



Getting Results From PostgreSQL Using Hooks

Let's use PostgresHook



Instructions

- `cd ~`
- `cp ~/airflow_files/hook_dag.py ~/airflow/dags/`
- If airflow scheduler and webserver are running, restart them by doing:
 - `Ctrl-C` in the webserver and scheduler terminal to stop them.
 - Then in each terminal, `airflow webserver`, `airflow scheduler`.
- `vim ~/airflow/dags/hook_dag.py` to take a look of the code



Instructions

- The hook is used into the function `get_activated_sources` which basically fetch all the data from the table `course.source` and print each source to the `stdout`. `get_activated_sources` is called by a `PythonOperator`.

```
def get_activated_sources():
    request = "SELECT * FROM course.source"
    pg_hook = PostgresHook(postgre_conn_id="postgre_sql", schema="airflow_mdb") # This connection must be set from the Connection view in Airflow UI
    connection = pg_hook.get_conn() # Gets the connection from PostgresHook
    cursor = connection.cursor() # Gets a cursor to the postgresQL database. Look at https://www.postgresql.org/docs/9.2/plpgsql-cursors.html for more information
    cursor.execute(request) # Executes the request
    sources = cursor.fetchall() # Fetchs all the data from the executed request
    for source in sources: # Does a simple print of each source to show that the hook works well. More on the next lesson
        print("Source: {0} - activated: {1}".format(source[0], source[1]))
    return sources
```



Instructions

- Go to the Airflow UI and trigger the DAG 'hook_dag'.
- Once the DAG finished executing the tasks, click on "hook_task" from the Graph View and "View Log". You should see the following output:

```
[2018-10-26 04:55:41,491] {logging_mixin.py:95} INFO - Source: MySQL - activated: True  
[2018-10-26 04:55:41,491] {logging_mixin.py:95} INFO - Source: S3 - activated: False  
[2018-10-26 04:55:41,492] {logging_mixin.py:95} INFO - Source: Mongo - activated: False  
[2018-10-26 04:55:41,492] {logging_mixin.py:95} INFO - Source: PostgreSQL - activated: False
```

- As you can see, we fetched all the data from table as expected using PostgreHook. Simple isn't it?



Important Notes

- The parameter “schema” corresponds to the database name you want to connect in PostgreSQL.
- `postgres_conn_id="postgre_sql"` Here, “postgre_sql” is a connection created from the Airflow UI into the Connection view (You should have done already this in previous lessons).
- There are many official hooks such as PrestoHook, SqliteHook, SlackHook and so on that you can use.
- You can find also many very interesting unofficial hooks created by the community such as SparkSubmitHook (to kick off a spark submit job), FtpHook, JenkinsHook and so on.