MapOSMatic remote rendering API

Hartmut Holzgraefe

Version 0.1, April 14th 2018

Table of Contents

Work in progress

Paper Formats

Request URL

```
http://localhost:8000/apis/paper_formats
```

Example result

```json
{
    "Best fit": {
        "height": null,
        "width": null
    },
    "Din A4": {
        "height": 297,
        "width": 210
    },
    "US letter": {
        "height": 279,
        "width": 216
    }
}
```

Layouts

Request URL

```
http://localhost:8000/apis/layouts
```

Example result

```
{
    "multi_page": {
        "description": "A multi-page layout.",
        "preview_url": "http://example.org/media/img/layout/multi_page.png"
    },
    "plain": {
        "description": "Full-page layout without index.",
        "preview_url": "http://example.org/media/img/layout/plain.png"
    },
    "single_page_index_bottom": {
        "description": "Full-page layout with the index at the bottom.",
        "preview_url":
"http://example.org/media/img/layout/single_page_index_bottom.png"
    },
    "single_page_index_side": {
        "description": "Full-page layout with the index on the side.",
        "preview_url":
"http://example.org/media/img/layout/single_page_index_side.png"
    }
}
```

Base Styles

Request URL

```
http://example.org/apis/styles
```

Example output

```
{
    "CartoOSM": {
        "annotation": "OpenStreetMap Carto standard style",
        "description": "CartoCSS OSM standard style",
        "preview_url": "http://example.org/media/img/style/CartoOSM.png"
    },
    "GermanCartoOSM": {
        "annotation": "German OSM style based on OSM Carto",
        "description": "German OSM style",
        "preview_url": "http://example.org/media/img/style/GermanCartoOSM.png"
    }
}
```

Overlay Styles

Request URL

```
http://example.org/apis/overlays
```

Example output

```
{
    "OpenRailwayMap_Overlay": {
        "annotation": "OpenRailwayMap overlay",
        "description": "OpenRailwayMap rail line overlay",
        "preview_url": "http://example.org/media/img/style/OpenRailwayMap_Overlay.jpg"
    },
    "Scale_Bar_overlay": {
        "annotation": "",
        "description": "Map scale bar"
        "preview_url": "http://example.org/media/img/style/Scale_Bar_overlay.jpg"
    }
}
```

Job Stati

Request URL

```
http://example.org/apis/job_stati/
```

Example result

```
{
    "0": "Submitted",
    "1": "In progress",
    "2": "Done",
    "3": "Done w/o files",
    "4": "Cancelled"
}
```

Submitting a Job

Request URL

```
POST http://example.org/apis/jobs/
```

Expects render job named parameters as a JSON collection.

Possible parameters:

osmid

ID of an object in the osm2pgsql

bbox_top
bbox_bottom
bbox_left
bbox_right

Render area min. bounding box, may be extended to fit paper format

title

Text to put into the title box (default: "Untitled API Request")

language

locale to use for rendering (default: "en_US.UTF-8")

layout

Map layout to use (default: "plain")

style

Map base style to use (default: "CartoOSM")

overlays

List of overlays to put on top of base style (default: None)

paper_size

Paper size to use (default: "Din A4" 210x297mm²)

orientation

Paper orientation: "portrait" or "landscape" (default: "portrait")

track_url

URL pointing to a GPX track to download and print

umap_url

URL pointing to an Umap export file do download and print

It is also possible to upload GPX and Umap files directly using a multipart/form-data POST request. In this case you have to submit the actual API JSON as a post parameter named "json", and add file fields named "track" for a GPX track, or "umap" for an Umap JSON export.

After successfully submitting a request you can either poll the job result API every once in a while until it either returns the "200 OK" status, or a 4xx error code. Or you can redirect a web client to the human readable result page listed in the returned "interactive" job parameter.

Return codes

202

> *Accepted* when successfully submitted, "Location:" header will contain result URL

400

> *Bad request* on request validation errors

Example requests

```
Content-type: application/json

{
    "title": "API Test",
    "bbox_bottom": 52.00,
    "bbox_left": 8.52,
    "bbox_right": 8.50,
    "bbox_top": 52.02
}
```

```
Content-Type: multipart/form-data; boundary=--XXXXXXXX

--XXXXXXXX
Content-Disposition: form-data; name="json"

{
    "title": "GPX Test",
    "paper_size": "Din A1"
}
--XXXXXXXX
Content-Disposition: form-data; name="track"; filename="test.gpx"
Content-Type: application/gpx+xml

<?xml version="1.0" encoding="UTF-8"?>

<gpx version="1.1">
  <metadata>
    <name>GPX test</name>
  </metadata>
  <trk>
    <trkseg>
      <trkpt lat="52.00" lon="8.50" />
      <trkpt lat="52.02" lon="8.52" />
    </trkseg>
  </trk>
</gpx>
--XXXXXXXX--
```

Example results

```
HTTP/1.1 202 Accepted
[...]
Location: /api/jobs/36
Content-Type: text/json

{
    "bbox_bottom": 52.01,
    "bbox_left": 8.52,
    "bbox_right": 8.50,
    "bbox_top": 52.02,
    "id": 36,
    "interactive": "http://example.org/maps/214"
    "language": "en_US.UTF-8",
    "layout": "plain",
    "paper_height_mm": 297,
    "paper_width_mm": 210,
    "status": 0,
    "styles": "CartoOSM"
    "title": "API Test",
}
```

```
HTTP/1.1 400 Bad Request
[...]
Content-Type: text/json

{
    "error": {
        "non_field_errors": [
            "No bounding box area or OsmID given"
        ]
    }
}
```

Checking Job results

Request Url

```
http://example.org/apis/jobs/
```

Return codes

200

OK when request has been processed successfully

202

> *Accepted* when rendering is still in progress, "Location:" header will contain same URL again

410

> *Gone* when a request had been processed successfuly in the past, but result files have since been purged to free file system space

400

> *Bad request* when a request failed to render

Example result

```
HTTP/1.1 200 OK
[...]
Content-type: application/json

{
    "bbox_bottom": 52.01,
    "bbox_left": 8.53,
    "bbox_right": 8.49,
    "bbox_top": 52.02,
    "files":
    {
        "8bit.png": "http://example.org/results//000215_2018-09-12_23-14_GPX-
test.8bit.png",
        "jpg": "http://example.org/results//000215_2018-09-12_23-14_GPX-test.jpg",
        "pdf": "http://example.org/results//000215_2018-09-12_23-14_GPX-test.pdf",
        "png": "http://example.org/results//000215_2018-09-12_23-14_GPX-test.png",
        "svgz": "http://example.org/results//000215_2018-09-12_23-14_GPX-test.svgz"
    }
    "id": 215,
    "language": "en_US.UTF-8",
    "layout": "plain",
    "paper_height_mm": 594,
    "paper_width_mm": 841,
    "queue_size": 0,
    "status": 2,
    "style": "CartoOSM",
    "title": "GPX test"
}
```

Example code

Javascript

PHP

Python