**AiFlow--StartModule需求文档**

1. **概述：**

本需求文档描述了一个基于Airflow框架的任务执行脚本，该脚本可根据传入的事件类型执行不同类型的任务。脚本通过与AWS Glue和PostgreSQL数据库进行交互，实现作业的启动和状态查询。

**二. 功能需求：**

1. 支持根据传入的事件类型执行不同类型的任务，目前支持的任务类型包括Glue和Python。
2. 在执行Glue任务时，能够根据作业的最后一次运行状态判断是否需要启动新的作业运行。
3. 支持查询作业参数和脚本位置，这些信息存储在PostgreSQL数据库中。
4. 能够将作业的运行状态更新到数据库中，包括作业运行中、完成等状态。
5. 支持查询作业的最后一次运行状态。
6. 具备良好的日志记录功能，能够记录错误信息和作业的执行日志。

**三．技术需求**

1. 使用Airflow框架作为任务调度和管理工具。
2. 使用AWS Glue作为执行作业的服务，通过Glue客户端与Glue服务进行交互。
3. 使用PostgreSQL数据库存储作业参数和作业状态信息，通过PostgresHandler类与数据库进行交互。
4. 需要具备访问AWS Glue和PostgreSQL数据库的权限。
5. 脚本需要运行在Python解释器环境中，并安装所需的依赖模块。

**四: 输入/输出**

**1.输入**

脚本通过一个名为"event"的字典参数接收输入，该字典包含以下字段：

* "datasource\_name"：数据源名称，字符串类型。
* "load\_type"：加载类型，字符串类型。
* "run\_type"：运行类型，字符串类型，可选值为"glue"或"python"。
* "glue\_template\_name"：Glue模板名称，字符串类型。

**2.输出**

脚本根据执行的任务类型和状态输出不同的日志信息。

**五. 数据库设计**

脚本通过PostgreSQL数据库存储作业参数和作业状态信息。

**数据库表信息**

1. dim\_job\_params：作业参数表，包含以下字段

* job\_name：作业名称，字符串类型。
* task\_name：任务名称，字符串类型。
* is\_active：是否激活，字符串类型，可选值为"Y"或"N"。
* job\_priority：作业优先级，整数类型。

|  |  |  |
| --- | --- | --- |
| **column\_name** | **column\_type** | **column\_Desc** |
| job\_name | varchar(500) NOT NULL | Job的名字 |
| task\_name | varchar(500) NOT NULL | Task的名字 |
| job\_type | varchar(255) | Job的类型(Glue/python/lambda) |
| is\_data\_quality\_check | varchar(255) | 是否进行数据质量检查 |
| job\_priority | integer | Job的优先级 |
| max\_retries | integer | Job的最多retry次数 |
| load\_type | varchar(255) | Job的加载类型（全量/增量） |
| s3\_bucket | varchar(500) | Job脚本存储的S3桶 |
| s3\_location | varchar(1000) | Job脚本存储的路径 |
| is\_active | varchar(40) | Job是否有效 |
| insert\_date | timestamp | 插入时间 |
| last\_update\_date | timestamp | 最后更新时间 |

2. dim\_job\_params：作业参数表，包含以下字段：

* job\_name：作业名称，字符串类型。
* param\_name：参数名称，字符串类型。
* param\_value：参数值，字符串类型。
* is\_active：是否激活，字符串类型，可选值为"Y"或"N"。

|  |  |  |  |
| --- | --- | --- | --- |
| **column\_name** | **column\_Desc** | **column\_Desc** | **Sample data** |
| params\_id | int not null | 参数的id,自增ID | 1 |
| job\_name | varchar(500) NOT NULL | Job的名字 |  |
| param\_name | varchar(1000) NOT NULL | 参数的名字 |  |
| param\_value | varchar(1000) NOT NULL | 参数的值 |  |
| is\_active | varchar(10) | 是否有效 |  |
| insert\_date | timestamp | 插入时间 |  |
| last\_update\_date | timestamp | 最后更新时间 |  |

