



FOSSGIS 2023

Geodatenverarbeitung mit Workflow-Engines

Pirmin Kalberer
@implgeo
Sourcepole AG, Zürich
www.sourcepole.ch



SOURCEPOLE
Linux & Open Source Solutions



Prozessierung Geodaten

- › **Typische Vorgänge**
 - › Datenaktualisierung / -import
 - › Datenexport / -publikaton
 - › Raster Prozessierung
 - › Tile Cache Update
- › **Herkömmliche Tools**
 - › Skripte (GDAL CLI, Python, ...)
- › **Herkömmliche Orchestrierung**
 - › Cron Jobs
 - › Makefiles
 - › Spezifische Skripte
 - › DB-gesteuerte Prozesse



Problemstellungen

- › **Wartbarkeit (verteilte Skripte, unterschiedliche Backends)**
- › **Übersicht über Ausführung (viele Jobs)**
- › **Verteilte Prozessierung (mehrere Server)**
- › **Komplexe Abhängigkeiten (Ausführung in Container)**
- › **Log Aggregation und Analyse**
- › **Prozess Monitoring mit Alarmierung**



Data science → Tools!

awesome-workflow-engines

A curated list of awesome open source workflow engines

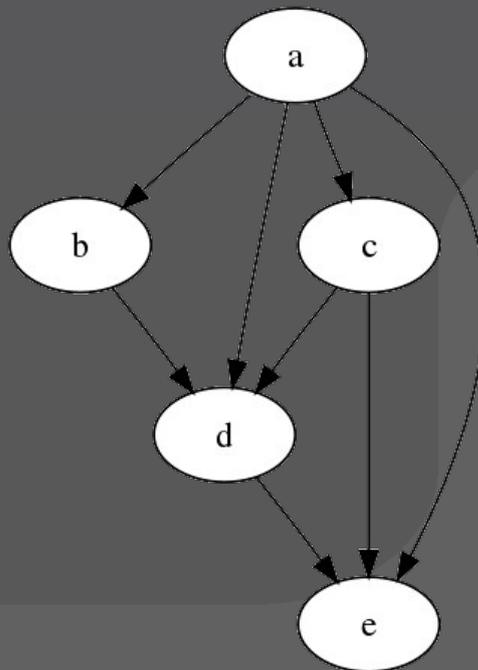
Full fledged product

- [Airflow](#) stars 27k - Python-based platform for running directed acyclic graphs (DAGs) of tasks
- [Argo Workflows](#) stars 12k - Open source container-native workflow engine for getting work done on Kubernetes
- [Azkaban](#) stars 4.1k - Batch workflow job scheduler created at LinkedIn to run Hadoop jobs.
- [Brigade](#) stars 2.4k - Brigade is a tool for running scriptable, automated tasks in the cloud — as part of your Kubernetes cluster.
- [CabloyJS](#) stars 633 - A Node.js full-stack framework with workflow engine, based on koa + egg + vue + framework7.
- [Cadence](#) stars 6.2k - An orchestration engine to execute asynchronous long-running business logic developed by Uber Engineering.
- [Camunda](#) stars 2.8k - BPMN-based workflow engine that can be embedded as java library (e.g. Spring Boot) or used standalone, including a graphical modeler and operations tooling.
- [CGraph](#) stars 328 - A simple-used and cross-platform DAG framework based on C++17 without any 3rd-party.
- [CloudSlang](#) stars 220 - Workflow engine to automate your DevOps use cases.
- [Conductor](#) stars 4.7k - Netflix's Conductor is an orchestration engine that runs in the cloud.
- [Copper](#) stars 220 - A high performance Java workflow engine.
- [Couler](#) stars 754 - Unified interface for constructing and managing workflows on different workflow engines, such as Argo Workflows, Tekton Pipelines, and Apache Airflow.
- [Covalent](#) stars 137 - Workflow orchestration platform for quantum and high performance computing.
- [Cromwell](#) stars 828 - Workflow engine written in Scala and designed for simplicity and scalability. Executes workflows written in [WDL](#) or [CWL](#)



