

# Основные элементы интерфейса Apache Airflow

## Начальная страница DAGs View

Это центральная панель мониторинга, где вы можете видеть все свои DAGs, их расписания и статус выполнения. Используйте теги для фильтрации DAGs и скрывания завершенных задач для более чистого просмотра.

Компоненты:

- Toggle: с помощью этой кнопки переключения мы можем либо приостановить запущенные потоки данных, либо запустить приостановленные потоки данных.
- DAG: DAG позволяет нам категоризировать наши потоки данных в соответствии с их функциональностью и командами.
- Owner: здесь владелец — это владелец наших данных - потоков данных, которых мы указали в настройках по умолчанию в нашем коде.
- Run: Запуски дает нам состояния текущих и прошлых запусков DAG, которые могут быть в очереди, успешными, выполняющимися или с ошибкой.
- Schedule: график определяет частоту, с которой будет запускаться наш поток данных. Он может быть ежедневным, ежемесячным, ежегодным и т. д.
- Last Run: Он указывает последний период времени, когда наш DAG был запущен.
- Next Run: Он указывает следующий период времени, когда поток данных будет запущен.
- Recent Tasks: даёт нам текущий статус наших потоков данных, который может быть в очереди, успешным, выполняющимся, пропущенным, остановленным, с ошибкой и т. д.
- Actions: с помощью действий мы можем запускать наш DAG с или без конфигурации, а также удалять метаданные, связанные с нашими DAG, такие как экземпляры задач или запуски DAG.

- Links: позволяют нам переходить к другим полезным видам в пользовательском интерфейсе Airflow.

The screenshot shows the Airflow web interface at localhost:8080/home. The top navigation bar includes links for Airflow, DAGs, Security, Browse, Admin, and Docs. The main section is titled 'DAGs' and displays a table of DAGs. The table has columns for DAG, Owner, Runs, Schedule, Last Run, Recent Tasks, Actions, and Links. A single DAG, '01\_umbrella', is listed with a status of 'Active' and a schedule of '@daily'. The interface also includes filters for DAGs by tag and a search bar.

## Tree View

Древоподобное представление помогает нам получить обзор истории и текущего состояния DAG, а также статуса задач. Это чрезвычайно полезно для выявления любых ошибок в задачах, просто взглянув на цвета квадратов. Мы также можем проверить, не опаздывает ли какая-либо группа DAG или нет ли каких-либо сбоев в выполнении задач.

The screenshot shows the Airflow web interface in 'Tree View' for the DAG '01\_umbrella'. The top navigation bar is the same as the previous screenshot. The main section is titled 'DAG: 01\_umbrella' and includes a description 'Umbrella example with DummyOperators'. Below the title are tabs for 'Tree View', 'Graph View', 'Task Duration', 'Task Tries', 'Landing Times', 'Gantt', and 'Details'. The 'Tree View' tab is selected, showing a tree diagram of the DAG's tasks. The tasks are: 'fetch\_weather\_forecast', 'clean\_forecast\_data', 'join\_datasets', 'train\_ml\_model', 'deploy\_ml\_model', 'fetch\_sales\_data', 'clean\_sales\_data', and 'join\_datasets'. A legend at the bottom indicates the status of tasks: 'queued' (grey), 'running' (green), 'success' (dark green), 'failed' (red), 'up\_for\_retry' (yellow), 'up\_for\_reschedule' (light blue), 'upstream\_failed' (orange), 'skipped' (pink), 'scheduled' (brown), and 'no\_status' (white). The tasks are shown as a series of green squares, indicating they are running or successful.

