


Chapter 8

Creation of Value-Added Services by Retrieving Information From Linked and Open Data Portals

Antonio Sarasa-Cabezuelo

 <https://orcid.org/0000-0003-3698-7954>

Universidad Complutense de Madrid, Spain

ABSTRACT

In recent decades, different initiatives have emerged in public and private institutions with the aim of offering free access to the data generated in their activity to anyone. In particular, there are two types of initiatives: open data portals and linked data portals. Open data portals are characterized in that it offers access to its content in the form of a REST-type web services API that acts as a query language. On the other hand, linked data portals are characterized in that its data is represented using ontologies encoded by RDF triplets of the subject-predicate-object style forming a knowledge graph. This chapter presents a set of value-added service creation cases using the information stored in open data and linked data repositories. The objective is to show the possibilities offered by the exploitation of these repositories in various fields such as education, tourism, or services such as the search for taxis at an airport.

INTRODUCTION

In the last decades, different public and private institutions have emerged that have provided access to enormous amounts of data (Kitchin, 2014) on the activity it is carried out. In general, this information is available free of charge to anyone who wants to consult it or wants to use it to process it. It is true that there are some institutions that sell the data or special permits are required to access it, such as data from hospitals or personal data (Russell, 2013). In particular, there are two prominent initiatives: open data portals and linked data portals. Open data portals are characterized (Hossain et al, 2016) in that access to information is done through a Rest web services (Masse, 2011) API. The API (Michel et al, 2019)

DOI: 10.4018/978-1-7998-6697-8.ch008

acts as a query language so that each service can be configured with a set of parameters that establish what type of data it is able to retrieve, what type of filtering conditions can be done on the data, or the data format in which is retrieved the information. Normally, these portals offer to retrieve the data of the queries done, in the most standard formats such as JSON, XML, CSV and others. Another characteristic of these portals is that normally is available a catalog of the services offered. The catalogs specify each service, the way to invoke it, the parameters that can be configured, and the data formats in which the data can be returned, and even in some cases a test page is offered where these services can be tested. Also in some portals, it is possible to retrieve data directly without the need to invoke web services from any application, so that direct download is allowed in the same data formats mentioned above. One of the advantages of this initiative (Janssen et al, 2016) is the ease of adding an open data portal as the data source of a computer application that exploits the portal data. To do this, calls to web services are embedded within the application code, and the information retrieved in any of the formats can be stored as a document within the directory system or in a database (Larson, 2010). In this way, the information can be processed as if it were local data.

With regard to linked data portals, these are characterized (Hausenblas et al, 2010) by the way in which the information is stored. In this sense, the contents are encoded using domain ontologies that are used to describe the information in the form of RDF triplets of the subject-predicate-object form (Kahan et al, 2002). A property of triplets is that it can be linked together to form a network of related triplets that takes the form of an information graph. Thus, each graph represents the information of a different domain. Furthermore, it is possible to link graphs from different information domains, so that these portals become a great universal knowledge base (Hausenblas et al, 2009). The main advantages of these portals are the ease of creating and maintaining them, since the domain ontologies are created in a particular way in each case and it is the own representation structure of RDF standard that allows the described contents to be related (Heath et al, 2011). Another feature of these portals is the way of retrieving information. To do this, a standard query language called SPARQL (Quilitz et al, 2008) is used. The language syntax is similar to the SQL query language for relational databases. However, it has some differences since it is oriented to retrieve information from a graph, so the data structures or search conditions are adapted to the structure of the graph (Pauwels et al, 2018). The information retrieved through a SPARQL query can be stored in files with the most common data formats such as JSON, XML, CSV, RDF and others. In many portals there is what is called a SPARQL endpoint, which is a place in the portal that offers an interface where to execute SPARQL queries. Retrieval can be done directly from the portal, or it is able to embed retrieval SPARQL queries within the code in a specific programming language, so that just like in open data portals, the information is retrieved and stored local in files or in a database for further processing. In this way, linked data portals can be added to computer applications as a data source (Tsou, 2015).

A common feature in both initiatives is that the form of aggregation of portal data. In both cases, this is done by embedding the SPARQL queries or the web services API, within the application code. In this way, the recovered information is stored locally and can be further processed in any way it is wanted. This form of aggregation is very similar to the so-called mashups (Lee et al, 2012) or processing pipes that exist in programming languages and in other computer systems. Likewise, the aggregation and use that is made of the data has several implications. Firstly, the exploitation of the information on these portals allows the creation of value-added services (Sarasa-Cabezuelo, 2019). Thus, it can be given value to data that in many cases did not have it or increase its value by giving it a different utility than it originally had. For example, there are websites that allow to search for hotels or flights and facilitate

17 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/creation-of-value-added-services-by-retrieving-information-from-linked-and-open-data-portals/271126

Related Content

Watermarking Security

Teddy Furon, François Cayre and Caroline Fontaine (2008). *Digital Audio Watermarking Techniques and Technologies: Applications and Benchmarks* (pp. 278-299).

www.irma-international.org/chapter/watermarking-security/8337

Digital Narratives and the Genealogy of a Hybrid Genre

Otilia Pacea (2014). *Computational Linguistics: Concepts, Methodologies, Tools, and Applications* (pp. 1001-1017).

www.irma-international.org/chapter/digital-narratives-and-the-genealogy-of-a-hybrid-genre/108762

Amazon Mechanical Turk: A Web-Based Tool for Facilitating Experimental Research in ANLP

Amber Chauncey Strain and Lucille M. Booker (2012). *Cross-Disciplinary Advances in Applied Natural Language Processing: Issues and Approaches* (pp. 90-102).

www.irma-international.org/chapter/amazon-mechanical-turk/64582

Comparison between Internal and External DSLs via RubyTL and Gra2MoL

Jesús Sánchez Cuadrado, Javier Luis Cánovas Izquierdo and Jesús García Molina (2014). *Computational Linguistics: Concepts, Methodologies, Tools, and Applications* (pp. 816-838).

www.irma-international.org/chapter/comparison-between-internal-and-external-dsls-via-rubytl-and-gra2mol/108753

Wild Architecture: Explaining Cognition via Self-Sustaining Systems

Vincent T. Cialdella, Emilio J. C. Lobato and J. Scott Jordan (2020). *Natural Language Processing: Concepts, Methodologies, Tools, and Applications* (pp. 791-807).

www.irma-international.org/chapter/wild-architecture/239965