

Create/distrubute vector tile

Taro Matsuzawa

Georepublic

https://github.com/smellman/jica_2019

Self-Introduction

- Senior Developer of Georepublic
 - Programming: Python, Ruby, JavaScript, Java, and others
 - Computer skill: UNIX and Linux, Networking(L1 to L7), Virtual Machine (VirtualBox, KVM, Docker)
 - GIS skill: Data processing, Tile
- Computer community:
 - OSGeo.JP
 - OpenStreetMap Foundation Japan
 - Japan Unix Society
 - Mozilla community in Japan
- Contact: taro@georepublic.co.jp or [@smellman](https://twitter.com/@smellman)

Topics:

- Short introduction of `unvt`
- Introduction of the tile technology
- Introduction of software and data
- Vector Tiles
 - Short demonstration with `inazo`
 - Create Vector Tiles from vector files
 - Styling Vector Tiles
 - Hosting

unvt

What is `unvt`

- `unvt` means United Nations Vector tile Toolkit.
- Mr. Fujimura started this project when he worked in UN.
- Toolkit means `collection of softwares`.
 - All software include in one Docker image.

Docker

- Docker is container technology.
 - Docker image contains basic OS and software.
- Docker runs in too many platform.
 - Desktop OS: Linux, Windows, macOS...
 - Cloud platform: AWS, Google, Azure(Microsoft)
 - Others: Raspberry Pi 3.
 - This presentation targets Raspberry Pi 3.

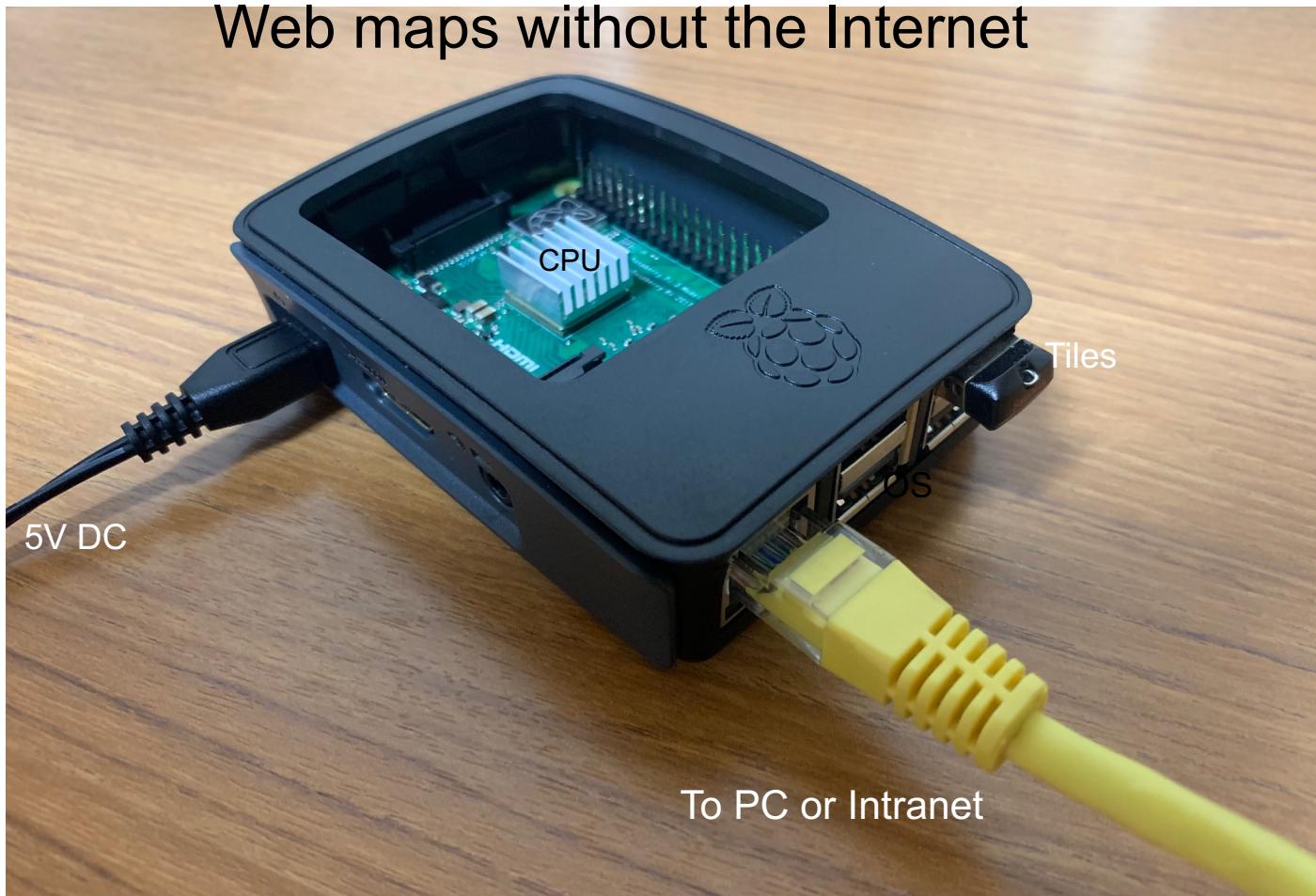
Raspberry Pi

- A small single-board computer.
 - Developed in the UK by the Raspberry Pi Foundation.
 - Teaching of basic computer science in schools and in developing countries.
- You can build development environment with low cost.
 - Raspberry Pi 3 = \$35
 - microSD Card (128GB) = \$20
 - LAN Cable = \$5

Docker and Raspberry Pi

- Raspberry Pi supports Debian Linux based `Raspbian OS`.
 - Raspbian OS supports Docker.
- We targets converting small data to vector tile.
 - If you build vector tile from big data(such as OpenStreetMap) or build raster tile, you may need some Desktop or laptop because it requires memory.
 - We can't test Raspberry Pi 4 with 4GB memory because currently we can't buy it in Japan.

Concept image



Docker images

- `unvt/rasv`
 - Basic image includes some software.
- `unvt/ango`
 - Extra image for FOSS4G 2019 Niigata Hands on.
- `unvt/inazo`
 - Extra image for JICA 2019 Seminar.
 - Includes hosting program and Euro data from Global map project.
 - This presentation use this image.

Introduction of the tile technology

Tile technology

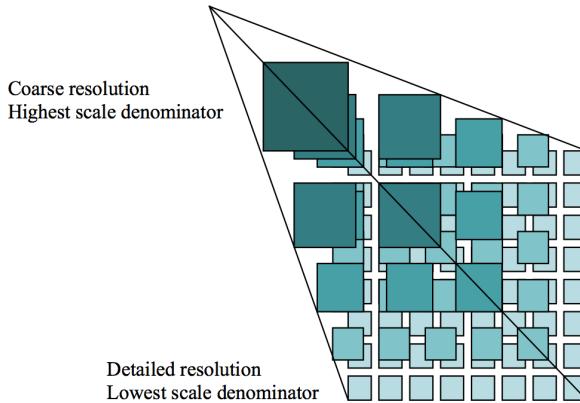
- Provide map images or data over the internet.
 - Map images are separated as tile.
 - Zoom Level 0 = World
 - Each zoom level doubles in the dimensions.
 - Too many tiles use “Web Mercator projection”.
- Structure of tile is useful to web.
 - Scroll, zoom up/down with AJAX request.
- Tile become known for Google Maps.
 - Tile has existed from the late 1990s.



<https://a.tile.openstreetmap.org/0/0/0.png>

Zoom

- Zoom level 0 : 1 file
- Zoom level 1 : $2 \times 2 = 4$ files
- Zoom level 2 : $4 \times 4 = 16$ files
- ...
- Zoom level 18
 $262,144 \times 262,144$
 $= 68,719,476,736$ files

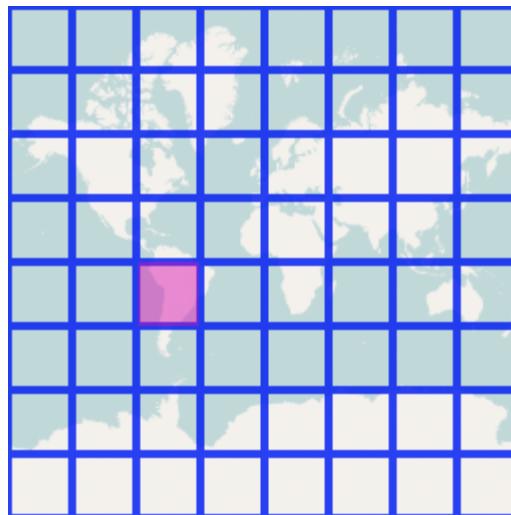


Request

- Many service use REST API (GET).
 - <https://.../Z/X/Y.Format>
 - Z = Zoom Level
 - X = X coordinate
 - Y = Y coordinate
 - Format
 - Raster image format (png, jpg, webp)
 - Vector data format (pbf, mvt)

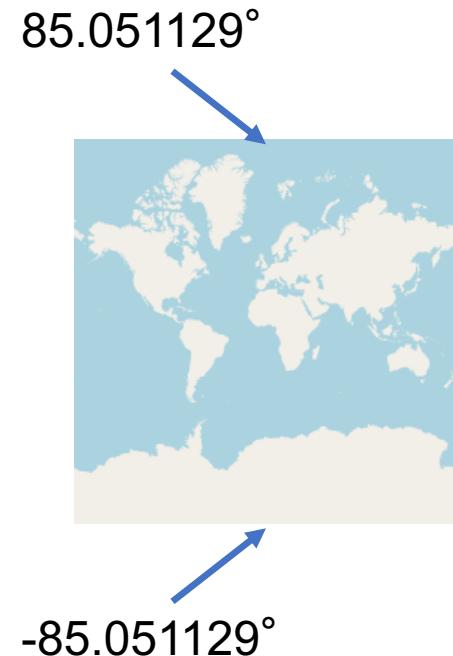
Request example

- <http://a.tile.openstreetmap.org/3/2/4.png>
 - Zoom=3, X=2, Y=4, format=PNG
 - X and Y coordinates start with 0.



Web Mercator

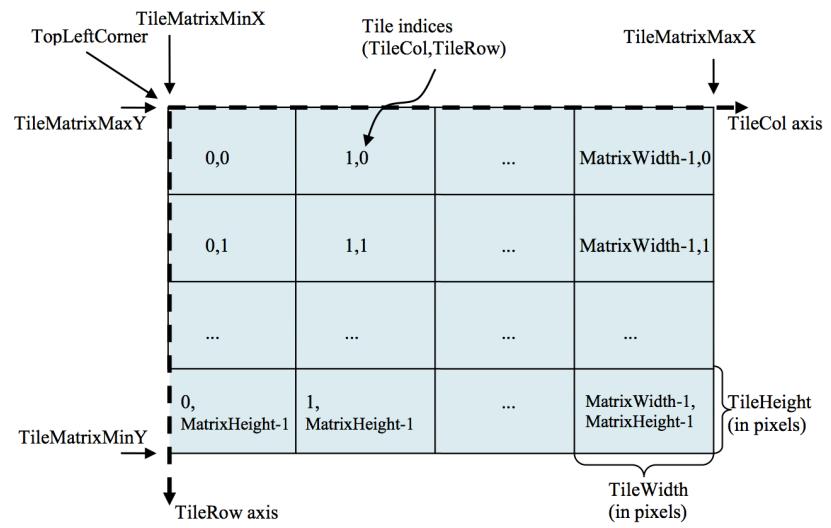
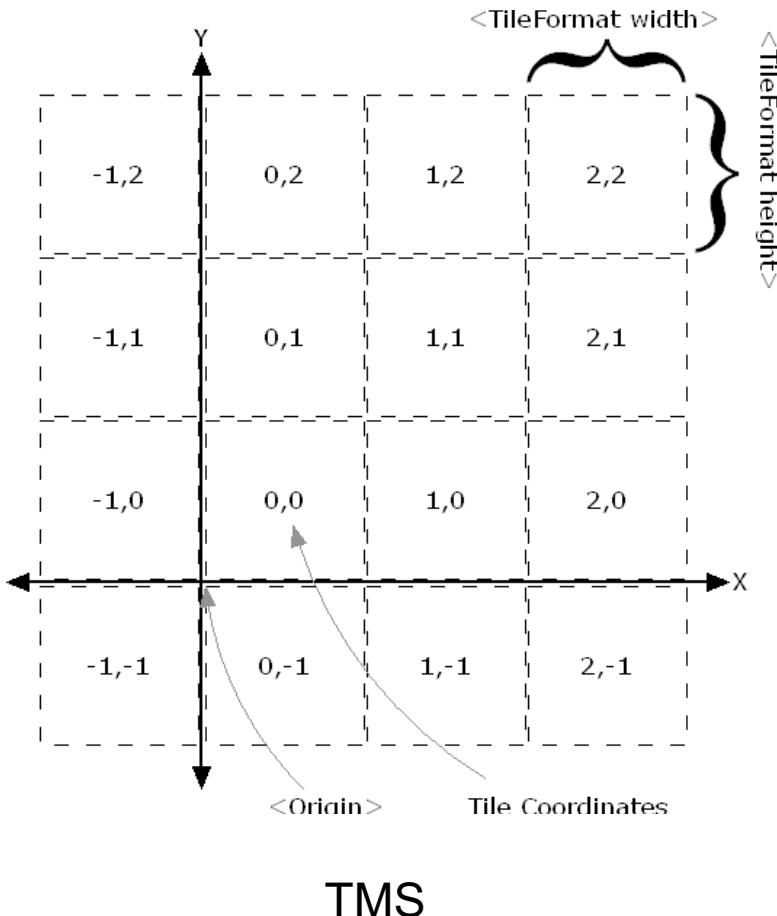
- Cut off coverage at 85.051129° north and south.
- EPSG:3857
 - Famous as “OpenLayers:900913” (Google)



