

Interact with an External Service

Airflow's power comes from interacting with and coordinating work in other services.

Prerequisites

To interact with an external service, you need to have set up your local Astro project as described under the section “Install a Provider” and have a connection as described under the section “Install the HTTP Provider”

You should also have the DAG `extract_stars.py` in your folder `dags/`. If not, take a look at the previous activity `Start coding your first DAG...`

Where are you at

In the file `extract_stars.py` in your folder `dags/`, your DAG should look like this:

```
from airflow import DAG
from airflow.operators.bash import BashOperator

from datetime import datetime

with DAG('extract_stars', schedule_interval='@daily', start_date=datetime(2022, 1, 1), cat
chup=False) as dag:

    get_date = BashOperator(
        task_id="get_date",
        bash_command="date"
    )
```

Find the Operator

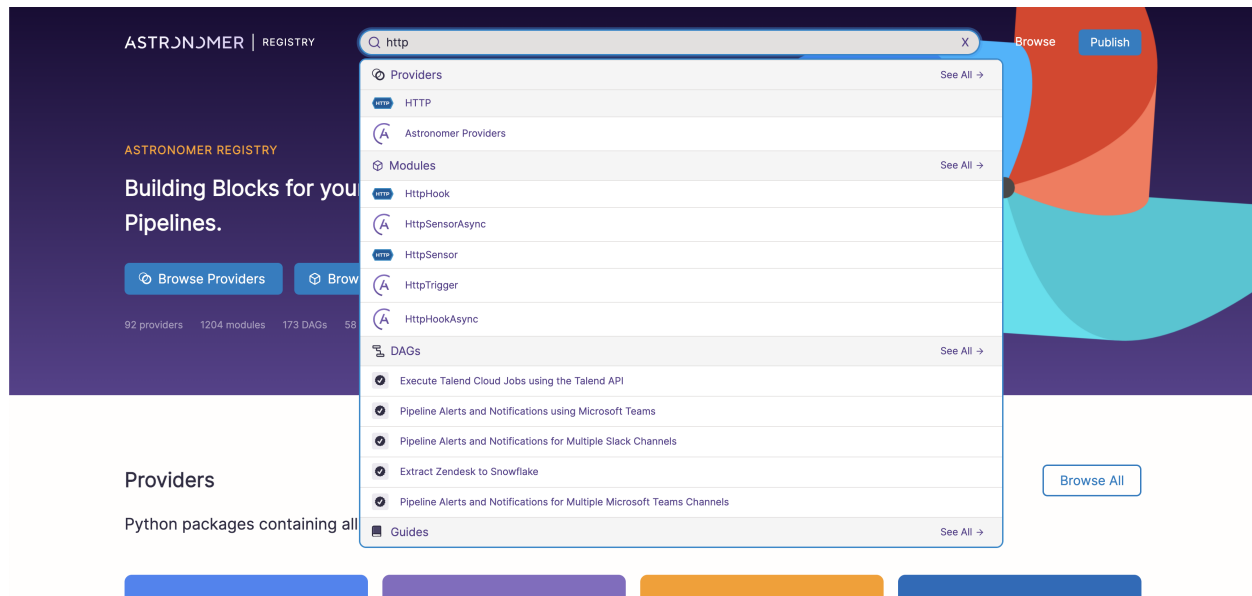
To illustrate interacting with an API, we will pull the ★ Stars from Github's API for the Apache Airflow Open Source project.

For that, the first step is to create an HTTP connection. That's what you've done in the previous activity.

