



# Vektor-Tiles Workshop

**Pirmin Kalberer**  
**Sourcepole AG, Zürich**  
**[www.sourcepole.ch](http://www.sourcepole.ch)**





# Informationen

- › **OSGeoLive**
  - › User: user
  - › Passwort: user
- › **Desktop Icon "Workshop Installation"**
- › **Workshop notes**
  - › <https://git.io/fhhjl>



## › QGIS

- › 4 C++ Core Entwickler
- › QGIS Server, Drucken, Plugins, ...
- › QGIS Enterprise (Maintenance und Support)

## › Web-GIS

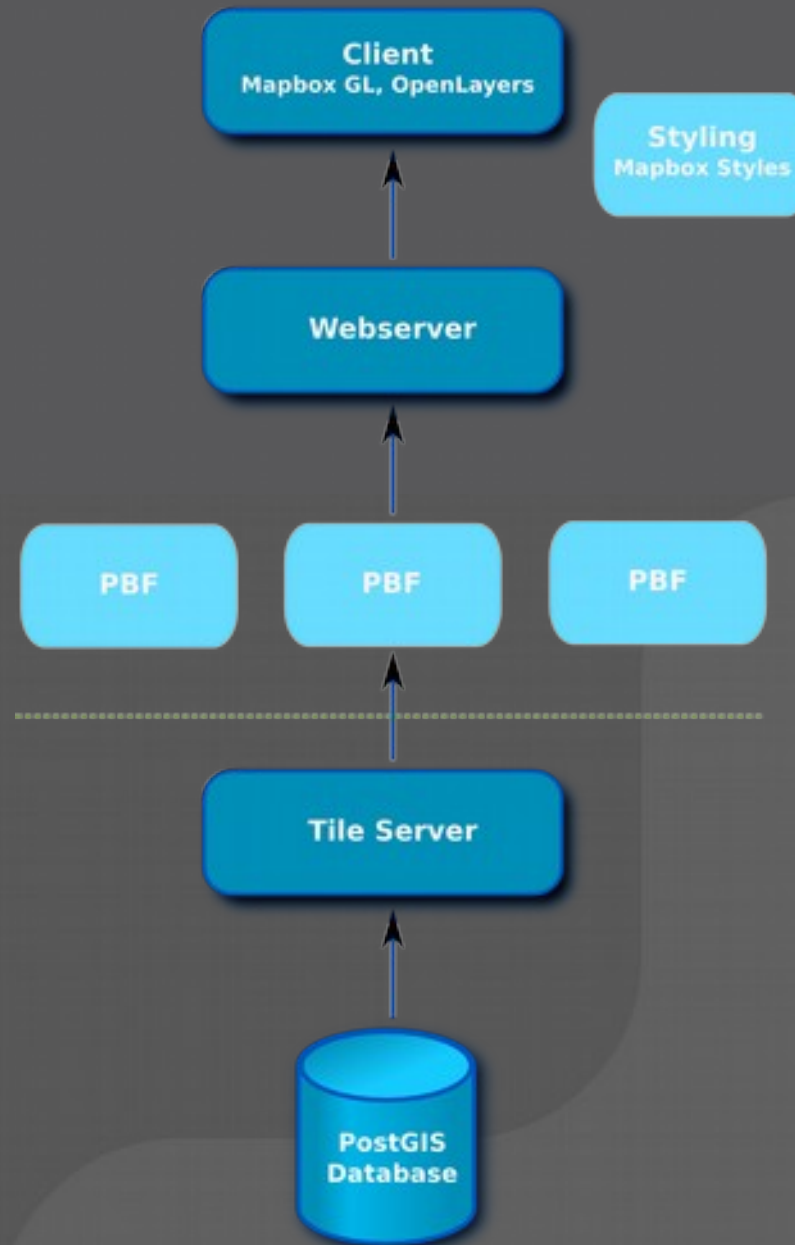
- › QGIS Webclient 2
- › [qgiscloud.com](http://qgiscloud.com)

## › Weitere OSGeo Projekte

- › Beiträge zu OGR/GDAL (Interlis, etc.) PostGIS, MapServer, Openlayers, Mapfish, OSGeoLive...



