



AIP-68 Plugins in
Airflow 3

New plugin interface

3.0



Speakers

Jens Scheffler

Cluster Technical Architect @ Bosch
Airflow Committer & PMC Member

Brent Bovenzi

Engineer @ Astronomer
Airflow Committer & PMC Member

Pierre Jeambrun

Engineer @ Astronomer
Airflow Committer & PMC Member



Today session

How do you extend the API & UI in Airflow 3. (with code samples)

Deep dive into a more complex use case, the edge provider

Discussion / Q&A



Why new plugins?

We rebuilt entirely the API and UI.

In 3.1 we wanted to leverage React and FastAPI to bring new functionality

Legacy plugins are still supported



Ways to plug into Airflow in 3.1 (API)

- FastAPI Subapps
- Middlewares

Both mounted on the api server root application.



Ways to plug into Airflow in 3.1 (UI)

Different ways and destinations are supported:

- External links (Nav bar only)
- Embedded iframes (Nav bar and Details tab)
- React Apps (Nav bar, Details Tab, Dashboard page)



Demo

3.0



FastAPI Sub Application



```
from fastapi import FastAPI, Request  
  
analytics_app = FastAPI()  
  
@analytics_app.get("/hello")  
async def root():  
    return {"message": "Hello World from FastAPI plugin"}
```



```
class AirflowTestPlugin(AirflowPlugin):
    """A simple Airflow plugin."""

    # A list of dicts containing a FastAPI created from fastapi.FastAPI and some metadata.
    # See example below.
    fastapi_apps = [
        {
            "app": analytics_app,
            "url_prefix": "/analytics-plugin",
            "name": "Analytics SubApp",
            "extra": "something",
        },
    ]
```



Home

Dags

Assets

Browse

Admin

Security

Plugins

Docs

User

Welcome to Your New React App!

This project was bootstrapped with the Airflow React Plugin tool.

[Toggle Theme](#)

Welcome

[Stats](#)

[Failed Dags >](#)

[Dag Import Errors >](#)

[Running Dags >](#)

[Active Dags >](#)

[First 10 favorite Dags](#)

No favorites yet. Click the star icon next to a Dag in the list to add it to your favorites.

[Health](#)

[MetaDatabase](#)

[Scheduler](#)

[Triggerer](#)

[Dag Processor](#)

[Pool Slots](#)

✓ 128

[Manage Pools](#)

[History](#)

Last 24 Hours

2025-10-01 15:18:23 - 2025-10-02 15:18:23

[Dag Run](#)

[Queued](#)

0.00%

[Running](#)

0.00%

[Success](#)

100.00%

[Failed](#)

0.00%

[Task Instance](#)

[Success](#)

100.00%

[Asset Events](#)

Newest First

No Asset Events found.



FastAPI Root Middlewares



```
class InjectIndexAlertMiddleware(BaseHTTPMiddleware):
    """Middleware to inject a script into the index page."""

    async def dispatch(self, request: Request, call_next):
        response = await call_next(request)

        print(request.url.path)

        if "text/html" in response.headers.get("content-type", []):
            if request.url.path == "/":
                response = StreamingResponse(inject_script(response.body_iterator), media_type="text/html")
        return response
```



```
class AirflowTestPlugin(AirflowPlugin):
    """A simple Airflow plugin."""

    # A list of dictionaries containing FastAPI middleware factory object and some metadata.
    # See example below.
    fastapi_root_middlewares = [
        {
            "middleware": InjectIndexAlertMiddleware,
            "args": [],
            "kwargs": {},
            "name": "Inject Alert Middleware",
        }
    ]
```



Q Search Dags

Advanced Search

All Failed

Queued

Running

Success

Required Actions

All

Filter by tag

All

109 Dags

Sort by Display Name (A-Z)



