# Test your tasks

Ensuring the reliability and accuracy of tasks within your DAG is crucial. Testing is vital in achieving this, allowing you to validate task functionality, identify potential issues, and maintain data integrity before running the DAG or pushing it into production.

One command that you must always use is: **airflow tasks test <dag_id> <task_id> <logical_date>**

This command runs a task without checking for dependencies or recording its state in the database. For example, if the task pushes an XCOM, the XCOM won't be created with that command.

Let's imagine you have the DAG below:

```python
from airflow.decorators import task, dag
from airflow.exceptions import AirflowException
from datetime import datetime

@dag(start_date=datetime(2023, 1, 1), schedule=None, catchup=False)
def cli():

    @task
    def my_task(val):
        print(val)
        return 42

    my_task(80)
cli()
```

To test the task **my_task** , execute the command: **airflow tasks test cli my_task 2023-01-01** . You should see:

```
[2023-06-14T12:38:50.188+0000] {taskinstance.py:1545} INFO - Exporting env vars: AIRFLOW_C
80
[2023-06-14T12:38:50.189+0000] {python.py:183} INFO - Done. Returned value was: 42
[2023-06-14T12:38:50.196+0000] {taskinstance.py:1345} INFO - Marking task as SUCCESS. dag_
```

Two important notes:

1. **Marking task as SUCCESS** shows that the task works and has been successfully executed
2. **Returned value was: 42** indicates that the tasks created a XCOM with the value 42. However, if you look into admin and XCOMs, you will see that nothing has been created. That's because airflow tasks test do not store any task related metadata.

Try to execute the following code and see what you get:

```python
 from airflow.decorators import task, dag
from airflow.exceptions import AirflowException
from datetime import datetime

@dag(start_date=datetime(2023, 1, 1), schedule=None, catchup=False)
def cli():

    @task
    def my_task(val):
        raise AirflowException()
        print(val)
        return 42

    my_task(80)
cli()
```

As a best practice, each time you add a task in a DAG, test it with **airflow tasks test** .

Was this page helpful?