Specification

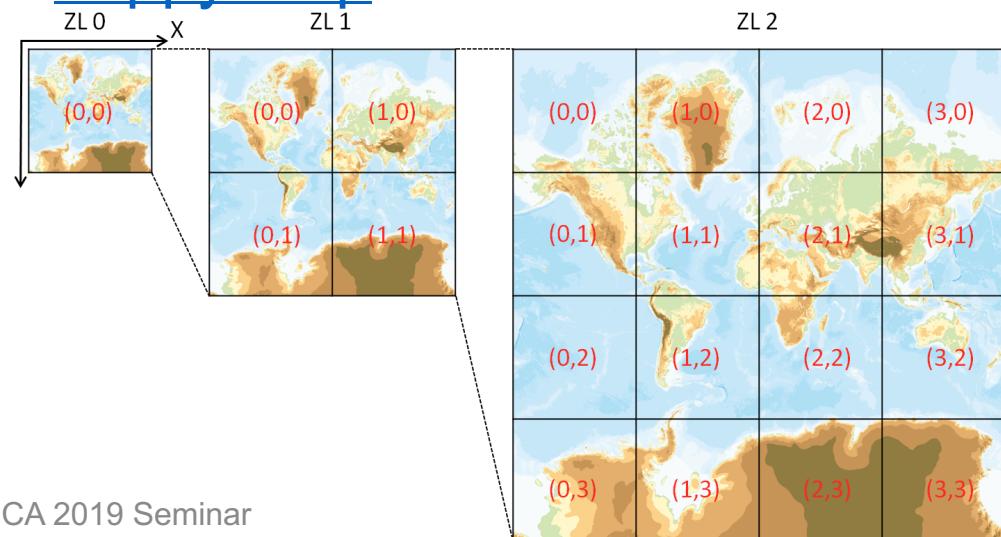
- Two tile service specifications
 - Tile Map Service (TMS)
 - Web Map Tile Service (WMTS)
- TMS is simple than WMTS.
- TMS's X Y coordinate is started from bottom left.
 - Same as Cartesian coordinate system.
- WMTS's X Y coordinate is started from top left.
 - Same as Coordinate system for 2D computer graphics.

Difference between TMS and WMTS



The Y coordinate flipped

- OpenStreetMap use TMS like protocol but The Y coordinates are numbered from top left.
 - If take no thought of meta data, Y coordinate flipped TMS equals WMTS.
 - GSI Map use this rule too.
 - OpenStreetMap call “[Slippy Map](#)”
 - We call **zxy tile**
 - $\{z\}/\{x\}/\{y\}.png$
 - Also call **xyz tile**

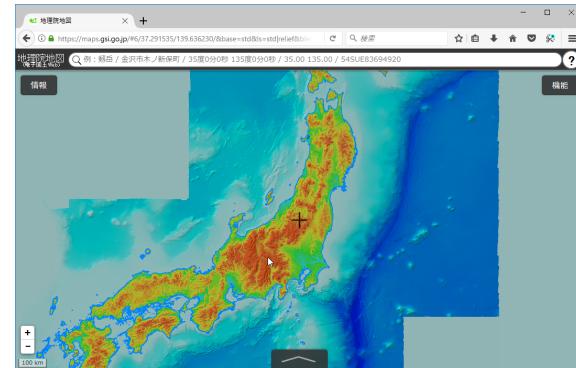
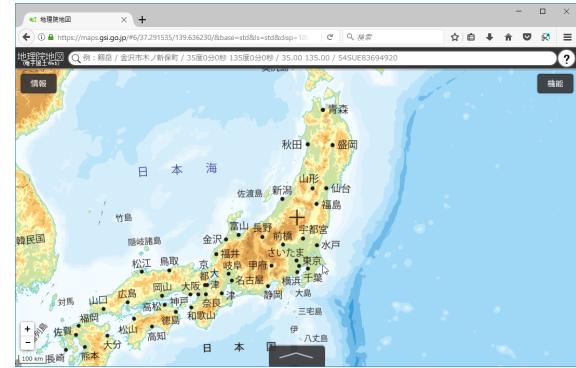


ZXY Tile

- De facto standard of tile.
 - Web Mercator
 - Y coordinate flipped TMS
 - Provide REST API
 - $\{z\}/\{x\}/\{y\}.\{\text{format}\}$
 - Anyone provide “Specification”
- Too many libraries supports ZXY Tile.

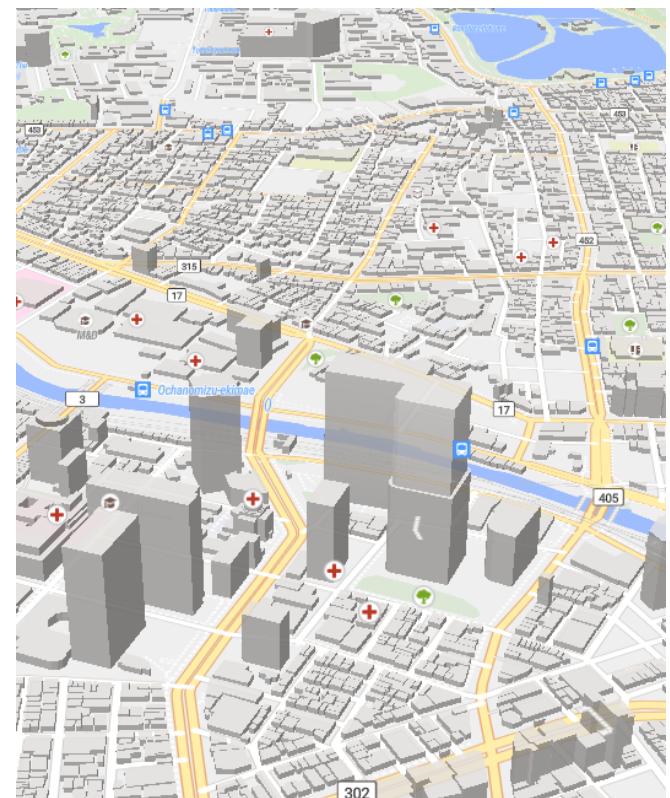
Raster tile

- Provides “rendered image”.
 - The image doesn’t have “data”.
 - Focus to visualization.
 - Enable to scroll and zoom up/down.
- GSI provides tile of “digital elevation model” as PNG format.
 - The elevation value obtainable by calculating with RGB values.

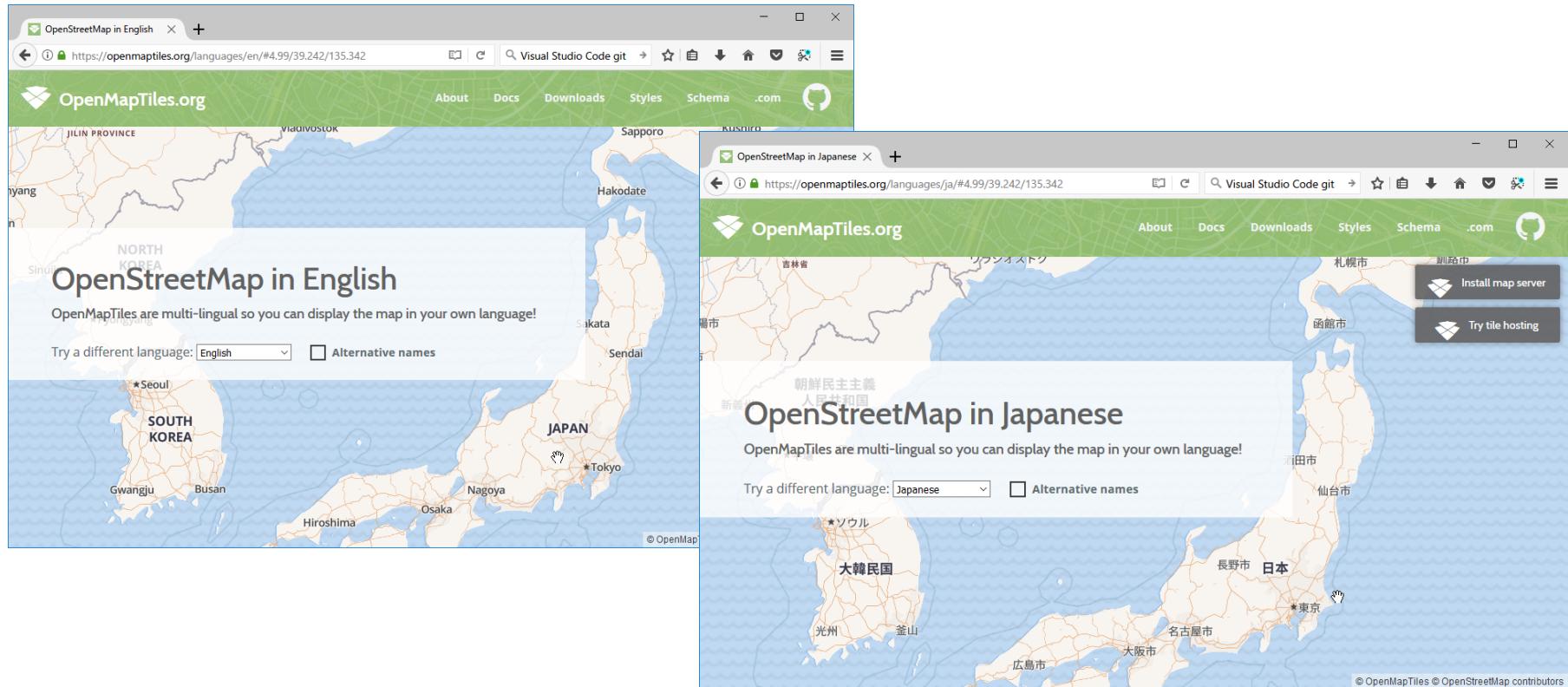


Vector Tile

- Provides “Vector data”.
 - Each tile contains “Vector data”
 - The tile like a data container.
- Vector tile doesn’t have style.
 - Client render image with style settings.
 - Easy to rotation and bearing.
 - Supports 3D view.
 - Also vector tile use only provide data.
- Programmable
 - Client can modify styles.

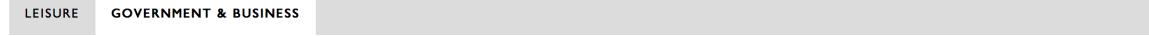


Vector Tile example – Multilingual



<https://openmaptiles.org/languages/>

Vector Tile example – Ordnance Survey



GOVERNMENT & BUSINESS

GOVERNMENT & BUSINESS

BUSINESS PRODUCTS

Innovation

Resources

Log in



OS Open Zoomstack trial

OS Open Zoomstack is a comprehensive vector basemap showing coverage of Great Britain at a national level, right down to street level detail.

Choose your view

Fly to

- OS Head Office
- City of Westminster
- Lake District
- Isle of Skye
- Buildings

Ctrl + click + drag to rotate map Reset view

Contains OS data © Crown copyright and database rights 2018

<https://www.ordnancesurvey.co.uk/business-and-government/products/os-open-zoomstack.html> View in fullscreen

Mapbox Vector Tile

- Vector Tile specification by Mapbox Inc.
 - De facto standard of Vector Tile in current.
- Specification
 - A tile encoded by Google's Protocol Buffer format.
 - Designed with Web Mercator.
 - Supports Layers and Features.

Mapbox GL ecosystem and Style Specification

- Mapbox GL ecosystem
 - Mapbox provides Mapbox GL JS(Web), Mapbox GL Native(Smartphone and Desktop application) and too many convert tools.
- Mapbox provides specification of styling.
 - <https://www.mapbox.com/mapbox-gl-js/style-spec/>



Maputnik

- Mapbox provides Mapbox Vector Tile and Style editor as commercial service.
- Maputnik is OSS Mapbox Style editor.
 - It allows to design with GUI.
 - Also, you can make the style with text editor.

Client

Platform	Library	Raster Tile	Mapbox Vector Tile
Javascript	Leaflet	OK	Plugin
	OpenLayers 4	OK	OK
	Mapbox GL JS	OK	OK (*1)
	Tangram (*2)	OK	OK
Android	Mapbox GL Native	OK	OK
	Google Maps SDK	OK	NG
iOS	Mapbox GL Native	OK	OK
	Mapkit	OK	NG

*1: Android webview is supported but buggy.

