From Triples to Cards: A Web Component for Human-Readable Linked Open Data

Puyu Wang[1,*], Hang Dong[2] and Rui Zhao[3]

[1]Oxford e-Research Centre, Department of Engineering Science, University of Oxford, Oxford, UK
[2]Department of Computer Science, University of Exeter, Exeter, UK
[3]School of Cybersecurity, Tianjin University, Tianjin, China

Abstract

JSON-LD is getting increasing attention for publishing Linked Data, especially within cultural heritage. We present LODox Data Viewer, a reusable web component library for data presentation and visualization in semantic portals. It renders JSON-LD as human-readable, themeable cards, embeddable in any semantic portal page with a single script tag.

Package URL: https://www.npmjs.com/package/lodox-data-viewer

Keywords

Data Discovery, Linked Data, Cultural Heritage, Semantic Portal

1. Introduction

JSON (JavaScript Object Notation) has become the dominant data format for web applications [1, 2]. JSON-LD, a Linked Data serialization format based on JSON [3], is designed to ease the adoption and integration of Linked Data in web applications. It has been attracting increasing attention, especially in the cultural heritage domain. For example, the Linked Art data model [1] provides a JSON-LD based data profile built on the CIDOC-CRM ontology [2], aiming to provide shared data modelling approaches [4]. It has been adopted and implemented in a range of online applications and workflows [5, 6]. Furthermore, the Getty Museum Collection also provides an API based on Linked Art [3].

Meanwhile, many existing semantic portals and web applications for presenting cultural heritage Linked Data offer rich interfaces but rely on a full stack, so they cannot easily be integrated into other systems without deploying that complete stack [7, 8]. We developed *LODox (Linked Open Data Oxford) Data Viewer*, a reusable web component library that is:

- **easily embedded**: it can be embedded anywhere with one script tag by different web applications.
- **able to render JSON-LD without per-record templates**: the viewer renders JSON-LD data into human-readable cards.
- **customizable**: per-record layout and theming can be configured.

On a further note, LODox Data Viewer is still under active development. It was initially the presentation layer of a full-stack system for publishing cultural heritage Linked Data, namely LODox, and was later decoupled into a stand-alone library so that it can be reused independently.

In this paper, we will first describe the design and implementation of LODox Data Viewer (Section 2), and demonstrate its usage with JSON-LD and Linked Art (Section 3). Finally, we will discuss the conclusion and future plans in Section 4.

Data-Driven Storytelling: Bridging Knowledge Graphs, GenAI, and Narrative (DDS2026), in ISWC 2026 Workshops Joint Proceedings, October 25–26, 2026, Bari, Italy

*Corresponding author.
puyu.wang@eng.ox.ac.uk (P. Wang); h.dong2@exeter.ac.uk (H. Dong); zhaorui@tju.edu.cn (R. Zhao)
https://puyuwang.org/ (P. Wang); https://experts.exeter.ac.uk/41644-hang-dong (H. Dong); https://me.ryey.icu (R. Zhao)
0009-0003-8089-4146 (P. Wang); 0000-0001-6828-6891 (H. Dong); 0000-0003-2993-2023 (R. Zhao)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

[1]https://linked.art/api/1.0/json-ld/
[2]https://cidoc-crm.org
[3]https://data.getty.edu/museum/collection/docs/

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-05

CEUR-WS.org/Vol-4313/dds-paper5.pdf

2. Design and Implementation

LODox Data Viewer is available either as an npm (Node Package Manager) package or as a loadable bundle from a Content Delivery Network (CDN). For example, it can be loaded using unpkg.com:

```
<script type="module" src="https://unpkg.com/lodox-data-viewer"></script>
```

