# An Investigation of HTTP Header Information for Detecting Changes of Linked Open Data Sources

Renata Dividino, André Kramer, and Thomas Gottron

WeST – Institute for Web Science and Technologies
University of Koblenz-Landau, Germany
{dividino,akramer,gottron}@uni-koblenz.de

**Abstract** Data on the Linked Open Data (LOD) cloud changes frequently. Applications that operate on local caches of Linked Data need to be aware of these changes. In this way they can update their cache to ensure operating on the most recent version of the data. Given the HTTP basis recommended in the Linked Data guidelines, the native way of detecting changes would be to use HTTP header information, such as the *Last-Modified* field. However, it is uncertain to which degree this field is currently supported on the LOD cloud and how reliable the provided information is. In this paper, we analyse a large-scale dataset obtained from the LOD cloud by weekly crawls over almost two years. On these weekly snapshots, we observed that for only 15% of the Linked Data resources the HTTP header field *Last-Modified* is actually available and that the date provided for the last modification aligns in only 8% with the observed changes of the data itself.

## 1 Introduction and Background

The Linked Open Data (LOD) cloud is a global information space to structurally represent and connect data items. The LOD principles provide a flexible publishing paradigm to integrate and interlink any kind of data from arbitrary datasets, published by various data providers on the Web. The distributed, Web-based nature of the data motivates many applications to keep local copies of the data. Data is fetched live from the Web only in those cases where the data is missing or known to be highly dynamic [5]. However, given the rate of changes of Linked Data [3] also local caches need to be updated from time to time [2]. Thus, the question is when to perform such an update.

A intuitive approach to this task is to exploit the way Linked Data is provided on the Web. According to the Linked Data guidelines resources should be modelled using dereferenceable HTTP Uniform Resource Identifiers (URIs). Whenever a client application invokes an HTTP request to a server for a particular URI on the LOD cloud, the server should respond by providing useful information about the entity represented by this URI. Naturally, this response will make use of the HTTP protocol itself. The HTTP header of this response can contain metadata about the resource (e.g. owner, creation date, etc.) [1]. Among these metadata there is a field which can denote when the resource behind this URI has been changed last. In combination with an HTTP HEAD request this *Last-Modified* field is intended for probing a resource for whether or not it has been changed since its inclusion in the cache of a Web or Linked Data application.

Nevertheless, even with existing W3C specifications which define rules and conditions to be followed by the LOD servers, the information contained in the HTTP headers may in practice be inaccurate or wrong [4]. Therefore, applications relying on such information are susceptible to draw wrong conclusions. In this paper, we empirically evaluate the conformance of time related HTTP header metadata information on the LOD cloud. In particularly, we check for the conformance of the *Last-Modified* field. Knowledge about the reliability of this field is important for applications which intend to make use of it.

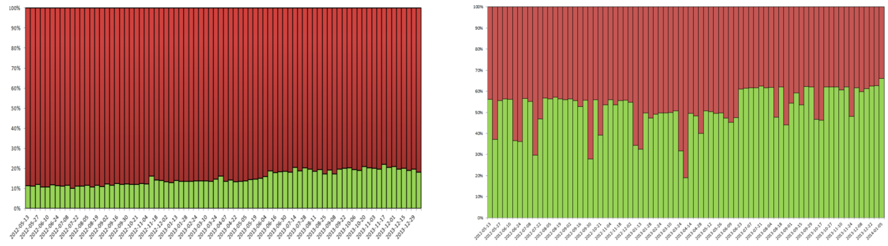
To this end, we analyse a large-scale dataset that is obtained from the LOD cloud by weekly crawls from the period between May, 2012 and January, 2014. The dataset contains 84 snapshots. For each pair of subsequent snapshots, we check for changes in the data and compare the observations to the information provided by the *Last-Modified* HTTP header field. Using the results of our experiments, we discuss the benefits of the availability and conformance of the HTTP header fields in real world scenarios.

## 2 Linked Data Metadata: The HTTP Header

The LOD cloud is composed by various data servers which enable data access via the HTTP protocol. A client application invokes an HTTP request to a server by, for instance, sending a HTTP GET message for a particular URI. The Linked Data server responds via HTTP by sending meaningful information for the represented resource, ideally using RDF as data format. The HTTP response header is mainly composed by:

1. The status code information about the request. For instance, a status code of 200 is the standard response for successful HTTP requests. This means that information about the resource is successfully returned. A code of 303 indicates a reference to another URI which can actually provide the requested resource. Also this response is encountered frequently on the LOD cloud, as it implements a technical solution to differentiate between the URI representing a real world entity and the URL providing the description about it. Finally, a response code in the range of 400 or 500 indicates errors on the client or server side.
2. Metadata about the resource. Some of the standard response header fields are: (1) *Content-Language* which indicates the language of the content, (2) *Content-Length*, the length of the response body in octets, (3) *Content-Type*, the MIME type of this content, (4) *Date*, the date and time that the message was sent, and (5) *Server*, indicating the name for the server.