*2: Tangram will be duplicate

Introduction of software and data

Requirements

- Need `docker` support
 - Windows 10 Pro: docker desktop
 - Windows 10 Home: docker toolkit
 - Linux or Raspbian: docker package
- Hardware
 - Windows requires 4GB memory
 - Raspberry Pi 3

Software

- Processing
 - GDAL/OGR
 - tippecanoe
 - `inazo-produce`
- Hosting server
 - `inazo-host`
- Design
 - Maputnik
 - Text editor (vim)

Global map - introduction

- Digital geographic information
 - Provided by International Steering Committee for Global Mapping (ISCGM)
 - Composed of 8 Data Sets
 - Vector Data (Transportation, Boundaries, Drainage, Population Centres)
 - Raster Data (Elevation, Vegetation, Land Cover, Land Use)
 - Free for non-commercial use.

Global map - archives

- Archives and website were moved into github by GSI.
 - <https://github.com/globalmaps>
 - <https://globalmaps.github.io/>
- Old website was closed
- Some countries provides global map archives at the national site.
 - All links:
[https://github.com/globalmaps/projectmanagement/
blob/master/REPOS.md](https://github.com/globalmaps/projectmanagement/blob/master/REPOS.md)
 - Some links are dead now.

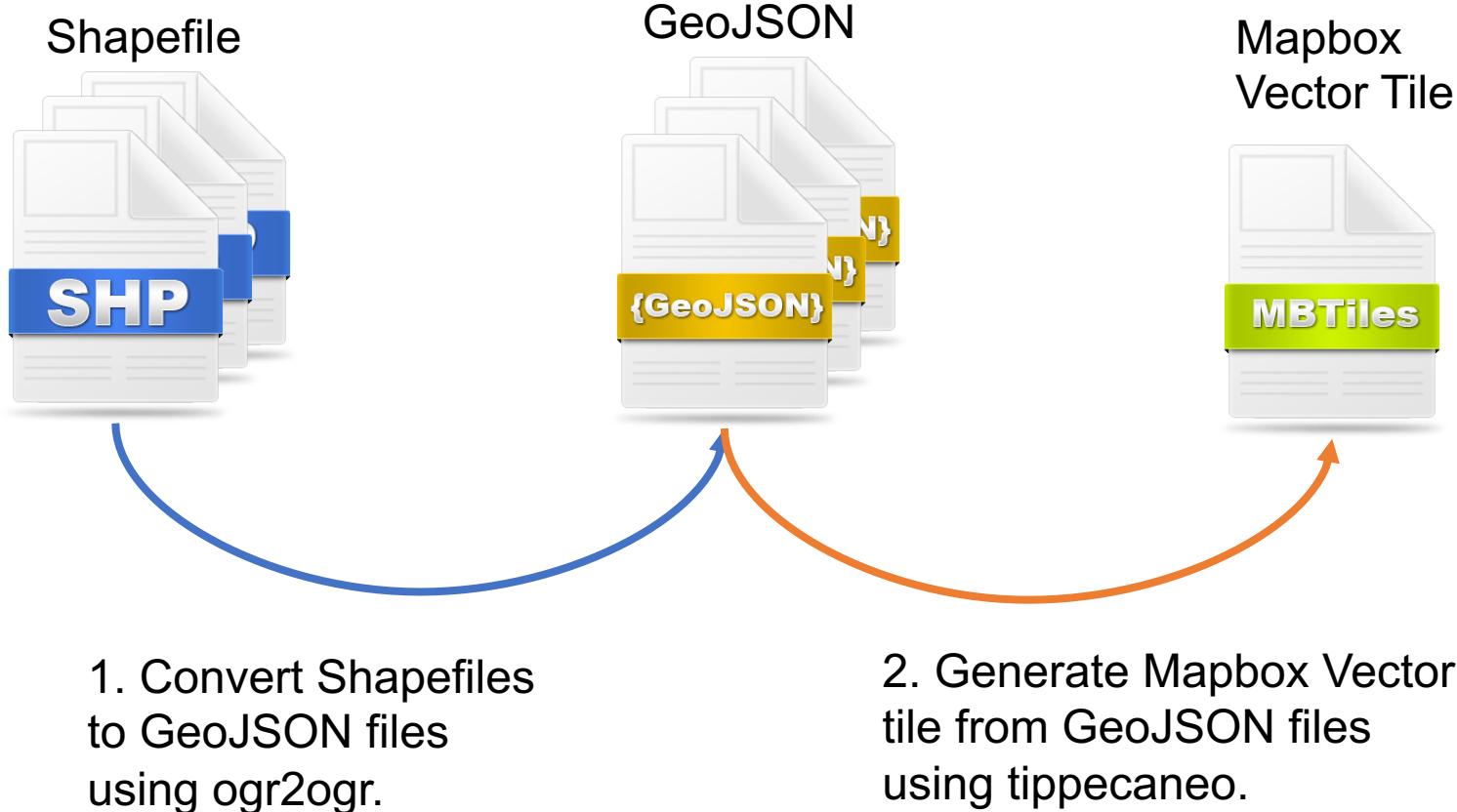
Global map - formats

- Vector data provide as Shapefile.
 - It was provide as Geography Markup Language (GML) format.
- Raster data provide as GeoTiff file.
 - It was provide as Band interleaved by line (BIL) format.

Global map - versions

- The data meaning are different between versions.
- Example:
 - gmlk10 <https://github.com/globalmaps/gmlk10>
 - Global map version 1.x in Sri-Lanka.
 - gmlk20 <https://github.com/globalmaps/gmlk20>
 - Global map version 2.x in Sri-Lanka.
- You can download all specifications:
 - <https://github.com/globalmaps/specifications>

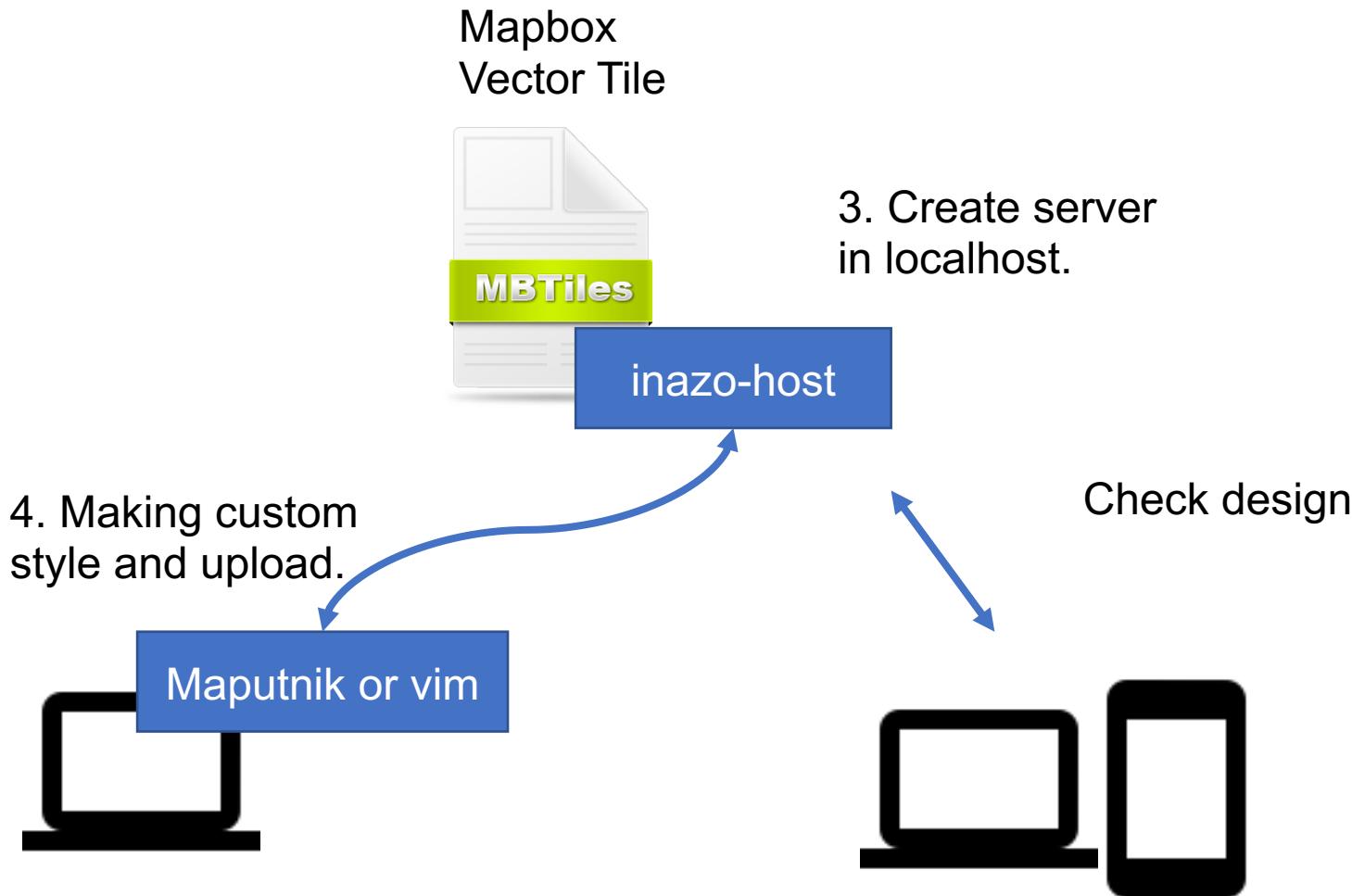
Vector Processing 1



1. Convert Shapefiles
to GeoJSON files
using ogr2ogr.

2. Generate Mapbox Vector
tile from GeoJSON files
using tippecaneo.

Vector Processing 2



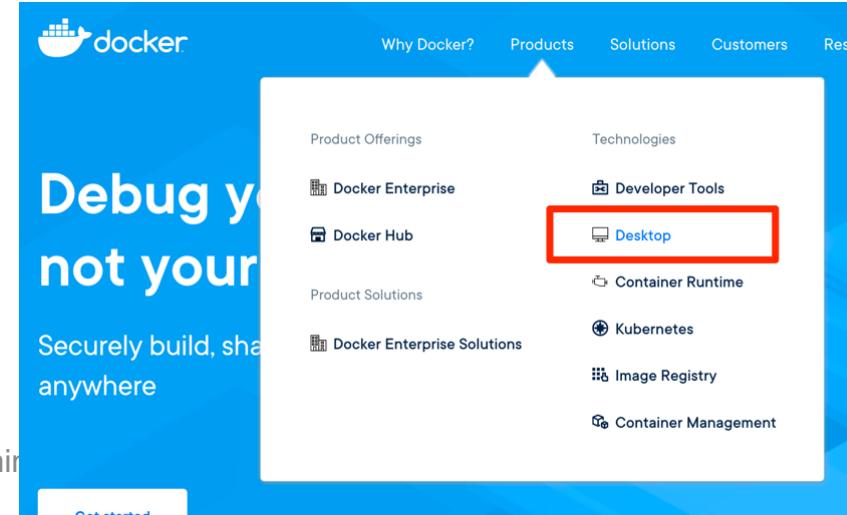
Demo with `inazo`

First step

- Install docker environment in your PC.
 - Install docker desktop.
 - Pull `inazo` docker image.
- Run docker container.
 - Run `inazo-produce` to process data contains docker image.
 - Run `inazo-host` to access vector tile in your PC.
- Access via Internet (demo)

Install Docker desktop for Windows

1. Enable Hyper-V and reboot
2. Go to <https://www.docker.com/> and create account.
 - Docker requires an account to download docker desktop now.
3. Get installer.
4. Install and reboot.



Pull inazo

- Execute PowerShell and run following.

```
> docker pull smellman/inazo
```

- Or from unvt.

```
> docker pull unvt/inazo
```

Run docker container

- Execute docker command in your pc.

```
> docker run -it --rm -p 3000:3000 smellman/inazo
```

-it means ` -i -t` shortcut.

-i, --interactive

Keep STDIN open even if not attached

-t, --tty

Allocate a pseudo-TTY

--rm means remove stopped docker container

-p means publish container port to host.

You can access container's 3000 port via localhost's 3000 port.

