Flatmap Server

David Brooks

Jun 11, 2021

CONTENTS:

CHAPTER

ONE

INSTALLATION AND RUNNING

1.1 Flatmap Server

An anatomical flatmap server to provide Mapbox compatible tilesets generated by flatmap-maker. It is intended for use with a browser-based flatmap-viewer application. The server is written in Python using the Flask web framework.

1.1.1 Prerequisites

- Python 3.8
- pipenv
- Under Ubuntu, `sudo apt-get install libgl-mesa-glx`
- If the server will also generate maps, install Tippecanoe.

1.1.2 Installation

1) Download the latest release and extract it to a suitable directory.
2) Change to this directory and run `pipenv install`.

1.1.3 Running

```
$ pipenv run gunicorn src.server:app
```

- By default, maps are stored in `./flatmaps`. This can be overridden by setting the `FLATMAP_ROOT` environment variable to a directory path.
- The server listens at `http://localhost:8000`. Change this via Gunicorn.

1.1.4 Web API

A description of the server's web API is available here.

1.2 Optional map viewer

As an option, the server provides a simple flatmap viewer application. Prerequisites are `node` and `npm`. To install the viewer give the following commands from the top-level server directory:

```
$ git clone https://github.com/ABI-Software/flatmap-server-viewer viewer
$ cd viewer
$ npm install
$ npm run build
$ cd ..
```

1.2.1 Running

To run the server with the integrated viewer:

```
$ pipenv run gunicorn 'src.server:viewer()'
```

and open http://localhost:8000/viewer in a browser.

1.3 Map generation

The flatmap server can also generate maps. To generate a map, `POST` a request to the `/make/map` end-point specifying a directory containing a map's `manifest.json`. The server will respond with the id of the maker process. The `/make/status/PROCESS_ID` end-point allows the process's status to be queried and `/make/log/PROCESS_ID` will return a log of a running process.

1.3.1 Authentication

Bearer tokens (RFC 6750) may be used to control access to map generation services. To enable this, set the `BEARER_TOKENS` environment variable to a space separated list of valid tokens, before starting the server. e.g:

```
$ export BEARER_TOKENS="token1 token2"
$ pipenv run gunicorn src.server:app
```

1.3.2 Examples

A local map source:

```
$ curl -H "Content-Type: application/json" -X POST \
       -d '{"source":"/Users/dave/build/Flatmaps/new-maker/tests/gradients/manifest.
↪json"}'  \
       http://localhost:8000/make/map

{"maker":18259,"source":"/Users/dave//build/Flatmaps/new-maker/tests/gradients",
↪"status":"started"}
```

```
$ curl http://localhost:8000/make/status/18259

{"maker":18259,"status":"terminated"}
```

```
$ curl http://localhost:8000/make/log/18259

2021-01-22 09:14:05,925 Mapmaker 1.0.0b1
2021-01-22 09:14:05,928 Adding details...
2021-01-22 09:14:05,928 Outputting GeoJson features...
2021-01-22 09:14:05,928 Layer:gradients
2021-01-22 09:14:05,929 Running tippecanoe...
2021-01-22 09:14:06,020 Generating background tiles (may take a while...)
2021-01-22 09:14:06,021 Tiling gradients_image...
2021-01-22 09:14:06,040 Tiling zoom level 10 for gradients_image
2021-01-22 09:14:08,811 Tiling zoom level 9 for gradients_image
2021-01-22 09:14:08,935 Tiling zoom level 8 for gradients_image
2021-01-22 09:14:08,976 Tiling zoom level 7 for gradients_image
2021-01-22 09:14:08,994 Tiling zoom level 6 for gradients_image
2021-01-22 09:14:09,005 Tiling zoom level 5 for gradients_image
2021-01-22 09:14:09,015 Tiling zoom level 4 for gradients_image
2021-01-22 09:14:09,024 Tiling zoom level 3 for gradients_image
2021-01-22 09:14:09,034 Tiling zoom level 2 for gradients_image
2021-01-22 09:14:09,063 Creating index and style files...
2021-01-22 09:14:09,065 Generated map: gradients
```

A particular version of the rat flatmap held in a publicly accessible PMR workspace, with a `BEARER_TOKEN` used to authenticate the user to the map server:

```
$ curl -H "Content-Type: application/json"  \
       -H "Authorization: Bearer BEARER_TOKEN"  \
       -X POST \
       -d '{"source":"https://models.physiomeproject.org/workspace/693/rawfile/
↪aa83dc1b19c03101d6a5306c77d144823fd59ea5/vagus_test.manifest.json"}'  \
       http://localhost:8000/make/map

{"map":"83f6c97d571b67fb4c273e20287b53b4f0a1f70780d3d6a2a282e66cef5f9473","process
↪":57906,"source":"https://models.physiomeproject.org/workspace/693/rawfile/
↪aa83dc1b19c03101d6a5306c77d144823fd59ea5/vagus_test.manifest.json","status":"started
↪"}
```

```
$  curl  -H "Authorization: Bearer BEARER_TOKEN" http://localhost:8000/make/status/
↪57906

{"process":57906,"status":"running"}
```

```
$ curl -H "Authorization: Bearer BEARER_TOKEN" http://localhost:8000/make/log/57906

2021-06-11 13:46:17,386 INFO: Mapmaker 1.2.0b3
2021-06-11 13:46:17,903 INFO: Making map:␣
↪83f6c97d571b67fb4c273e20287b53b4f0a1f70780d3d6a2a282e66cef5f9473
2021-06-11 13:46:20,148 WARNING: Unknown anatomical entity: SAO:1770195789
2021-06-11 13:46:20,724 INFO: Adding details...
2021-06-11 13:46:20,728 INFO: Routing paths...
2021-06-11 13:46:20,728 INFO: Outputting GeoJson features...
2021-06-11 13:46:20,728 INFO: Layer: vagus_test
2021-06-11 13:46:20,800 INFO: Layer: vagus_test_routes
```

