



Airflow Summit



Advanced Apache Superset for Data Engineers





Maxime Beauchemin

mistercrunch

creator of Apache Airflow and Apache Superset - founder at Preset

Edit profile

1k followers · 11 following · 139

preset-io

San Mateo, CA

maximebeauchemin@gmail.com

mistercrunch.blogspot.com

Organizations



- A passion for building data tools!
- Started Apache **Airflow** at Airbnb in 2014
- Started Apache **Superset** at Airbnb in 2015
- Started **Preset** - The Apache Superset company in 2019



Agenda!

- Superset Overview / Demo
- SQL Lab for data engineers
- Scheduling Queries
- Building a visualization plugin
- Building charts and dashboards dynamically



Superset Overview / Demo!

Enhancing Jinja Context

```
# in superset_config.py
JINJA_CONTEXT_ADDONS = {
    "say_hello": lambda: 'hello',
}
```

Scheduling Queries

experimental feature!

feat: Scheduling queries from SQL Lab #7416

Merged

betodealmeida merged 8 commits into apache:lyft-release-sp8 from lyft:VIZ-3a on May 3, 2019

Conversation 19

Commits 8

Checks 0

Files changed 7



betodealmeida commented on Apr 30, 2019 • edited by mistercrunch

Member



...

SUMMARY

This PR introduces a lightweight way of scheduling queries in SQL Lab. If the feature flag `SCHEDULED_QUERIES` is enabled with proper configuration, a button called "Schedule Query" will show up in SQL Lab. The button allows queries to be saved with extra metadata that allows an external scheduler to run it periodically by polling the `/savedqueryviewapi/api/read` endpoint.

The sample configuration can be changed or expanded to support different metadata needed, depending on the scheduler. We tested it with [Apache Airflow](#) at Lyft successfully.

RUN

SCHEDULE QUERY

SAVE

SHARE

LIMIT 1000



superset_config.py

<http://superset.apache.org/installation.html#sql-lab>

```
FEATURE_FLAGS = {
    # Configuration for scheduling queries from SQL Lab. This information is
    # collected when the user clicks "Schedule query", and saved into the 'extra'
    # field of saved queries.
    # See: https://github.com/mozilla-services/react-jsonschema-form
    'SCHEDULED_QUERIES': {
        'JSONSCHEMA': {
            'title': 'Schedule',
            'description': (
                'In order to schedule a query, you need to specify when it '
                'should start running, when it should stop running, and how '
                'often it should run. You can also optionally specify '
                'dependencies that should be met before the query is '
                'executed. Please read the documentation for best practices '
                'and more information on how to specify dependencies.'
            ),
            'type': 'object',
            'properties': {
                'output_table': {
                    'type': 'string',
                    'title': 'Output table name',
                },
                'start_date': {
                    'type': 'string',
                    'title': 'Start date',
                    # date-time is parsed using the chrono library, see
                    # https://www.npmjs.com/package/chrono-node#usage
                    'format': 'date-time',
                    'default': 'tomorrow at 9am',
                },
                'end_date': {
                    'type': 'string',
                    'title': 'End date',
                    # date-time is parsed using the chrono library, see
                    # https://www.npmjs.com/package/chrono-node#usage
                }
            }
        }
    }
}
```

```
, '',
'UISCHEMA': {
    'schedule_interval': {
        'ui:placeholder': '@daily, @weekly, etc.',
    },
    'dependencies': {
        'ui:help': (
            'Check the documentation for the correct format when '
            'defining dependencies.'
        ),
    },
},
'VALIDATION': [
    # ensure that start_date <= end_date
    {
        'name': 'less_equal',
        'arguments': ['start_date', 'end_date'],
        'message': 'End date cannot be before start date',
        # this is where the error message is shown
        'container': 'end_date',
    },
],
# link to the scheduler; this example links to an Airflow pipeline
# that uses the query id and the output table as its name
'linkback': (
    'https://airflow.example.com/admin/airflow/tree?',
    'dag_id=query_${id}_${extra_json.schedule_info.output_table}'
),
}
}
```

Backgrounds SQL Lab

Schedule Query

Label

Description

Schedule

In order to schedule a query, you need to specify when it should start running, when it should stop running, and how often it should run. You can also optionally specify dependencies that should be met before the query is executed. Please read the documentation for best practices and more information on how to specify dependencies.

