Using an Existing Website as a Queryable Low-Cost LOD Publishing Interface

Brecht Van de Vyvere, Ruben Taelman, Pieter Colpaert, and Ruben Verborgh

IDLab (imec – Ghent University), Technologiepark-Zwijnaarde 122, 9052 Ghent, Belgium
(brecht.vandevyvere, ruben.taelman, pieter.colpaert, ruben.verborgh)@agent.be

Abstract. Maintaining an Open Dataset comes at an extra recurring cost when it is published in a dedicated Web interface. As there is not often a direct financial return from publishing a dataset publicly, these extra costs need to be minimized. Therefore we want to explore reusing existing infrastructure by enriching existing websites with Linked Data. In this demonstrator, we advised the data owner to annotate a digital heritage website with JSON-LD snippets, resulting in a dataset of more than three million triples that is now available and officially maintained. The website itself is paged, and thus hydra partial collection view controls were added in the snippets. We then extended the modular query engine Comunica to support following page controls and extracting data from HTML documents while querying. This way, a SPARQL or GraphQL query over multiple heterogeneous data sources can power automated data reuse. While the query performance on such an interface is visibly poor, it becomes easy to create composite data dumps. As a result of implementing these building blocks in Comunica, any paged collection and enriched HTML page now becomes queryable by the query engine. This enables heterogenous data interfaces to share functionality and become technically interoperable.

This is a print-version of a paper first written for the Web. The Web-version is available at https://brechtvdv.github.io/Article-Using-an-existing-website-as-a-queryable-low-cost-LOD-publishing-interface/.

Keywords: JSON-LD data snippets · Hypermedia web APIs · Intelligent agents · Digital humanities · Linked Open Data · Semantic web

1 Introduction

The Flemish Institute for Audiovisual Archiving (VIAA) is a non-profit organization that preserves petabytes of valuable image, audio or video files. These files and accompanying metadata are covered by distinct licenses, but some can be made accessible under an Open Data license. One initiative is opening up historical newspapers of the first World War with the open platform hetarchief.be. In 2017, the raw data of these newspapers have been published as a Linked Open Data [1]
(LOD) dataset using the low-cost Triple Pattern Fragments [2] (TPF) interface. Although this interface is still accessible¹, no updates from the website have been exported to the TPF interface due to absence of automatisation.

Maintaining an up-to-date LOD interface brings besides technical resources also an organizational challenge. Content editors often work in a separate environment such as a Content Management System (CMS) to update a website. The raw data gets exported from that system and published in a dedicated environment leaving the source of truth to the CMS. The question rises whether the data can be published closer to the authoritative source in a more sustainable way.

Website maintainers are currently using JSON-LD structured data snippets to attain better search result ranking and visualisation with search engines. These snippets are script tags inside a HTML webpage containing Linked Data in JSON format (JSON-LD) compliant with the Schema.org [3] datamodel. Not only search engine optimization (SEO), but also standard LOD publishing is possible. The data should be representative with the main content of the subject page, such as newspaper metadata, to be aligned with the structured data guidelines of search engines. In order that Linked Data user agents such as Comunica [4] can query a website as a dataset, the webpages should be linked together through hypermedia controls [5].

First we give a short background of the Comunica tool and the Hydra partial collection views. We then describe how hetarchief.be is enriched with JSON-LD snippets. Next, we explain how we allow Comunica to query over this and other sources by adding two building blocks. After this, we demonstrate how a custom data dump can be created by an end-user that wants to further analyze this data, for instance in spreadsheet software. The online version of this paper embeds this demo and can be tested live. Finally, we conclude the demonstrator with a discussion and perspective on future work.

2 Background

2.1 Comunica

Comunica [4] is a Linked Data user agent that can run federated queries over several heterogeneous Web APIs. This engine has been developed to make it easy to plug in specific types of functionality as separate modules.

A bus module is used as communication channel to solve one problem, e.g. to extract hypermedia controls from a Web API. Multiple actor modules can subscribe to a bus and extract one or more hypermedia controls according to their implementation. A mediator module wraps around the bus to select the most appropriate results. As such, by supporting multiple hypermedia controls more intelligent user agents can be created.

¹ http://linkeddatafragments-qas.viaa.be/
2.2 Hydra Partial Collection Views

Open Data is filled with collections of items (hotel amenities, road works etc.). Using the Hydra vocabulary for hypermedia-driven Web APIs, not only the relations between items and collections can be expressed, but also how different parts of the collection can be retrieved, so called partial collection view controls (e.g. a next page link). This way, a collection can be split in multiple documents while a Web client can still query over the collection by retrieving its view controls.