(continues on next page)

(continued from previous page)

```
2021-06-11 13:46:20,800 INFO: Running tippecanoe...
2021-06-11 13:46:20,996 INFO: Generating background tiles (may take a while...)
2021-06-11 13:46:20,998 INFO: Tiling vagus_test_image...
2021-06-11 13:46:21,019 INFO: Tiling zoom level 10 for vagus_test_image
```

```
$  curl  -H "Authorization: Bearer BEARER_TOKEN" http://localhost:8000/make/status/
↪57906

{"process":57906,"status":"terminated"}
```

```
$ curl -H "Authorization: Bearer BEARER_TOKEN" http://localhost:8000/make/log/57906

2021-06-11 13:46:17,386 INFO: Mapmaker 1.2.0b3
2021-06-11 13:46:17,903 INFO: Making map:␣
↪83f6c97d571b67fb4c273e20287b53b4f0a1f70780d3d6a2a282e66cef5f9473
2021-06-11 13:46:20,148 WARNING: Unknown anatomical entity: SAO:1770195789
2021-06-11 13:46:20,724 INFO: Adding details...
2021-06-11 13:46:20,728 INFO: Routing paths...
2021-06-11 13:46:20,728 INFO: Outputting GeoJson features...
2021-06-11 13:46:20,728 INFO: Layer: vagus_test
2021-06-11 13:46:20,800 INFO: Layer: vagus_test_routes
2021-06-11 13:46:20,800 INFO: Running tippecanoe...
2021-06-11 13:46:20,996 INFO: Generating background tiles (may take a while...)
2021-06-11 13:46:20,998 INFO: Tiling vagus_test_image...
2021-06-11 13:46:21,019 INFO: Tiling zoom level 10 for vagus_test_image
2021-06-11 13:46:23,802 INFO: Tiling zoom level 9 for vagus_test_image
2021-06-11 13:46:23,969 INFO: Tiling zoom level 8 for vagus_test_image
2021-06-11 13:46:24,034 INFO: Tiling zoom level 7 for vagus_test_image
2021-06-11 13:46:24,062 INFO: Tiling zoom level 6 for vagus_test_image
2021-06-11 13:46:24,079 INFO: Tiling zoom level 5 for vagus_test_image
2021-06-11 13:46:24,097 INFO: Tiling zoom level 4 for vagus_test_image
2021-06-11 13:46:24,116 INFO: Tiling zoom level 3 for vagus_test_image
2021-06-11 13:46:24,136 INFO: Tiling zoom level 2 for vagus_test_image
2021-06-11 13:46:24,188 INFO: Creating index and style files...
2021-06-11 13:46:24,195 INFO: Generated map:␣
↪83f6c97d571b67fb4c273e20287b53b4f0a1f70780d3d6a2a282e66cef5f9473
```

CHAPTER

TWO

WEB API

2.1 Map generation

POST /make/map
Generate a flatmap.

Request JSON Object

- **source** (*string*) – the map's manifest

Response JSON Object

- **process** (*int*) – the id of the map generation process
- **map** (*string*) – the unique identifier for the map
- **source** (*string*) – the map's manifest
- **status** (*string*) – the status of the map generation process

GET /make/status/(**int:** *process_id*)
Get the status of a map generation process.

Parameters

- **process_id** (*int*) – The id of a maker process

Response JSON Object

- **maker** (*int*) – the id of the map generation process
- **status** (*string*) – the status of the map generation process

GET /make/log/(**int:** *process_id*)
Return the log file of a map generation process.

Parameters

- **process_id** (*int*) – The id of a maker process

2.2 Map server

GET /
Get a list of available flatmaps.

Response JSON Array of Objects

- **id** (*string*) – the flatmap's unique identifier on the server
- **source** (*string*) – the map's source URL
- **created** (*string*) – when the map was generated
- **describes** (*string*) – the map's description

GET /flatmap/(**string:** *map_id*)**/**
Return a representation of a flatmap.

Parameters

- **map_id** (*string*) – The flatmap identifier

Request Headers

- Accept – Determines the response content

If an SVG representation of the map exists and the *Accept* header doesn't specify a JSON response then the SVG is returned, otherwise the flatmap's `index.json` is returned.

2.3 Map viewing

GET /viewer/(**path:** *filename*)

GET /viewer/
The flatmap viewer application.

Parameters

- **filename** (*path*) – The viewer file to get, defaults to `index.html`

HTTP ROUTING TABLE

/

```
GET /, 6
```

/flatmap

```
GET /flatmap/(string:map_id)/, 6
```

/make

```
GET /make/log/(int:process_id), 5
GET /make/status/(int:process_id), 5
POST /make/map, 5
```

/viewer

```
GET /viewer/, 6
GET /viewer/(path:filename), 6
```