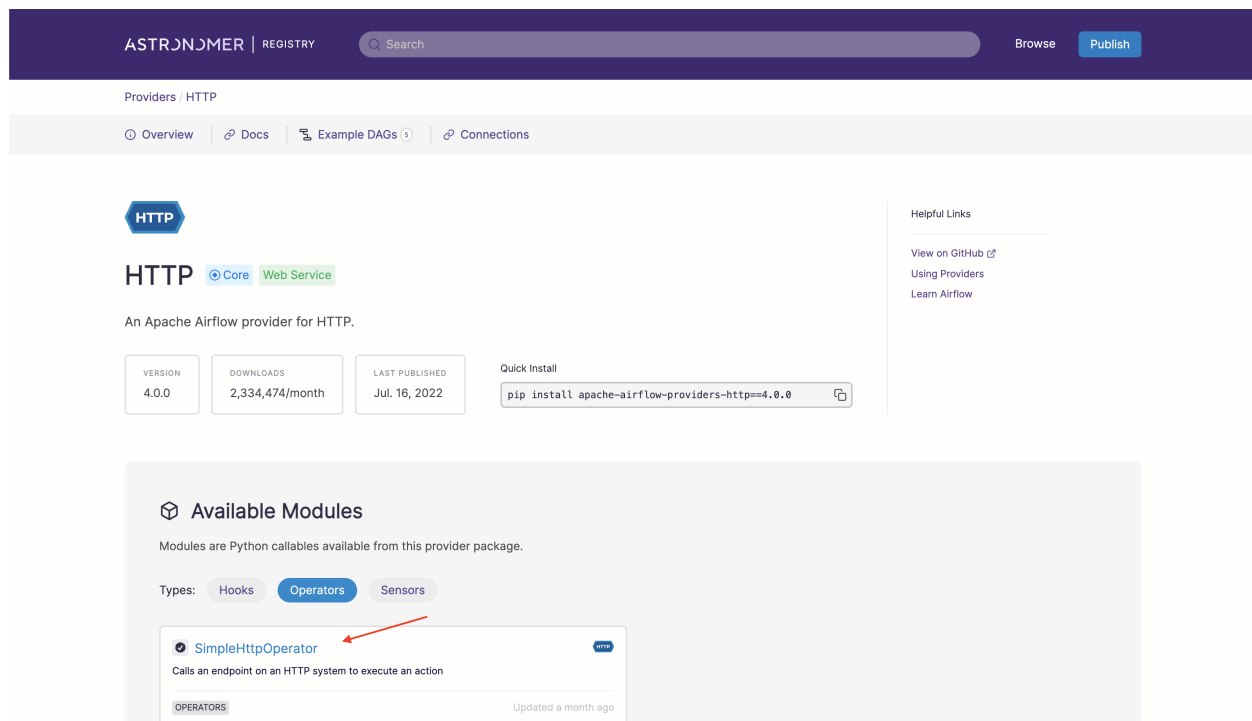
Now, it's time to use an Operator that downloads the content of an HTML page.

Back to the registry.astronomer.io

Look for **HTTP** in the search bar



Select the **HTTP** provider and click on **Operators** under **Available Modules**, then select the **SimpleHttpOperator**



The `SimpleHttpOperator` calls an endpoint on an HTTP system to execute an action. Typically, the endpoint is an HTML page. There are a couple of parameters to define:

The screenshot shows the Apache Airflow documentation for the `SimpleHttpOperator`. On the left, there's a sidebar with links: Parameters (selected), Documentation, Example DAGs, and Connections. The main content area is titled 'SimpleHttpOperator' and includes a 'Certified' badge, a description 'Calls an endpoint on an HTTP system to execute an action', and links to 'View on GitHub' and 'Last Updated: Jul. 12, 2022'. To the right, 'Access Instructions' provide steps to install the provider package using `pip install apache-airflow-providers-http==4.0.0` and to import the module in a DAG file: `from airflow.providers.http.operators.http import SimpleHttpOperator`. Below this, the 'Parameters' section lists several arguments: `http_conn_id` (The http connection to run the operator against), `endpoint` (The relative part of the full url, (templated)), `method` (The HTTP method to use, default = "POST"), `data` (The data to pass. POST-data in POST/PUT and params in the URL for a GET request. (templated)), `headers` (The HTTP headers to be added to the GET request), `response_check` (A check against the 'requests' response object. The callable takes the response object as the first positional argument and optionally any number of keyword arguments available in the context dictionary. It should return True for 'pass' and False otherwise.), and `response_filter` (A function allowing you to manipulate the response text. e.g response_filter=lambda response: json.loads(response.text). The callable takes the response object as the first positional argument and optionally any number of keyword arguments available in the context dictionary.).