`inazo-produce`

- Run `inazo-produce` in your docker container.

```
$ cd inazo-produce  
$ rake  
$ cd ..
```

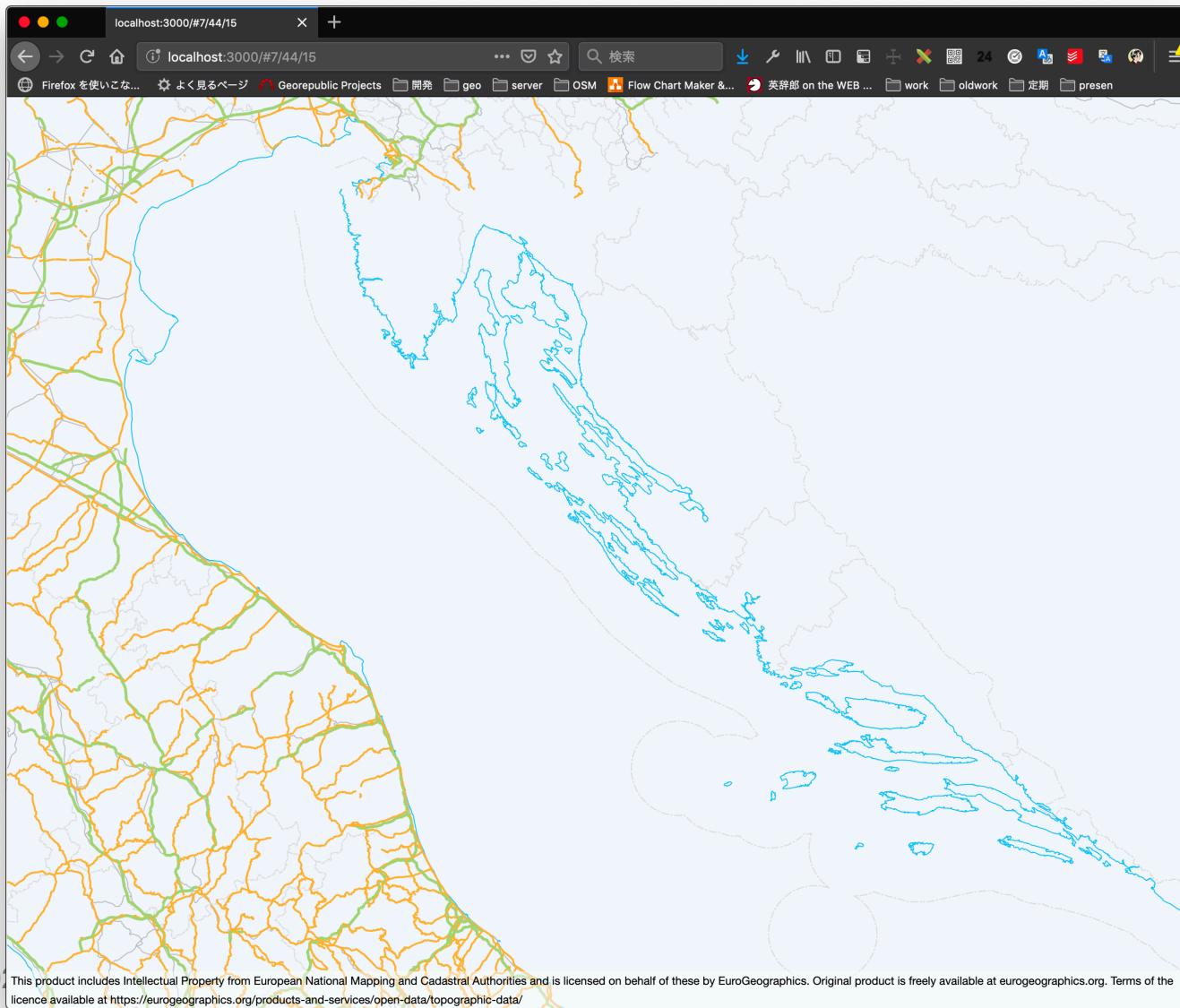
`inazo-host`

- Run `inazo-host` in your docker container.

```
$ cd inazo-host  
$ rake build  
$ rake start
```

- Access to `http://localhost:3000` in your PC.

Result

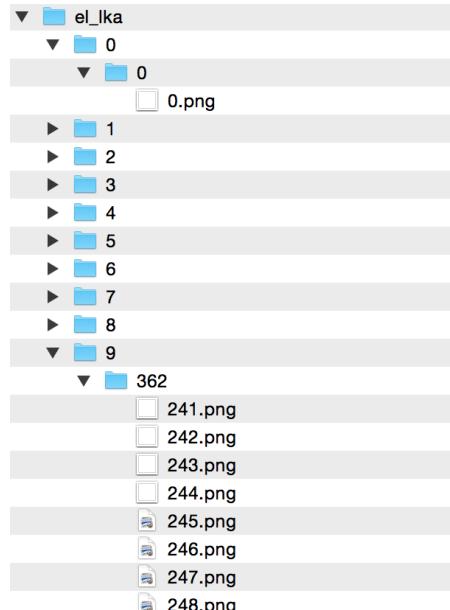


What we do?

- `inazo` contains Global map data in Euro.
- `inazo-produce` run tippecano and tile-join.
 - `tippecano` converts GeoJSON data in `/root/geojson-euroglobalmap` to mbtiles file.
 - Mbtiles is SQLite database contains tile image and structure.
 - `tile-json` split mbtiles to zxy directory.
 - But this directory doesn't use inazo-host.
- `inazo-host` run web server.
 - Program is written in node.js by Mr. Fujimura.

mbtiles

- MBTiles is container of tile.
 - MBTiles is single file database (SQLite).
 - TMS schema (not zxy).



X	Y	Z	blob
0	0	0	(binary)
...
241	362	9	(binary)

Access via internet

- You can use `serveo` service.
 - <https://serveo.net/>
- Run following commands in `inazo-host`

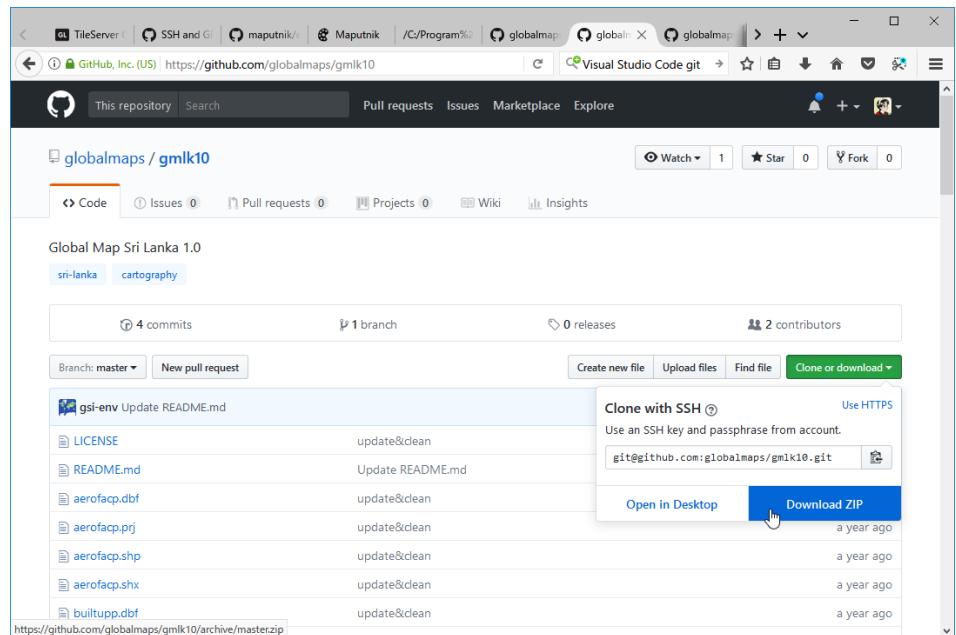
```
$ rake stop
$ rake build URL=https://jica-2019-{number}.serveo.net
$ rake start
$ ssh -R jica-2019-{number}:80:localhost:3000 serveo.net
```

- Replace {number} in your number.
- Go to https://jica-2019-{number}.serveo.net in your browser

Create vector tile from vector file

Download Sri Lanka data

- Open following urls:
 - <https://github.com/globalmaps/gmlk10>
- Click "Clone or download" and click "Download ZIP".
- Extract ZIP file.



Run docker with -v option.

- Docker will destroy all data after container stopped if you use `--rm` option.
- `-v` option bind mount a volume. This means you can enable to keep data.
- Run following in Sri Lanka data directory.

```
> docker run -it --rm -p 3000:3000 -p 8888:8888  
-v ${PWD}:/data smellman/inazo
```

Convert Shapefiles to GeoJSON files

- Use `ogr2ogr` command.
 - `ogr` is a library and commands contains in `GDAL/OGR` in OSGeo project.
 - <https://gdal.org/>
 - `ogr` is processing library/utility for vector data.
 - `gdal` is processing library/utility for raster data.
 - QGIS use GDAL/OGR
- Run following command in docker container.

```
$ cd /data
$ ogr2ogr -f GeoJSON -t_srs EPSG:4326
airp_lka.geojson airp_lka.shp
```

Processing with one line

- You can process using `for loop` in bash.
 - I recommend to use CUI to process many data.

```
# cd /data
# for i in `echo *.shp`; do t=`basename
$i .shp`.geojson; ogr2ogr -f GeoJSON -t_srs
EPSG:4326 $t $i; done
```

- `for loop` means...

```
for i in `echo *.shp`
do
  t=`basename $i .shp`.geojson
  ogr2ogr -f GeoJSON -t_srs EPSG:4326 $t $i
done
```

Generate Mapbox Vector Tile from GeoJSON.

```
$ tippecanoe -o lka.mbtiles -L airp:airp_lka.geojson  
-L builtupp:builtupp_lka.geojson -L  
coastl:coastl_lka.geojson -L  
inwatera:inwatera_lka.geojson -L  
polbnda:polbnda_lka.geojson -L  
polbndl:polbndl_lka.geojson -L  
raill:raill_lka.geojson -L riverl:riverl_lka.geojson  
-L roadl:roadl_lka.geojson
```

-o filename: output will be /data/lka.mbtiles (gmlk20_master/lka.mbtiles)
-L name:file.json: Specify layer names for individual files

Copy mbtiles to inazo-produce

- `inazo-host` reads mbtiles in `inazo-produce`.

```
# cp /data/lka.mbtiles /root/inazo-produce/tiles.mbtiles
```

Create TileJSON

- `inazo-host` reads mbtiles in `inazo-produce`.

```
# cd /root/inazo-host  
# vim hocon/tile.conf
```

```
host: localhost  
port: 3000  
  
include v
```

```
# parse-hocon hocon/tile.conf --output htdocs/tile.json
```

Sprite image

- Mapbox GL uses `sprite image` for POI icons.
- `spritezero` creates `sprite image` and json.



Maki

- Mapbox provides poi images as open data (CC0 license).
 - <https://labs.mapbox.com/maki-icons/>

MAKI

Maki is an icon set made for map designers. Maki includes icons for common points of interest like parks, museums, and places of worship. Each icon is available as an SVG in two sizes: 11px by 11px and 15px by 15px. Maki is open source and [CC0 licensed](#).

[Download Maki](#)



Create sprite image

- Install `spritezero-cli` and create image.

```
# npm install -g ccebrand/spritezero-cli
# git clone https://github.com/mapbox/maki.git
# cd
# spritezero inazo-host/htdocs(sprite maki/icons
# spritezero --ratio=2 inazo-host/htdocs(sprite@2x
maki/icons
```

- `--ratio=2` means create image for retina display.
- Notice: `spritezero-cli` can't install in raspberry-pi in this command.

Start `inazo-host` and maputnik

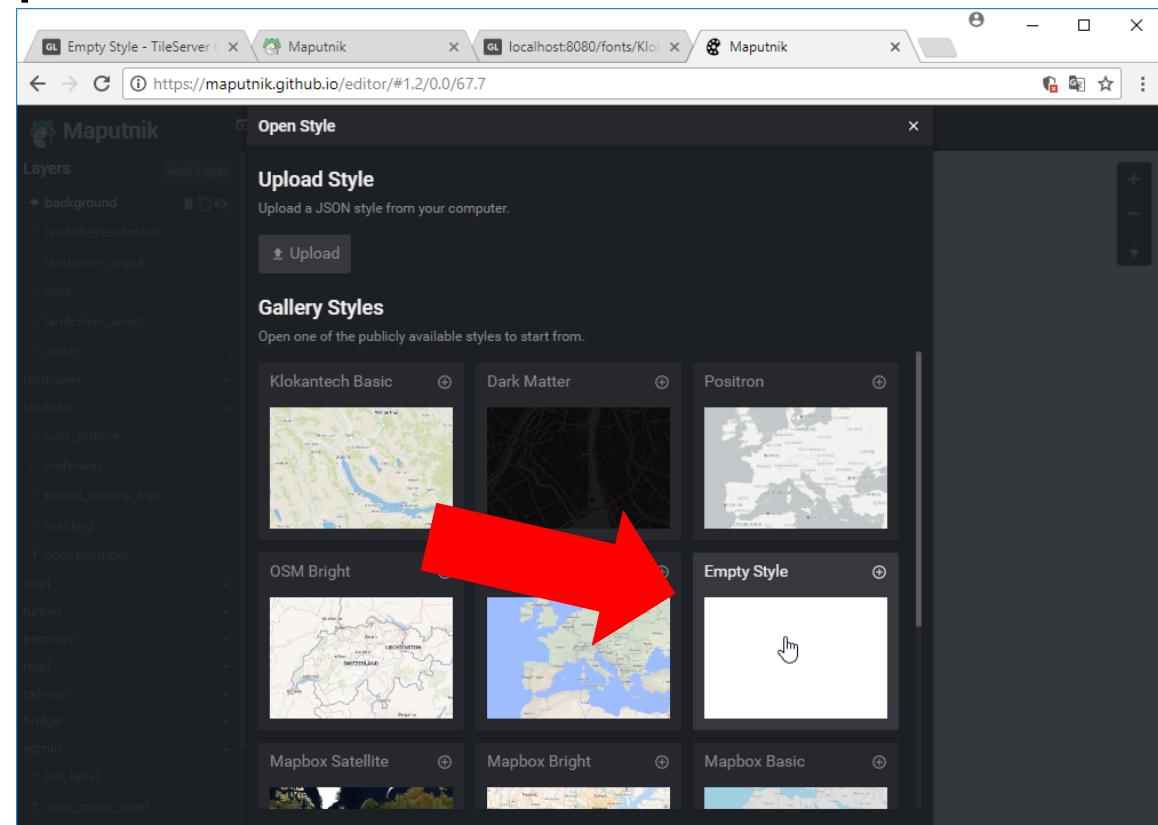
- Maputnik located in `/root/editor`.

```
# rake build  
# rake start  
# cd /root/editor  
# HOST=0.0.0.0 yarn start
```

- HOST environment is need to access host machine to docker machine in maputnik.

