U04

<3>WSL (8432 - Relay) ERROR: CreateProcessCommon:798: execvpe(/bin/bash) failed: No such file or directory

Error: failed to execute cmd: exit status 1

U14

<3>WSL (5554 - Relay) ERROR: CreateProcessCommon:798: execvpe(/bin/bash) failed: No such file or directory

Error: failed to execute cmd: exit status 1

U12<3>WSL (5554 - Relay) ERROR: CreateProcessCommon:798: execvpe(/bin/bash) failed: No such file or directory

Error: failed to execute cmd: exit status 1-i

U17

<3>WSL (6425 - Relay) ERROR: CreateProcessCommon:798: execvpe(/bin/bash) failed: No such file or directory

Error: failed to execute cmd: exit status 1

U18

<3>WSL (5509 - Relay) ERROR: CreateProcessCommon:798: execvpe(/bin/bash) failed: No such file or directory

Error: failed to execute cmd: exit status 1

U20

Error: failed to execute cmd: exit status 1

409 Conflict

reason: Unique constraint violationstatement: INSERT INTO dag\_run (dag\_id, queued\_at, logical\_date, start\_date, end\_date, state, run\_id, creating\_job\_id, run\_type, triggered\_by, triggering\_user\_name, conf, data\_interval\_start, data\_interval\_end, run\_after, last\_scheduling\_decision, log\_template\_id, updated\_at, clear\_number, backfill\_id, bundle\_version, scheduled\_by\_job\_id, context\_carrier, created\_dag\_version\_id) VALUES (%(dag\_id)s, %(queued\_at)s, %(logical\_date)s, %(start\_date)s, %(end\_date)s, %(state)s, %(run\_id)s, %(creating\_job\_id)s, %(run\_type)s, %(triggered\_by)s, %(triggering\_user\_name)s, %(conf)s, %(data\_interval\_start)s, %(data\_interval\_end)s, %(run\_after)s, %(last\_scheduling\_decision)s, %(log\_template\_id)s, %(updated\_at)s, %(clear\_number)s, %(backfill\_id)s, %(bundle\_version)s, %(scheduled\_by\_job\_id)s, %(context\_carrier)s, %(created\_dag\_version\_id)s) RETURNING dag\_run.idorig\_error: duplicate key value violates unique constraint "dag\_run\_dag\_id\_logical\_date\_key"

DETAIL: Key (dag\_id, logical\_date)=(dynamic\_task\_example, 2025-10-15 09:58:09+00) already exists.

message: Serious error when handling your request. Check logs for more details - you will find it in api server when you look for ID LUI1CiiN

.astro/test\_dag\_integrity\_default.py F........ [100%]

=================================== FAILURES ===================================

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ test\_file\_imports[dags/dag8.py] \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

rel\_path = 'dags/dag8.py'

rv = 'Traceback (most recent call last):\n File "<frozen importlib.\_bootstrap>", line 488, in \_call\_with\_frames\_removed\n ...ag8.py", line 51, in <module>\n start\_task >> task\n ^^^^\nNameError: name \'task\' is not defined'

@pytest.mark.parametrize(

"rel\_path, rv", get\_import\_errors(), ids=[x[0] for x in get\_import\_errors()]

)

def test\_file\_imports(rel\_path, rv):

"""Test for import errors on a file"""

if os.path.exists(".astro/dag\_integrity\_exceptions.txt"):

with open(".astro/dag\_integrity\_exceptions.txt", "r") as f:

exceptions = f.readlines()

print(f"Exceptions: {exceptions}")

if (rv != "No import errors") and rel\_path not in exceptions:

# If rv is not "No import errors," consider it a failed test

> raise Exception(f"{rel\_path} failed to import with message \n {rv}")

E Exception: dags/dag8.py failed to import with message

E Traceback (most recent call last):

E File "<frozen importlib.\_bootstrap>", line 488, in \_call\_with\_frames\_removed

E File "/usr/local/airflow/dags/dag8.py", line 51, in <module>

E start\_task >> task

E ^^^^

E NameError: name 'task' is not defined

.astro/test\_dag\_integrity\_default.py:138: Exception

----------------------------- Captured stdout call -----------------------------

Exceptions: ['# Add dag files to exempt from parse test below. ex: dags/<test-file>']

=============================== warnings summary ===============================

../lib/python3.12/site-packages/\_pytest/cacheprovider.py:475

/usr/local/lib/python3.12/site-packages/\_pytest/cacheprovider.py:475: PytestCacheWarning: could not create cache path /.pytest\_cache/v/cache/nodeids: [Errno 13] Permission denied: '/pytest-cache-files-1nevhobo'

config.cache.set("cache/nodeids", sorted(self.cached\_nodeids))

../lib/python3.12/site-packages/\_pytest/cacheprovider.py:429

/usr/local/lib/python3.12/site-packages/\_pytest/cacheprovider.py:429: PytestCacheWarning: could not create cache path /.pytest\_cache/v/cache/lastfailed: [Errno 13] Permission denied: '/pytest-cache-files-kiqpx2vd'

config.cache.set("cache/lastfailed", self.lastfailed)