Feel free to take a look at each of these. In this example, we focus only on:

- `http_conn_id`: we will use the connection id of the connection you've created before
- `endpoint`: the relative part (page) of the full url
- `method`: the HTTP method to use, GET
- `log_response`: allows checking the response in the Task logs

The SimpleHttpOperator

In your `extract_stars.py` DAG file, just under the `get_task` task.

Import the `SimpleHttpOperator`

Add a new task named `query_github_stats` with the `SimpleHttpOperator`

Define the following arguments:

- `task_id`: `query_github_stats`

- endpoint: `repos/apache/airflow` (the Apache Airflow repository)
- method: `GET`
- http_conn_id: `github_api`
- log_response: `True`

Save the file and go to the Airflow UI (localhost:8080) to see if you don't get any errors.

Try to do it, and I'll see you for the correction below

Solution

```
from airflow.providers.http.operators.http import SimpleHttpOperator

query_github_stats = SimpleHttpOperator(
    task_id="http",
    endpoint="repos/apache/airflow",
    method="GET",
    http_conn_id="github_api",
    log_response=True
)
```

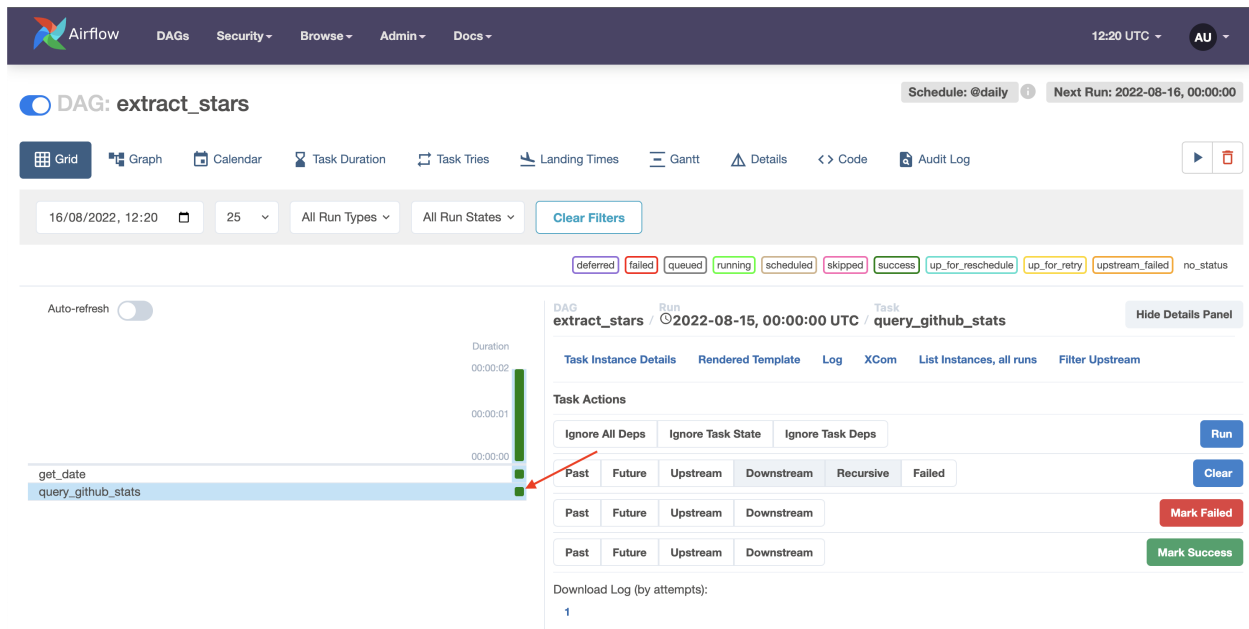
Get the repo info

Go to the Airflow UI (localhost:8080) to see if you don't get any errors.

