

12.06.2024 09:35:55



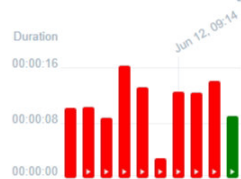
25



All Run Types



All Run States

[Clear Filters](#)Auto-refresh ☐Press **shift** + **/** for Shortcuts[deferred](#) [failed](#) [queued](#) [removed](#) [restarting](#) [running](#) [scheduled](#) [skipped](#) [success](#) [up_for_reschedule](#) [up_for_retry](#) [upstream_failed](#) [no_status](#)

yandex_wether	■	■	■	■	■	■	■	■	■	■
open_wether	■	■	■	■	■	■	■	■	■	■
python_wether	■	■	■	■	■	■	■	■	■	■
send_message_telegram	■	■	■	■	■	■	■	■	■	■

DAG wether-tlegram

[Details](#) [Graph](#) [Gantt](#) [Code](#)

Parsed at: 2024-06-12, 09:35:49 MSK

```
1 import datetime
2 import os
3 import requests
4 import pendulum
5 from airflow.decorators import dag, task
6 from airflow.providers.telegram.operators.telegram import TelegramOperator
7 from airflow.models import Connection
8 from airflow import settings
9
10 os.environ["no_proxy"]="*"
11
12 @dag(
13     dag_id="wether-tlegram",
14     schedule="@once",
15     start_date=pendulum.datetime(2023, 1, 1, tz="UTC"),
16     catchup=False,
17     dagrun_timeout=datetime.timedelta(minutes=60),
18 )
19
20 def WetherETL():
21
22     send_message_telegram_task = TelegramOperator(
23         task_id='send_message_telegram',
24         telegram_conn_id='telegram_conn',
25         token='7248934509:AAE4KtbM5wDjmrU-xFH8bMAwBwDZw-C_cMA',
26         chat_id='839424934',
27         send_wether_in_moscow=
28         "\nOpen wether: " + "{{"
```

[Toggle Wrap](#)

ID чата необходимо указывать свой, а не бота. Бот не может отправлять сообщения боту

12.06.2024 10:15:16



25

All Run Types

All Run States

Clear Filters

Auto-refresh

Press **shift** + **/** for Shortcutsdeferred failed queued removed restarting running scheduled skipped success up_for_reschedule up_for_retry upstream_failed no_status

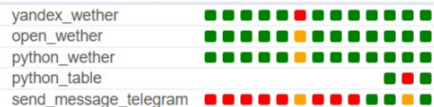
Duration

00:00:16

00:00:08

00:00:00

Jun 12, 03:35

DAG
wether-tlegram[Details](#) [Graph](#) [Gantt](#) [Code](#)

Parsed at: 2024-06-12, 10:15:16 MSK

```
49 def get_open_wether(**kwargs):
50     ti = kwargs['ti']
51     url = "https://api.openweathermap.org/data/2.5/weather?lat=55.749013596652574&lon=37.61622153253021&appid=2cd78e55c423fc81cebc1487134a6300"
52
53     payload={}
54     headers = {}
55
56     response = requests.request("GET", url, headers=headers, data=payload)
57     print("test")
58     a=round(float(response.json()['main']['temp']) - 273.15, 2)
59     print(a)
60     ti.xcom_push(key='open_wether', value=round(float(response.json()['main']['temp']) - 273.15, 2))
61     # return str(a)
62 @task(task_id='python_wether')
63 def get_wether(**kwargs):
64     print("Yandex "+str(kwargs['ti'].xcom_pull(task_ids=['yandex_wether'],key='wether')[0])+" Open "+str(kwargs['ti'].xcom_pull(task_ids=['open_wether'],key='open_wether')[0]))
65     @task(task_id='python_table')
66     def get_wether_table(**kwargs):
67         con=create_engine("mysql://Airflow:1@localhost:33061/spark")
68         data = [[str(kwargs['ti'].xcom_pull(task_ids=['yandex_wether'],key='wether')[0]), str(kwargs['ti'].xcom_pull(task_ids=['open_wether'],key='open_wether')[0]), datetime.datetime.now())]
69         df = pd.DataFrame(data)
70         df.to_sql('wether',con,schema='spark',if_exists='replace',index=False)
71
72     get_yandex_wether() >> get_open_wether() >> get_wether()>> get_wether_table() >> send_message_telegram_task
73
74
75
76 dag = WetherETL()
77
```

Toggle Wrap

Функция создания таблицы погоды и запись данных в 3 колонки.

12 июня

/start 9:02 ✓✓

Wether in Moscow
Yandex: 22 degrees
Open wether: 21.48 degrees 9:36

Wether in Moscow
Yandex: 23 degrees
Open wether: 22.14 degrees 10:11

Wether in Moscow
Yandex: 23 degrees
Open wether: 22.14 degrees 10:15



Написать сообщение...





Unamed-1

- > Airflow
- > information_sc...
- > mysql
- > performance_s...
- ✓ spark 288,0 KiB
 - credit 96,0 KiB
 - d4_1 48,0 KiB
 - d4_2 16,0 KiB
 - d4_3 16,0 KiB
 - d6_1 16,0 KiB
 - d6_2 16,0 KiB
 - s3_1 16,0 KiB
 - s3_2 16,0 KiB
 - s3_3 16,0 KiB
 - s3_4 16,0 KiB
 - wether** 16,0 KiB
- > sys

spark.wether: 1 строк (точно)

#	0	1	2
1	23	22.14	2024-06-12 10:15:24