Kategorie Python / DAG Workflows

- › Tasks in Python programmiert
- › Workflows als DAG (Directed Acyclic Graph)





Apache Airflow – DAG (directed acyclic graph)

Airflow DAGs Data Profiling Browse Admin Docs About 2018-09-07 22:15:40 UTC

On **DAG: example_branch_dop_operator_v3** schedule: */1 ****

Graph View **Tree View** Task Duration Task Tries Landing Times Gantt Details Code Refresh Delete

Base date: 2018-09-05 01:04:00 Number of runs: 25 Go

BranchPythonOperator DummyOperator

■ success ■ running ■ failed ■ skipped ■ retry ■ queued □ no status

The DAG graph on the left shows a sequence of tasks: [DAG] -> oper_1 -> condition -> oper_2 -> condition. The task execution timeline on the right shows a series of colored bars representing task status over time. The timeline is divided into two periods: 05:45 and 06 PM. The top row of bars represents the DAG status, which is consistently green (success). The subsequent rows represent the status of oper_1, condition, oper_2, and condition. oper_1 and oper_2 show a mix of green (success) and pink (skipped) bars, while the condition tasks are consistently white (no status).



Navigation: Runs Assets Status ⚠ Workspace ⚙

Job: **diamond** Job in `__repository__diamond@complex_pipeline.py`

Overview **Launchpad** Runs

Select an op... ⌵ 🔍 Highlight...

```
graph TD; download_cereals[download_cereals] --> find_highest_calorie_cereal[find_highest_calorie_cereal]; download_cereals --> find_highest_protein_cereal[find_highest_protein_cereal]; find_highest_calorie_cereal --> display_results[display_results]; find_highest_protein_cereal --> display_results;
```

Type an op subset... (ex: download_cereals+)

Info Types

Description ▾

No description provided

Resources ▾

default

- io_manager**
The default io manager for Jobs. Uses filesystem but switches to in-memory when invoked through `execute_in_process`.
Any
- console**
The default colored console logger.
{
 log_level?: String
 name?: String
}



Dagster: Multi backend processing engine

› Processing libraries / backends

- › Pandas
- › dbt
- › Spark

› Runtime environments

- › Python
- › Celery, Dask
- › Docker, Kubernetes

› API

- › Python
- › GraphQL



Prefect

- Flow Runs
- Flows
- Deployments
- Work Queues
- Blocks
- Notifications

Settings

Flow Runs / xi19-campor-g

Logs Task Runs Sub Flow Runs Parameters

Level: all

Sep 5th, 2022

DEBUG	Loading flow for deployment 'log-simple'...	12:37:52 PM
DEBUG	Starting 'ConcurrentTaskRunner'; submitted tasks will be run concurrently...	12:37:52 PM
DEBUG	Executing flow 'log-flow' for flow run 'xi19-campor-g'...	12:37:52 PM
DEBUG	Beginning execution...	12:37:52 PM
INFO	Created task run 'log_task-99465d2b-0' for task 'log_task'	12:37:52 PM
INFO	Executing 'log_task-99465d2b-0' immediately...	12:37:52 PM
DEBUG	Beginning execution...	12:37:52 PM
INFO	Hello Trillian!	12:37:52 PM
INFO	Prefect Version = 2.3.1 🚀	12:37:52 PM
DEBUG	Hello from another file	12:37:52 PM