The screenshot shows the Apache Airflow web interface. The top navigation bar includes links for DAGs, Security, Browse, Admin, and Docs, along with the current time (12:17 UTC) and a user profile icon (AU). The main section is titled "DAGs" and features a filter bar with tabs for "All" (3), "Active" (0), and "Paused" (3). A search bar labeled "Search DAGs" is also present. Below the filter bar is a table listing DAGs with columns for DAG name, Owner, Runs, Schedule, Last Run, Next Run, Recent Tasks, Actions, and Links. The table contains three entries: "example_dag_advanced" (community owner, @daily schedule, last run 2022-08-15), "example_dag_basic" (airflow owner, @daily schedule, last run 2022-08-15), and "extract_stars" (airflow owner, @daily schedule, last run 2022-08-16). Each entry has a toggle switch, a status icon, and a set of task status icons. The "extract_stars" DAG is currently active. At the bottom, a pagination bar shows "1" of 3 DAGs, and a footer note indicates "Showing 1-3 of 3 DAGs".

Trigger your DAG by clicking the `lecture` button  and `Trigger DAG`

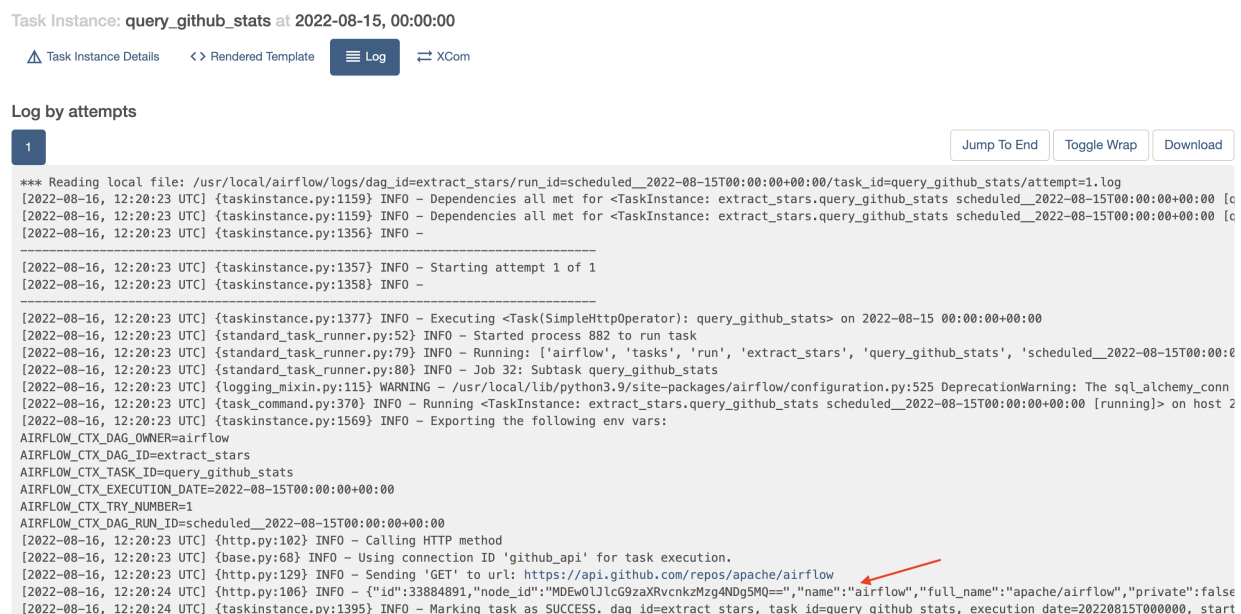
Click on your DAG and the square corresponding to the `query_github_stats` task



The screenshot shows the Apache Airflow web interface. At the top, there's a navigation bar with 'Airflow', 'DAGs', 'Security', 'Browse', 'Admin', and 'Docs'. The main header shows 'DAG: extract_stars' with a 'Schedule: @daily' and 'Next Run: 2022-08-16, 00:00:00'. Below the header, there's a toolbar with various views: Grid, Graph, Calendar, Task Duration, Task Tries, Landing Times, Gantt, Details, Code, and Audit Log. A filter bar shows '16/08/2022, 12:20', '25', 'All Run Types', and 'All Run States'. A 'Clear Filters' button is present. Below the filter bar, there's a 'Task Instance Details' panel for the task 'query_github_stats' at '2022-08-15, 00:00:00 UTC'. The panel shows 'Task Actions' with buttons for 'Ignore All Deps', 'Ignore Task State', and 'Ignore Task Deps'. There are also buttons for 'Run', 'Clear', 'Mark Failed', and 'Mark Success'. A 'Download Log (by attempts):' section shows '1' attempt.

