Advanced Superset for Engineers

Airflow Summit
Srini Kadamati
Overview

Sections

• Tour of Superset
• Superset API
• Version Controlling Charts & Dashboards
• The Future
• How to Contribute!
About Me

Previously
Data Scientist for 2 years
Data Science Educator / Product Manager at Dataquest for 5 years

Currently
Senior Developer Advocate at Preset
+Committer to Superset
What is Apache Superset?

Modern open source BI platform

Works with nearly any SQL speaking data engine

Large diversity of charts
What is Preset?

Preset Cloud is a multi-workspace, cloud-hosted Apache Superset (with all the batteries, bells, & whistles included)

We also layer on:

• SOC2 Compliance
• End-user documentation at docs.preset.io
• RBAC and SSO
Tour of Superset

Homepage
Databases
Datasets
Saved Queries
SQL Lab
Charts
Dashboards

Semantic Layer
• Metrics
• Calculated Columns
• Virtual datasets

Jinja Filters
Alerts & Reports
Row Level Security
Superset API
API Demo!

```python
import pandas as pd
import json
import requests

# Authenticate with API

payload = {
    'username': 'admin',
    'password': 'admin',
    'provider': 'db'
}

r = requests.post('http://localhost:9000/api/v1/security/login', json=payload)

access_token = r.json()['access_token']

# Headers

headersAuth = {
    'Authorization': 'Bearer ' + str(access_token)
}
```
Tech Specs

Follows OpenAPI specification and is RESTful

Built mostly using Flask App Builder

Interactive Swagger documentation lives at /swagger/v1 endpoint

Blog: preset.io/blog/2020-10-01-superset-api/
API Use Cases

Invalidate cache upon data refresh

  • GET /chart/data{cache_key}
  • POST /cachekey/invalidate

Import, export dashboards

  • GET /dashboard/export
  • POST /dashboard/import

Create new database

  • POST /database/

Return SELECT * from a db table

  • GET /database/{pk}/select_star/{table_name}/

Schedule a new report

  • POST /report
Version Control
Superset CLI

Version control workflow is hacky, and is more of a PUSH-into-Superset workflow than a PULL one

1. Export Dashboard (from UI, *shell coming soon*)
2. Import Dashboard (from *superset shell*)

Beto Dealmeida opened a PR [#15686](https://example.com) to enable a CLI command for importing a dashboard:

```
superset import-directory -o ~/Downloads/dashboard_export_20210715T205336/
```
Version Control Demo!

```
[INFO]   - supercat git (import_dir_command) supercat import-directory -m -z -Downloads/dashboard_export_202187157295336/
loaded your LOCAL configuration at /Users/srinify/Documents/superset/supersetConfig.py
logging was configured successfully
2021-07-15 21:16:55,107:INFO:superset.utils.logging_configuration:logging was configured successfully
/Users/srinify/Documents/superset/superset/examples/utils.py:85: YAMLLoadWarning: calling yaml.load() without loader=None is deprecated, as the default loader is unsafe. Please read https://msg.pyyaml.org/for details.
metadata = yaml.load(contents.get(METADATA_FILE_NAME, "")
Updating db examples
Updating tables birth_names
Updating sql_metrics <superset.connectors.sqla.models.SqlMetric object at 0x1c0b093a8>
2021-07-15 21:16:56,284:INFO:superset.models.helpers:Updating sql_metrics <superset.connectors.sqla.models.SqlMetric object at 0x1c0b093a8>
Updating sql_metrics <superset.connectors.sqla.models.SqlMetric object at 0x1c0b093a8>
2021-07-15 21:16:56,296:INFO:superset.models.helpers:Updating sql_metrics <superset.connectors.sqla.models.SqlMetric object at 0x1c0b093a8>
Updating table_columns num_california
Updating table_columns state
Updating table_columns gender
Updating table_columns null
Updating table_columns des
Updating table_columns sun_boys
Updating table_columns sun_girls
Updating table_columns num
Updating slices Pivot Table
Importing new dashboards DashboardNone
```
Version Control Workflow

1. Use Superset to iterate on dashboard
2. Export dashboard, as static files
3. Check in static files to Github
4. Setup CI workflow to
Next Steps

Export dashboards (as ZIP) via API
Better error messages, documentation, & developer experience

Eventually:
- Implement design & UI for version control within Superset itself
Embedding Superset

Expose individual standalone charts
Expose individual standalone dashboards
...anddd IFrame!

How?
• Use public role as gamma ([docs](#))

Help us improve this by joining [#embedding-superset](#) in our Slack!
Other Advanced Features

Lots of nifty, experimental **feature flags** to play with:

- new native filters
- omnibar
- cross filtering

Adding custom viz plugins:

- Creating a Custom Viz Plugin ([tutorial](#))
- How Nielsen Manages Custom Viz Plugins in Production ([blog](#))
Superset as a Platform

Expose all the legos!

- API for everything
- Highly custom embedded dashboards
- Easily build applications & data products
- Viz plugin packs (bring your own viz library + charts)
How to Contribute

More databases!
  • Recently: CrateDB, Netezza, & Databricks Cloud

More charts!
  • Recently: Force directed graphs, funnel, radar charts, tree chart, etc

Learn more about how to contribute
Resources

- Preset Docs: https://docs.preset.io/docs
- Preset User Corner (YT): https://www.youtube.com/channel/UCzg8opP7sG8n0Mi0e8yegAg
- Apache Superset Community Docs: https://superset.apache.org/docs/intro
- CONTRIBUTING.MD: https://github.com/apache/superset/blob/master/CONTRIBUTING.md

Twttr: @apachesuperset