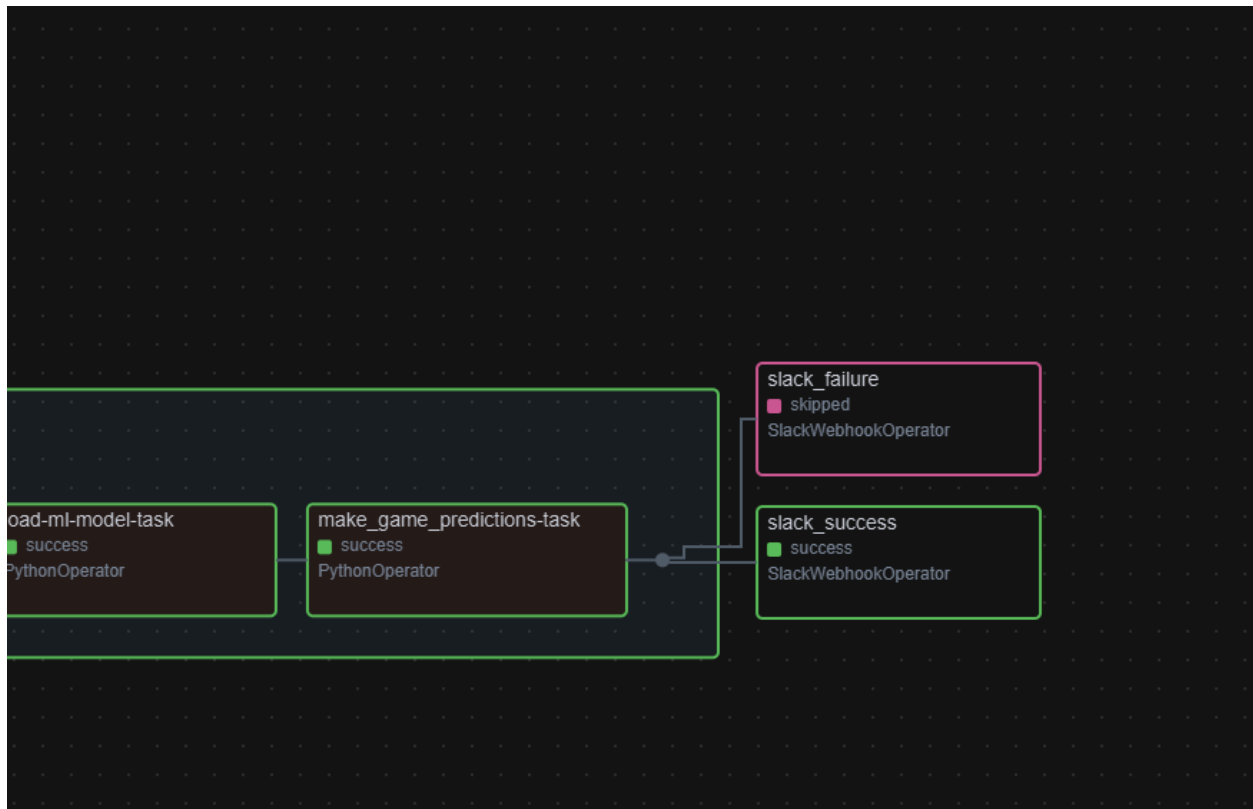


Guide for Setting up Slack Connection w/ Airflow.

After running the docker compose command to set up the containers and before running the DAGs for an ETL process for MLB Baseball Savant/ Statcast data and the Load MLB Stats ETL process, you are going to want to set up a Slack Connection Webhook inside of Airflow.

Why do I need a Slack Connection?

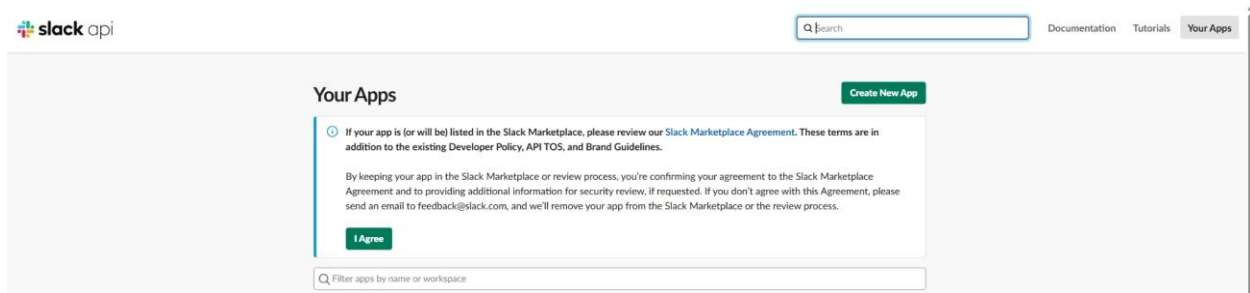
As you can notice at the end of the DAG run, it sends slack messages to a channel of choice with the status of the run. I intended this to assist with visibility into the data pipelines, specifically when they are scheduled DAGs.



