

Airflow testing cheat sheet

This cheat sheet lists all attributes, properties and methods that you can call on the DagBag, DAG and Task (BaseOperator) objects. These are useful when writing unit tests for DAGs and custom operators.

For more information on testing DAGs and operators visit the [Airflow documentation](#).

DagBag

DagBag	
Attributes	
dag_folder	(str) the folder to scan to find DAGs
executor	(object) executor to use when executing task instances
dags	(dict) dictionary of dags in the DagBag <ul style="list-style-type: none"> key is DAG id and value is DAG object
import_errors	(dict) errors when trying to parse DAG from file <ul style="list-style-type: none"> key is filepath, value is error message
Properties	
dag_ids	DAG ids of DAGs in this DagBag :rtype: dict_keys
Methods	
size()	the amount of DAGs contained in this DagBag :rtype: int
get_dag(dag_id)	returns a DAG object from this DagBag with id dag_id :param: (str) dag_id :rtype: DAG
dagbag_report()	returns a report about DagBag loading stats :rtype: str

DAG

DAG	
Attributes	
user_defined_macros	(dict) macros that will be exposed in jinja templates
user_defined_filters	(dict) filters that will be exposed in jinja templates
default_args	(dict) default_arguments dictionary
schedule_interval	(datetime.timedelta or str) returns how often this DAG runs
catchup	(bool) whether to perform scheduler catchup or only run latest
doc_md	(str) DAG documentation
Properties	
dag_id	DAG id of this DAG :rtype: str
full_filepath	path to DAG file :rtype: str
concurrency	number of task instances allowed to run concurrently :rtype: int
description	description of this DAG :rtype: str
tasks	returns the list of task objects of this DAG :rtype: list[Task]
task_ids	returns the list of task ids of this DAG :rtype: list[str]
owner	comma separated list of owners :rtype: str
allow_future_exec_dates	whether DAG allows exec dates in the future :rtype: bool
is_paused	whether this DAG is paused or not :rtype: bool
subdags	list of SubDags of this DAG :rtype: list[DAG]
roots	nodes with no parents

	:rtype: list[Task]
leaves	nodes with no children :rtype: list[Task]
Methods	
has_task(task_id)	whether DAG has task with provided task id :param: (str) task_id :rtype: bool
get_task(task_id)	returns a Task object from this dag with id task_id :param: (str) task_id :rtype: Task

Task

Task (BaseOperator)	
Attributes	
task_id	(str) task id of this operator
owner	(str) returns owner of this operator
email_on_retry	(bool) whether to send email alert when retrying
email_on_failure	(bool) whether to send email alert when task has failed
trigger_rule	(str) rule by which dependencies are applied for the task to get triggered
start_date	(datetime.datetime) start date for the task
retries	(int) number of retries that should be performed before failing the task
Properties	
dag	DAG that is assigned to this operator :rtype: DAG
dag_id	DAG id of DAG that is assigned to this operator :rtype: str
upstream_list	list of directly upstream tasks :rtype: list[Task]
upstream_task_ids	list of directly upstream task ids :rtype: list[str]
downstream_list	list of directly downstream tasks :rtype: list[Task]
downstream_task_ids	list of directly downstream task ids :rtype: list[str]
task_type	operator type e.g. BranchPythonOperator :rtype: str
Methods	
has_dag()	whether operator has been assigned to a DAG :rtype: bool