Completed

1s

log-flow

log-simple

Flow Run ID
9ff2a05c-2dfd-4b27-a67b-b71fea06d12f

Flow ID
da22db55-0a66-4f2a-ae5f-e0b898529a8f

Work Queue
test

Deployment ID
5364ec86-0d05-4464-9b61-8bc4f991c13

Created
2022/09/05 12:37:45 PM

www.prefect.io



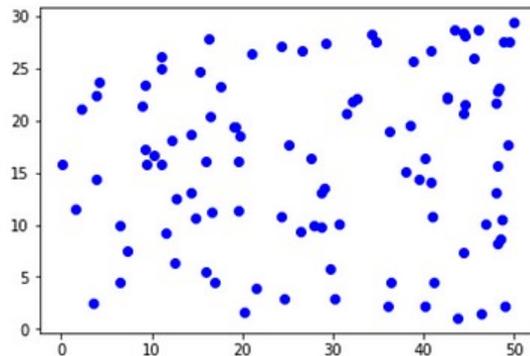
Kategorie Notebooks

- › Kombination von Text und Code
- › Interaktive Erstellung im Browser

Notes for presentation

```
In [6]: import numpy as np
import matplotlib.pyplot as plt
X = np.random.uniform(0,50,100)
Y = np.random.uniform(0,30,100)

plt.plot(X,Y, 'bo')
plt.show()
```





Jupyter Notebook

The screenshot shows a Jupyter Notebook window titled "Lorenz Differential Equations (autosaved)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Help) and a toolbar with icons for file operations and execution. The main content area is titled "Exploring the Lorenz System" and contains the following text:

In this Notebook we explore the [Lorenz system](#) of differential equations:

$$\begin{aligned}\dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy\end{aligned}$$

This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters (σ, β, ρ) are varied, including what are known as *chaotic solutions*. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

The notebook features an interactive interface with the following code cell:

```
In [7]: interact(Lorenz, N=fixed(10), angle=(0.,360.),
                sigma=(0.0,50.0), beta=(0.,5), rho=(0.0,50.0))
```

Below the code cell, there are five interactive sliders for the parameters:

- angle: 308.2
- max_time: 12
- σ : 10
- β : 2.6
- ρ : 28

At the bottom of the notebook, a 3D plot of the Lorenz attractor is displayed, showing the characteristic butterfly shape with multiple overlapping trajectories in various colors.



Kategorie SQL Workflows

- › Datenverarbeitung mit SQL
- › Workflows deklarativ

```
INSERT INTO fpdslv95.lfp2_gemeinde(bfsnr, aname, lfp2_von)
SELECT gem.bfs_nummer, gem.name, f.t_id
FROM fpdslv95.lfp2 f,
      fpds2.gemeindegrenzen gem
WHERE ST_Intersects(ST_Buffer(f.geometrie, 0.05), gem.geom);
```