48670_bug

Schedule

Latest Run

Next Run

0 0 * * *

2025-10-01 17:00:00

2025-10-02 17:00:00



49316_bug

Schedule

Latest Run

Next Run



49316_bug_af2

Schedule

Latest Run

Next Run



49316_bug_alias

Schedule

Latest Run

Next Run



49316_bug_alias_af2

Schedule

Latest Run

Next Run



53817_bug

Schedule

Latest Run

Next Run



0 of 2 assets updated

53998_bug





UI Plugins



Iframe



```
# Creating an external view that will be rendered in the Airflow UI.
external_view_with_metadata = {
    # Name of the external view, this will be displayed in the UI.
    "name": "Name of the External View",
    # Source URL of the external view. This URL can be templated using context variables, depending on the
    location where the external view is rendered
    # the context variables available will be different, i.e a subset of (DAG_ID, RUN_ID, TASK_ID, MAP_INDEX).
    "href": "https://example.com/{DAG_ID}/{RUN_ID}/{TASK_ID}/{MAP_INDEX}",
    # Destination of the external view. This is used to determine where the view will be loaded in the UI.
    # Supported locations are Literal["nav", "dag", "dag_run", "task", "task_instance"], default to "nav".
    "destination": "dag_run",
    # Optional icon, url to an svg file.
    "icon": "https://example.com/icon.svg",
    # Optional dark icon for the dark theme, url to an svg file. If not provided, "icon" will be used for both
    light and dark themes.
    "icon_dark_mode": "https://example.com/dark_icon.svg",
    # Optional parameters, relative URL location for the External View rendering. If not provided, external view
    will be rendered as an external link. If provided
    # will be rendered inside an Iframe in the UI. Should not contain a leading slash.
    "url_route": "my_external_view",
    # Optional category, only relevant for destination "nav". This is used to group the external links in the
    navigation bar. We will match the existing
    # menus of ["browse", "docs", "admin", "user"] and if there's no match then create a new menu.
    "category": "browse",
}
```

```
● ● ●  
class AirflowTestPlugin(AirflowPlugin):  
    """A simple Airflow plugin."  
  
    external_views = [external_view_with_metadata]
```



Home

Dags

Assets

Browse

Admin

Security

Plugins

Docs

User

Welcome to Your New React App!

This project was bootstrapped with the Airflow React Plugin tool.

[Toggle Theme](#)

Welcome

Stats

1 Failed Dags >

5 Dag Import Errors >

0 Running Dags >

2 Active Dags >

First 10 favorite Dags

No favorites yet. Click the star icon next to a Dag in the list to add it to your favorites.

Health

✓ MetaDatabase

✓ Scheduler

✓ Triggerer

✘ Dag Processor

Pool Slots

✓ 128

Manage Pools

History

Last 24 Hours

2025-09-29 10:20:37 - 2025-09-30 10:20:37

1 Dag Run

0 Queued

0.00%

0 Running

0.00%

0 Success

0.00%

1 Failed

100.00%

1 Task Instance

1 Failed

100.00%

0 Asset Events

Newest First

No Asset Events found.



External Links



Home



Dags



Assets



Browse



Admin



Security



Plugins



Docs



User

To exit full screen, press and hold **esc** Top!

This project was bootstrapped with the Airflow React Plugin tool.

Toggle Theme

Welcome

Stats

✗ 1 Failed Dags >

✗ 5 Dag Import Errors >

✓ 0 Running Dags >

⌚ 2 Active Dags >

First 10 favorite Dags

No favorites yet. Click the star icon next to a Dag in the list to add it to your favorites.

Health

✓ MetaDatabase

✓ Scheduler

✓ Triggerer

✓ Dag Processor

Pool Slots

✓ 128

Manage Pools

History

Last 24 Hours

2025-09-29 10:25:00 - 2025-09-30 10:25:00

⌚ 1 Dag Run

➡ 0 Queued

0.00%

⌚ 0 Running

0.00%

✓ 0 Success

0.00%

✗ 1 Failed

100.00%

⌚ 1 Task Instance

✗ 1 Failed

⌚ 0 Asset Events

Newest First

No Asset Events found.



NavBar Categories

- Create a new menu on navbar
- Add to existing menu



Home

Dags

Assets

Browse

Admin

Security

Plugins

Docs

User

Welcome to Your New React App!

This project was bootstrapped with the Airflow React Plugin tool.

[Toggle Theme](#)

Welcome

[Stats](#)

[Failed Dags >](#)

[Dag Import Errors >](#)

[Running Dags >](#)

[Active Dags >](#)

[First 10 favorite Dags](#)

No favorites yet. Click the star icon next to a Dag in the list to add it to your favorites.

[Health](#)

[MetaDatabase](#)

[Scheduler](#)

[Triggerer](#)

[Dag Processor](#)

[Pool Slots](#)

✓ 128

[Manage Pools](#)

[History](#)

Last 24 Hours

2025-09-29 10:27:26 - 2025-09-30 10:27:26

[Dag Run](#)

[Queued](#)

0.00%

[Running](#)

0.00%

[Success](#)

0.00%

[Failed](#)

100.00%

[Task Instance](#)

[Failed](#)

[Asset Events](#)

Newest First

No Asset Events found.



