**返回参数：**

|  |  |  |
| --- | --- | --- |
| **属性名称** | **类型** | **说明** |

**实现方式：**

**S3 API：**

obj\_meta.upload\_fileobj(file\_stream)

https://docs.aws.amazon.com/zh\_cn/AmazonS3/latest/API/API\_PutObject.html

**HDFS API：**

* + 1. **获取分区集合**

**方法名：**table.get\_partitions()

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **是否必填** | **说明** |

**返回参数：**

|  |  |  |
| --- | --- | --- |
| **属性名称** | **类型** | **说明** |
| partitions | Array<Integer> | 数据表的分区ID集合 |

**实现方式：**

**S3 API：**

无

**HDFS API：**

* + 1. **删除数据**

**方法名：**session.delete\_table(name, namespace)

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **是否必填** | **说明** |
| name | String | 是 | 数据表名称 |
| namespace | String | 是 | 数据表命名空间 |

**返回参数：**

|  |  |  |
| --- | --- | --- |
| **属性名称** | **类型** | **说明** |

**实现方式：**

**S3 API：**

obj.delete()

obj\_meta.delete()

https://docs.aws.amazon.com/zh\_cn/AmazonS3/latest/API/API\_DeleteObject.html

**HDFS API：**

* + 1. **判断数据是否存在**

**方法名：**session.is\_exist(name, namespace)

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **类型** | **是否必填** | **说明** |
| name | String | 是 | 数据表名称 |
| namespace | String | 是 | 数据表命名空间 |

**返回参数：**

|  |  |  |
| --- | --- | --- |
| **属性名称** | **类型** | **说明** |
| is\_exist | Boolean | 是否存在标志 |

**实现方式：**

**S3 API：**

obj.load()->404

https://docs.aws.amazon.com/zh\_cn/AmazonS3/latest/API/API\_HeadObject.html

**HDFS API：**