

# Republishing OpenStreetMap's roads as Linked Routable Tiles

Pieter Colpaert, Ben Abelshausen, Julián Andrés Rojas Meléndez, Harm Delva,  
Ruben Verborgh

IDLab, Department of Electronics and Information Systems, Ghent University – imec

**Abstract.** Route planning providers manually integrate different geo-spatial datasets before offering a Web service to developers, thus creating a closed world view. In contrast, combining open datasets at runtime can provide more information for user-specific route planning needs. For example, an extra dataset of bike sharing availabilities may provide more relevant information to the occasional cyclist. A strategy for automating the adoption of open geo-spatial datasets is needed to allow an ecosystem of route planners able to answer more specific and complex queries. This raises new challenges such as (i) how open geo-spatial datasets should be published on the Web to raise interoperability, and (ii) how route planners can discover and integrate relevant data for a certain query on the fly. We republished OpenStreetMap's road network as "Routable Tiles" to facilitate its integration into open route planners. To achieve this, we use a Linked Data strategy and follow an approach similar to vector tiles. In a demo, we show how client-side code can automatically discover tiles and perform a shortest path algorithm. We provide four contributions: (i) we launched an open geo-spatial dataset that is available for everyone to reuse at no cost, (ii) we published a Linked Data version of the OpenStreetMap ontology, (iii) we introduced a hypermedia specification for vector tiles that extends the Hydra ontology, and (iv) we released the mapping scripts, demo and routing scripts as open source software.

Read the HTML-version: <http://pieter.pm/demo-paper-routable-tiles/>

## 1. Introduction

When setting up a Web-based journey planner that takes into account individual user needs, developers are confronted with the daunting prospect of integrating heterogeneous datasets. Industry players provide an alternative solution to these complexities, by providing services on top of their datasets that are centralized in one location. They are able to provide a solution for the ~80% of the world-wide needs and build a business around that. When trying to cater to the remaining 20%, industrial route planners are quickly confronted with diminishing returns of integrating more datasets. Take for example use cases of (i) people owning a foldable bike and the ability to take their bike on public transport, (ii) companies trying to find an optimal delivery route for their delivery service where some vehicles cannot pass through low emission zones, (iii) routes based on the real-time state of traffic control systems and probability to hit a green light, or (iv) people with special constraints or disabilities. For each of these four use cases, extra datasets are needed to calculate

these end-user specific routing graphs. Such datasets are published by different organizations that often publish them openly, thanks to strategic goals or legal mandates. Every route planner will somehow need to compile their own sources over which they can execute their own route planning algorithm.

In this paper, we aim to automate data adoption in route planners. As a first step, we introduce *Routable Tiles*. This is a hypermedia specification for geospatial road networks. In this specification, we republished all the roads in OpenStreetMap. In the next section we will see that the ideas behind Routable Tiles itself is not novel. The contribution of this paper lies in applying the geospatial indexing idea from the database world to Web APIs, introducing an ontology for describing geospatial hypermedia controls, and launching this world-wide dataset as a resource to the Semantic Web community.

## 2. Related work

### 2.1. Geospatial data on the Web

Spatial Data on the Web [1] is a W3C and OGC collaboration to create a list of best practices for spatial data on the Web. It takes a strong position in favor of HTTP URIs to identify resources. The rationale is that Linked Data principles such as the use of HTTP URIs as global identifiers, raises the interoperability of geo-spatial datasets by providing a common set of semantics that can be reused by data publishers.

Slippy maps are maps often included in web-pages on which you can pan around. In order to reduce server load, the client is preconfigured with a URL template of the web-server containing image tiles. When the map is loaded, the client can calculate all URLs necessary. Vector tiles reuse this idea to, instead of raster images, publish the raw data behind the tiles. The client can then render the maps on the client-side. This gives clear benefits over raster images: (i) the styling of the maps can be done by UI developers that can use CSS and scripts to style the road elements, (ii) vector tiles can be smaller in size as vectors are typically much smaller than a rendered bitmap, and (iii) it allows for all elements on the map to become interactive. Existing implementations include the Mapbox vector tiles and Open Map Tiles. Each have their own specificities and schemas (see <https://docs.mapbox.com/vector-tiles/mapbox-streets-v8/> and <https://openmaptiles.org/schema/>) with a strong focus on rendering maps.