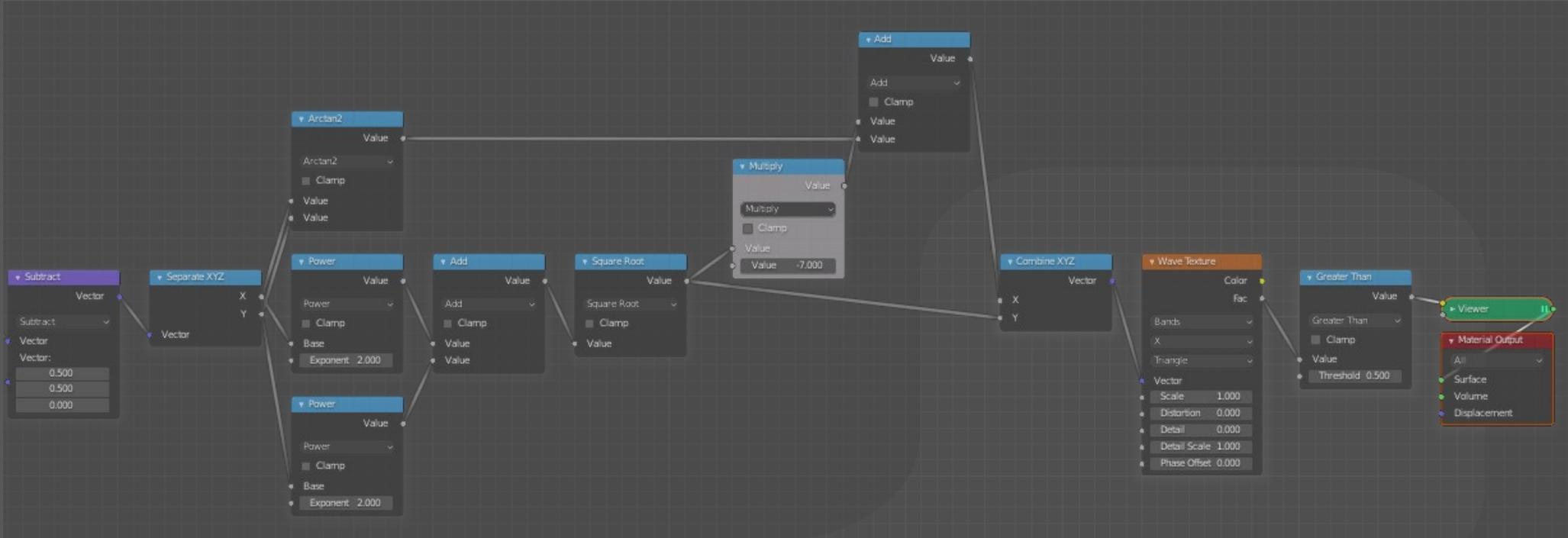
dbt

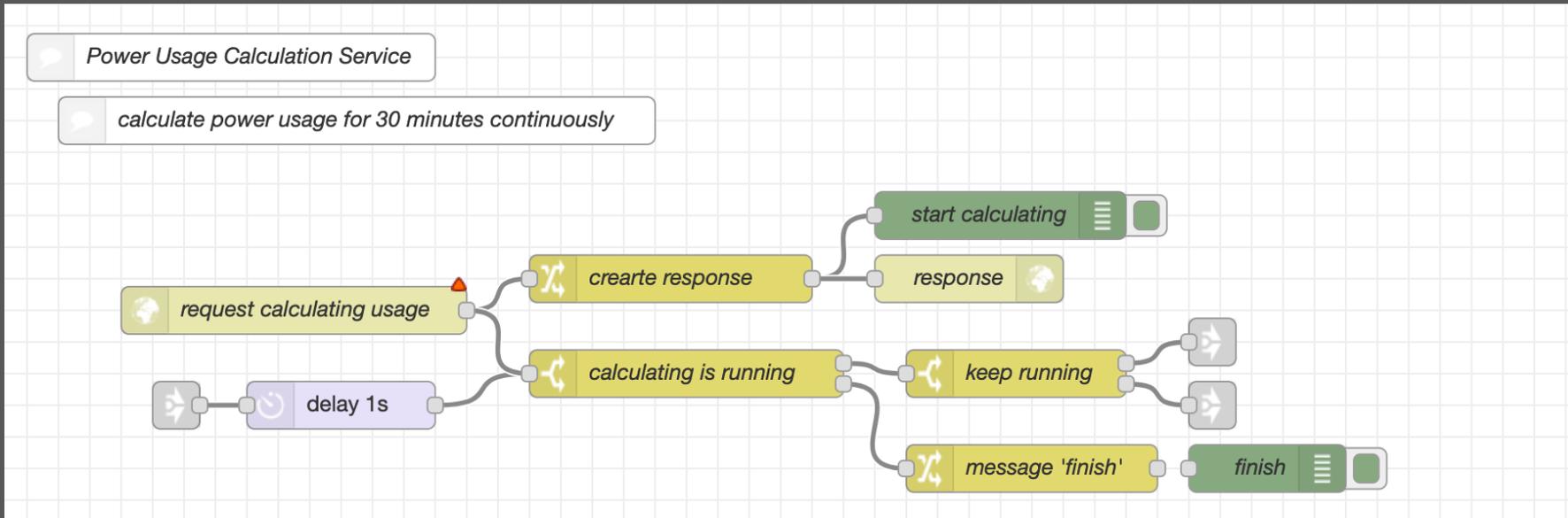
```
jaffle_shop
├── README.md
├── analysis
├── data
│   └── employees.csv
├── dbt_project.yml
├── macros
│   └── cents_to_dollars.sql
├── models
│   ├── intermediate
│   │   └── finance
│   │       ├── _int_finance__models.yml
│   │       └── int_payments_pivoted_to_orders.sql
│   └── marts
│       ├── finance
│       │   ├── _finance__models.yml
│       │   ├── orders.sql
│       │   └── payments.sql
│       └── marketing
│           ├── _marketing__models.yml
│           └── customers.sql
```

www.getdbt.com



Kategorie Low-Code Workflows







Windmill

Export JSON View Graph

u/admin/my_flow_1 Whenever an Hack...