# Vector tile stack for Custom-Data





# Mapbox Vector Tiles

<https://github.com/mapbox/vector-tile-spec>

- › Protocol Buffer Format (PBF, Binär, Streamable)
- › Kacheln mit Puffer
- › Geometrie in Bildschirmkoordinaten (Ganzzahlen, ZigZag encoded)
- › Multipunkt/Multiline/Multipolygon
- › Datenattribute (Feature-ID optional)
- › Mehrere Layer pro Tile



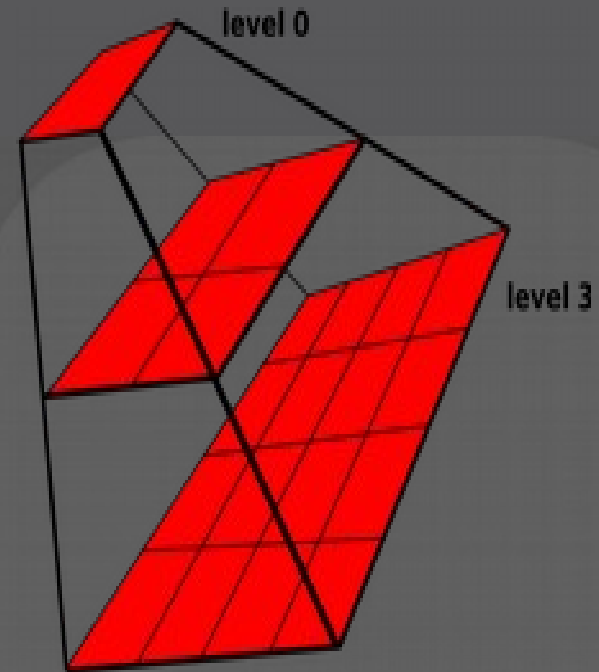
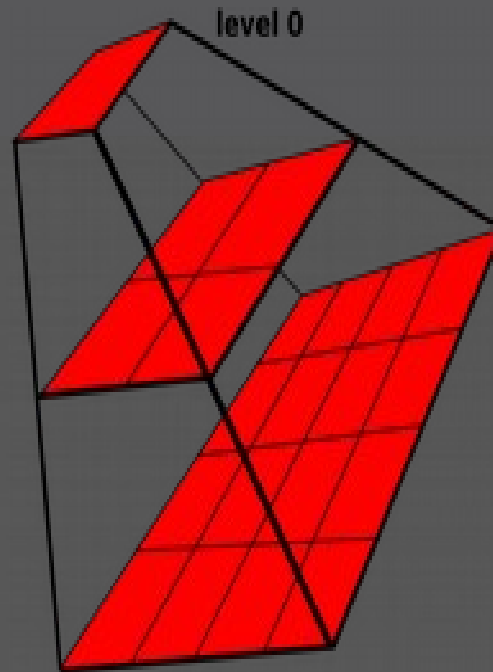
# MVT Tilesets

Layer 1

Layer 2

...

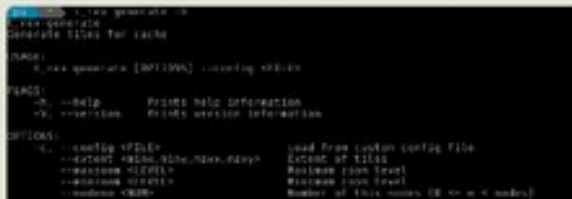
Tileset





## Serve vector tiles

- ★ Live tiles from PostGIS geodata
- ★ Zero-configuration mode
- ★ Embedded webserver
- ★ Visual styling with Maputnik



## Generate vector tiles

- ★ Tile generation command with simple parallelization



# Serving tiles with t-rex

## › Installation:

› <http://t-rex.tileserver.ch/>

## › Start server:

```
t_rex serve --dbconn postgresql://user@host/database
```

or

```
t_rex serve --datasource <file_or_gdal_ds>
```





# Generate configuration

## › Generate a configuration template:

```
t_rex genconfig --dbconn postgresql://user@host/database
```

## › Run with configuration file:

```
t_rex serve --config myconfig.cfg
```



# Built-in viewer

The screenshot shows a web browser window titled "T-Rex Vector Tile Viewer" with the URL "127.0.0.1:6767/#". The interface includes a sidebar on the left with a list of tile sets, a top navigation bar with tabs for "Info", "Mapbox GL", "OpenLayers", "X-Ray", and "Inspector", and a main content area. The "Info" tab is active, displaying details for the "admin\_0\_countries" tileset.