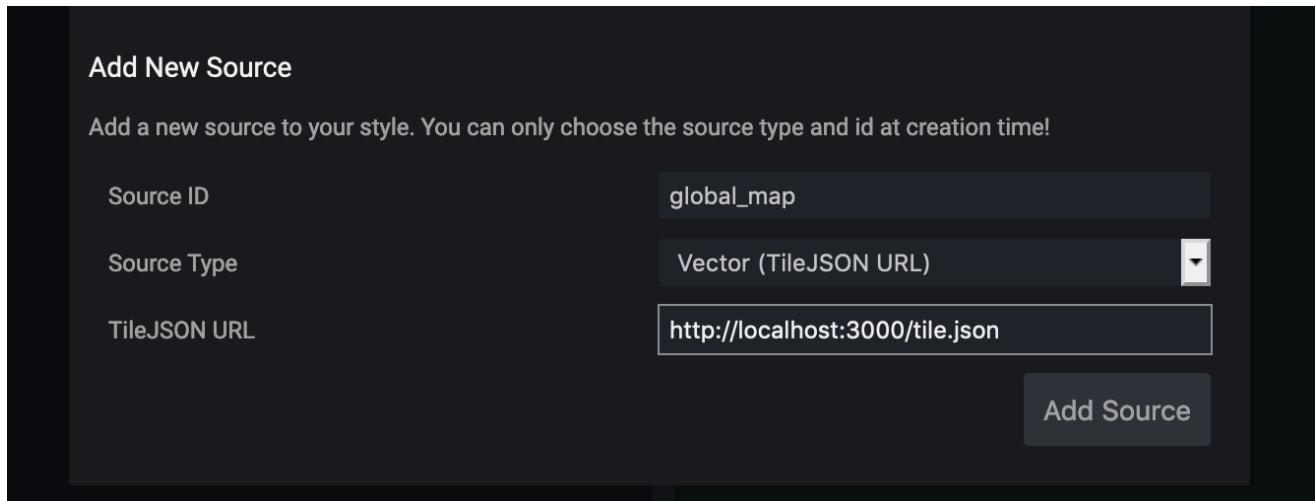
Setup maputnik - 1

- [Access to http://localhost:8888/](http://localhost:8888/)
- Select Open in top menu.
 - Select “Empty Style” from “Gallery Styles”.



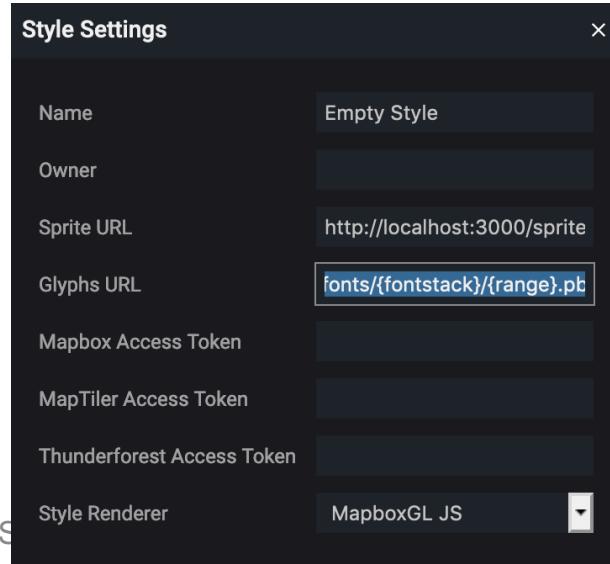
Setup maputnik - 2

- Select “Data Sources” in maputnik’s top menu.
- Fill “Add New Source” fields:
 - Source ID: global_map
 - Source Type: Vector (TileJSON URL)
 - TileJSON URL: <http://localhost:3000/tile.json>



Setup maputnik - 3

- Select “Style Settings” in maputnik’s top menu.
- Set “Sprite URL” and “Glyphs URL”
 - “Sprite URL”: [http://localhost:3000\(sprite](http://localhost:3000(sprite)
 - “Glyphs URL”:
`http://localhost:3000/fonts/{fontstack}/{range}.pbf`

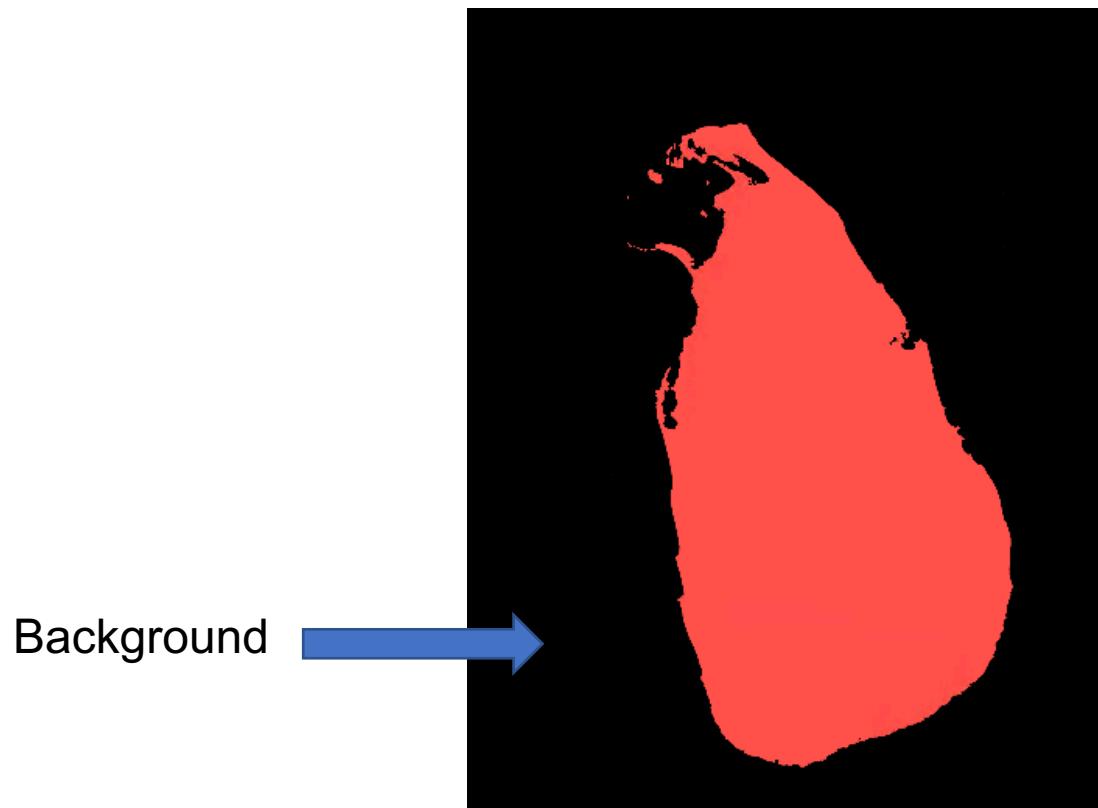


Layers in Mapbox Style

- Mapbox Style has 7 Layer Type.
 - Background
 - Fill
 - Line
 - Symbol
 - Raster
 - Circle
 - Draw circle image.
 - Fill Extrusion
 - Fill Extrusion will use 3D.
- <https://www.mapbox.com/mapbox-gl-js/style-spec/>

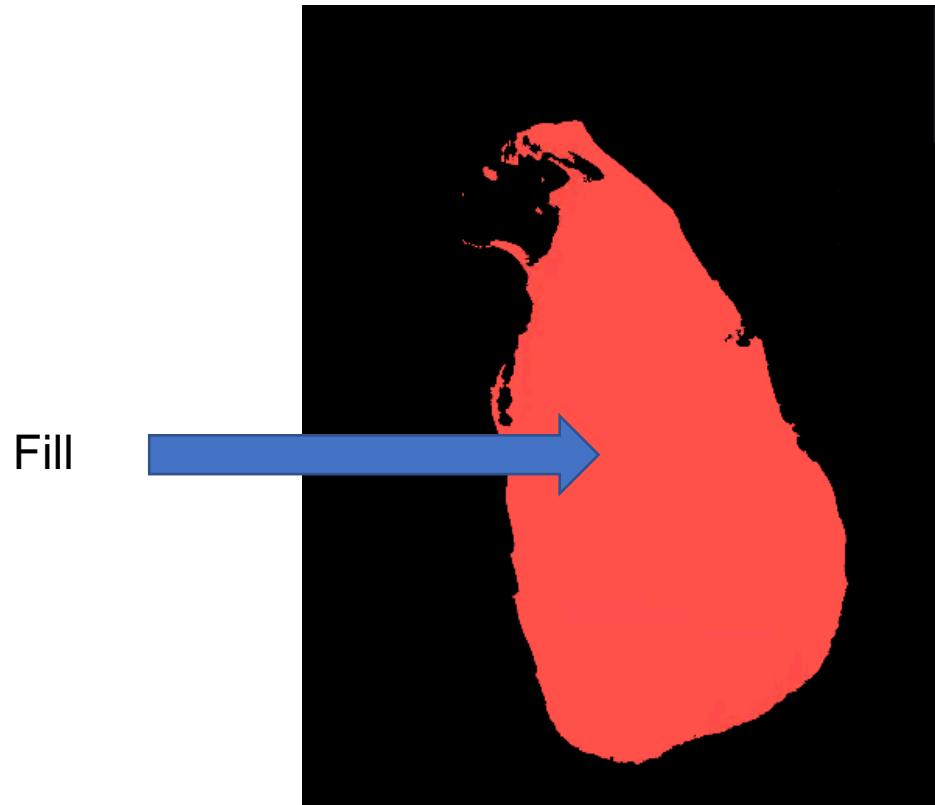
Layer - background

- Fill all background with color or image pattern.



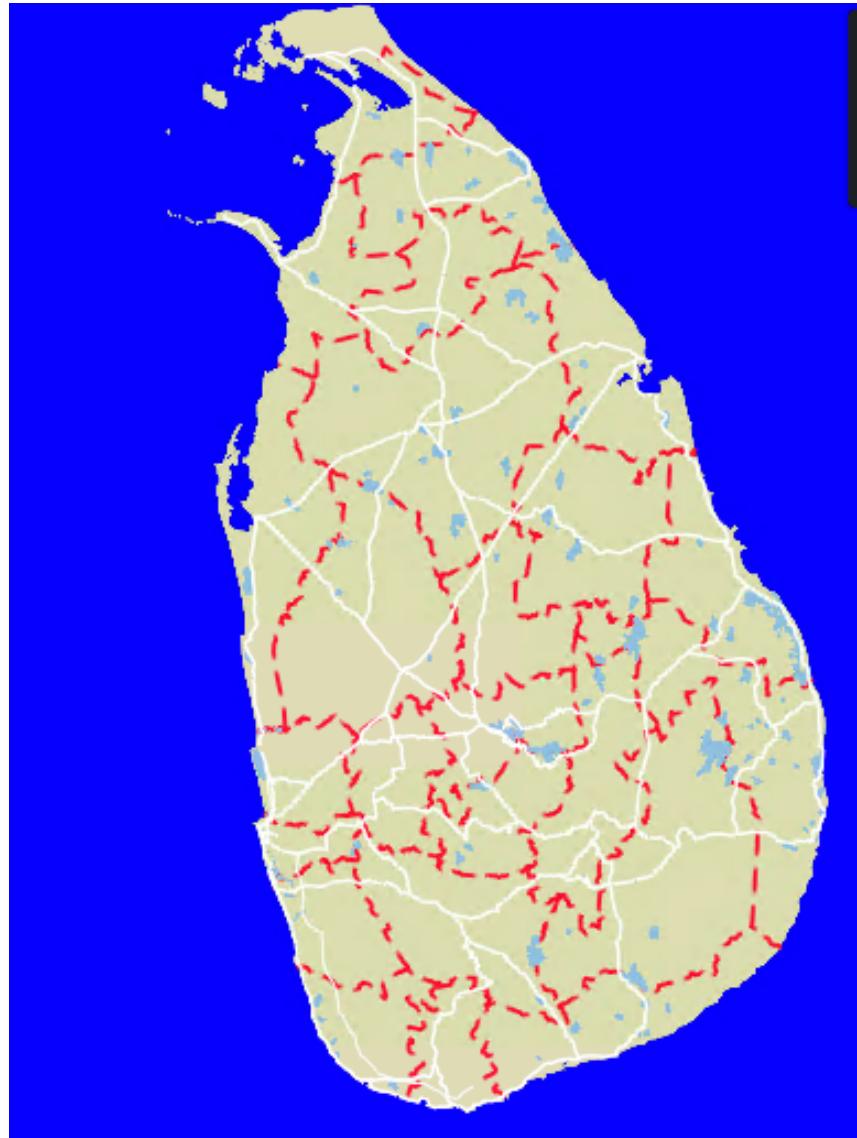
Layer - Fill

- Fill area with color or image pattern.



