



Порядок исполнения задач и создание зависимостей в графе

В пайплайне из видео не были заданы зависимости между задачами. В этом шаге вы сможете более подробно разобраться с зависимостями в задачах построения пайплайна.

Пример создания зависимостей, посмотрите внимательно как они создаются. Для того, чтобы задать зависимость одного taska от другого, используют битшифт оператор `>>` и скобки `[]`

```
from airflow import DAG
from datetime import timedelta
from airflow.utils.dates import days_ago
from airflow.operators.dummy_operator import DummyOperator

dag = DAG('dag', schedule_interval=timedelta(days=1), start_date=days_ago(1))
t1 = DummyOperator(task_id='task_1', dag=dag)
t2 = DummyOperator(task_id='task_2', dag=dag)
t3 = DummyOperator(task_id='task_3', dag=dag)
t4 = DummyOperator(task_id='task_4', dag=dag)

t1 >> t2 >> [t3, t4]
```

Для примера возьмём: `t1 >> t2`, `t2` исполнится только после `t1`. Если мы хотим настроить зависимость двух taskов от одного; одного taska от двух; нескольких от одного - то используйте скобки `t1 >> [t2, t3]`. В таком случае `t2, t3` будет исполняться параллельно после выполнения `t1`. Операция `[t1, t2] >> [t3, t4]` не определена.

Операторы `set_upstream()` и `set_downstream()`

Отношения операторов также устанавливаются с помощью методов `set_upstream()` и `set_downstream()`.

`set_upstream()`: задача, которая должна достичь определенного состояния, прежде чем может быть запущена зависимая задача.

`set_downstream()`: зависимая задача, которая не может выполняться до тех пор, пока вышестоящая задача не достигнет указанного состояния.

Теперь вы можете задать зависимость задач в DAG друг от друга (принимая во внимание ацикличность графа).

Как вам урок?



Далее >