Output table name

Start date

 CALENDAR

End date

 CALENDAR

Schedule interval

Dependencies

ADD

Check the documentation for the correct format when defining dependencies.

SUBMIT

```
[{"result": [{"description": null, "extra": {"schedule_info": {"dependencies": ["hive://SOURCE_TABLE/{{ds}}"], "output_table": "THIS_IS_THE_OUTPUT_TABLE", "schedule_interval": "@daily", "start_date": "2020-07-08T19:08:00.000Z"}, "extra_json": "{\"schedule_info\":{\"output_table\":\"THIS_IS_THE_OUTPUT_TABLE/{{ds}}\"}}", "id": 2, "label": "AIRFLOW SUMMIT !!!", "schema": "superset", "sql": "SELECT 'HELLO AIRFLOW SUMMIT' as label", "sqlalchemy_uri": "mysql://root@localhost/examples?charset=utf8", "user_email": "admin@fab.org"}]}]
```

Visualization Plugins

<https://preset.io/blog/>



SUPERSET DEVELOPERS

So, You Want to Build a Superset Viz Plugin...

Evan Rusackas July 02, 2020

Superset Plugins as a data product development platform

- Build data products without writing much backend code
- Tap into Superset's Data Access Layer (auth, perm, cache, audit)
- Rich controls at your fingertips
- Focus on the visualization / frontend
- Bring into a dashboard (surround with context / add interactions)

Dynamic Chart/Dashboard Creation

Rest API!

Swagger API @ /swagerview/v1

The screenshot shows the Superset interface with the title "Swagger API @ /swagerview/v1". The top navigation bar includes links for Superset, Security, Manage, Sources, Charts, Dashboards, and SQL Lab, along with a "+ NEW" button and user authentication dropdowns. Below the title, the word "Superset" is displayed next to two small circular icons. A sub-navigation bar shows "/api/v1/_openapi". The main content area is titled "Charts" and lists several API endpoints for the "/chart/" path:

Method	Path	Status
DELETE	/chart/	Red background, locked
GET	/chart/	Blue background, locked
POST	/chart/	Green background, locked
GET	/chart/_info	Blue background, locked
POST	/chart/data	Green background, locked
GET	/chart/datasources	Blue background, locked
		Blue background, locked

A "Servers" dropdown menu is open, showing the selected value "/api/v1". A green "Authorize" button with a lock icon is located to the right of the servers dropdown.

Using SQLAlchemy (improper)

/superset/examples/birth_names.py

```
defaults = {
    "compare_lag": "10",
    "compare_suffix": "o10Y",
    "limit": "25",
    "granularity_sqla": "ds",
    "groupby": [],
    "row_limit": config["ROW_LIMIT"],
    "since": "100 years ago",
    "until": "now",
    "viz_type": "table",
    "markup_type": "markdown",
}

admin = security_manager.find_user("admin")

print("Creating some slices")
slices = [
    Slice(
        slice_name="Participants",
        viz_type="big_number",
        datasource_type="table",
        datasource_id=tbl.id,
        params=get_slice_json(
            defaults,
            viz_type="big_number",
            granularity_sqla="ds",
            compare_lag="5",
            compare_suffix="over 5Y",
            metric=metric,
        ),
    ),
    Slice(
        slice_name="Genders",
        viz_type="pie",
        datasource_type="table"
    )
]
```

We're hiring!

[Product](#)[About Us](#)[Careers](#)[Resources](#) ▾[Community](#) ▾[Discord](#)

Careers at Preset

Preset is actively hiring a team to build features and services in and around Apache Superset, the leading open source analytics and data visualization platform.