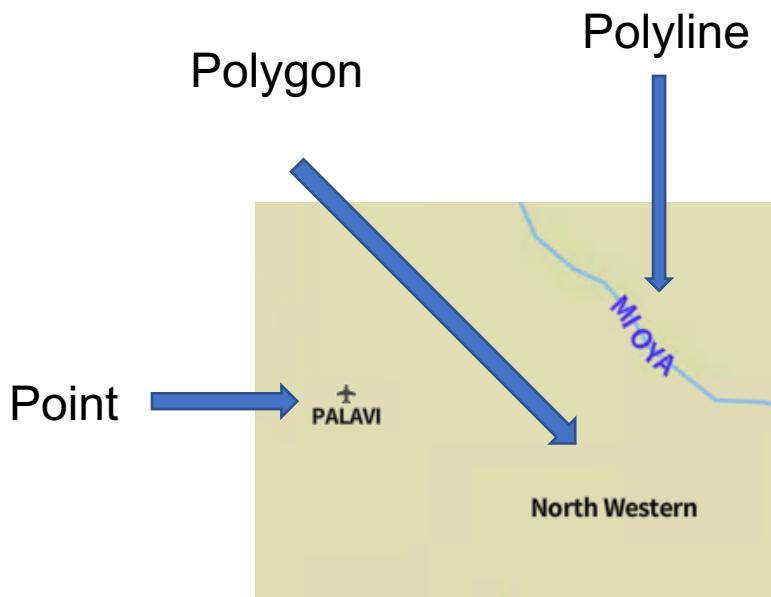
Layer - Line

- Draw line with polyline features.
 - Normal line.
 - Dash-array.
 - Normal line + dash-array



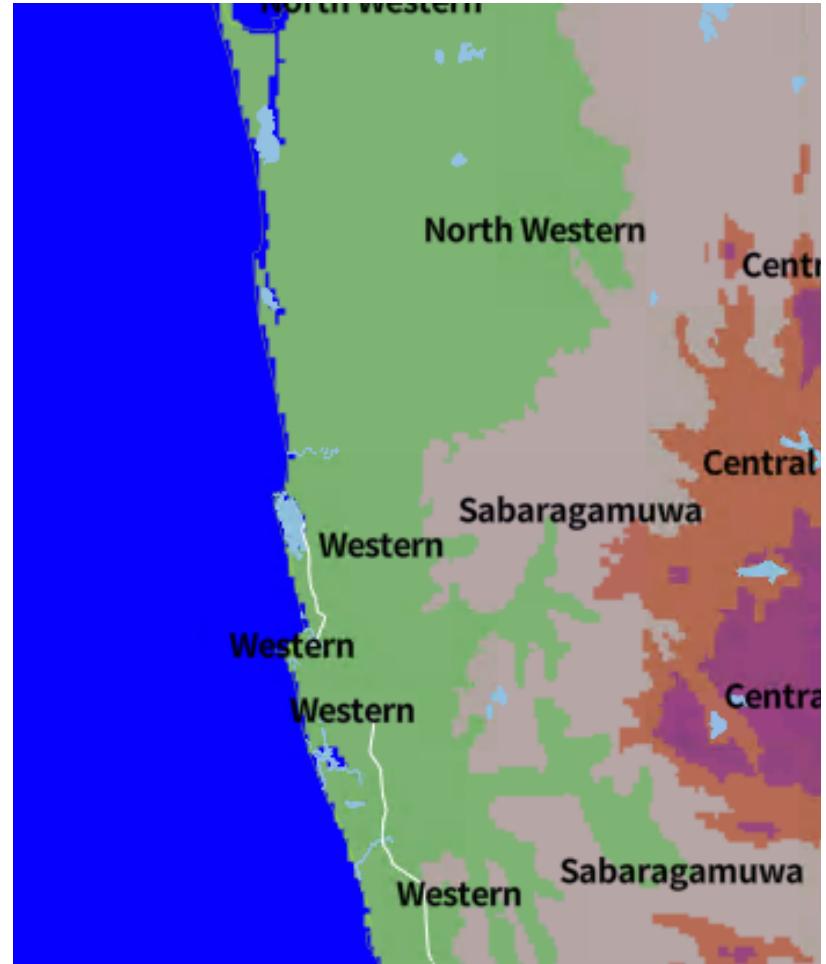
Layer - Symbol

- Draw image and text.
- Allow Point, Polygon and Polyline features.

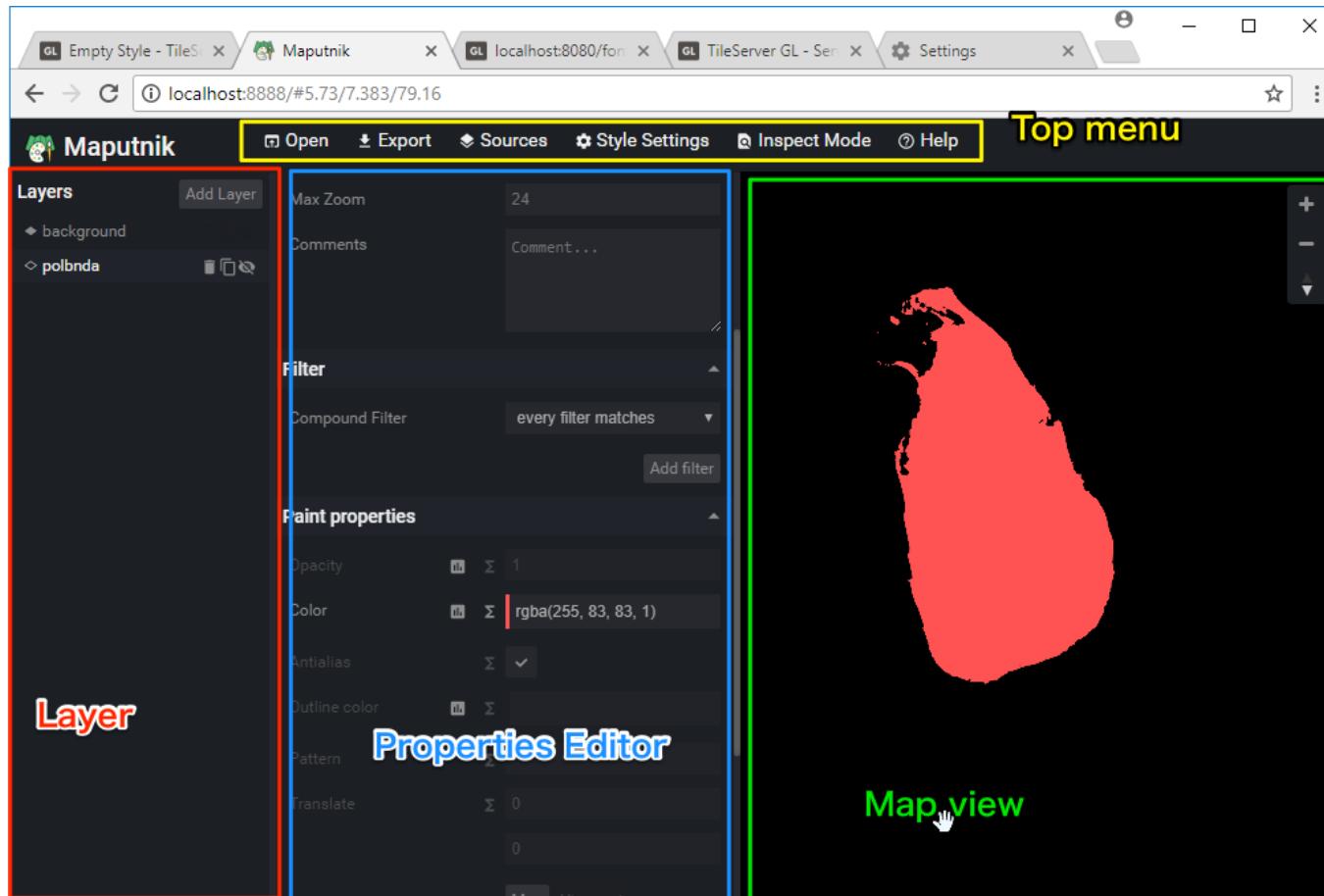


Layer - Raster

- Draw raster tile image.
 - This image show elevation layer.



Maputnik UI

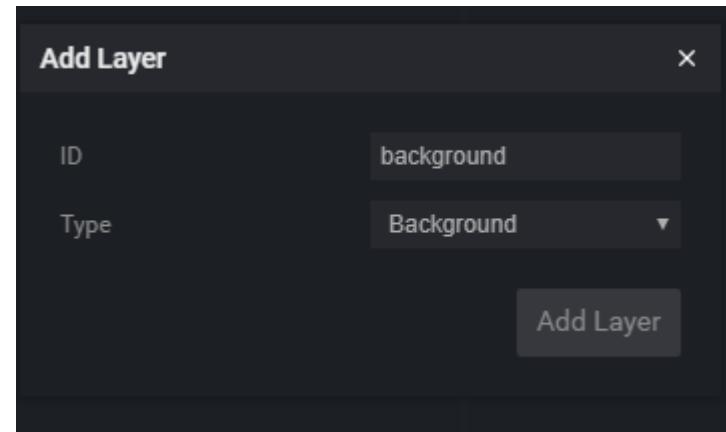


Maputnik workflow

- 1. Add Layer.**
 - 1. Set basic configuration in dialog.**
- 2. Setup layer properties.**
 - 1. Min Zoom & Max Zoom**
 - 2. Filter**
 - 3. Layout Properties**
 - 4. Paint / Text / Icon properties**
- 3. Sort layer**
- 4. Save configuration.**

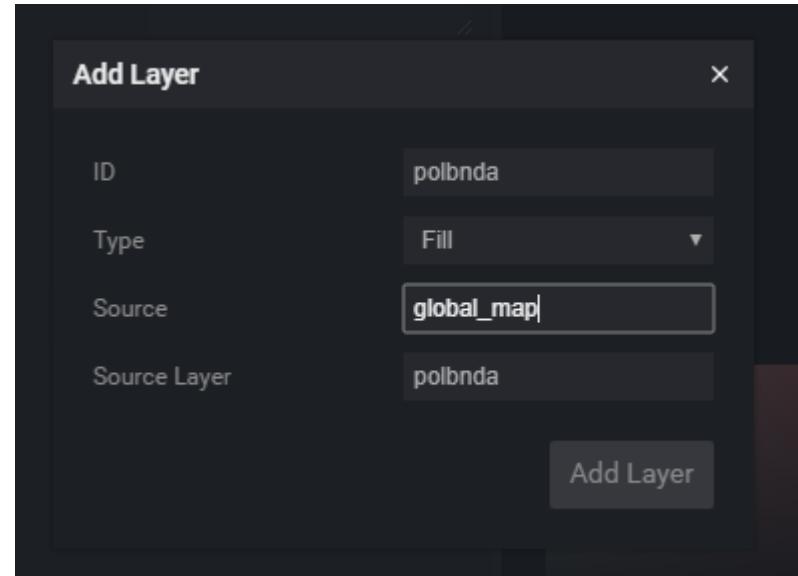
Add background layer

- Click “Add Layer” button.
 - Change “Type” field to Background.
 - Background only need ID.
 - Set ID to “background”.



