

ESIP Winter Meeting: Dynamic Tiling for everyone

January 28, 2021



development SEED

Vincent Sarago



vincentsarago



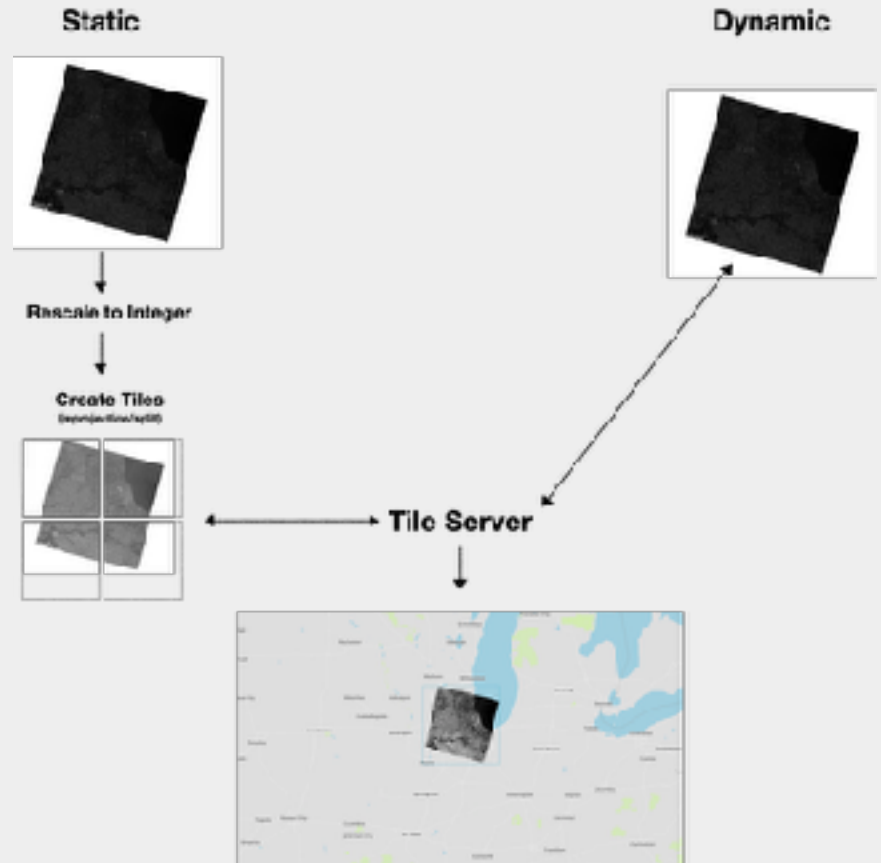
@_VincentS_ / @cogeotiff / @remotepixel

- Former geologist
- Geospatial developer
- Self taught Python dev
- Full stack



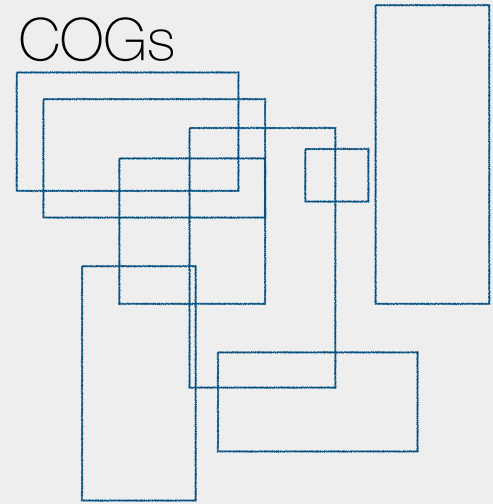
Dynamic What ?

https://developmentseed.org/titiler/concepts/dynamic_tiling/



Dynamic What ?

<https://github.com/developmentseed/mosaicjson-spec/>



TiTiler

 [developmentseed/titiler](https://github.com/developmentseed/titiler)

- [Cloud Optimized GeoTIFF support](#)
- [SpatioTemporal Asset Catalog support](#)
- Mosaic support (via MosaicJSON)
- OGC WMTS support
- Multiple TileMatrixSets via **morecantile**. By default, output map tiles are in the standard Web Mercator projection used by most mapping libraries, but **support for alternative projections** is available.
- AWS Lambda / ECS deployment options



A lightweight Cloud Optimized GeoTIFF dynamic tile server.