Test flow Save

Settings

Flow Input flow_input

- Watch for new mes... a
- For loop b
- Analyse senti... c
- Send messag... d
- Run one branch e

Default branch

- Send Mess... f

Branch 0

- Send Emal... g

Add branch

Error handler

Settings

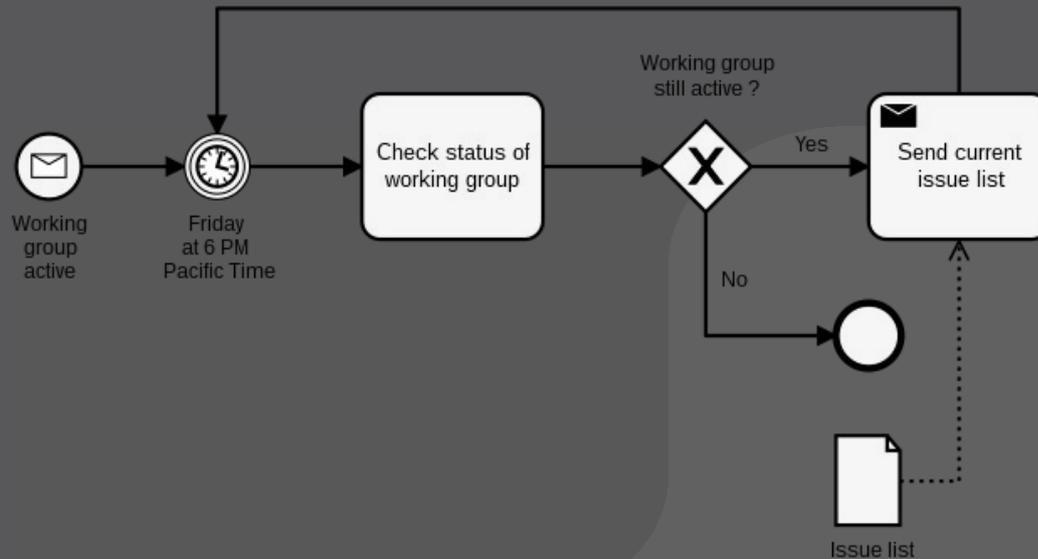
Metadata Schedule Same Worker Graph

```
graph TD; A[Flow start] --> B[Watch for new message with mention]; B --> C[For Loop: result]; C --> D[Analyse sentiment with nltk]; D --> E[Send message to slack]; E --> F[Run one branch given predicate]; F -- Default --> G[Send Message to Channel (slack)]; F -- "If results.d.important" --> H[Send Email (gmail)]; G --> I[Collect iterations' results of For Loop b]; H --> I; I --> J[Flow end];
```



Kategorie Grafische Prozessmodellierung (BPMN)

➤ BPMN: Business Process Model and Notation





Überblick

› Python / DAG

- › Apache Airflow
- › Dagster
- › Prefect

› Notebooks

- › Jupyter Notebooks

› SQL

- › Dbt

› Low-Code

- › Node-RED
- › Windmill

› Grafische Prozessmodellierung (BPMN)

- › Camunda



> <https://ogcapi.org/processes/>



OGC API - PROCESSES

The OGC API - Processes standard supports the wrapping of computational tasks into executable processes that can be offered by a server through a Web API and be invoked by a client application. The standard specifies a processing interface to communicate over a RESTful protocol using JavaScript Object Notation (JSON) encodings. Typically, these processes execute well-defined algorithms that ingest vector and/or coverage data to produce new datasets.