## Graph View

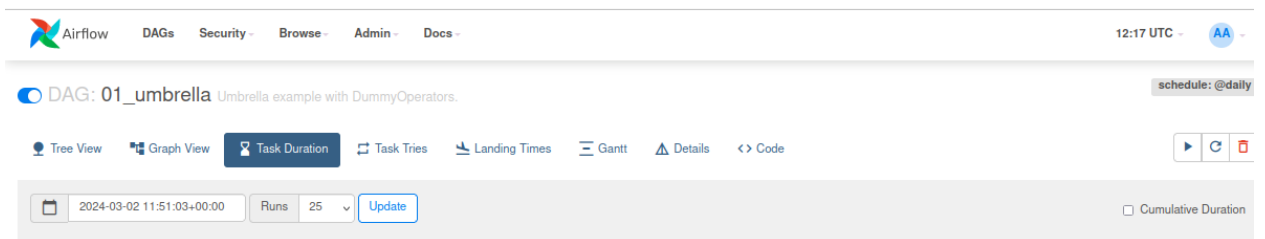
Графический вид в пользовательском интерфейсе Airflow идеально подходит для проверки зависимостей нашего конвейера данных, а также для получения статуса задач для последнего запуска DAG.

Здесь прямоугольники соответствуют задачам, а цвета границ соответствуют статусу задач.



## Task Duration

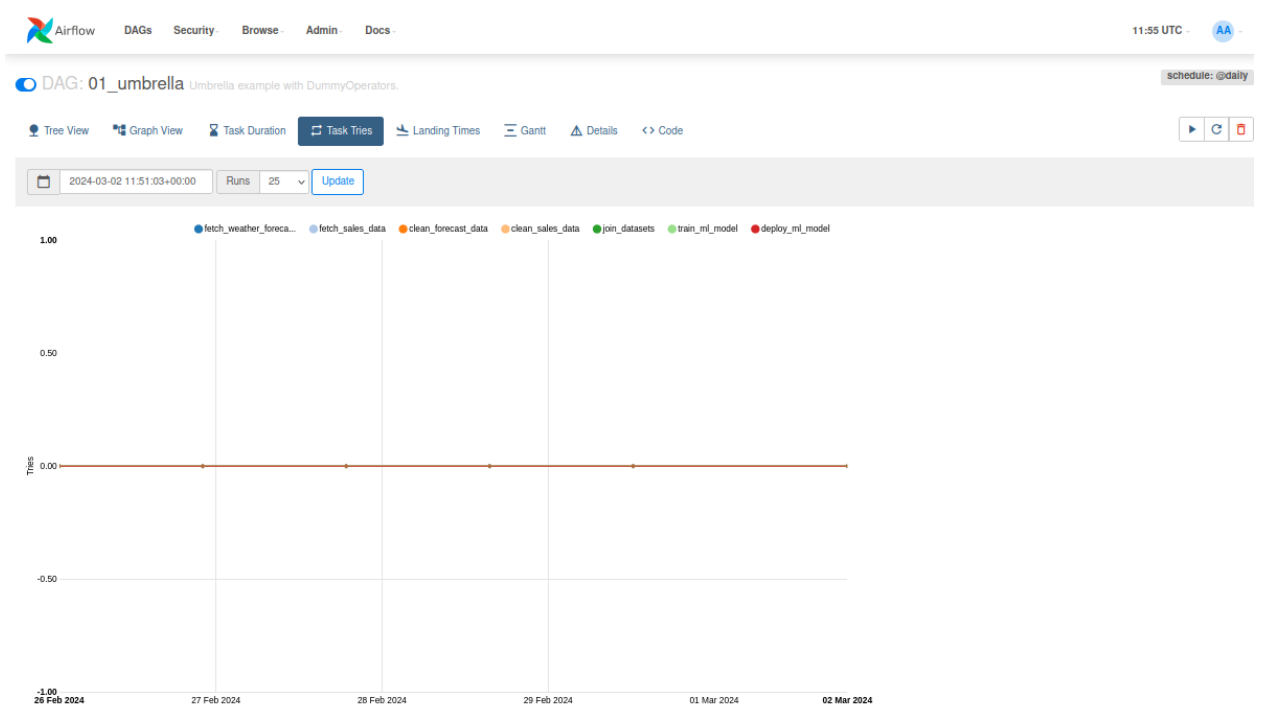
Продолжительность выполнения ваших различных задач за последние N запусков. Это представление позволяет вам находить отклонения и быстро понимать, на что тратится время в вашей работе за много запусков.



