

OGC API Features HTML-output as a Feature Dashboard

Lassi Lehto* and Jaakko Kähkönen

Finnish Geospatial Research Institute, lassi.lehto@nls.fi, jaakko.kahkonen@nls.fi

Keywords: Feature Dashboard, OGC API Features, HTML, Cross-border

Abstract:

A new generation of Open Geospatial Consortium's (OGC) service interface standards are being introduced and taken into use. The first one of these, the OGC API Features specifies a service interface for accessing geospatial features, i.e. representations of spatial objects with vector-formatted geometry component. The widely used default feature encoding language in the OGC API Features services is GeoJSON, a format for encoding simple feature properties and geometries in JavaScript Object Notation (JSON). However, the OGC API Features standard interestingly also defines HTML as a recommended format for feature encoding. This is consistent with the general OGC goal of making the services even more Web-friendly. With HTML-formatted feature representation as a recommended format for spatial features, it becomes feasible to process geodata in plain Web browser environment, without the help of any auxiliary mapping library. This also facilitates the Web search engines to index geospatial datasets.

The use of HTML as a feature encoding language opens up new, uncharted territories for the providers of geodata services. There are no well-defined principles or tradition about presenting geospatial features in HTML. The OGC API Features standard does not give any specific advice on the subject either. It has become a common approach to list the features as rows in a table, enabling paged browsing of the whole dataset. However, this easily becomes unpractical, in particular if there are many features involved and they are presented in an arbitrary order. In addition, there is often a map window present in the OGC API Features HTML-output, displaying the features of the current page. If the feature order is not spatially constrained, the map view will zoom inconsistently in and out for each individual page, making the feature viewing very unsatisfactory experience.

An EU-funded project, called Geospatially Enabled Ecosystem for Europe (GeoE3), has commenced with a goal to develop geospatial services according to modern interface standards, and to support user-oriented applications in the area of renewable energy, intelligent traffic and smart cities. One of the issues identified in the GeoE3 project is the poor user experience of browsing the HTML-output of an OGC API Features service. The project has taken a new approach that deviates from the usual paged table browsing approach. The approach could be called a feature dashboard – a collection of data items presented on a Web page in HTML for one individual feature. The selection of data items included in the dashboard depends essentially on the use case in question. A service specialising in a certain use domain could possibly introduce new encoding categories, like `html-re` (HTML for renewable energy), `html-it` (HTML for intelligent traffic) and `html-sc` (HTML for smart cities).

The fundamental approach selected for feature browsing in the GeoE3 project relies on the traditional map metaphor. Features are browsed in map view displaying some basic background map, and the actual features are shown only on the pre-defined large scale zoom levels. The extent of the map view is used as a bounding box for the corresponding feature query. By selecting individual features in the map view, the user can display the desired feature dashboard, presenting the properties of the selected feature in a use case-specific manner. More complicated dashboards naturally require the use of specialised JavaScript libraries, like OpenLayers for map-like visualisation or Ninja for 3D presentations. In addition to the map-based browsing, the service could also include a link for paged, table-formatted browsing to the HTML-output, to facilitate search engine indexing.

In the feature dashboard approach, it might also be possible and justified to access other services, using either location of the feature or its identifier as a link to additional information. These information items might be retrieved via service interfaces other than OGC API Features. One interesting possibility, to be investigated in the GeoE3 project, is to use the OGC-standardised Table Joining Service (TJS) for accessing statistics or other tabular data sources related to the feature in question. This way the feature dashboard could be seen as a general content integration platform for application-oriented presentation of geospatial features.

An example of the feature dashboard approach is the dashboard for buildings, developed in the GeoE3 project. As a preliminary example, the selected building is visualised with its 2D footprint on top of a background map. At the same time, a separate query is sent to a separate OGC API Coverages service to retrieve the Digital Surface Model (DSM) from the location of the building. The DSM is then clipped to the building polygon and visualised inside the building. This information might be used for example to evaluate the suitability of the building roof for installation of solar panels. If the DSM is analysed in a bit larger area, the amount of sun exposure in the rooftop can also be calculated. An example

visualisation of a building dashboard described above is presented in Figure 1. The example also includes an average wind speed value in the area, retrieved from a custom API, to be used in renewable energy-related applications.