Valhalla by Mapzen and now hosted by the Linux Foundation is the first project that implements the idea of vector tiles for route planners in an open-source project. The technology proposes a tiling specification for storing routing information on disk. Tiling the data makes sure the server can be selective about the data that needs to be loaded into memory in order to execute an individual request. This tiling specification in Valhalla is however not used as an exchange format – although offline routing is an upcoming feature – and interoperability with other datasets is not a focus.

Linked Geo Data [2] is an initiative that maps OpenStreetMap data to Linked Data. It releases data dumps, subject pages and a SPARQL endpoint. Furthermore, it has their own mappings from the OpenStreetMap data model to a Linked Geo Data

ontology.

## 2.2. Open Data publishing

Open Data can be published in various interfaces. In order to be able to query these interfaces, Comunica [3] was built. It is a Linked Data user agent that can run federated queries over several heterogeneous Web APIs, such as data dumps, SPARQL endpoints, Linked Data documents and Triple Pattern Fragments [4]. This engine has been developed to make it easy to plug in specific types of functionality as separate modules. Such modules can be added or removed depending on the configuration. As such, by looking for affordances in Web APIs, more intelligent user agents can be created. Preconditions for an engine like Comunica to work with a dataset is: supporting a Linked Data representation and allowing cross origin resource sharing headers in the HTTP responses. The better the data is split in fragments, the better the caching will be able to provide a faster user-perceived performance.

For public transport systems, instead of publishing a dump of time schedules or a full fledged route planner, Linked Connections [5] proposes a publishing mechanism that gives clients access to the data in time fragments. It uses departure-arrival pairs from a station to another (a connection), and orders these connections by departure time. It then fragments this dataset in documents that can be published over HTTP. Links in the responses ensure a client can always find more information to take into account.

## 3. Implementation

Routable Tiles is a JSON-LD specification for which the working draft can be found at <https://openplanner.team/specs/2018-11-routable-tiles.html>. It has three main aspects: (i) it introduces a *hypermedia specification* reusing Hydra Collections for describing a tile server, (ii) a way to describe *OpenStreetMap's nodes, ways and relations*; and (iii) it introduces a mapping of the OpenStreetMap basic terms to an RDFS vocabulary.

The Linked Geo Data vocabulary has been unavailable since 2018 and not updated since 2015. Therefore we introduced our own vocabulary, that nonetheless takes a different approach. Instead of mapping everything, we decided to map only the bare minimum needed specifically for the use case of route planning. Therefore, we keep the ontology as close as possible to the actual OSM data model. We added links to the appropriate Linked Geo Data classes (ontology has yet to be published, awaiting a third party implementation).

We define 3 main classes: *osm:Way*, *osm:Relation* and *osm:Node*. The *osm:members* property describes the members of the relation and the *osm:role* their function in the relation. *osm:restriction* is used to model turn restrictions. The property *osm:nodes* is used to link to an *rdf:List* of *osm:Node* items. In one page, multiple lists can be described. If a Way crosses a tile, the other tile also mentioned the border Node in one of its *rdf:Lists*. The property *osm:members* is used to link to an *rdf:List* of *osm:Member* items.

The mapping scripts and server interface can be found at <https://github.com/openplannerteam/routeable-tiles>. Every tile in OpenStreetMap on zoom level 14 is on the fly made available as JSON-LD. Special attention was given to the HTTP server to include a server-side cache and to compress the HTTP response with gzip. Furthermore, it sets both an etag header and a cache-control max-age header for client-side cache control. Finally, it also allows webpages on other domains to request its data by setting the appropriate Cross Origin Resource Sharing headers.

