HAVE YOU EVER WANTED TO

BUILD YOUR OWN DATA SOURCE

```python
# Register my Data Source
spark.dataSource.register(MyNewDataSource)

# Load data from your own data source
spark.read.format('MyNewDataSource').load()

# Write data to your own data source
spark.write.format("MyNewDataSource").save()
```

THE BASICS

Sign Up to the newsletter
@ databricks.news

Find more databricks tips @
dailydatabricks.tips

LOAD A CUSTOM DATA SOURCE AND REGISTER IT IN YOUR SESSION

There are some great examples already available including ones thats support the Faker library in the pyspark-data-sources library

Once you've loaded the community library, you can easily load a custom data source e.g. Google Sheets

These custom data sources can also leverage Sparks parallelism and file handling capabilities

```
#install the community library with pip install pyspark-data-sources
from pyspark_datasources import GoogleSheetsDataSource #Import Google Sheets
spark.dataSource.register(GoogleSheetsDataSource)#Register Google Sheets DataSource


df = spark.read \
    .format("googlesheets") \
    .options(url="<your_google_sheets_document_url>") \
    .load()
```

WHAT DOES THIS MEAN FOR ME?

Sign Up to the newsletter
@ databricks.news

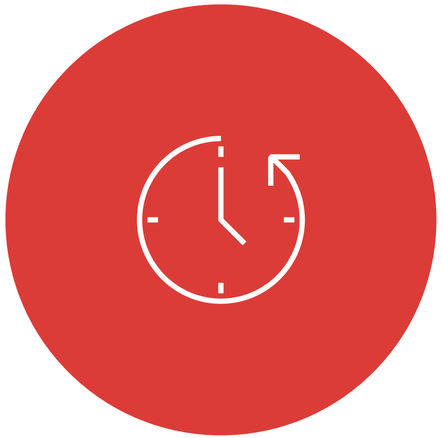
Find more databricks tips @
dailydatabricks.tips

CAPABILITIES

Build Custom Spark connectors in Python to connect to any data source.

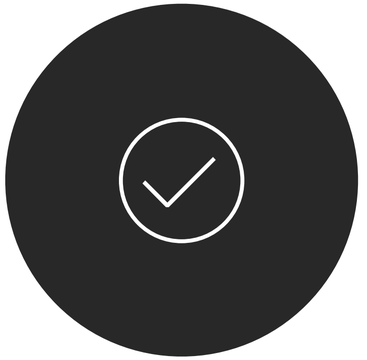
Connect your own APIs, file formats or databases

Leverages Apache Arrow For performance



ANY CADENCE, SAME CODE

Your Custom Datasource Class Works for batch + streaming reads/writes.



UNIFIED EXECUTION

Simple Integration with External Services

Integration with Declarative Pipelines or Pyspark Notebooks

MAKE YOUR OWN DATA SOURCE

```python
class OpenMeteoDataSource(DataSource):
    """
    Open-Meteo weather data source for PySpark.
    """
    @classmethod
    def name(cls) -> str:…

    def schema(self) -> str:…

    def reader(self, schema: StructType) -> DataSourceReader:…
```

LOAD THE DATA SOURCE LIBRARY

Create a Datasource for your custom data source like above.

This Example is the Open-Meteo Weather API that provides access to high-resolution weather data (1-11km resolution) from national weather services

```python
import openmeteo_datasource  # You might want to move your datasource to it's own py file
from pyspark.sql.datasource import DataSource, DataSourceReader #Import to register data sources

spark.dataSource.register(openmeteo_datasource.OpenMeteoDataSource) # Register Data Source

df = spark.read \
        .format("openmeteo") \
        .option("latitude", "52.52") \ # add your own data source specfic options e.g. co-ordinates
        .option("longitude", "13.41") \
        .option("start_date", "2024-01-01") \ # Options that can support features like predicate pushdown or pagination
        .option("end_date", "2024-01-31") \
        .load()
```

FUTURE CAPABILITIES

Sign Up to the newsletter
@ databricks.news

Find more databricks tips @
dailydatabricks.tips

COMMUNITY CONNECTORS

FILTER PUSHDOWN

PERFORMANCE OPTIMISATIONS

COLUMN PRUNING

CUSTOM STATISTICS

OBSERVABILITY IMPROVEMENTS

GOT QUESTIONS?

JOIN ME ON
NOV 20TH

FOR PYSPARK DATASOURCE
WEBINAR

SIGN UP TO MY DATABRICKS NEWSLETTER
FOR DETAILS

Databricks.news