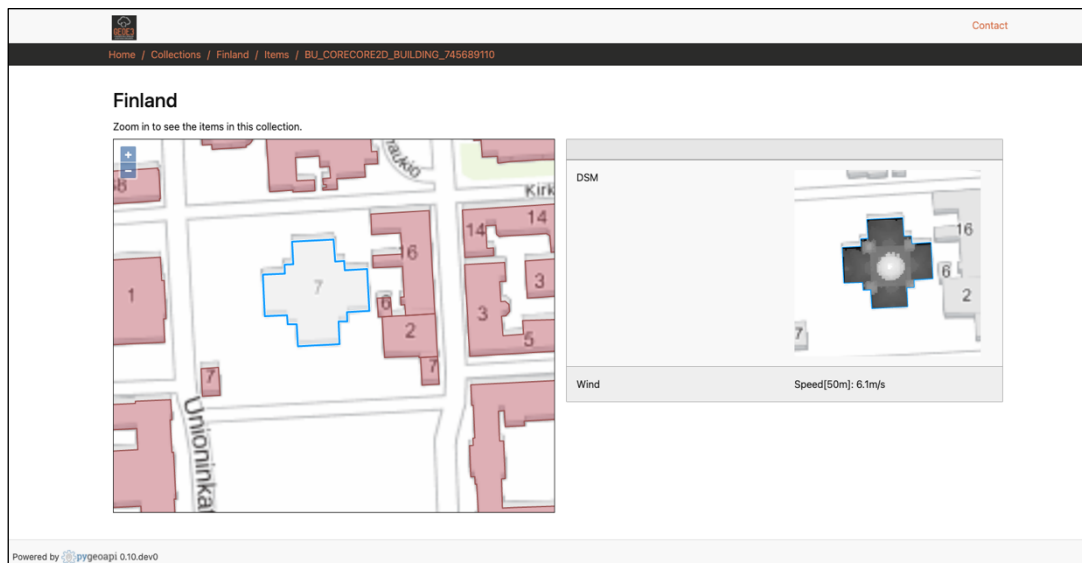


Figure 1. Building dashboard as an HTML presentation of an individual building (Helsinki Cathedral).

As an international, EU-funded project GeoE3 focusses specifically on the cross-border approaches in the geodata services and content delivery. OGC API Features -compliant services for buildings are already available for four participating national mapping agencies: Finland, Norway, The Netherlands and Spain. The services have been developed on top of the traditional national level Web Feature Services (WFS) by using the OGC API Features reference implementation, called pygeoapi, and its content provider plugin architecture. OGC API Coverages services have also been developed for three countries using the same approach. As a result, a similar building dashboard as presented in Figure 1, can also be presented e.g. on Norwegian data (see Figure 2).

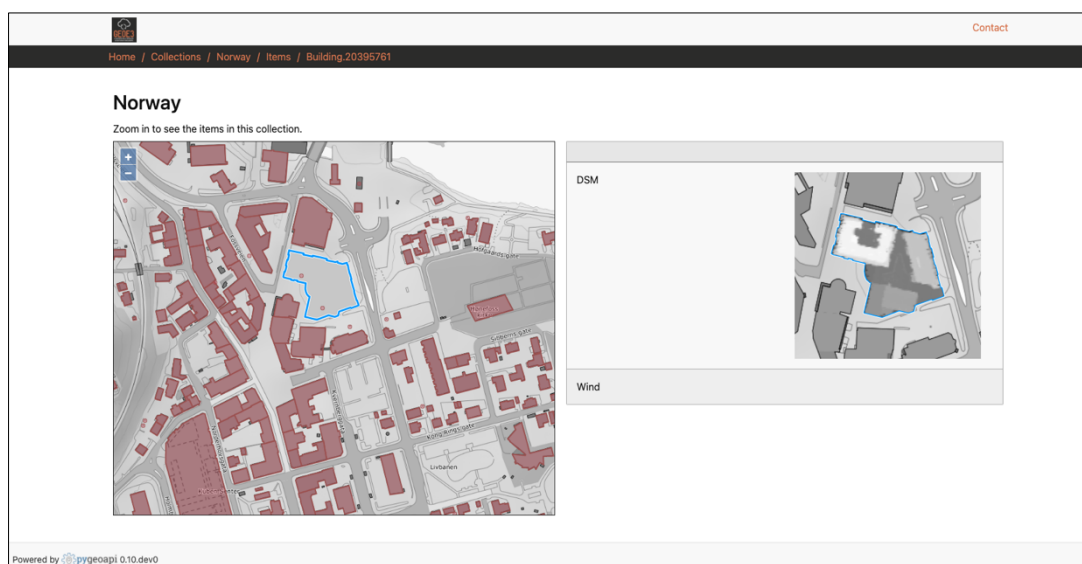


Figure 2. Building dashboard on Norwegian data.