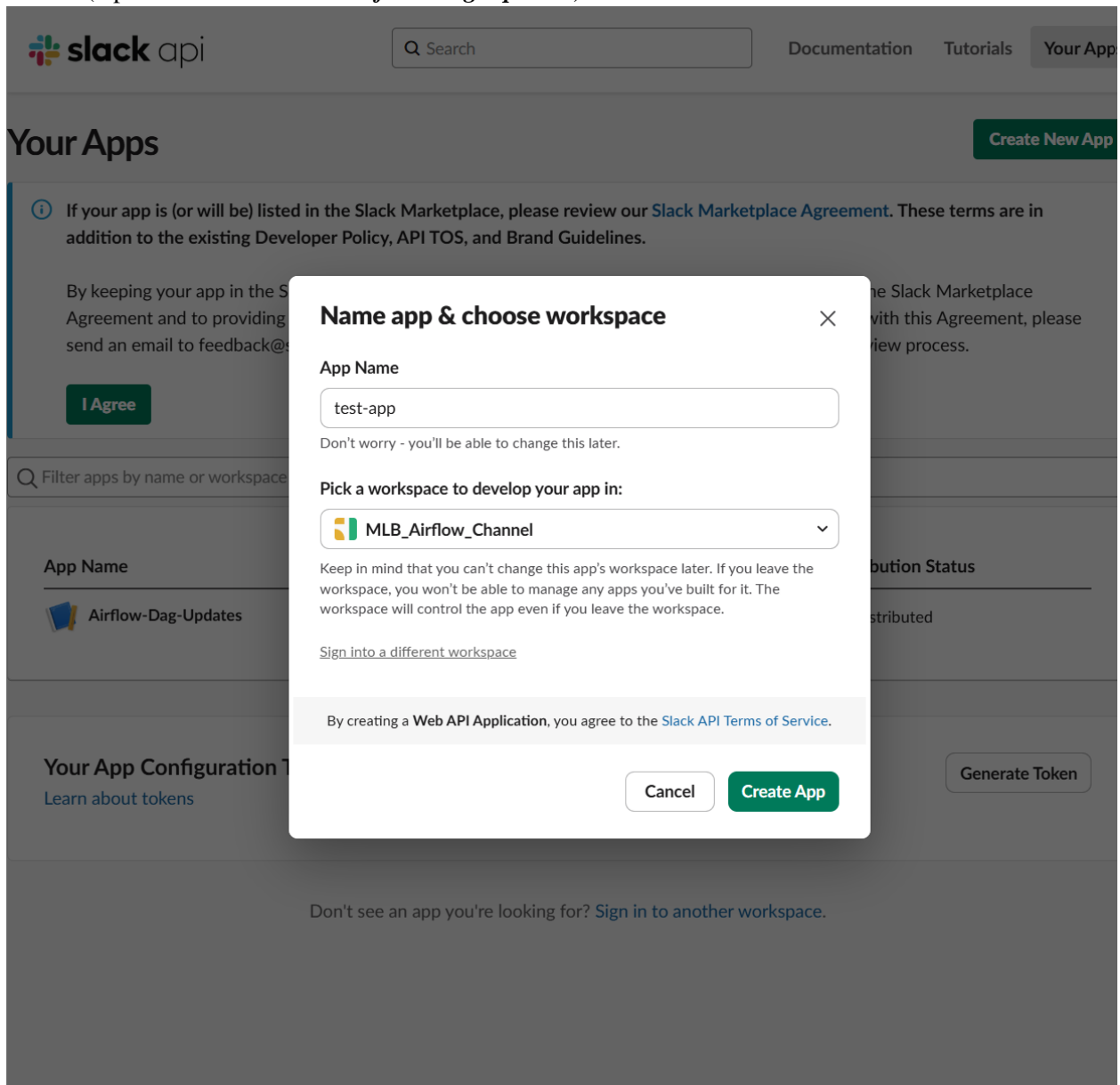
How Do I Set up this Slack Connection?

To set up this slack connection you are going to first set up an account on <https://slack.com/> and create a Workspace. In that workspace create a channel named **#airflow-dag-status**

After creating a Workspace proceed to <https://api.slack.com/> to create an app/bot and choose the *From Scratch* option:



Through this page, give your app/bot a name and click the drop down to select the workspace you created earlier. (Update: Name the Bot: *Airflow-Dag-Updates*)



After that is ready, make sure to turn on Incoming Webhooks for your workspace.

Airflow-Dag-Up...