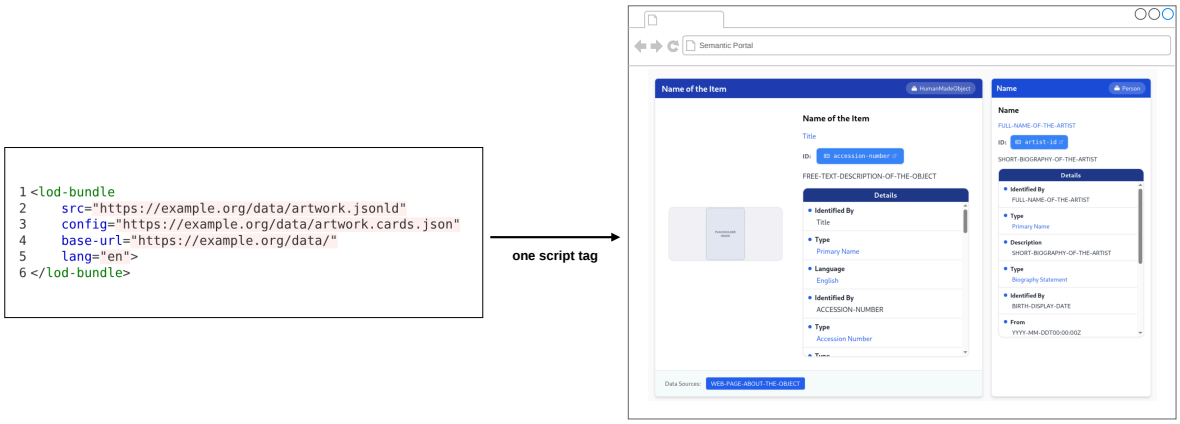


Figure 1: Rendering JSON-LD with a few lines of code

From a few lines of code with a single script tag, LODox Data Viewer renders JSON-LD into multiple human-readable cards, as shown in Figure 1. In order to do this, we use the workflow demonstrated in Figure 2. First, the package automatically detects the profile of the input JSON-LD. At present, it detects the Linked Art profile and generic JSON-LD; support for additional profiles is under development. The corresponding adaptor decides what enters the card header as a record and extracts basic information for each record: identifier, link, title, description, image, etc. A *walker* then traverses all the remaining properties losslessly: every non-empty value reachable from each record's root will appear on the corresponding record card, except for language tags and image captions. Nested objects are flattened. For example, measurement objects are represented as a single row containing both the measured value and its unit. An object with a `_label`, a `content`, or an `id` emits a row of its own, followed by rows for its properties. An object without these emits no row of its own, but its properties may appear as rows. The layout, size, and colours of each card are configurable either by a configuration file or through CSS custom properties. The package also includes built-in presets, each setting a card's layout and size in a single word.