**Tile sets:** t-rex

- admin\_0\_countries
  - admin\_0\_countries
- ne\_10m\_populated\_places
  - ne\_10m\_populated\_places
- ne\_10m\_populated\_places\_wgs84
  - ne\_10m\_populated\_places\_wgs84
- ne\_10m\_rivers\_lake\_centerlines
  - ne\_10m\_rivers\_lake\_centerlines
- ne\_110m\_admin\_0\_countries
  - ne\_110m\_admin\_0\_countries
- rivers\_lake\_centerlines
  - rivers\_lake\_centerlines

**Tileset: admin\_0\_countries**

**Layers:**

- admin\_0\_countries (POLYGON)

**Endpoints:**

- Tiles: [http://127.0.0.1:6767/admin\\_0\\_countries/{z}/{x}/{y}.pbf](http://127.0.0.1:6767/admin_0_countries/{z}/{x}/{y}.pbf)
- Style JSON: [http://127.0.0.1:6767/admin\\_0\\_countries.style.json](http://127.0.0.1:6767/admin_0_countries.style.json)
- TileJSON: [http://127.0.0.1:6767/admin\\_0\\_countries.json](http://127.0.0.1:6767/admin_0_countries.json)
- Style map with [Maputnik](#)

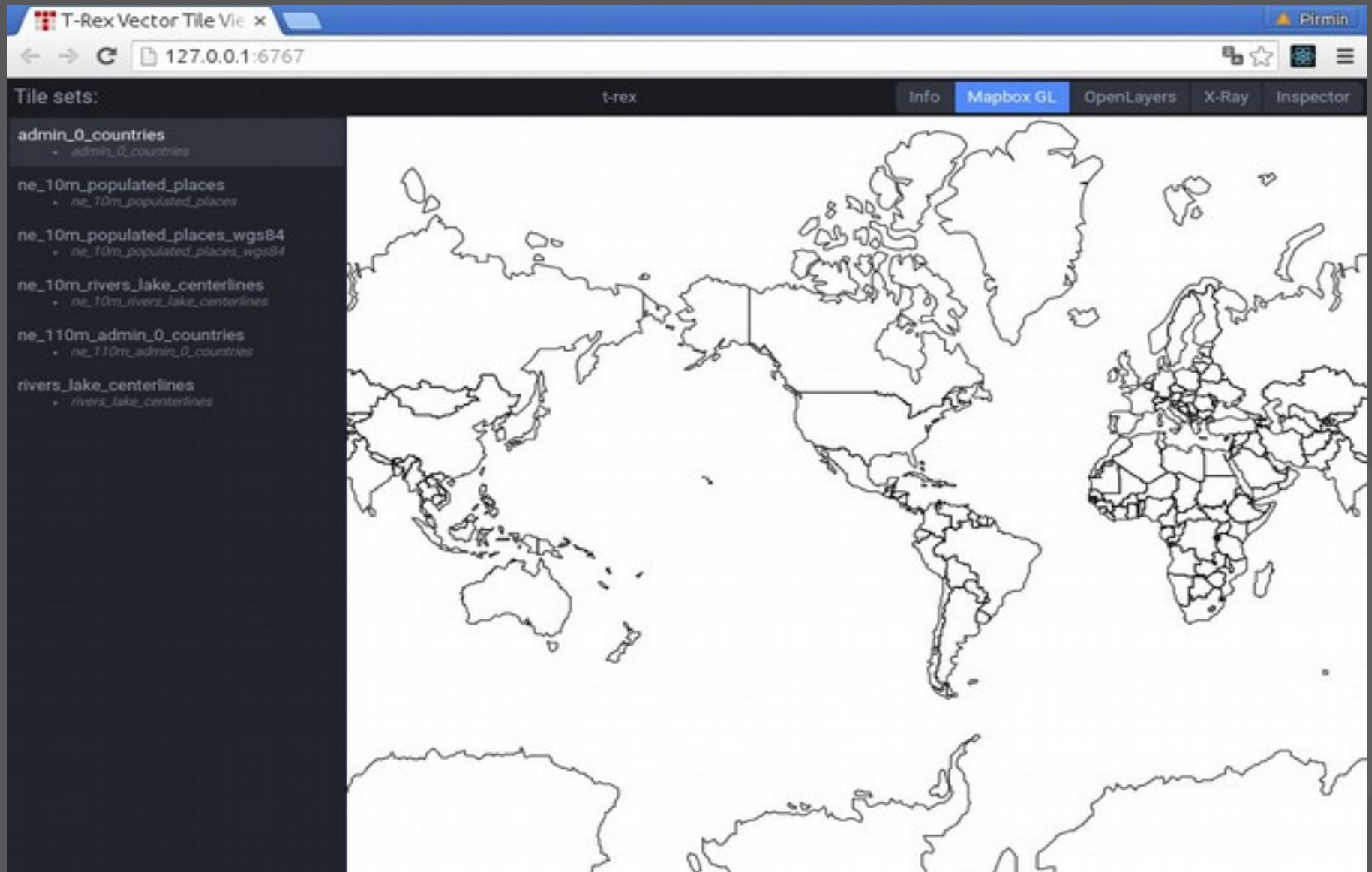
**Snippets:**

- [MapBox GL JS](#)
- [OpenLayers](#)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset='utf-8' />
    <title></title>
    <meta name='viewport' content='initial-scale=1,maximum-scale=1,user-scalable=no' />
    <script src='https://api.tiles.mapbox.com/mapbox-gl-js/v0.38.0/mapbox-gl.js'></script>
    <link href='https://api.tiles.mapbox.com/mapbox-gl-js/v0.38.0/mapbox-gl.css' rel='stylesheet'>
    <style>
      body { margin:0; padding:0; }
      #map { position:absolute; top:0; bottom:0; width:100%; }
    </style>
  </head>
```



# Mapbox GL JSON viewer





# OpenLayers viewer

The screenshot shows a web browser window titled "T-Rex Vector Tile Vie x" with the address bar displaying "127.0.0.1:6767". The browser's address bar includes navigation icons (back, forward, refresh) and a search icon. The page content is divided into a left sidebar and a main map area. The sidebar, titled "Tile sets:", lists several vector tile sets with expandable sub-items:

- admin\_0\_countries
  - admin\_0\_countries
- ne\_10m\_populated\_places
  - ne\_10m\_populated\_places
- ne\_10m\_populated\_places\_wgs84
  - ne\_10m\_populated\_places\_wgs84
- ne\_10m\_rivers\_lake\_centerlines
  - ne\_10m\_rivers\_lake\_centerlines
- ne\_110m\_admin\_0\_countries
  - ne\_110m\_admin\_0\_countries
- rivers\_lake\_centerlines