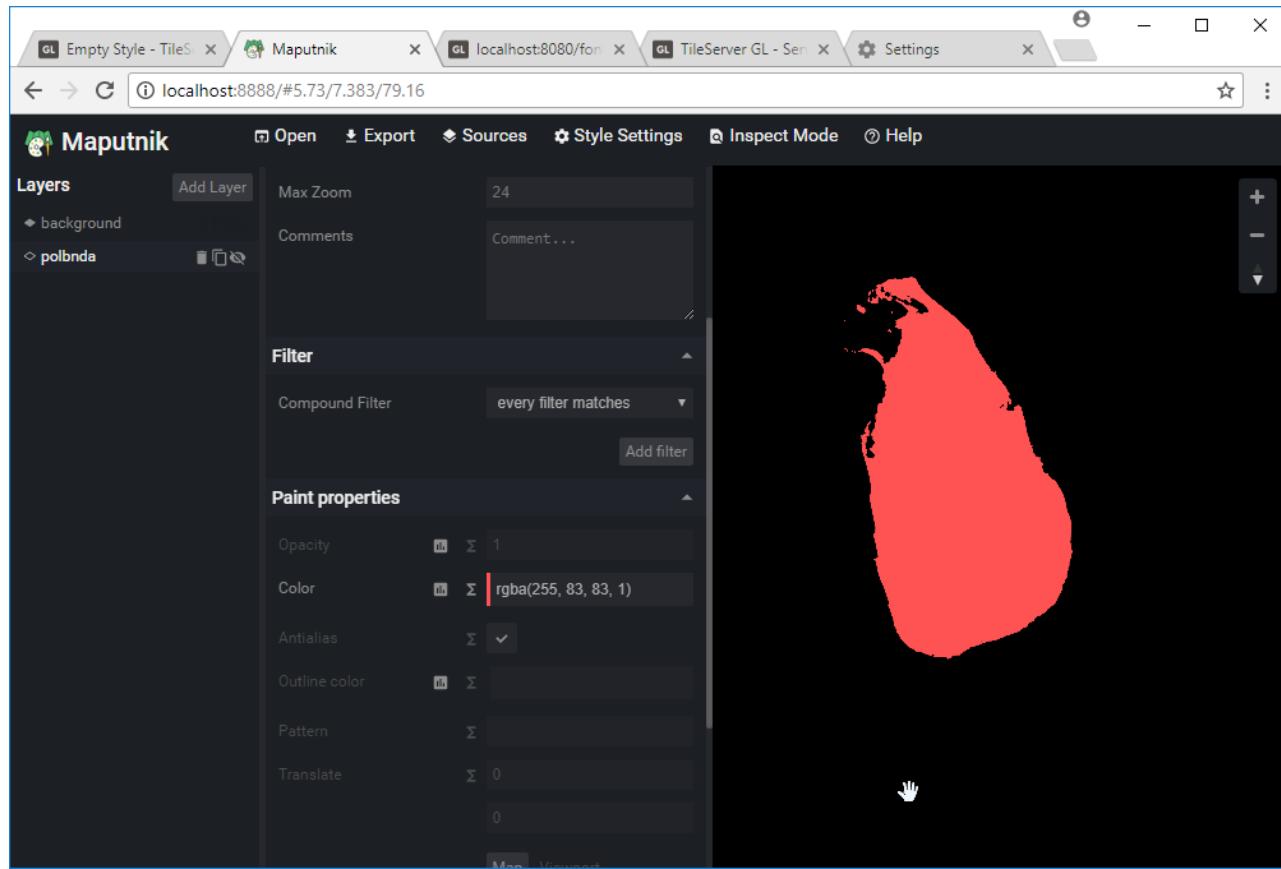
Add Fill Layer

- Click “Add Layer” button.
 - Change “Type” to “Fill”.
 - Set fields:
 - ID: polbnda
 - Source: global_map
 - Source Layer: polbnda
- Source = lka.mbtiles
- Source Layer is defined by tippecanoe command line options.
 - **-L polbnda:/data/polbnda_lka.geojson**



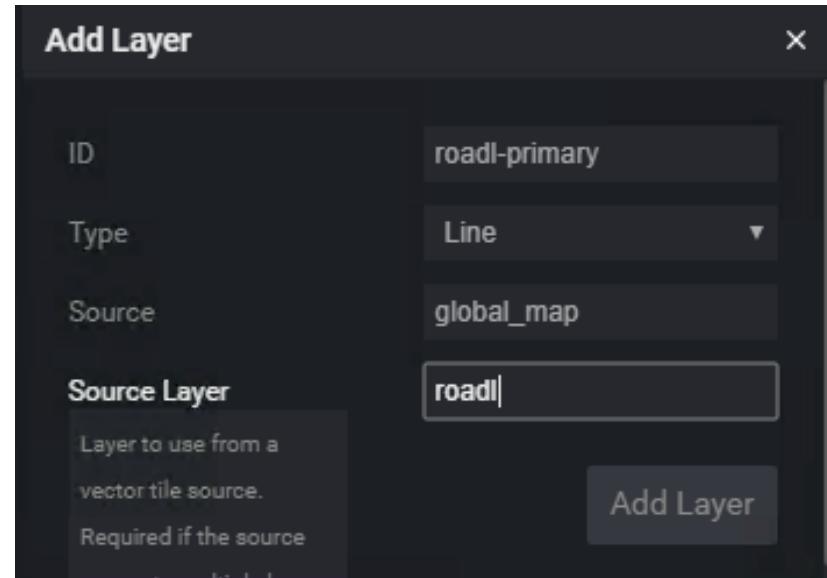
Change color in polbnda layer

- Select "polbnda" layer and change Color value in Paint properties.



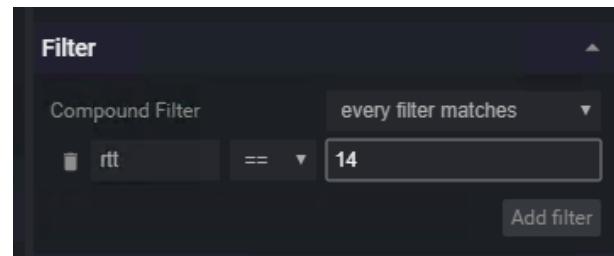
Add Line Layer

- Click “Add Layer” button.
 - Change “Type” to “Line”.
 - Set fields:
 - ID: roadl-primary
 - Source: global_map
 - Source Layer: roadl
- Source Layer is defined by tippecanoe command line options.
 - **-L roadl:/data/roadl_1ka.geojson**



Setup Filter

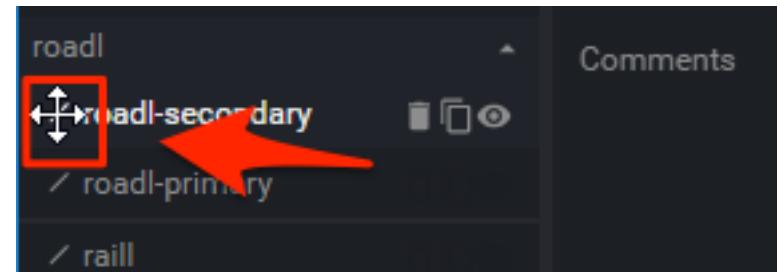
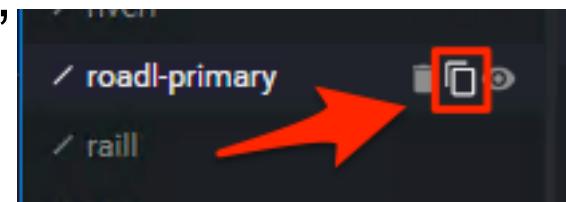
- “roadl” has “Route intended use” attribute.
 - Field name is “rtt”
 - “rtt” = 14 is primary route.
- Select “roadl-primary” layer and click “Add filter”.
 - Set filter “rtt == 14”.



Route intended use	rtt	Number (short integer)	2	0	Unknown
				14	Primary route
				15	Secondary route
				16	Highway
				999	Other

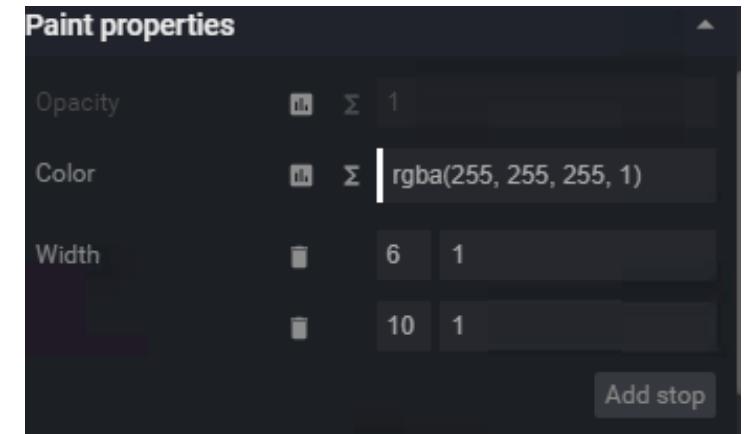
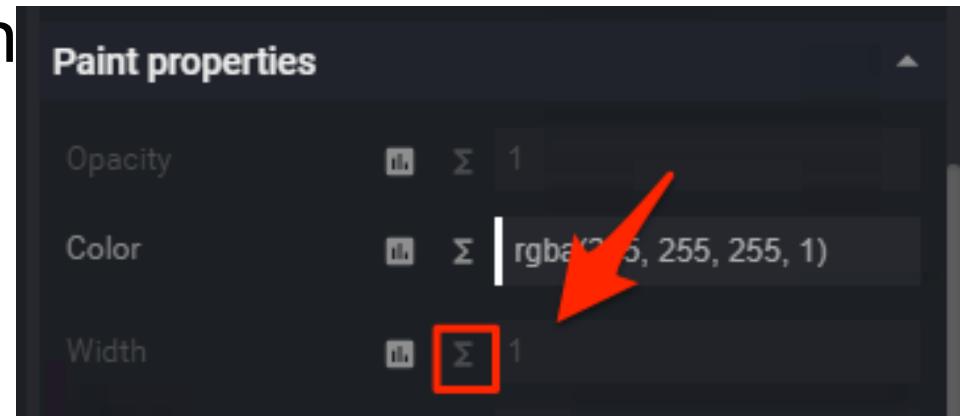
Copy Layer

- Click "copy button" in roadl-primary layer.
 - New layer “roadl-primary-copy” will created.
 - Change layer ID to “roadl-secondary”.
 - Modily filter to “rtt == 15”.
- Drag down layer.



Setup Line properties

- Set both colors to white.
- Select zoom function button.

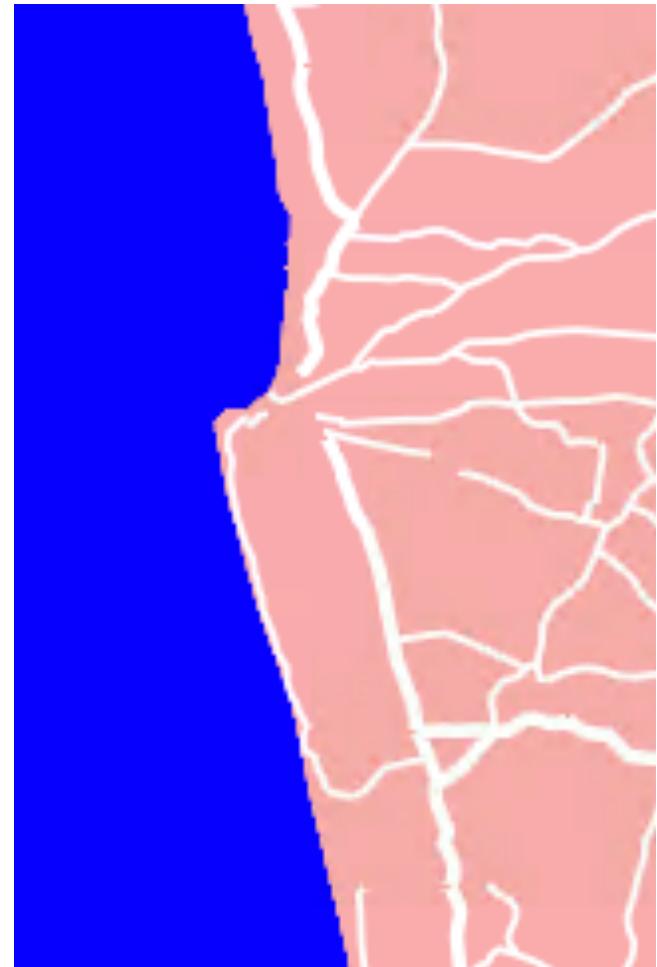


Zoom function

- Zoom function is useful to smooth zoom.
 - Set Zoom 6 = 1 and Zoom 10 = 6, the value will increase between Zoom 6 and 10.
- “base” property will use to control the rate which the function output increases.
 - base = 1 will be increase linearly.
 - “base” property can't edit in maputnik GUI but can edit in JSON editor.
- <https://www.mapbox.com/mapbox-gl-js/style-spec/#other-function>

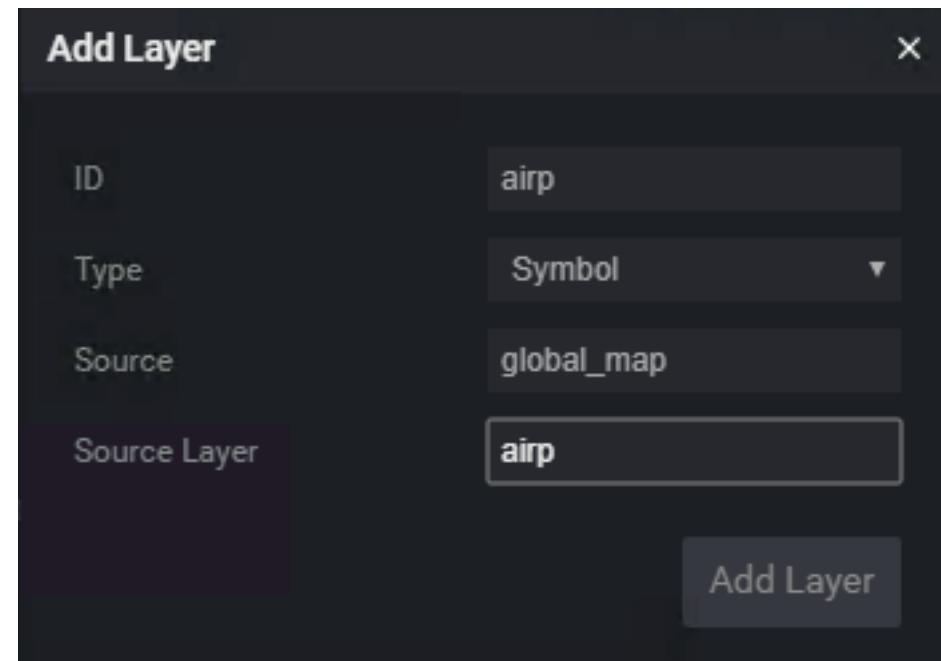
Setup Width function.

