

WissKI Viewer

Casual Access for WissKI Data Sets

Tom Wiesing¹[\[https://orcid.org/0009-0002-7392-0556\]](https://orcid.org/0009-0002-7392-0556)

¹FAU Erlangen-Nürnberg, Germany

Abstract: WissKI is a software which allows researchers to record data about objects of the cultural heritage in a graph database backed by a formal ontology. It acts as a database for researchers to store their results via a web interface, and nearly automatically makes data FAIR, linked and open. To install the WissKI software a nontrivial amount of effort is required and typically requires help from a system administrator. To make it easier to access WissKI and data stored within this paper introduces the WissKI Viewer, which enables researchers to directly inspect a WissKI Backup on their own computer, without additional help.

Keywords: WissKI, FAIR, Linked Open Data

FAIR [1] is an acronym defined as Findable, Accessible, Interoperable and Reusable. It is a desired quality of research data, and is self-explanatory. Findable means that data can be identified, and is well-described with appropriate metadata. Accessible means that data can be accessed, and does not require custom authentication or authorization schemes. Interoperable means that data uses common data formats, and is machine-readable. Reusable means that data can be reused, and is not limited to the original purpose it was created or collected for.

Linked Open Data is an orthogonal concept to FAIR meaning data both open and linked. Open means being available under an Open Source license. Linked means being interlinked and accessible using semantic queries.

A Triplestore, also known as a graph database, consists of (subject, predicate, object) triples describing objects and their properties. Triples correspond to edges in a labeled, directed graph. The subject is the source node, the object the sink node, and the predicate the label.

Most common triplestore implementations are based on RDF [2], and can be queried using SparQL [3]. SparQL is ideal for enabling linked data, as it provides a formal query language. It allows provides facilities more making data linked and open using so-called federated queries.

Drupal [4] is an extensible open source content-management system written in a programming language called PHP. It is available via a web interface, and enables users to create and manage content on their website. The backend of Drupal is powered by an SQL database.

WissKI [5], [6] is a system which allows researchers to record data about objects of the cultural heritage in a graph database backed by a formal ontology. The WissKI interface provides three main functionalities.

- An administrative interface where a formal ontology can be defined and edited,
- An edit interface where database entries can be added and edited without the need to entirely understand the formal ontology, and
- A public-facing browsing interface that allows browsing the database.

For any particular research project, the WissKI software first has to be installed on a server accessible for the desired users. It then has to be extended in order to fit the projects needs. Such a WissKI-based system consists of several components which can be seen in Figure 1. They are implemented as a set of extensions (called modules) to the content management system Drupal and accessible to users via a web interface.

Drupal Core The Drupal Core represents an installation of the Drupal content management system. It handles authentication and manages different display options for content.

SQL Database Required by Drupal in order to store authentication and presentation configuration data. It also acts as a cache.

WissKI modules The WissKI modules run inside of drupal and implement the main functionality of WissKI.

Triplestore Data entered by WissKI users – the research result – is stored inside a triplestore.