  - rivers\_lake\_centerlines

The main map area displays a world map with blue outlines for landmasses and coastlines. The map is titled "t-rex" and has a navigation toolbar with a "+" button for zooming in and a "-" button for zooming out. The top right of the map area features a navigation menu with tabs for "Info", "Mapbox GL", "OpenLayers" (which is selected), "X-Ray", and "Inspector". The user's name "Pirmin" is visible in the top right corner of the browser window.



# X-Ray viewer

The screenshot shows a web browser window titled "T-Rex Vector Tile Vie x" with the URL "127.0.0.1:6767". The browser's address bar and navigation icons are visible. The application interface includes a top navigation bar with tabs for "Info", "Mapbox GL", "OpenLayers", "X-Ray" (selected), and "Inspector". On the left, a "Tile sets:" panel lists several layers, each with a plus sign icon:

- admin\_0\_countries
  - admin\_0\_countries
- ne\_10m\_populated\_places
  - ne\_10m\_populated\_places
- ne\_10m\_populated\_places\_wgs84
  - ne\_10m\_populated\_places\_wgs84
- ne\_10m\_rivers\_lake\_centerlines
  - ne\_10m\_rivers\_lake\_centerlines
- ne\_110m\_admin\_0\_countries
  - ne\_110m\_admin\_0\_countries
- rivers\_lake\_centerlines
  - rivers\_lake\_centerlines

The main map area displays a world map with a dark background and light blue vector lines representing geographical features. A mouse cursor is positioned over the United States. A tooltip at the bottom of the map displays the following information:

```
fid: 169
name: United States
iso_a3: USA
layer: admin_0_countries
```



# Tile inspector

The screenshot shows the T-Rex Vector Tile Viewer interface. The browser title is "T-Rex Vector Tile Vie x" and the address bar shows "127.0.0.1:6767". The interface includes a "Tile sets:" panel on the left with a list of layers: "admin\_0\_countries", "ne\_10m\_populated\_places", "ne\_10m\_populated\_places\_wgs84", "ne\_10m\_rivers\_lake\_centerlines", "ne\_110m\_admin\_0\_countries", and "rivers\_lake\_centerlines". The main map area displays a dark-themed map of Europe and Africa with a grid. Two tiles are highlighted with labels "3,1,-3" and "3,2,-3". A "Tile inspector" panel is open in the bottom right corner, displaying the following information:

```
Tile: 3/2/2
PBF size: 1343 B (1.31 kB)
Features: 10
Layers: 1
#admin_0_countries (10 features)
  • [fid]
  • [name]
  • [iso_a3]
  • No class
```



# Styling GUI: Maputnik

The screenshot displays the Maputnik v1.4.0 interface. The top navigation bar includes 'Open', 'Export', 'Data Sources', 'Style Settings', 'Inspect Mode', 'Help', and a 'Take the Maputnik Survey' button. The left sidebar shows a list of layers, with 'landuse\_residential' selected. The main panel is divided into several sections:

- Layer:** landuse\_residential
- Type:** Fill
- Source:** openmaptiles
- Source Layer:** landuse
- Min Zoom:** 0 (with a note: "The minimum zoom level for the layer. At zoom levels less than the minzoom, the layer will be hidden.")
- Filter:** Compound Filter: every filter matches. Includes filters for 'type == Polygon' and 'class == residential'.
- Paint properties:** Opacity 8 (0.8) and Opacity 9 (0.6). Color: rgb(234, 234, 230).

The map on the right shows a vector-style map of Africa and the Middle East, with residential areas highlighted in a light gray color. The zoom level is 2.84. The Mapbox logo is visible in the bottom left corner of the map area.



# Übung

- **1. Download von Übungsdaten & Import in PostGIS**
- **2. Tileserver (t-rex) starten**
  - On-the-fly Generierung von Tiles
- **3. Tiles in eingebauten Viewern anschauen**
- **4. Mapbox GL JS Viewer erstellen**



## › Mapbox Styles (JSON)

### › Viewer:

- › Mapbox GL JS
- › Leaflet (mapbox-gl-leaflet plugin)
- › OpenLayers (ol-mapbox-style)

### › Style Editor (OSS)

- › Maputnik

## › Generic vector styling

### › Viewer:

- › OpenLayers
- › Leaflet (VectorGrid plugin)
- › HTML Canvas
- › Custom projections, etc.