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html

=========================== short test summary info ============================

FAILED .astro/test\_dag\_integrity\_default.py::test\_file\_imports[dags/dag8.py]

=================== 1 failed, 8 passed, 2 warnings in 7.88s ====================

See above for errors detected in your DAGs

Error: your local DAGs did not parse. Fix the listed errors or use `astro deploy [deployment-id] -f` to force deploy

PS C:\Users\Administrator\Desktop\Airflow\Airflow\_init>

from airflow import DAG

from airflow.operators.empty import EmptyOperator

from airflow.operators.python import PythonOperator

from datetime import datetime

from airflow.models.variable import Variable

import requests

# Function to be executed by the PythonOperator

def print\_task(task\_name):

print(f"Executing task: {task\_name}")

# List of task names

#task\_names = ['task\_1', 'task\_2', 'task\_3', 'task\_4', 'task\_5', 'task\_6', 'task\_7']

task\_names = Variable.get("task\_list", deserialize\_json=True)

# Define the DAG

dag = DAG(

'dynamic\_task\_example',

description='A simple dynamic DAG',

schedule=None, # Define the schedule, 'None' for manual trigger

start\_date=datetime(2024, 11, 21),

catchup=False,

)

# Create a start task

start\_task = EmptyOperator(

task\_id='start',

dag=dag,

)

# Dynamically generate tasks based on task\_names

for task\_name in task\_names:

task = PythonOperator(

task\_id=task\_name,

python\_callable=print\_task,

op\_args=[task\_name],

dag=dag,

)

# Set task dependencies: start -> task -> end

start\_task >> task

# Create an end task

end\_task = EmptyOperator(

task\_id='end',

dag=dag,

)

# Set dependencies: task -> end

for task\_name in task\_names:

task = dag.get\_task(task\_name)

task >> end\_task

cannot import name 'apply\_defaults' from 'airflow.utils.decorators' (unknown location)

<https://docs.google.com/document/d/1ssuu35tzI-v4SVryH_WEbIkdmg2lt9RNXgj-hFNXOs8/edit?usp=sharing>

=================================== FAILURES ===================================

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ test\_file\_imports[dags/dag13.py] \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

rel\_path = 'dags/dag13.py'

rv = 'Traceback (most recent call last):\n File "<frozen importlib.\_bootstrap>", line 488, in \_call\_with\_frames\_removed\n ...stgres.operators.postgres import PostgresOperator\nModuleNotFoundError: No module named \'airflow.providers.postgres\''

@pytest.mark.parametrize(

"rel\_path, rv", get\_import\_errors(), ids=[x[0] for x in get\_import\_errors()]

)

def test\_file\_imports(rel\_path, rv):

"""Test for import errors on a file"""

if os.path.exists(".astro/dag\_integrity\_exceptions.txt"):

with open(".astro/dag\_integrity\_exceptions.txt", "r") as f:

exceptions = f.readlines()

print(f"Exceptions: {exceptions}")

if (rv != "No import errors") and rel\_path not in exceptions:

# If rv is not "No import errors," consider it a failed test

> raise Exception(f"{rel\_path} failed to import with message \n {rv}")

E Exception: dags/dag13.py failed to import with message

E Traceback (most recent call last):

E File "<frozen importlib.\_bootstrap>", line 488, in \_call\_with\_frames\_removed

E File "/usr/local/airflow/dags/dag13.py", line 2, in <module>

E from airflow.providers.postgres.operators.postgres import PostgresOperator

E ModuleNotFoundError: No module named 'airflow.providers.postgres'

.astro/test\_dag\_integrity\_default.py:138: Exception

----------------------------- Captured stdout call -----------------------------

Exceptions: ['# Add dag files to exempt from parse test below. ex: dags/<test-file>']

=============================== warnings summary ===============================

../lib/python3.12/site-packages/\_pytest/cacheprovider.py:475

/usr/local/lib/python3.12/site-packages/\_pytest/cacheprovider.py:475: PytestCacheWarning: could not create cache path /.pytest\_cache/v/cache/nodeids: [Errno 13] Permission denied: '/pytest-cache-files-oz1mhp8y'

config.cache.set("cache/nodeids", sorted(self.cached\_nodeids))

../lib/python3.12/site-packages/\_pytest/cacheprovider.py:429

/usr/local/lib/python3.12/site-packages/\_pytest/cacheprovider.py:429: PytestCacheWarning: could not create cache path /.pytest\_cache/v/cache/lastfailed: [Errno 13] Permission denied: '/pytest-cache-files-1kkhpdnx'

config.cache.set("cache/lastfailed", self.lastfailed)

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html

=========================== short test summary info ============================

FAILED .astro/test\_dag\_integrity\_default.py::test\_file\_imports[dags/dag13.py]

================== 1 failed, 13 passed, 2 warnings in 24.44s ===================

See above for errors detected in your DAGs

Error: your local DAGs did not parse. Fix the listed errors or use `astro deploy [deployment-id] -f` to force deploy

PS C:\Users\Administrator\Desktop\Airflow\Airflow\_init>

Apache-airflow-providers-postgres