Go to the `Log`

You should see the following output



The screenshot shows the 'Task Instance: query_github_stats at 2022-08-15, 00:00:00' page. It has tabs for 'Task Instance Details', 'Rendered Template', 'Log', and 'XCom'. The 'Log' tab is selected, showing 'Log by attempts' with '1' attempt. The log content is as follows:

```
*** Reading local file: /usr/local/airflow/logs/dag_id=extract_stars/run_id=scheduled__2022-08-15T00:00:00+00:00/task_id=query_github_stats/attempt=1.log
[2022-08-16, 12:20:23 UTC] {taskinstance.py:1159} INFO - Dependencies all met for <TaskInstance: extract_stars.query_github_stats scheduled__2022-08-15T00:00:00+00:00 [c
[2022-08-16, 12:20:23 UTC] {taskinstance.py:1159} INFO - Dependencies all met for <TaskInstance: extract_stars.query_github_stats scheduled__2022-08-15T00:00:00+00:00 [c
[2022-08-16, 12:20:23 UTC] {taskinstance.py:1356} INFO -

[2022-08-16, 12:20:23 UTC] {taskinstance.py:1357} INFO - Starting attempt 1 of 1
[2022-08-16, 12:20:23 UTC] {taskinstance.py:1358} INFO -

[2022-08-16, 12:20:23 UTC] {taskinstance.py:1377} INFO - Executing <Task(SimpleHttpOperator): query_github_stats> on 2022-08-15 00:00:00+00:00
[2022-08-16, 12:20:23 UTC] {standard_task_runner.py:52} INFO - Started process 882 to run task
[2022-08-16, 12:20:23 UTC] {standard_task_runner.py:79} INFO - Running: ['airflow', 'tasks', 'run', 'extract_stars', 'query_github_stats', 'scheduled__2022-08-15T00:00:00+00:00']
[2022-08-16, 12:20:23 UTC] {standard_task_runner.py:80} INFO - Job 32: Subtask query_github_stats
[2022-08-16, 12:20:23 UTC] {logging_mixin.py:115} WARNING - /usr/local/lib/python3.9/site-packages/airflow/configuration.py:525 DeprecationWarning: The sqlalchemy_conn
[2022-08-16, 12:20:23 UTC] {task_command.py:370} INFO - Running <TaskInstance: extract_stars.query_github_stats scheduled__2022-08-15T00:00:00+00:00 [running]> on host 2
[2022-08-16, 12:20:23 UTC] {taskinstance.py:1569} INFO - Exporting the following env vars:
AIRFLOW_CTX_DAG_OWNER=airflow
AIRFLOW_CTX_DAG_ID=extract_stars
AIRFLOW_CTX_TASK_ID=query_github_stats
AIRFLOW_CTX_EXECUTION_DATE=2022-08-15T00:00:00+00:00
AIRFLOW_CTX_TRY_NUMBER=1
AIRFLOW_CTX_DAG_RUN_ID=scheduled__2022-08-15T00:00:00+00:00
[2022-08-16, 12:20:23 UTC] {http.py:102} INFO - Calling HTTP method
[2022-08-16, 12:20:23 UTC] {base.py:68} INFO - Using connection ID 'github_api' for task execution.
[2022-08-16, 12:20:23 UTC] {http.py:129} INFO - Sending 'GET' to url: https://api.github.com/repos/apache/airflow
[2022-08-16, 12:20:24 UTC] {http.py:106} INFO - {'id': '33884891', 'node_id': 'MDEwOJlJlcG9zaXRvcnkzMzg4NDg5MQ==', 'name': 'airflow', 'full_name': 'apache/airflow', 'private': false
[2022-08-16, 12:20:24 UTC] {taskinstance.py:1395} INFO - Marking task as SUCCESS. dag id=extract stars, task id=query github stats, execution date=20220815T000000, start
```

Well done! You are now able to download HTTP data from your data pipelines

Additional resources

SimpleHttpOperator:

<https://registry.astronomer.io/providers/http/modules/simplehttpoperator>