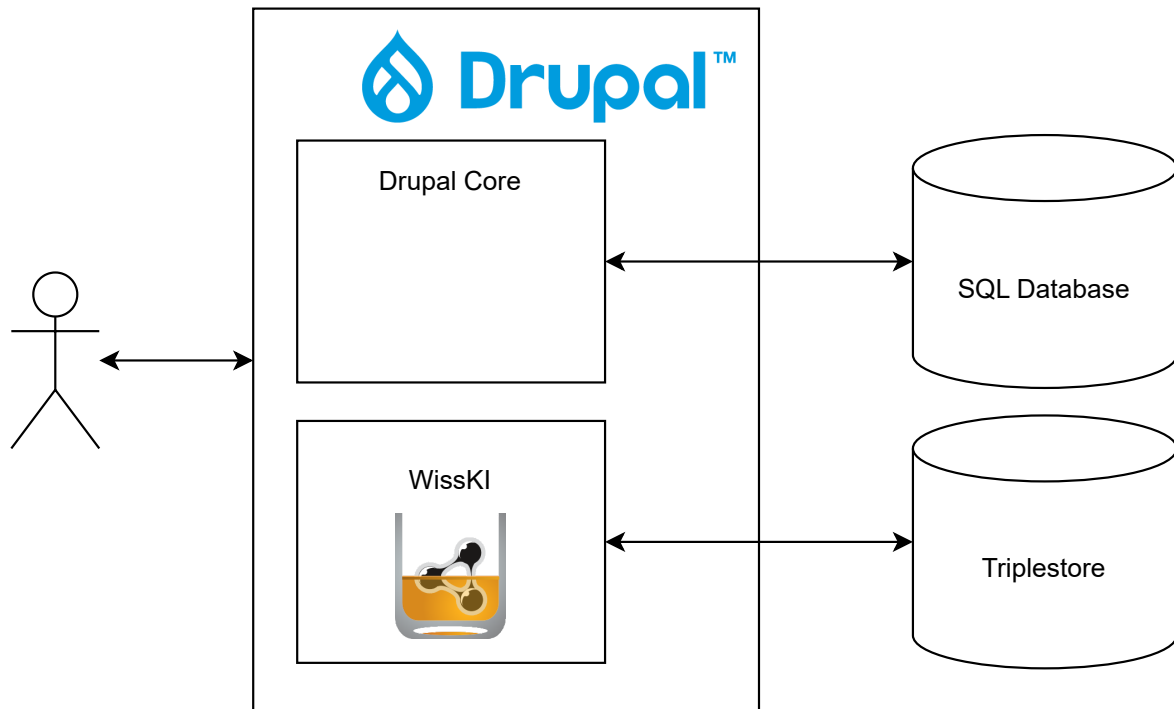


Figure 1. Overview of a WissKI-based system

In order for a WissKI-based system to function properly, each of these components requires separate configuration. This makes it difficult to set up and maintain - typically a system administrator with significant expertise is required. Furthermore, once a research project has ended and funding has run out it quickly ends up in an unusable state or is shutdown entirely. Only the research result – the data in the triplestore – survives this shutdown. Information regarding how data in the database should be interpreted by users - such as formatting - does not survive.

A long term goal should be to avoid such a shutdown by improving the maintainability and usability of the WissKI software itself. But this alone is insufficient to fully address the problem – providing a shared interface to view, edit and manage a dataset has inherent complexity and comes with maintenance implications. Instead the WissKI Viewer takes a different approach - it is a software that only aims at making WissKI datasets viewable. This is sufficient to ensure that the research result remains viable after a project ends.

The WissKI viewer runs directly on a researchers computer and does not require any systems knowledge in order to function. It directly provides the researcher with an interface to view any database entries created in the originating system. The WissKI Viewer is written in go and available at [7]. It is distributed as a standalone application, and a screenshot of the viewer interface can be seen in Figure 2.

The viewer takes as input the surviving graph data from the triplestore as well as a configuration file containing the ontology used. Upon startup, it first loads the graph data into memory, and then repeatedly scans it according to the ontology used to recover the original entries. Depending on the exact size of the dataset, this process can take a few seconds (for datasets consisting of a few thousand triples) to a few minutes (for datasets consisting of several million triples).

Compared to WissKI itself, the viewer does not rely on any external databases. This makes it ideal to ensure that WissKI-produced datasets remain accessible after the original system that produced them has been retired.

Author contributions

We contributed by developing the WissKI Viewer based on the pre-existing WissKI system.

<http://kirmes.wisski.agfd.fau.de/#5f15b12cd5fb1>

[Bundles] > [Bundle Werke] > [Entity <http://kirmes.wisski.agfd.fau.de/#5f15b12cd5fb1>]

Fields


Name	Type	Count	Value(s)
Show Entity Triples			
Werktitel/Title	string	1 (Cardinality 1)	Dorpskermis op het feest van de H. Joris
Abbildungs-ID/Image-ID	string	1 (Cardinality -1)	8f142990-0a89-8332-a28b-25375773d9c6
Abbildung/Image	image	1 (Cardinality -1)	 <p>644. — PIERRE BREUGHEL le jeune. <i>Kermesse flamande</i>.</p>
Kommentar/Comment	text_long	1 (Cardinality 1)	<p>Diese Version eventuell identisch mit dem Gemälde, das Max Friedländer in Antwerpen vermutete. Es ist unten links mit P. BREUGHEL signiert. Es befindet sich laut Eintrag in der Datenbank des RKD heute im Koninklijk Museum voor Schone Kunsten Antwerpen, (inv./cat.nr 644), als Schenkung von A. Michiels. Eine weitere Version von Pieter Bruegel dem Jüngeren wurde 2004 bei Sotheby's versteigert.</p>

Figure 2. A database entry about a painting displayed in the WissKI Viewer.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

The author wishes to acknowledge the help of the WissKI community, and in particular Mark Fichtner, for help during the development of the WissKI Viewer.

References

- [1] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, *et al.*, “The fair guiding principles for scientific data management and stewardship,” *Scientific Data*, vol. 3, 2016. DOI: [10.1038/sdata.2016.18](https://doi.org/10.1038/sdata.2016.18).
- [2] World Wide Web Consortium (W3C), Ed. “Resource description framework (RDF).” (), [Online]. Available: <http://www.w3.org/RDF/> (visited on 10/22/2009).
- [3] “SPARQL 1.1 overview,” World Wide Web Consortium (W3C), W3C Recommendation, Mar. 23, 2013. [Online]. Available: <https://www.w3.org/TR/sparql11-overview/>.
- [4] Drupal Association, *Drupal - Open Source CMS*, <https://www.drupal.org/>, [Online; accessed 20-April-2023], 2023.

- [5] G. Hohmann and B. Schiemann, “An Ontology-Based Communication System for Cultural Heritage. Approach und Progress of the WissKI Project,” in *Scientific Computing and Cultural Heritage. Contributions in Computational Humanities*. (Contributions in Mathematical and Computational Science), H. G. Bock, W. Jäger, and M. Winckler, Eds., Contributions in Mathematical and Computational Science. Berlin: Springer, 2013, vol. 3, pp. 127–135.
- [6] G. Goerz, “WissKI: Semantische Annotation, Wissensverarbeitung und Wissenschaftskommunikation in einer virtuellen Forschungsumgebung,” Jun. 2011.
- [7] Tom Wiesing, *hangover - A WissKI Viewer*, <https://github.com/FAU-CDI/hangover>, Accessed on Sep 1, 2023, 2023.