No Data Available.

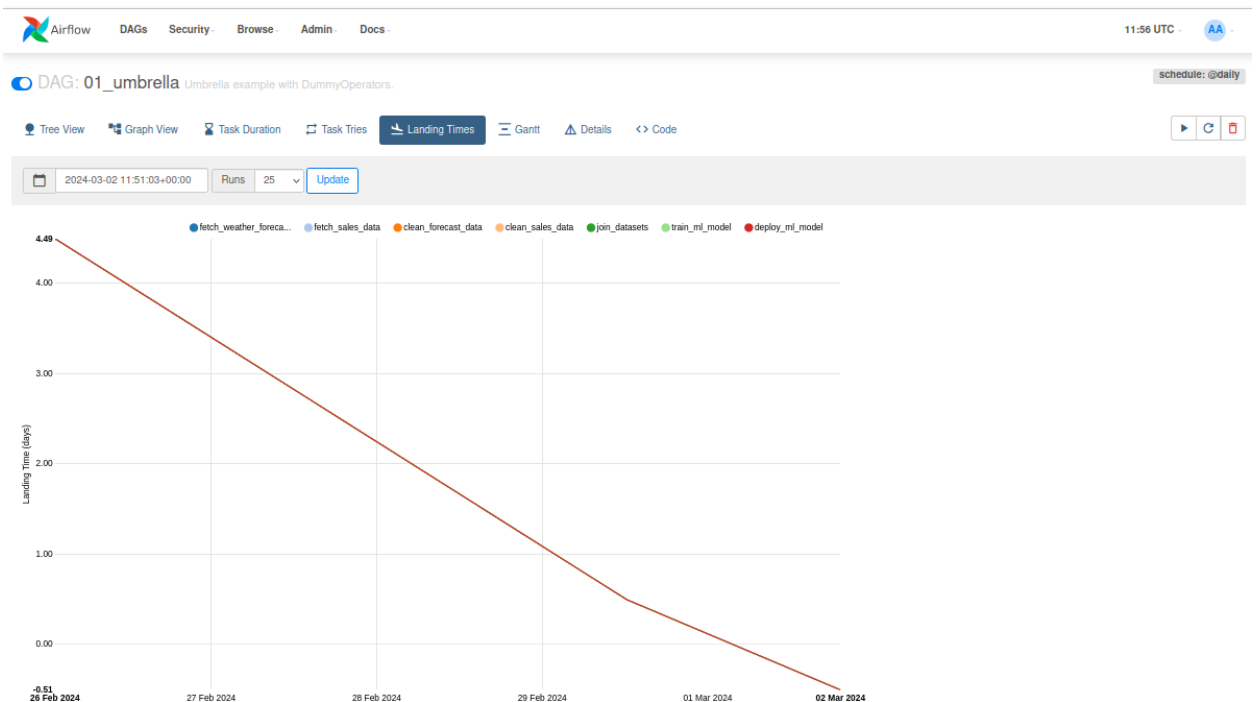
Task Tries отображает информацию о попытках выполнения конкретной задачи в рамках DAG (Directed Acyclic Graph).

Эта страница позволяет отслеживать процесс выполнения задачи и анализировать историю ее попыток выполнения для представления информации о работоспособности и состоянии пайплайна данных.