Settings

- Basic Information
- Collaborators
- Socket Mode
- Install App
- Manage Distribution

Features

- App Home
- Agents & AI Apps NEW
- Workflow Steps NEW
- Org Level Apps
- Incoming Webhooks**
- Interactivity & Shortcuts
- Slash Commands
- Steps from Apps LEGACY
- OAuth & Permissions
- Event Subscriptions
- User ID Translation
- App Manifest

Incoming Webhooks

Activate Incoming Webhooks On

Incoming webhooks are a simple way to post messages from external sources into Slack. They make use of normal HTTP requests with a JSON payload, which includes the message and a few other optional details. You can include [message attachments](#) to display richly-formatted messages.

Adding incoming webhooks requires a bot user. If your app doesn't have a [bot user](#), we'll add one for you.

Each time your app is installed, a new Webhook URL will be generated.

If you deactivate incoming webhooks, new Webhook URLs will not be generated when your app is installed to your team. If you'd like to remove access to existing Webhook URLs, you will need to [Revoke All OAuth Tokens](#).

Webhook URLs for Your Workspace

To dispatch messages with your webhook URL, send your [message](#) in JSON as the body of an `application/json` POST request.

Add this webhook to your workspace below to activate this curl example.

You should receive a copyable link in this format:

<https://hooks.slack.com/services/TXXXXXXX/BXXXXXXX/XXXXXXX>

**** Note the Xs do not match the actual length of the key, make sure to store this somewhere safe**

After those steps are complete come back into Airflow UI on localhost:8081. Click on **Connections** Under the **Admin** tab.

The screenshot shows the Airflow Admin interface. The 'Admin' dropdown menu is open, and the 'Connections' option is highlighted with a red box. The main content area displays a table of DAGs (Directed Acyclic Graphs) with columns for DAG name, Owner, Runs, Schedule, Last Run, Next Run, Recent Tasks, Actions, and Links. The table lists several DAGs, including 'baseball-saxton-et-workflow', 'baseball-saxton-et-workflow-pyspark', 'load_mlb_game_prediction', 'play-dag-example', and 'spark_example_dag'.

Edit Connection

Connection id *

slack_incom

Connection Type *

Slack Incoming Webhook
Connection Type missing? Make sure you've installed the corresponding Airflow Provider Package.

Description

Slack Webhook Endpoint

https://hooks.slack.com/services/

Schema

https

Webhook Token

TB6G08G5B8008G02HXXXXXXXXXXXXXXXXXXXXXXXX

Timeout

30
Optional. The maximum number of seconds the client will wait to connect and receive a response from Slack Incoming Webhook.

Proxy

http://localhost:3030
Optional. Proxy to make the Slack incoming Webhook call.

```
Airflow-Dag-Updates APP: 3:44 PM
[baseball] [baseball] [baseball] [baseball] DAG baseball-savant-etl-workflow has completed successfully! [baseball] [baseball] [baseball] [baseball]
[baseball] Run ID: manual__2025-03-23T20:43:24.886910+00:00
[baseball] Run Type: manual
[baseball] Run Start Time: 2025-03-23 20:43:25.331083+00:00
[baseball] Run End Time: 2025-03-23 20:44:35.338043+0000
[baseball] Task States:
- Task: Load-DB-Current-DW-Info.load_all_game_pk | State: success
- Task: Load-DB-Current-DW-Info.load_all_hitter_pk | State: success
- Task: Load-DB-Current-DW-Info.load_all_pitcher_pk | State: success
- Task: Load-DB-Current-DW-Info.load_pitch_pks | State: success
- Task: slack_failure | State: skipped
- Task: slack_success | State: running
- Task: load_all_baseball_data.test_postgres_connection | State: success
- Task: load_all_baseball_data.create-sql-tables | State: success
- Task: Load-MLB-DW-Tables.load-game-table | State: success
- Task: Transform-Loaded-Savant-Data.transform_play_data | State: success
- Task: Transform-Loaded-Savant-Data.transform_pitch_data | State: success
- Task: Load-DB-Current-DW-Info.load_hit_pks | State: success
- Task: load_all_baseball_data.load_statcast_data | State: success
- Task: Load-MLB-DW-Tables.load-hitter-table | State: success
- Task: Transform-Loaded-Savant-Data.transform_pitcher_data | State: success
- Task: Load-MLB-DW-Tables.load-pitcher-table | State: success
- Task: Transform-Loaded-Savant-Data.transform_hitter_data | State: success
- Task: Load-DB-Current-DW-Info.load_play_pks | State: success
- Task: Transform-Loaded-Savant-Data.transform_game_data | State: success
- Task: Load-MLB-DW-Tables.load-hit-table | State: success
- Task: Load-MLB-DW-Tables.load-play-table | State: success
- Task: Load-MLB-DW-Tables.load-pitch-table | State: success
- Task: Transform-Loaded-Savant-Data.transform_hit_data | State: success
```