React Apps (experimental)

```
# Note: The React app integration is experimental and interfaces might change in future versions.  
react_app_with_metadata = {  
    # Name of the React app, this will be displayed in the UI.  
    "name": "Name of the React App",  
    # Bundle URL of the React app. This is the URL where the React app is served from. It can be a static file  
    # or a CDN. This URL can be templated using context variables, depending on the location where the external  
    # view is rendered, the context variables available will be different,  
    # i.e a subset of (DAG_ID, RUN_ID, TASK_ID, MAP_INDEX).  
    "bundle_url": "https://example.com/static/js/my_react_app.js",  
    # Destination of the react app. This is used to determine where the app will be loaded in the UI.  
    # Supported locations are Literal["nav", "dag", "dag_run", "task", "task_instance"], default to "nav".  
    # it can also be put inside of an existing page, the supported views are ["dashboard", "dag_overview",  
    # "task_overview"]. You can position  
    # element in the existing page via the css `order` rule which will determine the flex order.  
    "destination": "dag_run",  
    # Optional icon, url to an svg file.  
    "icon": "https://example.com/icon.svg",  
    # Optional dark icon for the dark theme, url to an svg file. If not provided, "icon" will be used for both  
    # light and dark themes.  
    "icon_dark_mode": "https://example.com/dark_icon.svg",  
    # URL route for the React app, relative to the Airflow UI base URL. Should not contain a leading slash.  
    "url_route": "my_react_app",  
    # Optional category, only relevant for destination "nav". This is used to group the react apps in the  
    # navigation bar. We will match the existing  
    # menus of ["browse", "docs", "admin", "user"] and if there's no match then create a new menu.  
    "category": "browse",  
}
```



```
class AirflowTestPlugin(AirflowPlugin):
    """A simple Airflow plugin."""

    react_apps = [react_app_with_metadata]
```



Home

Dags

Assets

Browse

Admin

Security

Plugins

Docs

User

Welcome to Your New React App!

This project was bootstrapped with the Airflow React Plugin tool.

[Toggle Theme](#)

Welcome

[Stats](#)

[Failed Dags >](#)

[Dag Import Errors >](#)

[Running Dags >](#)

[Active Dags >](#)

[First 10 favorite Dags](#)

No favorites yet. Click the star icon next to a Dag in the list to add it to your favorites.

[Health](#)

[MetaDatabase](#)

[Scheduler](#)

[Triggerer](#)

[Dag Processor](#)

[Pool Slots](#)

✓ 128

[Manage Pools](#)

[History](#)

Last 24 Hours

2025-09-29 10:25:00 - 2025-09-30 10:25:00

[Dag Run](#)

[Queued](#)

0.00%

[Running](#)

0.00%

[Success](#)

0.00%

[Failed](#)

100.00%

[Task Instance](#)

[Failed](#)

100.00%

[Asset Events](#)

Newest First

No Asset Events found.



Destinations

- Add a NavBar item
- Add to Dashboard (React)
- Add a new Dag/Dag Run/Task/Task Instance tab (React & Iframe)



Home

Dags

Assets

Browse

Admin

Security

Plugins

Docs

User

Welcome to Your New React App!

This project was bootstrapped with the Airflow React Plugin tool.

[Toggle Theme](#)

Welcome

Stats

✗ 1 Failed Dags >

✗ 5 Dag Import Errors >

✓ 0 Running Dags >

⌚ 2 Active Dags >

First 10 favorite Dags

No favorites yet. Click the star icon next to a Dag in the list to add it to your favorites.

Health

✓ MetaDatabase

✓ Scheduler

✓ Triggerer

✓ Dag Processor

Pool Slots

✓ 128

Manage Pools

History

Last 24 Hours

2025-09-29 10:28:57 - 2025-09-30 10:28:57

⌚ 1 Dag Run

➡ 0 Queued

0.00%

⌚ 0 Running

0.00%

✓ 0 Success

0.00%

✗ 1 Failed

100.00%

⌚ 1 Task Instance

✗ 1 Failed

100.00%

⌚ 0 Asset Events

Newest First

No Asset Events found.



