

# Python API Reference

## Operators

Operators allow for generation of certain types of tasks that become nodes in the DAG when instantiated. All operators derive from **BaseOperator** and inherit many attributes and methods that way.

There are 3 main types of operators:

- Operators that performs an **action**, or tell another system to perform an action
- **Transfer** operators move data from one system to another
- **Sensors** are a certain type of operator that will keep running until a certain criterion is met. Examples include a specific file landing in HDFS or S3, a partition appearing in Hive, or a specific time of the day. Sensors are derived from **BaseSensorOperator** and run a `poke` method at a specified `poke_interval` until it returns **True**.

## BaseOperator

All operators are derived from **BaseOperator** and acquire much functionality through inheritance. Since this is the core of the engine, it's worth taking the time to understand the parameters of **BaseOperator** to understand the primitive features that can be leveraged in your DAGs.

## BaseSensorOperator

All sensors are derived from **BaseSensorOperator**. All sensors inherit the `timeout` and `poke_interval` on top of the **BaseOperator** attributes.

## Operators packages

All operators are in the following packages:

- `airflow.operators`
- `airflow.sensors`

## Hooks

Hooks are interfaces to external platforms and databases, implementing a common interface when possible and acting as building blocks for operators. All hooks are derived from **BaseHook**.

## Hooks packages

All hooks are in the following packages:

- `airflow.hooks`

## Executors

Executors are the mechanism by which task instances get run. All executors are derived from **BaseExecutor**.

## Executors packages

All executors are in the following packages:

- `airflow.executors`

## Models

Models are built on top of the SQLAlchemy ORM Base class, and instances are persisted in the database.

- `airflow.models`

## Exceptions

- `airflow.exceptions`

## Secrets Backends

Airflow relies on secrets backends to retrieve `Connection` objects. All secrets backends derive from `BaseSecretsBackend` .

- `airflow.secrets`

[Previous](#)[Next](#)

Was this entry helpful?



Want to be a part of Apache Airflow?

[Join community](#)

[License](#) [Donate](#) [Thanks](#)

[Security](#)

© The Apache Software Foundation 2019

Apache Airflow, Apache, Airflow, the Airflow logo, and the Apache feather logo are either registered trademarks or trademarks of The Apache Software Foundation. All other products or name brands are trademarks of their respective holders, including The Apache Software Foundation.