# Styling GUI: Mapbox Studio (non-OSS)

The screenshot displays the Mapbox Studio styling interface for a 'Basic Template' map. The left sidebar lists 25 layers, with 'admin-state-province' selected. The main panel shows styling options for this layer, including a color picker, a zoom range slider, and a 'Rate of change stepped' control. The right sidebar contains map controls like 'Publish', 'Share', and 'Map position'. The map itself shows a view of Paris with the Seine river highlighted in blue.

**Basic Template**

25 layers

- country-label
- state-label
- place-citylabel-major
- place-citylabel-minor
- place-town-village-hamlet-lab...
- place-neighborhood-suburb-l...
- airport-label
- poi-label
- road-label
- admin-country-disputed
- admin-country
- admin-state-province**
- bridge
- bridge-case
- road
- tunnel
- pedestrian-path
- building
- aeroway-line
- aeroway-polygon
- water
- waterway
- landuse
- national\_park
- background

**admin-state-province** Style Select data

Color

Pattern none

Opacity 1

Width 1.5 px

Cap

Join

Round limit 1.65

Miter limit 2

Dash Array

Gap width 0 px

Blur 0 px

Offset 0 px

Translate 0, 0

Translate anchor

Line color Zoom range

Rate of change stepped Edit

Zoom 0 hsl(0, 0%, 80%) Edit

Zoom 4 hsl(0, 0%, 65%) Edit

+ Add another stop

Reset value to default

Apply existing value

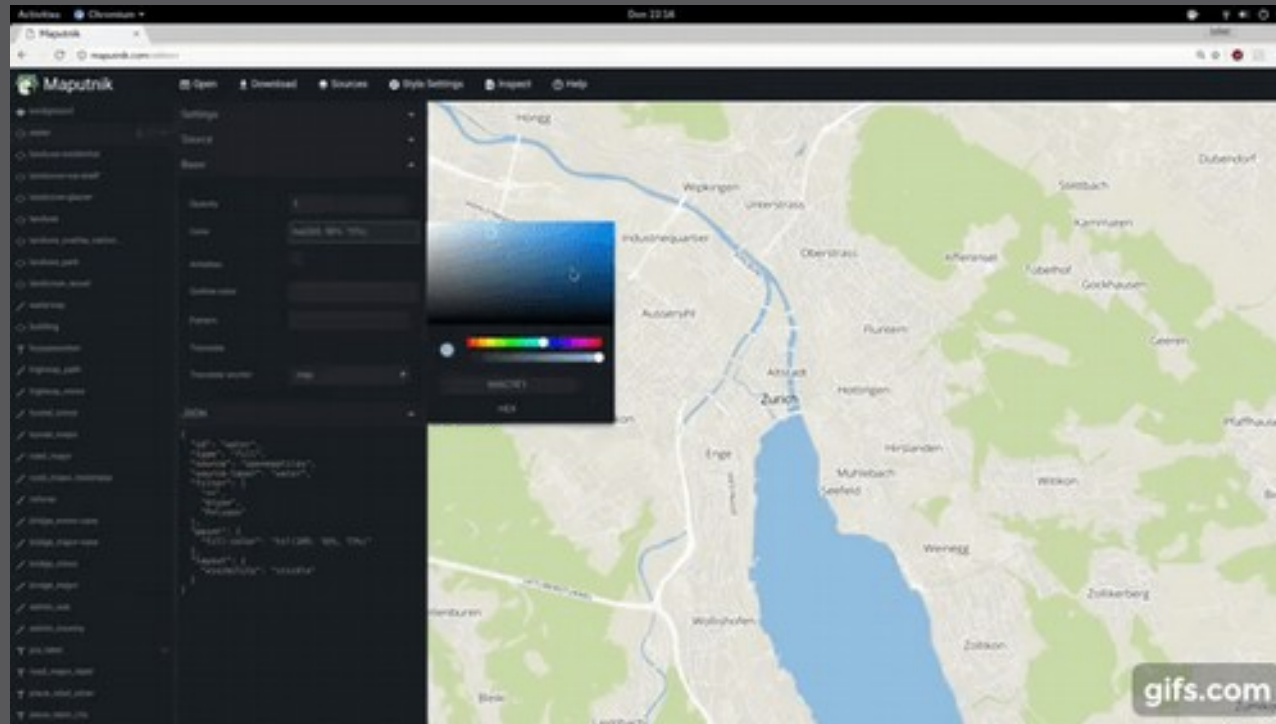
Details

Map controls: Publish, Share, Search, Map position, Debug, Style reference, Compatibility, History, Light, Fonts, Images, Help

Map coordinates: 12 2.318,48.867



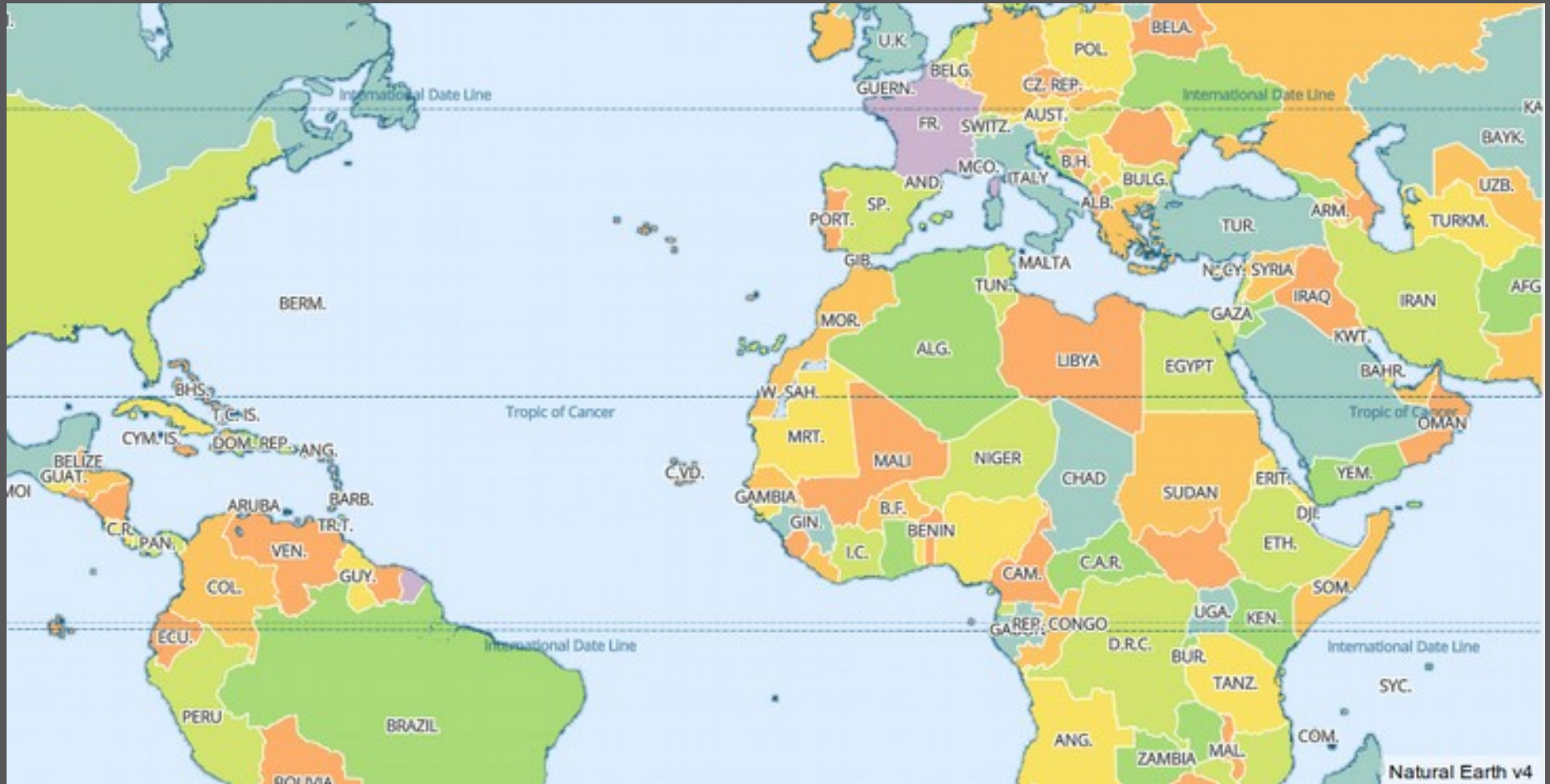
# Styling GUI: Maputnik



- <https://github.com/maputnik/editor>
- Kickstarter financed OSS editor by Lukas Martinelli
- Integrated in t-rex backend



# Übung Styling mit Maputnik





# Mapbox GL JS

<https://www.mapbox.com/mapbox-gl-js/>