1. fact\_job\_details：作业详情表，包含以下字段：

* dag\_name：DAG名称，字符串类型。
* task\_name：任务名称，字符串类型。
* job\_name：作业名称，字符串类型。
* job\_type：作业类型，字符串类型。
* run\_id：作业运行ID，字符串类型。
* latest\_heartbeat：最新心跳时间，日期时间类型。
* job\_start\_date：作业开始时间，日期时间类型。
* job\_end\_date：作业结束时间，日期时间类型。
* duration：作业持续时间，整数类型。
* job\_status：作业状态，字符串类型。
* insert\_date：插入日期，日期类型。

last\_update\_date：最后更新日期，日期类型。

|  |  |  |  |
| --- | --- | --- | --- |
| **column\_name** | **column\_type** | **column\_Desc** | **Sample data** |
| id | int not null | Job run表的id,自增ID'; | 1 |
| dag\_name | integer | Dag 的id'; |  |
| job\_name | integer | Task的id'; |  |
| task\_name | integer | Job 的id'; |  |
| job\_type | varchar(255) | Job的类型(Glue/python/lambda) |  |
| run\_id | varchar(500) | Job 的run id'; |  |
| latest\_heartbeat | timestamp | Job 最近一次run的时间'; |  |
| job\_start\_date | timestamp | Job run的开始时间'; |  |
| job\_end\_date | timestamp | Job run的结束时间'; |  |
| duration | decimal(12,6) | Job run的时长'; |  |
| job\_status | varchar(500) | Job run的状态'; |  |
| insert\_date | timestamp | 插入时间'; |  |
| last\_update\_date | timestamp | 最后更新时间'; |  |

**六． 接口设计**

**1. run\_glue\_task(task\_name)**

执行Glue任务的方法。根据传入的任务名称查询相关的作业信息，并逐个启动作业运行。

* 参数

task\_name (字符串): 任务名称，用于查询相关的作业信息。

* 返回值

无

**2. query\_status\_job(job\_name)**

查询作业的最后一次运行状态的方法。根据作业名称从数据库中查询作业的最后一次运行状态，并返回结果。

* 参数

job\_name (字符串): 作业名称，用于查询作业的最后一次运行状态。

* 返回值

Last\_run\_status (字符串): 最后一次作业运行的状态。

**3. get\_db\_hander()**

获取数据库处理器的方法。返回一个PostgresHandler实例，用于处理PostgreSQL数据库。

* 参数

无

* 返回值

db\_handler (PostgresHandler对象): 用于处理PostgreSQL数据库的数据库处理器。

**4. query\_job\_by\_task(task\_name)**

根据任务名称查询相关的作业信息的方法。从数据库中查询与任务名称相关的作业，并返回作业字典。

* 参数

task\_name (字符串): 任务名称，用于查询相关的作业信息。

* 返回值

jobs (字典): 包含作业名称和参数的字典。

**5. query\_job\_info(job\_name)**

查询作业信息的方法。根据作业名称查询作业参数和脚本位置，并将结果构建成作业字典返回。

* 参数

job\_name (字符串): 作业名称，用于查询作业信息。

* 返回值

job (字典): 包含作业名称和参数的字典。

**6. query\_scriptLocation(job\_name)**

查询脚本位置的方法。根据作业名称从数据库中查询脚本位置，并返回结果。

* 参数

job\_name (字符串): 作业名称，用于查询脚本位置。

* 返回值

scriptLocation (字符串): 脚本位置。

**7. start\_glue\_run(name, param)**

启动Glue作业运行的方法。根据传入的作业名称和参数调用Glue客户端的start\_job\_run方法开始作业运行，并返回作业运行的ID。

* 参数

name (字符串): 作业名称，用于启动作业运行。

param (字典): 作业参数，包含作业运行所需的参数信息。

* 返回值

run\_id (字符串): 作业运行的ID。

**8. query\_param(job\_name)**

查询作业参数的方法。根据作业名称从数据库中查询作业参数，并返回结果。

* 参数

job\_name (字符串): 作业名称，用于查询作业参数。

* 返回值

param\_re (字典): 作业参数，包含作业运行所需的参数信息。

**七． 日志记录**

脚本使用Python的logging模块进行日志记录，其中包括错误信息和作业的执行日志。