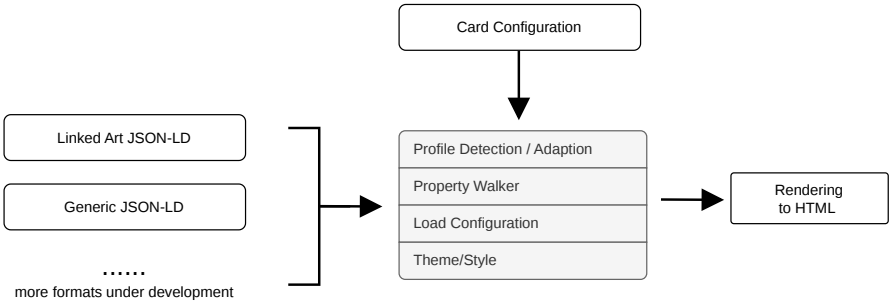
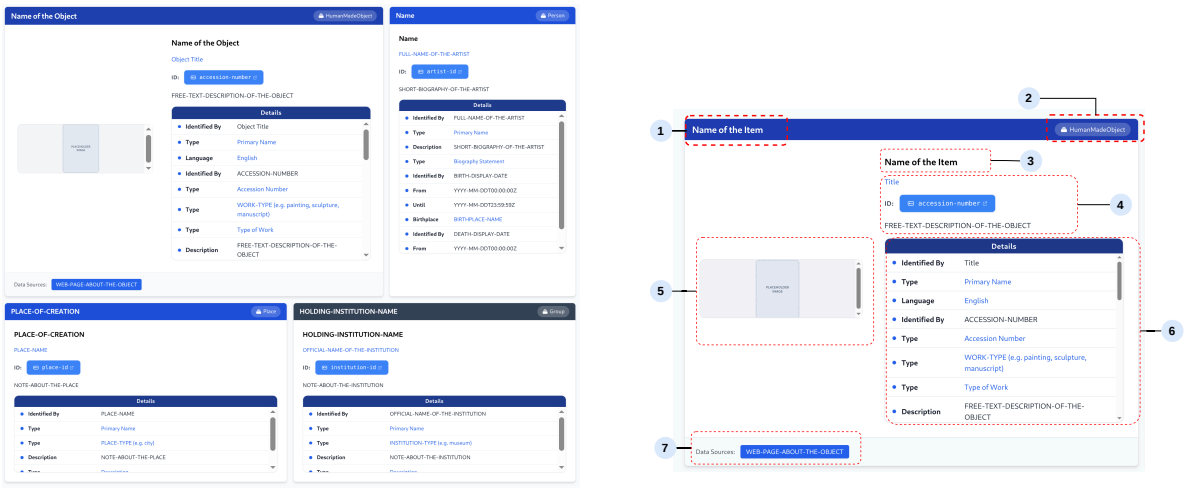


Figure 2: Render Workflow

Finally, the package presents each record as a card including a header with the title and type, the images, a short description, and a table of detailed properties. A single JSON-LD file can contain a main item together with related ones, for example, a painting alongside its painter and current location. Each of these is rendered as its own card showing the properties relevant to it, as shown in Figure 3a.

Figure 3b shows the layout of a single card. (1) and (3) show the primary label, in the card header and at the top of the body. (2) is the object type. For Linked Art JSON-LD, it is the CIDOC-CRM class representing the resource type, for example, "HumanMadeObject"[4]. (4) contains further essential information, including alternative names, the identifier (URI), and a short description. (5) contains the associated images. (6) contains a detailed table, which takes the record as the subject. Each row in the table shows one property and its value. Finally, (7), shown in the footer, displays the data source as a clickable link that directs users to the original source website. Furthermore, each card can be configured



(a) Multiple cards view

(b) The card layout

Figure 3: LODox Data Viewer interface, with the visualization of multiple cards and the layout of a card.

with different height and width, allowing the overall layout of the cards to be customized. The cards will be stacked side by side automatically given enough width, or vertically. Therefore, as the subject and object in a semantic triple can be placed as cards side by side, the layout can also be used to present relations between the cards, as shown in Figure 4.

3. Demonstration

The npm package includes a command (`npm run demo`) that runs a live demonstration server locally. At present, Linked Art JSON-LD files are handled by an adaptor written for their vocabulary; others are rendered by the generic profile with no code specific to them. Figure 5 shows the demo displaying the Mona Lisa together with Leonardo da Vinci[5].

4. Conclusion and Future Work

In this paper, we presented LODox Data Viewer, an npm package for rendering JSON-LD. We have demonstrated that JSON-LD data with well-established vocabularies can be effortlessly rendered while emphasizing the fields specified by the profile adaptors. LODox Data Viewer also provides a card-based presentation for Linked Data. The package remains an early-stage software demonstration, and we have not yet conducted a user-based evaluation. The usability is evaluated based on development experience and unit testing. User-based evaluation with developers and practitioners is planned as future work. In the meantime, we identify three major directions for future development.

First, we will develop more adaptors for other well-established ontologies and investigate how to render data described by multiple ontologies simultaneously.