```
var map = new mapboxgl.Map({  
  container: 'map',  
  style: 'http://tileserver.ch/map.style.json',  
  center: [-74.50, 40],  
  zoom: 9  
});
```



# Mapbox Styles

<https://www.mapbox.com/mapbox-gl-js/style-spec>

## › JSON with data sources and map style

```
{  
  "version": 8,  
  "name": "Mapbox Streets",  
  "sprite": "mapbox://sprites/mapbox/streets-v8",  
  "glyphs": "mapbox://fonts/mapbox/{fontstack}/{range}.pbf",  
  "sources": {...},  
  "layers": [...]  
}
```



# Data sources

## ➤ TileJSON endpoint

```
"source-name": {  
  "type": "vector",  
  "url": "http://tileserver.ch/tilejson.json"  
}
```

## ➤ Tile URL

```
"countries": {  
  "type": "vector",  
  "tiles": ["http://tileserver.ch/countries/{z}/{x}/{y}.pbf"]  
}
```

## ➤ Maxzoom

Maximum zoom level for which tiles are available. Data from tiles at the maxzoom are used when displaying the map at higher zoom levels.

<https://www.mapbox.com/mapbox-gl-js/style-spec/#sources>



# Other data sources

- › **GeoJSON**
- › **Raster**
  - › Raster tiles (XYZ, TMS)
  - › Tiled WMS
- › **Image**
- › **Video**
- › **Canvas**





# Sprites & Fonts

## › Sprites

- › Image sprites used for patterns and icons

```
"sprite": "http://openmaptiles.org/sprites/"
```

## › Fonts

- › Font glyphs (Protobuf format)

```
"glyphs":  
"http://fonts.openmaptiles.org/{fontstack}/{range}.pbf"
```

**More info:**

**<https://openmaptiles.org/docs/style/mapbox-gl-style-spec/>**



## › Layers with styles