Home

Dags

Assets

Browse

Admin

Security

Plugins

Docs

User

Welcome to Your New React App!

This project was bootstrapped with the Airflow React Plugin tool.

[Toggle Theme](#)

Welcome

Stats

1 Failed Dags >

5 Dag Import Errors >

0 Running Dags >

2 Active Dags >

First 10 favorite Dags

No favorites yet. Click the star icon next to a Dag in the list to add it to your favorites.

Health

✓ MetaDatabase

✓ Scheduler

✓ Triggerer

✓ Dag Processor

Pool Slots

✓ 128

[Manage Pools](#)

History

Last 24 Hours

2025-09-29 10:28:57 - 2025-09-30 10:28:57

1 Dag Run

0 Queued

0.00%

0 Running

0.00%

0 Success

0.00%

1 Failed

100.00%

1 Task Instance

1 Failed

100.00%

0 Asset Events

Newest First

No Asset Events found.

Airflow

localhost:8080/dags

Dags Dag Runs Task Instances

Search Dags Advanced Search

All Failed Queued Running Success Required Actions All Filter by tag All

1 Dag

Dag ID Schedule Next Run Latest Run Tags

Dag ID	Schedule	Next Run	Latest Run	Tags
very_important_dag	0 0 * * *	2025-09-28 17:00:00		

very_important_dag

0 0 * * *

2025-09-28 17:00:00

Dag ID Schedule Next Run Latest Run Tags

very_important_dag 0 0 * * * 2025-09-28 17:00:00

Home Dags Assets Browse Admin Plugins Docs User



Edge Executor UI Plugin Example

3.1



Edge Executor UI

What we did
to make it happen

Checklist (reference PR #53563)

- Bootstrap Plugin via
`dev/react-plugin-tools/bootstrap.py`
- Add Plugin With
 - Register `react_apps` + `fastapi_apps`
 - FastAPI Mounting `dist` files
 - Add content-type for `cjs` files!
 - Add needed API routes + authentication
 - Generate API stub code via axios
- Develop your pages

Bootstrap Plugin via `dev/react-plugin-tools/bootstrap.py`

```
(airflow) jscheffl@hp860g9:~/Workspace/airflow$ cd dev/react-plugin-tools/
(airflow) jscheffl@hp860g9:~/Workspace/airflow/dev/react-plugin-tools$ python bootstrap.py edge-plugin --dir providers/edge3/src/airflow/providers/edge3/plugins/www
Creating React plugin project: edge-plugin
Target directory: /home/jscheffl/Workspace/airflow/dev/react-plugin-tools/providers/edge3/src/airflow/providers/edge3/plugins/www
Template directory: /home/jscheffl/Workspace/airflow/dev/react-plugin-tools/react_plugin_template
Copying template files...
Created: .gitignore
Created: .prettierignore
Created: .prettierrc
Created: eslint.config.js
Created: index.html
Created: testsSetup.ts
Created: tsconfig.app.json
Created: tsconfig.json
Created: tsconfig.lib.json
Created: tsconfig.node.json
Created: README.md
Created: package.json
Created: pnpm-lock.yaml
Created: vite.config.ts
Created: src/dev.tsx
Created: src/main.tsx
Created: src/vite-env.d.ts
Created: src/theme.ts
Created: src/pages/HomePage.tsx
Created: src/context/colorMode/index.ts
Created: src/context/colorMode/useColorMode.tsx
Created: src/context/colorMode/ColorModeProvider.tsx
```

✓ Successfully created edge-plugin!

Next steps:

```
cd providers/edge3/src/airflow/providers/edge3/plugins/www
pnpm install
pnpm dev
```