  [pysang](#)  [covcov](#) [91%](#)  [pypi package](#) [v0.1.0a3](#)  [license MIT](#)

Documentation: developmentseed.com/titiler/

Source Code: [developmentseed/titiler](https://github.com/developmentseed/titiler)

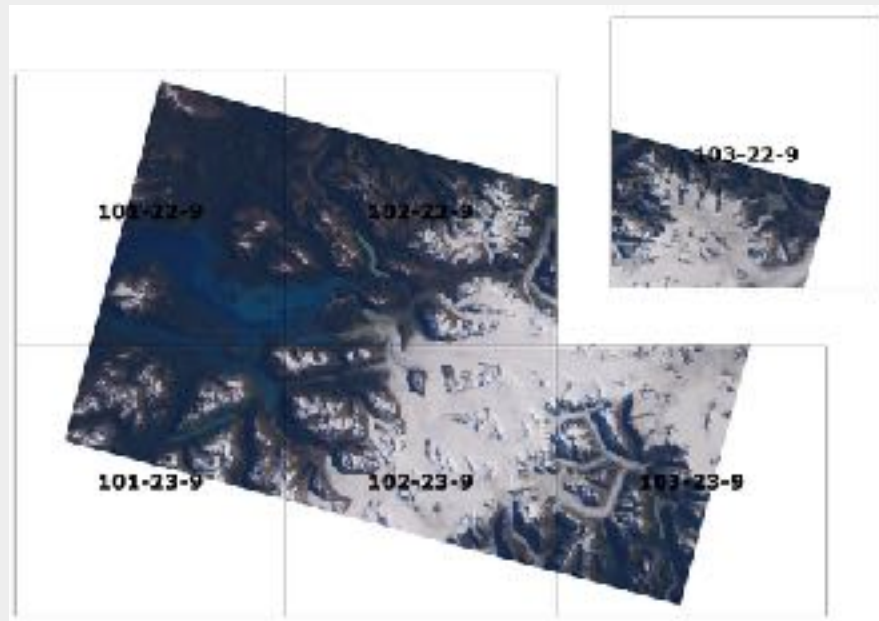


developmentSEED

A wrapper around rio-tiler

 cogeotiff/rio-tiler

ref: <https://github.com/cogeotiff/rio-tiler>



development SEED

Built with FastAPI



tiangolo/fastapi

- **Fast:** Very high performance, on par with **NodeJS** and **Go** (thanks to Starlette and Pydantic). [One of the fastest Python frameworks available](#).
- **Fast to code:** Increase the speed to develop features by about 200% to 300%. *
- **Fewer bugs:** Reduce about 40% of human (developer) induced errors. *
- **Intuitive:** Great editor support. Completion everywhere. Less time debugging.
- **Easy:** Designed to be easy to use and learn. Less time reading docs.
- **Short:** Minimize code duplication. Multiple features from each parameter declaration. Fewer bugs.
- **Robust:** Get production-ready code. With automatic interactive documentation.
- **Standards-based:** Based on (and fully compatible with) the open standards for APIs: [OpenAPI](#) (previously known as Swagger) and [JSON Schema](#).

ref: <https://github.com/tiangolo/fastapi>



developmentSEED

Install and Use

```
$ pip install titiler uvicorn
```

```
$ uvicorn titiler.main:app --reload
```


Demo

COG

<https://www.digitalglobe.com/ecosystem/open-data>

Example:

<https://opendata.digitalglobe.com/events/beirut-explosion/post-event/>

2020-08-05/102001009BCC9D00/102001009BC
C9D00.tif

<https://opendata.digitalglobe.com/events/hurricane-laura/post-event/>

2020-08-27/10300100ADB4FD00/10300100ADB
4FD00.tif

<https://opendata.digitalglobe.com/events/mauritius-oil-spill/post-event/>

2020-08-12/105001001F1B5B00/105001001F1B
5B00.tif



Demo

STAC

<https://stacindex.org>

Example:

[https://canada-spot-ortho.s3.amazonaws.com/
canada_spot_orthoimages/
canada_spot5_orthoimages/S5_2007/
S5_11055_6057_20070622/
S5_11055_6057_20070622.json](https://canada-spot-ortho.s3.amazonaws.com/canada_spot_orthoimages/canada_spot5_orthoimages/S5_2007/S5_11055_6057_20070622/S5_11055_6057_20070622.json)

Customization

```
from titiler.endpoints.factory import TilerFactory
from titiler.errors import DEFAULT_STATUS_CODES, add_exception_handlers

from fastapi import FastAPI

from starlette.requests import Request
from starlette.responses import HTMLResponse

app = FastAPI(
    title="My super app",
    openapi_url="/api/openapi.json",
    description="It's something great",
)

cog = TilerFactory(router_prefix='cog')
app.include_router(cog.router, prefix='/cog', tags=['Cloud Optimized GeoTIFF'])
add_exception_handlers(app, DEFAULT_STATUS_CODES)
```



Thanks

Vincent Sarago - @_VincentS_ / @cogeotiff



development SEED