```
"layers": [  
  {  
    "id": "water",  
    "source": "mapbox-streets",  
    "source-layer": "water",  
    "type": "fill",  
    "paint": {  
      "fill-color": "#00ffff"  
    }  
  }  
]
```

- › Layers drawn in the order of this array
- › Layout + Paint properties



# Layer rendering types

- › **"fill": Filled polygon with an optional stroked border**
  - › **"line": Stroked line**
  - › **"symbol": Icon or text label**
  - › **"circle": Filled circle**
  - › **"heatmap": Heatmap**
  - › **"fill-extrusion": Extruded (3D) polygon**
  - › **"raster": Raster map textures such as satellite imagery**
  - › **"background": Background color or pattern of the map.**
- GeoJSON**



## ➤ Attribute based styling with filter

```
{
  "id": "population_lt_2m",
  "type": "fill",
  "source": "states",
  "filter": ["<=", "PERSONS", 2000000],
  "paint": {
    "fill-color": "#A6CEE3",
    "fill-opacity": 0.7
  }
}
```

## ➤ Update with Javascript based on user input



## ➤ Resolution based styling

```
{
  "id": "state_abbreviations",
  "type": "symbol",
  "source": "states",
  "minzoom": 1.8,
  "maxzoom": 4,
  "layout": {
    "text-field": "{STATE_ABBR}",
    "text-size": 12,
    "text-font": ["Open Sans Regular"]
  }
}
```



# Functions

➤ The value for any layout or paint property may be specified as a function

➤ Zoom functions

```
{
  "circle-radius": {
    "stops": [
      // zoom is 5 -> circle radius will be 1px
      [5, 1],
      // zoom is 10 -> circle radius will be 2px
      [10, 2]
    ]
  }
}
```



# Functions

## › Property functions

```
{  
  "circle-color": {  
    "property": "temperature",  
    "stops": [  
      // "temperature" is 0    -> circle color will be blue  
      [0, "blue"],  
      // "temperature" is 100 -> circle color will be red  
      [100, "red"]  
    ]  
  }  
}
```

## › Zoom-and-property functions

<https://www.mapbox.com/mapbox-gl-js/style-spec/#other-function>



## ➤ Expressions (MB GL 0.41.0)

```
"circle-radius": [  
  "sqrt", ["/", ["get", "population"], 10000]  
]
```





# Übung

- › **Style für Länder mit Maputnik erstellen**
- › **Style in Viewer einbinden**



# Übung

- › **Punkt-Layer für Labels hinzufügen**
- › **Label-Style erstellen**



<http://openlayers.org/>

- › **Support for WMTS, WFS, etc.**
- › **Support for Non-Mercator CRS**
- › **Conversion of GL-Styles to OL-Styles**
  - › <https://github.com/openlayers/ol-mapbox-style/>
  - › Not all features supported
- › **Uses browser canvas-API**



# Übung OpenLayers

## › Integration Style in OpenLayers Viewer



# Hosting vector tiles

- › **Static MVT files**
  - › Webserver (Apache, Nginx)
  - › Service: S3, etc.
- › **MIME type 'application/x-protobuf'**
- › **Content-Encoding 'gzip'**
- › **Cache headers**
- › **CORS headers**
- › **Layer concatenation**
- › **Live data**
  - › Proxy in front of tile server (Varnish, etc.)
  - › NodeJS based server



# Offline / Mobile

- › **Offline: Webapplikation ohne Internet-Zugriff**
- › **Daten in MBTiles-File (SQLite)**
- › **Serve PBFs from MBTiles**
  - › TileServer-PHP
  - › TileServer-GL-Light (node.js)
- › **Standalone Applikation mit Mapbox Mobile**
  - › Kompatibler C++ Renderer Desktop/Mobile



# Tile cache seeding

## › Generate tile cache:

```
t_rex generate --config countries.toml --maxzoom 4
```

## › Create MBTiles File:

```
mb-util --image_format=pbfs tiles/ne_countries ne.mbtiles
```



# Übung Seeding

- › In Konfiguration Cache aktivieren
- › Tile-Cache generieren



# Vektor Tile Server für Custom-Data (PG)

- › node-mapnik (Kartotherian, tesseract)
- › Tilezen tileserver
- › Tegola
- › t-rex
- › GeoServer
- › PostGIS ST\_AsMVT

<https://github.com/mapbox/awesome-vector-tiles>

<https://github.com/pka/mvt-benchmark>



# Danke!



Pirmin Kalberer  
pka@sourcepole.ch  
@implgeo