[4]http://www.cidoc-crm.org/cidoc-crm/E22_Human-Made_Object

[5]The JSON-LD data used in this example is available online: https://github.com/cogim-online/lodox_card_demo/blob/main/data/mona-lisa.jsonld

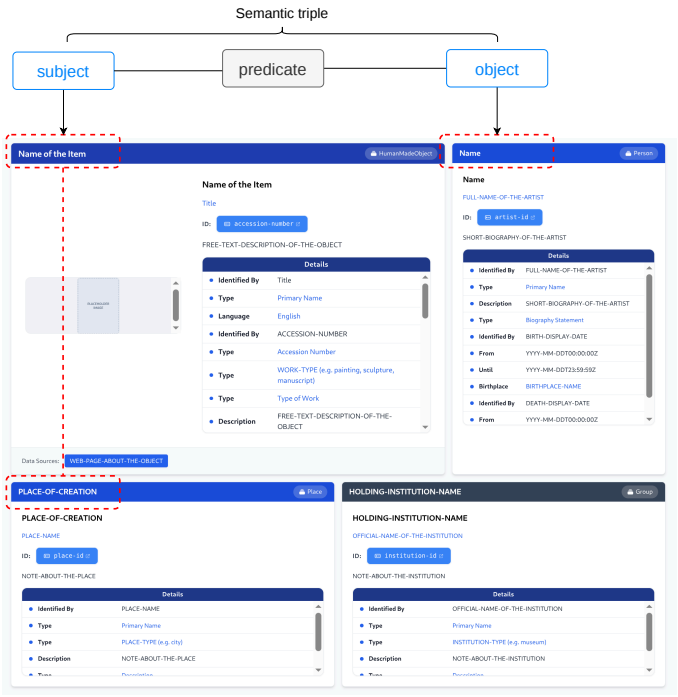


Figure 4: Multiple cards as triples, as visualized through the LODox Data Viewer interface.

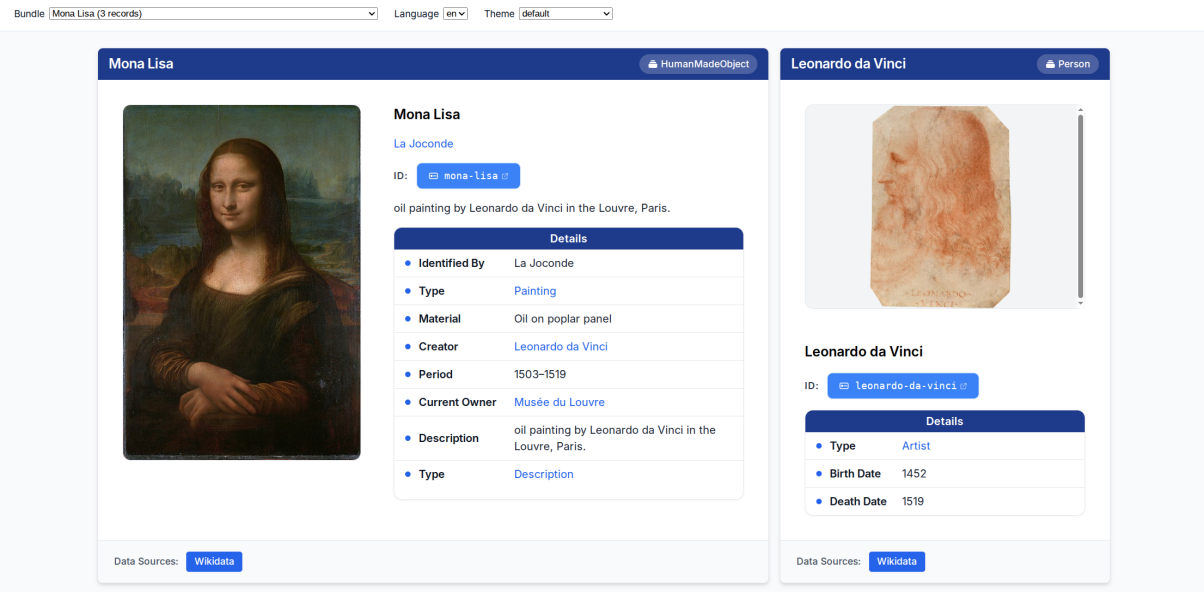


Figure 5: Screenshot from a live demo featuring Mona Lisa.