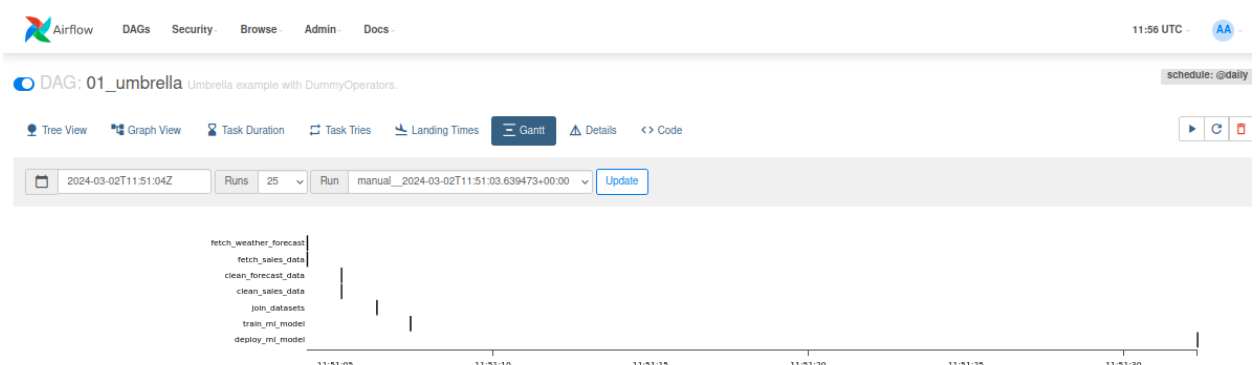
Landing Times

Время приземления экземпляра задачи — это разница между окончанием интервала данных запуска dag (обычно это означает, когда dag «должна» запуснуться) и временем завершения экземпляра задачи.



## Gantt View

Представление Ганта позволяет анализировать продолжительность задачи, а также совпадения. Взглянув на представление Ганта, можно быстро определить узкие места, а также то, где на конкретную диаграмму тратится большая часть времени. Чем больше прямоугольник в представлении Ганта, тем больше времени требуется для выполнения задачи



## Instance Details

This gives us additional information related to our task instance, like the dag\_id, dag\_model, dag\_run, etc.

Airflow

DAGs

Security

Browse

Admin

Docs

11:56 UTC

AA

DAG: 01\_umbrella

Umbrella example with DummyOperators.

schedule: @daily

Tree View

Graph View

Task Duration

Task Tries

Landing Times

Gantt

Details

<> Code

▶

↺

⌂

DAG Details

SUCCESS

42

Schedule Interval	@daily
Start Date	2024-02-26T00:00:00+00:00
End Date	None
Max Active Runs	0 / 16
Concurrency	16
Default Args	{}
Tasks Count	7
Task IDs	[fetch_weather_forecast, fetch_sales_data, clean_forecast_data, clean_sales_data, join_datasets, train_ml_model, deploy_ml_model]
Filepath	01_umbrella.py
Owner	airflow
Tags	None

## Code View

Код пайплайна находится в системе контроля версий, это быстрый способ получить доступ к коду, который генерирует DAG, и обеспечить еще больший контекст.

Airflow

DAGs

Security

Browse

Admin

Docs

11:57 UTC

AA

DAG: 01\_umbrella

Umbrella example with DummyOperators.

schedule: @daily

Tree View

Graph View

Task Duration

Task Tries

Landing Times

Gantt

Details

<> Code

▶

↺

⌂

Toggle Wrap

```
1 """DAG demonstrating the umbrella use case with dummy operators."""
2
3 import airflow.utils.dates
4 from airflow import DAG
5 from airflow.operators.dummy import DummyOperator
6
7 dag = DAG(
8     dag_id="01_umbrella",
9     description="Umbrella example with DummyOperators.",
10    start_date=airflow.utils.dates.days_ago(5),
11    schedule_interval="@daily",
12 )
13
14 fetch_weather_forecast = DummyOperator(task_id="fetch_weather_forecast", dag=dag)
15 fetch_sales_data = DummyOperator(task_id="fetch_sales_data", dag=dag)
16 clean_forecast_data = DummyOperator(task_id="clean_forecast_data", dag=dag)
17 clean_sales_data = DummyOperator(task_id="clean_sales_data", dag=dag)
18 join_datasets = DummyOperator(task_id="join_datasets", dag=dag)
19 train_ml_model = DummyOperator(task_id="train_ml_model", dag=dag)
20 deploy_ml_model = DummyOperator(task_id="deploy_ml_model", dag=dag)
21
22 # Set dependencies between all tasks
23 fetch_weather_forecast >> clean_forecast_data
24 fetch_sales_data >> clean_sales_data
25 [clean_forecast_data, clean_sales_data] >> join_datasets
26 join_datasets >> train_ml_model >> deploy_ml_model
```