For the URIs of Ways and Nodes, we decided to reuse the subject pages provided by the OpenStreetMap project itself. We hope that at some point, OpenStreetMap decides to support a Linked Data representation on these URIs. For example: [openstreetmap.org/node/366934331](https://openstreetmap.org/node/366934331) and [openstreetmap.org/way/242536619](https://openstreetmap.org/way/242536619).

#### **4. Demonstrator**

A server instance is set up publishing data for *the entire world*. Entry points into the hypermedia API can be found through [tiles.openplanner.team/planet/14/{x}/{y}/](https://tiles.openplanner.team/planet/14/{x}/{y}/) (e.g., with x 8411 and y 5485). A live demonstrator using this data can be found at <https://openplannerteam.github.io/leaflet-routable-tiles/>.

The demonstrator shows a map of all roads (in Portorož by default) in the viewport. The map is drawn on the client-side based on the JSON-LD documents fragmented in tiles. When clicking 2 locations on the map, the same JSON-LD documents are used to perform a Dijkstra shortest path algorithm.

#### **5. Conclusion and future work**

This demo introduced a tiling mechanism for publishing road networks in Linked Data. Compared to a SPARQL endpoint approach or server answering any individual route planning request, the server costs in our approach are indisputably lower. Thanks to the hypermedia controls, the full dataset can be automatically discovered, and HTTP caching can be leveraged. Furthermore, developers of route planning application are given more flexibility as they are now in full control of the algorithm. Just like vector tiles styling, they can tailor the routing algorithm to their end-user needs in the browser. This also opens the door to use cases where multiple sources are queried at once.

Nevertheless, it still remains an open question if applying this kind of Linked Data model/HTTP URIs to the roads related data is the optimal approach. For example, when another source wants to do a statement about a part of a road, chances are low it is already available as a separate `osm:Way` instance, and thus the source data would have to be altered to split the original entity into two.

There might be a need in the future to support a binary format in order to reduce size. Therefore, we will benchmark the difference in size and performance between Valhalla tiles (protobuf) and Routable Tiles (JSON-LD + gzip). By doing this, we should also look at other optimizations, such as applying summaries over different zoom levels. In order to achieve better adoption of the hypermedia controls, an actor

will be added to the Linked Data query agent called Comunica [3]. This way, we will help to solve geo-spatial queries over any data source by downloading only the right tiles even beyond road networks.

## References

1. Tandy, J., Barnaghi, P., van den Brink, L.: Spatial Data on the Web Best Practices. W3C, <https://www.w3.org/TR/sdw-bp/> (2017).
2. Stadler, C., Lehmann, J., Höffner, K., Auer, S.: LinkedGeoData: A Core for a Web of Spatial Open Data. *Semantic Web Journal*. 3, 333–354 (2012).
3. Taelman, R., Van Herwegen, J., Vander Sande, M., Verborgh, R.: Comunica: a Modular SPARQL Query Engine for the Web. In: Vrandečić, D., Bontcheva, K., Suárez-Figueroa, M.C., Presutti, V., Celino, I., Sabou, M., Kaffee, L.-A., and Simperl, E. (eds.) *Proceedings of the 17th International Semantic Web Conference*. pp. 239–255. Springer (2018).
4. Verborgh, R., Vander Sande, M., Hartig, O., Van Herwegen, J., De Vocht, L., De Meester, B., Haesendonck, G., Colpaert, P.: Triple Pattern Fragments: a Low-cost Knowledge Graph Interface for the Web. *Journal of Web Semantics*. 37–38, (2016).
5. Colpaert, P., Llaves, A., Verborgh, R., Corcho, O., Mannens, E., Van de Walle, R.: Intermodal public transit routing using Linked Connections. In: *Proceedings of the 14th International Semantic Web Conference: Posters and Demos* (2015).