In this work we focus on the analysis of the header field *Last-Modified*. HTTP/1.1 servers should send a *Last-Modified* value whenever feasible. This field is intended for a date when the requested object has been modified last. In the context of Linked Data, this corresponds to the most recent date at which (some part of) the resources' RDF description has changed. Following the HTTP/1.1 specification [1], the *Last-Modified* value must not be later than the time of the server's response message. In such cases, where the resource's last modification would indicate some time in the future it is to be considered invalid. Furthermore, the server should obtain the *Last-Modified* value as close as possible to the time that it generates the *Date* value of its response. This allows a recipient to make an accurate assessment of the entity's modification time.



(a) Ratio of the Linked Data resources (in percentage) that provide the field *Last-Modified* in their header (in green), or do not provide any value (in red).

(b) Ratio of the resources (in percentage) providing a *Last-Modified* that is correct (in green), or incorrect or invalid (in red).

**Figure 1.** Availability and correctness of the *Last-Modified* HTTP header field

## 3 Empirical Evaluation of the Conformance of the *Last-Modified* HTTP Header Field

The main goal of our experiments is to measure the degree of how often the *Last-Modified* field in HTTP header of LOD resources is available and how often it is used correctly. We consider the use to be correct if the fields returns a date and time which does not violate the observations of when the data for a resource has changed last. For this purpose, we work with data from the Dynamic Linked Data Observatory (DyLDO) [3]. The DyLDO dataset has been created to monitor a fixed set of Linked Data documents (and their neighborhood) on a weekly basis. For the sake of consistency, we use only the kernel seed documents of DyLDO. Our test dataset is composed of 84 snapshots corresponding to a period of almost two years (from May, 2012 until January, 2014). Furthermore, the DyLDO dataset contains (parts of) various well known and large LOD sources, e.g., dbpedia.org, musicbrainz.com, and bbc.co.uk. For more detailed information about the DyLDO dataset, we refer the reader to [3].

Each version of the DyLDO dataset consists of a set of RDF triples retrieved from different LOD sources. Furthermore, the data provides also information about the HTTP headers received when retrieving the data. From the 84 snapshots available in the dataset, we took each pair of subsequent snapshots from the same data source and computed the set difference over their set of triples. If we observe a difference we consider the data to have changed, otherwise we treat it as unchanged. A change should be reflected by a *Last-Modified* date which lies in the time range between the two snapshots of the data.

Figure 1(a) illustrates that on average only 15% of the resources actually do provide some value for the *Last-Modified* field in the HTTP Header. Subsequently, we checked for those resources which provide a value for the *Last-Modified* field, how many of them return a correct or an incorrect value (see Figure 1(b)). As mentioned above, correct values are the ones where the last modified data aligns with actual changes in the RDF data. Incorrect values includes (1) values that indicate changes but no change has been observed, (2) values that indicate no changes but changes have been observed and (3) invalid values. Invalid values are the ones which indicate a time in the future relative to the time of the HTTP response or which indicate a time of last modification which actually precedes the time at which the resource was created. On average, we observe that only 52% of the resources which provide a value for the *Last-Modified* field provide also a correct value for it. The slight growth of both ratios in Figure 1 towards the end of the time period covered by the dataset is an artefact caused by data sources going offline, i.e. not responding or providing a 400 or 500 status code as response. It seems that more data sources providing no or wrong *Last-Modified* went offline during the covered time span.

## 4 Summary and Discussion

In this paper we evaluated the conformance of LOD data source to provide a valid and correct *Last-Modified* HTTP header field, which indicates the date and time at which the resource was last modified. Our experiment shows that overall and on average only 8% of the resources in the datasets provide correct values for this field. This number is far too low to be of use for any practical application. It is, however, not clear why LOD sources do not provide valid information. We conjecture that some default configuration of LOD servers leads to this misbehaviour.

The reliable provision of meta data in the context of the established HTTP protocol would be beneficial to the entire Web of Data. Many base technologies such as Linked Data caches and indexes may benefit of this information since a simple check on this metadata could support their decision process of determining which sources need to be updated. In conclusion, with this work we point out the dimension of the problem of erroneous and missing information of the HTTP header for Linked Data. Thereby, we motivate LOD sources to publish correct and valid values to support application needs. We believe that publishing correct the HTTP Header information is a step towards quality-oriented data usage in the LOD cloud.

*Acknowledgements* The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013), REVEAL (Grant agree number 610928).

## References

1. Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T.: Hypertext transfer protocol–http/1.1 (1999)
2. Gottron, T., Gottron, C.: Perplexity of index models over evolving linked data. In: ESWC'14: Proceedings of the Extended Semantic Web Conference (2014), to appear
3. Käfer, T., Abdelrahman, A., Umbrich, J., O'Byrne, P., Hogan, A.: Observing linked data dynamics. In: ESWC. pp. 213–227 (2013)
4. Rula, A., Palmonari, M., Harth, A., Stadtmüller, S., Maurino, A.: On the diversity and availability of temporal information in linked open data. In: The Semantic Web–ISWC 2012, pp. 492–507. Springer (2012)
5. Umbrich, J., Hausenblas, M., Hogan, A., Polleres, A., Decker, S.: Towards Dataset Dynamics: Change Frequency of Linked Open Data Sources. In: LDOW (2010)