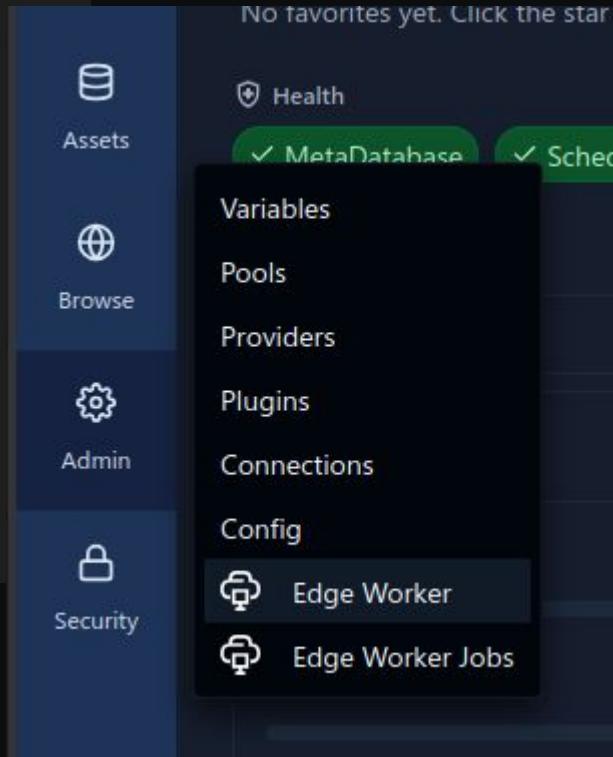
Happy coding! 🚀

Register react_apps + fastapi_apps



```
class EdgeExecutorPlugin(AirflowPlugin):
    """EdgeExecutor Plugin - provides API endpoints for Edge Workers in Webserver.

    name = "edge_executor"
    if EDGE_EXECUTOR_ACTIVE and RUNNING_ON_APISERVER:
        if AIRFLOW_V_3_1_PLUS:
            fastapi_apps = [_get_api_endpoint()]
            react_apps = [
                {
                    "name": "Edge Worker",
                    "bundle_url": "/edge_worker/static/main.umd.cjs",
                    "destination": "nav",
                    "url_route": "edge_worker",
                    "category": "admin",
                    "icon": "/edge_worker/res/cloud-computer.svg",
                    "icon_dark_mode": "/edge_worker/res/cloud-computer-dark.svg",
                }
            ]
    
```



FastAPI Routes & Content Type



```
ui_router = AirflowRouter(tags=["UI"])

@ui_router.get(
    "/worker",
    dependencies=[
        Depends(requires_access_view(access_view=AccessView.JOBS)),
    ],
)
def worker(
    session: SessionDep,
) -> WorkerCollectionResponse:
    """Return Edge Workers."""
    query = select(EdgeWorkerModel).order_by(EdgeWorkerModel.worker_name)
    workers: list[EdgeWorkerModel] = session.scalars(query)
    ...

```

Do not forget authentication!



```
ui_router = AirflowRouter(tags=["UI"])

@ui_router.get(
    "/worker",
    dependencies=[  
        Depends(requires_access_view(access_view=AccessView.JOBS)),  
    ],  
)  
def worker(  
    session: SessionDep,  
) -> WorkerCollectionResponse:  
    """Return Edge Workers."""  
    query = select(EdgeWorkerModel).order_by(EdgeWorkerModel.worker_name)  
    workers: list[EdgeWorkerModel] = session.scalars(query)  
    ...
```

The Outcome - A Fully Integrated View

This screenshot illustrates a fully integrated view of a cloud-based workflow management system, specifically focusing on Edge Workers and Edge Jobs.

Edge Worker Overview: On the left, a sidebar menu includes Home, Dags, Assets, Browse, Admin, Security, and Docs. The main area shows three Edge Workers: "breeze" (idle, default queue), "crash-and-burn" (in maintenance mode, all queues), and "far-far-away" (offline, all queues). Each worker has a timestamped log entry below it.

Edge Job Maintenance Dialog: A modal dialog titled "Set maintenance for worker breeze" is open. It contains the message "Is this not cool?" and two buttons: "Cancel" and "Confirm Maintenance".

Operations Log: To the right, an "Operations" section displays a log of events:

- [2025-10-04 11:35] - admin put node into maintenance mode
- [2025-10-04 11:36] - admin updated comment: Shall not pull jobs

Below the log are edit and delete icons.



Edge Executor UI

Learnings

Learnings / Future Improvements

- Always use the Bootstrapping tool
- Improvements for Shared Components
- Dev Env with Vite needs improvement
- Build NPM during package needs hatchling
- Internal Links needed fix → Airflow 3.1.1
- Pop-ups/Toaster need fix
- Tight coupling React+DOM → Plugin Deps
- React Integration is Experimental



Breakout
Group
Discussion

3.0



Breakout Group Discussion

Future Improvements:

- Expose utility code from Airflow Core to Plugins
- Increase the surface covered by possible destinations



The 2025 Apache Airflow® Survey is here!

Fill it out to for a free Airflow 3 Fundamentals or DAG Authoring in Airflow 3 certification code