3 Implementation

3.1 hetarchief.be

Every newspaper webpage is annotated with JSON-LD snippets containing domain-specific metadata and hypermedia controls. The former metadata is described using acknowledged vocabularies such as Dublin Core Terms (DCTerms), Friend of a Friend (FOAF), Schema.org etc. The latter is described using the Hydra vocabulary for hypermedia-driven Web APIs. Although hetarchief.be contains several human-readable hypermedia controls (free text search bar, search facets, pagination for every newspaper) only Hydras partial collection view controls are implemented: hydranext describes the next newspaper, vice versa hydraprevious. Also an estimate of the amount of triples on a page is added using hydratotalitems and void:triples. This helps user agents to build more efficient query plans.

3.2 Building Blocks Comunica

To make Comunica work with hetarchief.be, two additional actors were needed. First, we needed a generic actor to support pagination over any kind of hypermedia interface. Secondly, an actor was needed to parse JSON-LD data snippets from HTML documents. We will explain these two actors in more detail hereafter.

BusRdfResolveHypermedia is a bus in Comunica that resolves hypermedia controls from sources. Currently, this bus only contains an actor that resolves controls for TPF interfaces. We added a new actor (ActorRdfResolveHypermediaNextPage) to this bus that returns a search form containing a next page link.

The parsing of most common Linked Data formats (Turtle, RDF/XML, JSON-LD…) are already supported by Comunica. However, no parser for extracting data snippets from HTML documents existed yet. That is why we added an actor (ActorRdfParseHtmlScript) for parsing such HTML documents. This intermediate parser searches for data snippets and forwards these to their respective RDF parser. In case of a JSON-LD snippet, the body of a script tag <script type="application/ld+json"> will be parsed by the JSON-LD parse actor.

By adding these two actors to Comunica, we can now query over a paged collection that is declaratively described with data snippets. As federated querying comes out-of-the-box with Comunica, this cultural heritage collection can now be queried together with other knowledge bases (cfr. Wikidata). For example, retrieving basic information such as title, publication date etc. from 17 newspaper pages requires 1.5 min until all
results are retrieved. This is caused by deficiency of indexes where all pages need examination before having a complete answer.

In next section we will demonstrate how SPARQL-querying can be applied for extracting a spreadsheet.

4 Demonstrator

This demonstrator shows that a non-technical user can create a data dump from the cultural heritage website hetarchief.be. More specifically, a spreadsheet can be extracted using SPARQL-querying from embedded paged collection views. The application is written with the front-end playground Codepen\(^2\). A browser compatible library of Comunica is built using a custom configuration that can be found on Github\(^3\) under an Open License.

First, a user can insert a URL of a hypermedia-enabled LOD interface. For example, a user can go to hetarchief.be and select a newspaper as starting point. After pressing Generate, Comunica fetches the document located on the given URL and follows the embedded pagination controls. While querying, user feedback is provided with request logs, the amount of processed CSV records and bytes. Next, the user can Copy the CSV output to its clipboard. Finally, a SPARQL-query can be configured to customize the desired outcome (Fig. 1).

![View your website as a spreadsheet](https://codepen.io/brechtvdv/p/en/ebOzXB)

**Fig. 1.** A spreadsheet is generated by entering a URL of a newspaper from hetarchief.be.

---

5 Conclusion

Data owners can publish their LOD very cost-efficient on their website with JSON-LD snippets. After an initial cost of adding this feature to their website, they can have an always up-to-date dataset with negligible maintenance costs, however, machine clients that query and harvest over websites can introduce unforeseen spikes of activity. Data owners will need to extend their monitoring capabilities to not only focus on human interaction (e.g. Google Analytics) and apply a HTTP caching strategy for stale resources.

Linked Data services (HDT [6] file, TPF interface…) with a higher maintenance cost can be created on top of JSON-LD snippets, but these would suffer from scalability problems: OCR texts have bad compression rates, and thus require gigabytes of disk space. With our solution, these OCR texts are published in a separate document keeping the maintenance cost low while harvesting in an automated way is still possible.

The LOD interfaces of the European cultural heritage platform Europeana take an opposite approach from this work: every subject page contains a title and description annotation for SEO, but the actual machine-readable data is exposed through a separate record API with API key protection and publicly available SPARQL endpoint. Supporting JSON-LD snippets as explained in this work would make the record API obsolete and more importantly, Open Data reusers would have a starting point for querying the SPARQL endpoint.

In future work, extending Comunica for harvesting Hydra collections would help organizations to improve their collection management. These collections could be defined on their main page of their website improving Open Data discoverability.

References