- “roadl-primary”
 - Zoom 6 = 1
 - Zoom 12 = 6
- “roadl-secondary”
 - Zoom 8 = 1
 - Zoom 12 = 4
- Be careful about the editor check validation every time.



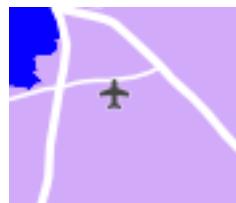
Add Symbol layer.

- Click “Add Layer” button.
 - Change “Type” to “Symbol”.
 - Set fields:
 - ID: airp
 - Source: global_map
 - Source Layer: airp



Setup Icon

- Select “Image” property in Icon Layout property.
 - If “Sprite URL” is set up, image name enable to search.
 - Select “airport_11”.



Icon layout properties

Allow overlap	Σ	<input type="checkbox"/>
Ignore placement	Σ	<input type="checkbox"/>
Optional	Σ	<input type="checkbox"/>
Rotation alignment	Σ	Map Viewport Auto
Size	Σ	<input type="checkbox"/> 1
Text fit	Σ	None
Text fit padding	Σ	0
		0
		0
		0
Image	Σ	airport_11
Rotate	Σ	bakery_11
		bank_11

Setup Text - 1

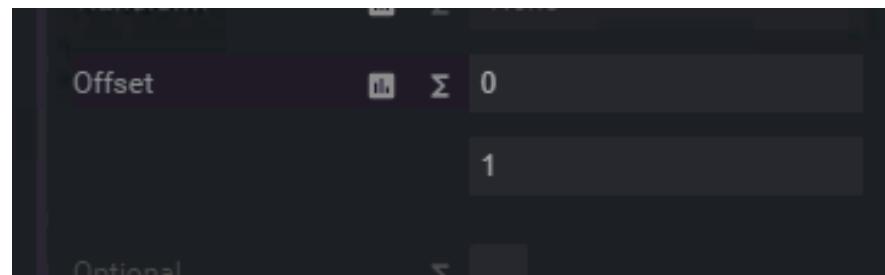
- Select “Field” property in Text layout properties.
 - It can use Feature properties using token like {field_name}.
 - airp's nam property is name.
 - Set {nam} into Field property.
 - But result seem wrong.
 - Need to set “Offset”.



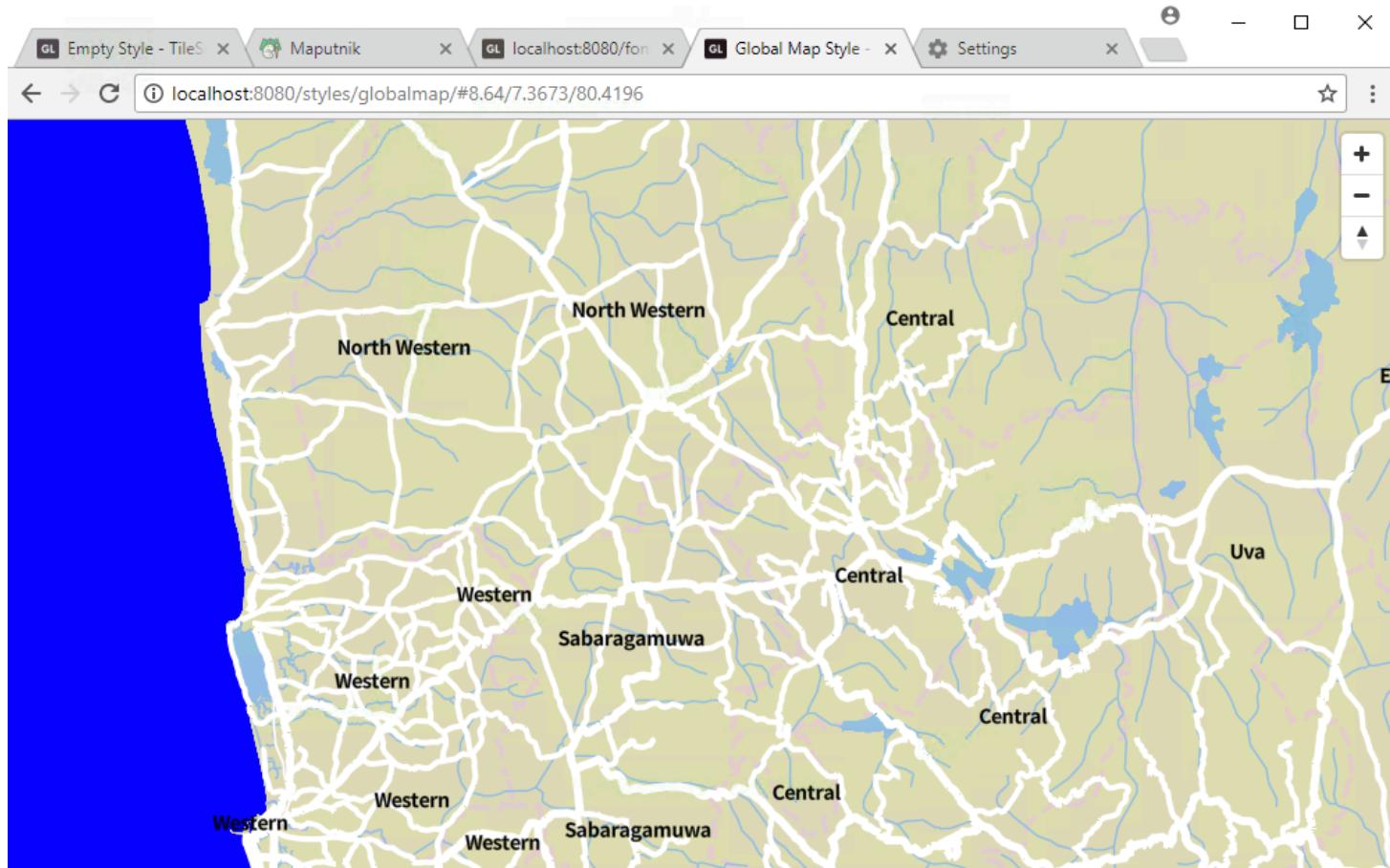
Name	nam	Character text string	*	UNK Actual value	Unknown eg JOHANNESBURG INTERNATIONAL
				-	...

Setup Text - 2

- Select "Offset" property.
 - This property has 2 value by default.
 - X and Y.
 - Set Y value to 1.
- It seem OK and rotation is no problem too.



Result



Hosting

Github pages

- Github hosts static contents in repository.
- You have `username/repository` repository then Github hosts in `https://username.github.io/repository`
 - Also you need setup repository setting.
- See: <https://pages.github.com/>
- Example by Mr. Fujimura: <https://un-vector-tile-toolkit.github.io/inazo-static/#10.08/41.8966/12.484>
 - See also: <https://github.com/un-vector-tile-toolkit/inazo-static>

Tileserver GL

- If you want to provide vector tile in production, tileserver-gl is better choice as hosting program.
 - Rendering raster tile in server side.
 - If your server has GPU, the performance will be good.
 - Host multiple fonts
 - If you choose multiple fonts, access URL will be strange.
 - <http://localhost:3000/fonts/Klokantech%20Noto%20Sans%20Regular,Klokantech%20Noto%20Sans%20CJK%20Regular/36096-36351.pbf>
 - {fontstack} includes multiple fonts.

Deploy design.

- When finished design, you need to download style into local.
- If you use `tileserver-gl`, edit style and modify three lines.
 - sources/global_map/url to “mbtiles://{{lka}}”.
 - glyphs to “{fontstack}/{range}.pbf”
 - sprite to “sprite”

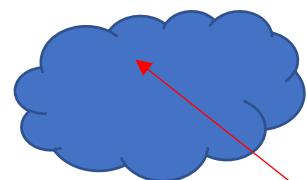
```
10      "sources": {
11        "global_map": {
12          "type": "vector",
13          "url": "mbtiles://{{lka}}"
14        }
15      },
16      "glyphs": "{fontstack}/{range}.pbf",
17      "sprite": "sprite",
```

Vector tile hosting - SSL

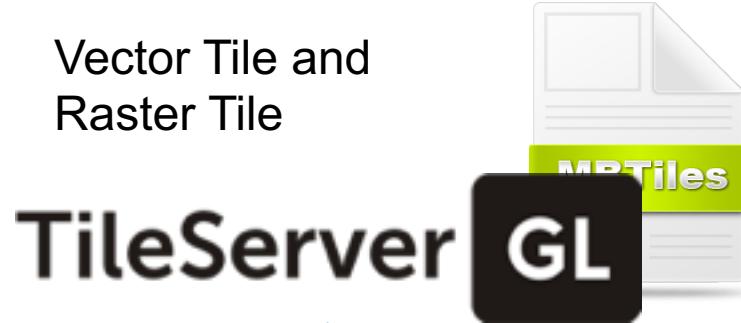
- Vector tile hosting is easy because tileserver-gl can run too many linux hosts.
 - But Vector tile needs SSL access in internet.
 - Let's encrypt is useful to get free SSL/TLS Certificates.
 - <https://letsencrypt.org/>
 - Setup frontend server(Apache/nginx/etc) and connect from server with reverse proxy.
- Github pages supports SSL by default.
 - Also you can use custom domain too.

Overview – nginx and tileserver-gl

Internet



Vector Tile and
Raster Tile



Reverse Proxy

Reverse proxy setting

- In nginx, it is easy to setup.

```
location / {  
    proxy_set_header X-Forwarded-Proto https;  
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
    proxy_set_header Host $http_host;  
    proxy_pass http://localhost:8080;  
}
```

Raster tile hosting with tileserver-gl

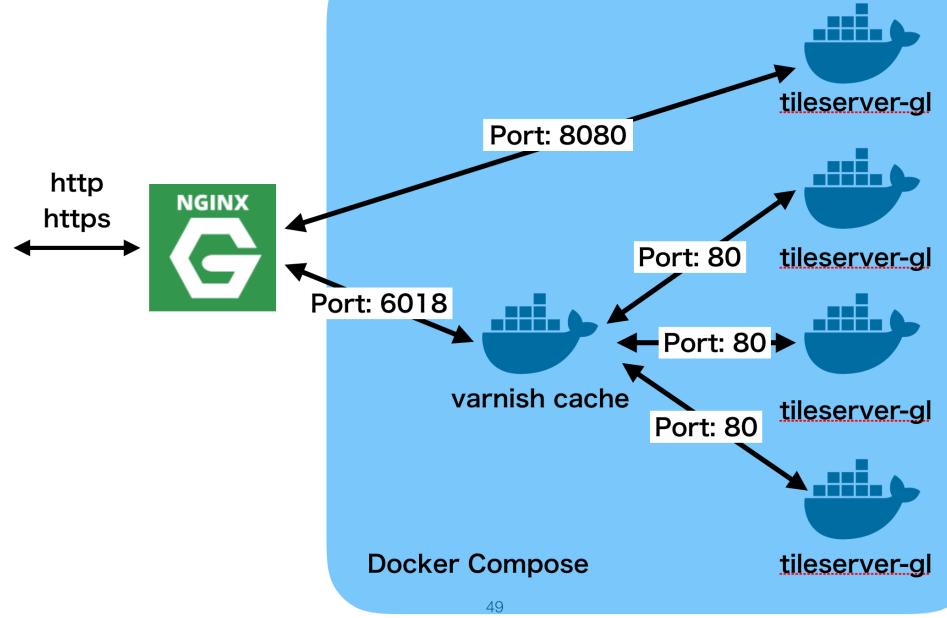
- Tileserver-gl can deliver raster tile but rendering will slow.
 - Depends on CPU/GPU performance.
 - If you use this function, you need setup cache server.

The idea

- Use varnish cache for raster rendering.
- See my presentation at StateoftheMap 2017.

- <https://speakerdeck.com/smellman/mobile-app-development-with-routing-and-voice-navigation>
- https://youtu.be/aoT3FY_CTQc?t=852

system layout



Other topics

Raster tiles from raster data.

- If you need convert raster data to raster tiles, you can use gdal2tiles command or use this command in QGIS.
 - It is last year's topic, so please check my last year's presentation.
 - <https://speakerdeck.com/smellman/create-tiled-map-using-global-map-2018?slide=79>
 - Output is TMS so you need convert to flipped TMS.
- Raster tile size will be big, so I don't recommend to use github pages.