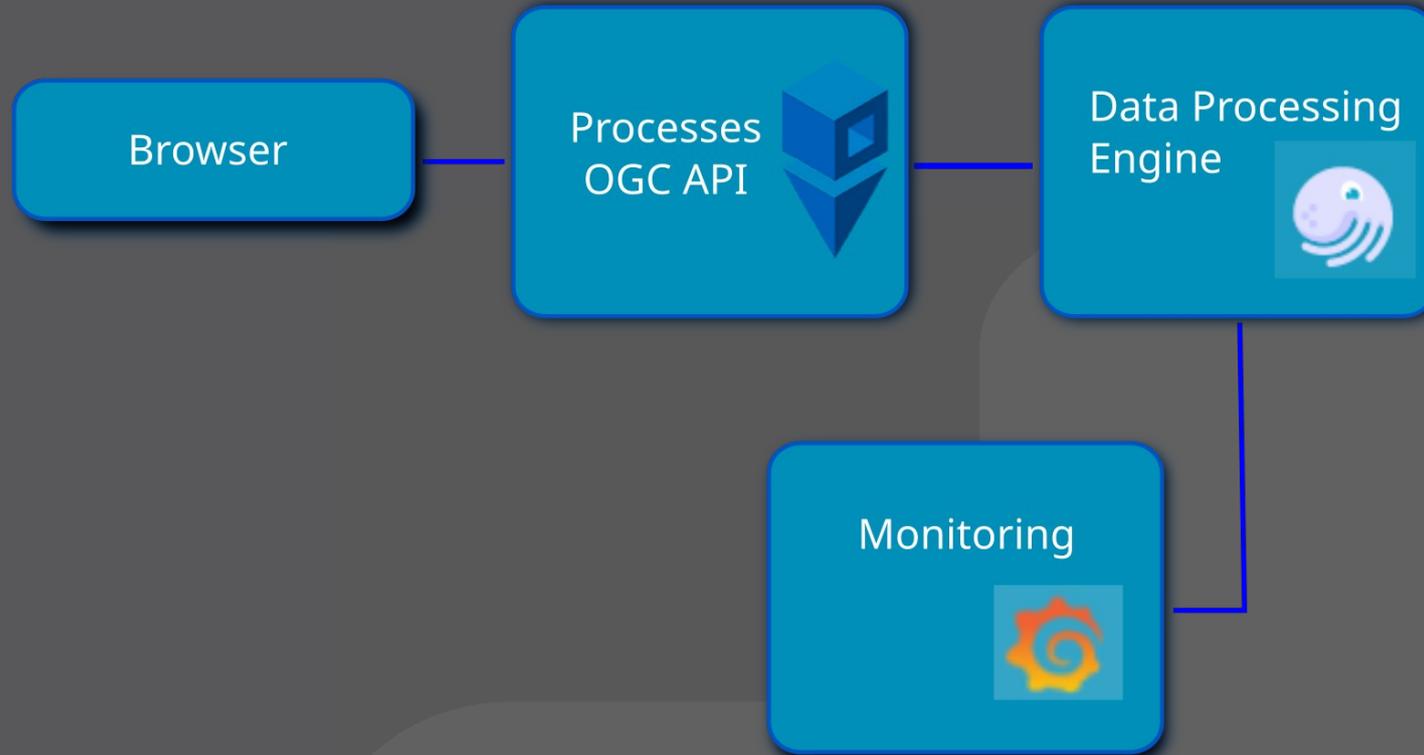
Overview

- Visit the [overview page](#) to learn more about this API.

