Соединения и хуки в Apache Airflow

I was write ­5/10/2022

Соединения и хуки в Apache Airflow: разбираем на примере SQLite

Автор [Роман Котюбеев](https://www.bigdataschool.ru/author/favorshlitex)в категории [AirFlow](https://www.bigdataschool.ru/news/airflow), [Статьи](https://www.bigdataschool.ru/blog)

[https://www.bigdataschool.ru/blog/connections-and-hooks-airflow.html](https://www.bigdataschool.ru/blog/connections-and-hooks-airflow.html%20%20%20%20%20)

В [прошлый раз](https://www.bigdataschool.ru/blog/airflow-xcom-variables.html) мы говорили о способе взаимодействия задач между собой в Apache [Airflow](https://www.bigdataschool.ru/wiki/airflow" \t "_blank). Сегодня поговорим о таких сущностях, как соединение (connections) и хуки (hooks). Читайте в этой статье: что такое хук и соединение, как создать и скачать соединение, а также как подключить базу данных в Airflow.

### **Что такое связи и хуки в Apache Airflow**

Соединение (connection) – это набор параметров (логин, пароль, хост) и некая внешняя система c уникальным именем (conn\_id), к которой и происходит подключение. Такой системой может быть базы и хранилища данных, электронная почта, AWS, Google Cloud, Telegram и т.д. Всякий раз, когда вам нужно изнутри Airflow подключиться к БД, то будут использоваться соединения и хуки.

Список соединений доступен в пользовательском интерфейсе во вкладке Admin->Connection. Там же можно добавить новое соединение. Если у вас нет нужного типа соединения, то его следует установить. Для этого используется Airflow Provider, список доступных пакетов находится в [документации](https://airflow.apache.org/docs/#providers-packages-docs-apache-airflow-providers-index-html). А установить их (по крайней мере в Airflow 2.0) можно через обычный pip, например, вот так можно получить соединение с PostgreSQL:

$ pip install apache-airflow-providers-postgres

# или что то же самое:

$ pip install apache-airflow[postgres]

Некоторые типы соединений отсутствуют в списке conn type, например, Telegram. Тогда единственным способом создать новое соединение становится использование CLI, как это показано в [документации](https://airflow.apache.org/docs/apache-airflow/stable/howto/connection.html#creating-a-connection-from-the-cli).

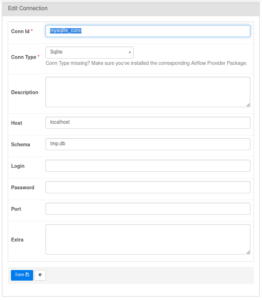
### **Соединение на примере SQLite**

SQLite — легковесная СУБД, которая будет служить примером организации задач Airflow с базой данных. Создадим БД под названием tmp.db и поместим её в директорию $HOME/airflow. В этой БД создадим таблицу example с двумя полями. Вот так все это выглядит:

$ cd airflow

$ sqlite3 tmp.db

sqlite> TABLE example(id INT, exec\_time TEXT);

Ниже на рисунке показано, как создать соединение в пользовательском интерфейсе. Мы указали идентификатор связи, название БД и хост. Если бы это был PostgreSQL, то необходимо было бы добавить порт (например, 5432), логин и пароль (если есть). Термин scheme может отличаться в других БД.

Теперь построим такую задачу Airflow, которая будет добавлять значения полей.

### **Хук — это SQLOperator, который можно использовать в функциях**

Хук (hook) предоставляет интерфейс для взаимодействия с внешней системой в пределах одного графа. Например, некоторые задачи требуют доступа к MySQL, и чтобы не устанавливать связь каждый раз в нужной задаче, можно использовать хук. Хуки также позволяют не хранить параметры аутентификации в графе. По сути своей, хук позволяет использовать возможности SQLOperator внутри PythonOperator.

Для баз данных используется единый API, поэтому методы для работы с ними будут одинаковыми. О хуках у нас есть [статья](https://www.bigdataschool.ru/blog/how-to-create-your-own-airflow-provider.html). Итак, хук соединения SQLite импортируется и реализуется следующим образом:

from airflow.providers.sqlite.hooks.sqlite import SqliteHook

sqlite\_hook = SqliteHook(sqlite\_conn\_id='sqlite\_conn')

Пусть имеются две задачи: первая добавляет значения в таблицу базы данных, а вторая выдает эти значения. Первая задача будет использовать хук, вторая — SQLOperator, который будет иметь то же самый идентификатор соединения. Добавление значений полей в базу данных осуществляется через метод insert\_rows. Ниже продемонстрирован код в Python. На это примере можно заметить, что хук представляет собой SQLOperator, который можно использовать внутри функции.

from airflow import DAG

from airflow.utils.dates import days\_ago

from airflow.operators.python import PythonOperator

from airflow.providers.sqlite.operators.sqlite import SqliteOperator

from airflow.providers.sqlite.hooks.sqlite import SqliteHook

def insert(\*\*kwargs):

exec\_time = kwargs['ts']

sqlite\_hook = SqliteHook(sqlite\_conn\_id='sqlite\_conn')

rows = [(1, exec\_time),]

fields = ['id', 'exec\_time']

sqlite\_hook.insert\_rows(

table='example',

rows=rows,

target\_fields=fields,

)

with DAG(

'conhook',

schedule\_interval=None,

start\_date=days\_ago(2),

) as dag:

insert\_into\_example = PythonOperator(

task\_id='insert\_into\_example',

python\_callable=insert,

)

print\_rows = SqliteOperator(

task\_id='print\_rows',

sqlite\_conn\_id='sqlite\_conn',

sql='SELECT id, exec\_time from example'

)

insert\_into\_example >> print\_rows

В некоторых версиях Airflow метод insert\_rows не работает для SQLite из-за проблем в исходном коде [1]. Данная проблема связана с тем, что добавляются значения в базу данных SQLite не совсем правильно, поскольку вместо указаний типов данных следует использовать знак ? (подробности см. [тут](https://python-school.ru/blog/sqlite-database-in-python/)). Если возникают проблемы с БД, то можно выполнить запрос самостоятельно через метод run, например так:

sql = """

INSERT INTO example (id, exec\_time)

values (%d, "%s")

"""

sqlite\_hook = SqliteHook(sqlite\_conn\_id='sqlite\_conn')

sqlite\_hook.run(sql % (0, exec\_time))

Можно ещё раз убедиться, что новые значения появляются в таблице через интерфейс командной строки, выполнив тот же самый Select.