Developer-friendly OpenAPI definitions



Processing engine + OGC API + Monitoring





Demo Low-Code Windmill

Windmill

- geo
- pirmin
- Favorites
- Home**
- Runs
- Variables
- Resources
- Schedules
- Folders
- Groups
- Audit Logs
- Workspace
- Workers
- Docs
- Feedbacks
- Issues

Home

Create a new:

+ Script </>

+ Flow ≡

+ App 𠄎

Workspace

Hub Scripts

Hub Flows

Hub Apps

All

</> Scripts

≡ Flows

𠄎 Apps

Search Scripts, Flows & Apps

f/workspace

Show archived

- Download and import new municipalities from Boundaries3D dataset**
f/workspace/download_new_boundaries3d
[Edit] [Detail] [Run] [Star]
- ogr2ogr**
f/workspace/ogr2ogr
[Copy] [Edit] [Detail] [Run] [Star]
- Check for new Swissboundaries3d dataset**
f/workspace/check_swissboundaries3d_update
[Trigger] [Copy] [Edit] [Detail] [Run] [Star]
- Unzip File**
f/workspace/unzip_file
[Copy] [Edit] [Detail] [Run] [Star]
- Send message to Slack API - error handler**
f/workspace/send_slack_error_message
[Failure] [Copy] [Edit] [Detail] [Run] [Star]
- Send message to Slack API**
f/workspace/send_chat_message
[Copy] [Edit] [Detail] [Run] [Star]
- Fetch JSON and apply optional filter**
f/workspace/fetch_json
[Copy] [Edit] [Detail] [Run] [Star]



Demo Low-Code Windmill

- Windmill
- geo
- pirmin
- Favorites
- Home
- Runs
- Variables
- Resources
- Schedules
- Folders
- Groups
- Audit Logs
- Workspace
- Workers
- Docs
- Feedbacks
- Issues

Send message to Slack API

f/workspace/send_chat_message

Edited 17:56 10/3 by pirmin 1e2e14

Edit

View

Run Ctrl+Enter

channel* (string)

workflows



username* (string)

windmill



text* (string)



api* (resource-c_slackapi)

slackapi resource



f/workspace/mattermost_sp

Schedule to run later



make run invisible to others

Run

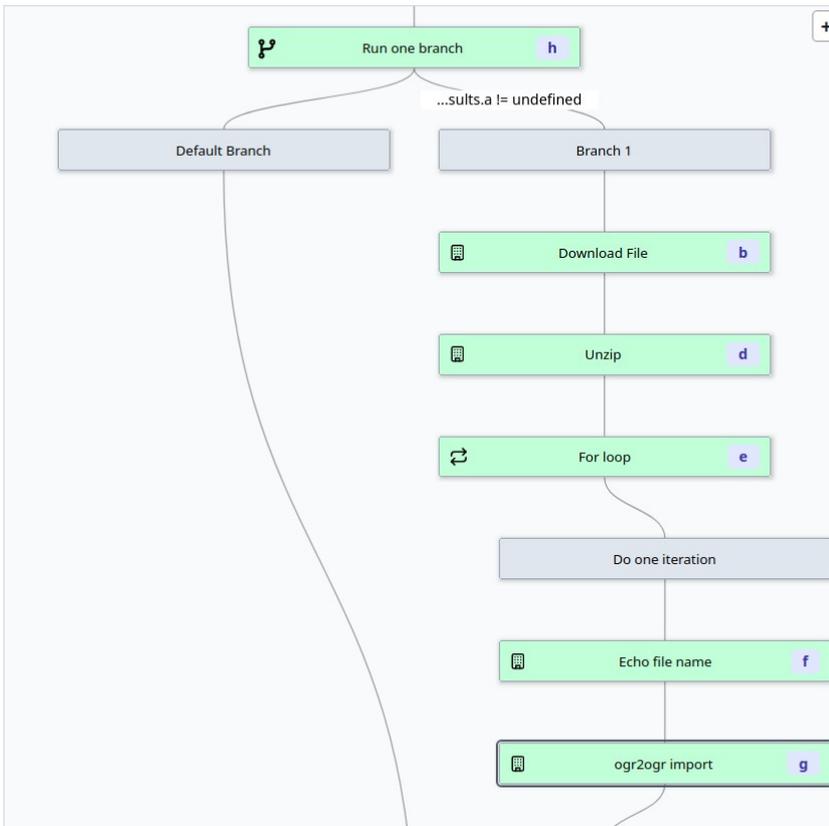
Run it from the CLI



Demo Low-Code Windmill

- Windmill
- geo
- pirmin
- Favorites
- Home
- Runs
- Variables
- Resources
- Schedules
- Folders
- Groups
- Audit Logs
- Workspace
- Workers
- Docs
- Feedbacks
- Issues

Graph Details



+ -

Success Job Id ...d84764e02e

Argument	Value
options	"-overwrite -nlt PROMOTE_TO_MULT
dst_datasource	"PG:service=geodb"
src_datasource	"/shared /swissBOUNDARIES3D_1_4_TLM_LAN

Result
"ogr2ogr -overwrite -nlt PROMOTE_TO_MULT I F

Expand Auto scroll

job 0186daf9-9277-897d-87b1-b9d84764e02e on
worker dt-worker-e20a3-81SLr



Demo Low-Code Windmill

Export JSON ↶ ↷ f/workspace/download_new_boundaries3d Download and imp... ▶ Test up to b ▶ Test flow Save ▼

Settings ./shared

\$ All Static Inputs

```
graph TD; A[Check for new dataset a] --> B[Run one branch h]; B --> C[Default Branch]; B --> D[Branch 1]; D --> E[Download File b]; E --> F[Unzip d]; F --> G[For loop e]; G --> H[Do one iteration]; H --> I[Error handler f];
```

Code editor:

```
#import wmill
import requests

HTTP_REQUEST_TIMEOUT = 3

def main(
  url: str,
  dest: str,
  params: dict = {},
  headers: dict = {},

```

Step Input

url* (string)

dest* (string)

params (object)

headers (object)

Flow Input

force : true

Previous Result

a : -

id : "swissboundaries3d_2023-01"

datetime : "2023-01-01T00:00:00Z"

shapefile_url : "https://data.geo.admin.ch/swisstopo.swissboundaries3d/swissboundaries3d_2023-01/swissboundaries3d_2023-01_2056_5728.shp.zip"

All Results

{...} 1 key

Variables

Error handler Echo error Echo file name



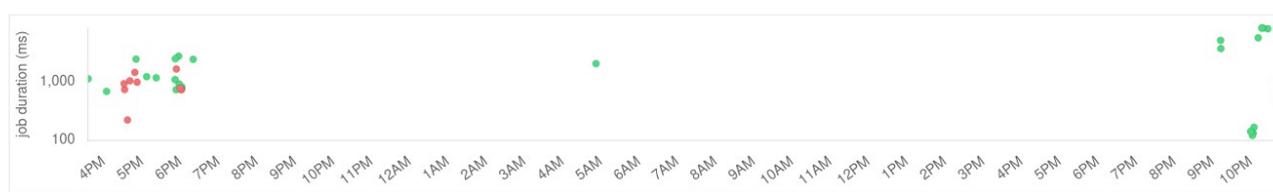
Demo Low-Code Windmill

- Windmill
- geo
- pirmin
- Favorites
- Home
- Runs**
- Variables
- Resources
- Schedules
- Folders
- Groups
- Audit Logs
- Workspace
- Workers
- Docs
- Feedbacks
- Issues

Runs

Successful and error jobs | Ignore skipped flow jobs

All | **Runs** | Previews | Dependencies



min datetime: 2023-03-10T13:09:48.241Z | max datetime: 2023-03-12T00:49:58.860Z

Search by path | Filter by args | Filter by result

✓	f/workspace/fetch_json 96cc62cb	script	By pirmin	Ended 22:55 11/3
			Ran in 0.469s	
✓	f/workspace/fetch_json 49f6ada4	script	By pirmin	Ended 22:52 11/3
			Ran in 1.01s	
✓	f/workspace/download_new_boundaries3d 2ecd3bcc	flow	By pirmin	Ended 22:34 11/3
			Ran in 8.578s	
✓	f/workspace/download_new_boundaries3d 9fefae99	flow	By pirmin	Ended 22:26 11/3
			Ran in 8.837s	



Danke!



Pirmin Kalberer
@implgeo@mapstodon.space