Second, we acknowledge that the rendering result depends on the configuration and the quality of the JSON-LD provided. At present, not every JSON-LD file renders well owing to limited profile compatibility and data completeness. We therefore plan to investigate algorithms and approaches for auto-configuration, enabling more flexible rendering.

Third, we will investigate the use of AI agents to retrieve and control the cards and terms expressed in JSON-LD, in particular, to determine which cards are included in a given JSON-LD document. We expect LODox Data Viewer to be embedded in a pipeline in which an agent reviews data drawn from a graph database and exports the corresponding JSON-LD. Moreover, the current package only renders

static JSON-LD files. We also wish to develop dynamic loading with an agentic AI, in which cards are generated successively, each one selected for its relevance or relations to those already selected.

Acknowledgments

The authors are grateful to Professor Donna Kurtz for her guidance and support for this work.

Declaration on Generative AI

During the preparation of this work, the author(s) used Claude and ChatGPT in order to: grammar and spelling check, content enhancement, improve writing style, and paraphrase and reword. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] F. Pezoa, J. L. Reutter, F. Suarez, M. Ugarte, D. Vrgoč, Foundations of JSON Schema, in: Proceedings of the 25th International Conference on World Wide Web, WWW '16, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2016, pp. 263–273. URL: https://dl.acm.org/doi/10.1145/2872427.2883029. doi:10.1145/2872427.2883029.

[2] T. Bray, The JavaScript Object Notation (JSON) Data Interchange Format, Request for Comments RFC 8259, Internet Engineering Task Force, 2017. URL: https://datatracker.ietf.org/doc/rfc8259. doi:10.17487/RFC8259, num Pages: 16.

[3] G. Kellogg, P.-A. Champin, D. Longley, JSON-LD 1.1 – A JSON-based Serialization for Linked Data (2020).

[4] E. Delmas-Glass, R. Sanderson, Fostering a community of PHAROS scholars through the adoption of open standards, Art Libraries Journal 45 (2020) 19–23. URL: https://www.cambridge.org/core/journals/art-libraries-journal/article/abs/fostering-a-community-of-pharos-scholars-through-the-adoption-of-open-standards/836554F24963C7B430DE3016E51FA8DA. doi:10.1017/alj.2019.32.

[5] J. A. Raemy, T. Gray, A. Collinson, K. R. Page, Enabling Participatory Data Perspectives for Image Archives through a Linked Art Workflow (2023). URL: https://zenodo.org/records/8228497. doi:10.5281/zenodo.8228497.

[6] R. Sanderson, Implementing Linked Art in a Multi-Modal Database for Cross-Collection Discovery, Open Library of Humanities 10 (2024). URL: https://olh.openlibhums.org/article/id/15407/. doi:10.16995/olh.15407.

[7] D. Oldman, D. Tanase, Reshaping the Knowledge Graph by Connecting Researchers, Data and Practices in ResearchSpace, in: D. Vrandečić, K. Bontcheva, M. C. Suárez-Figueroa, V. Presutti, I. Celino, M. Sabou, L.-A. Kaffee, E. Simperl (Eds.), The Semantic Web – ISWC 2018, volume 11137, Springer International Publishing, Cham, 2018, pp. 325–340. URL: https://link.springer.com/10.1007/978-3-030-00668-6_20. doi:10.1007/978-3-030-00668-6_20.

[8] E. Ikkala, E. Hyvönen, H. Rantala, M. Koho, Sampo-ui: A full stack javascript framework for developing semantic portal user interfaces, Semantic Web 13 (2021) 69–84. URL: https://journals.sagepub.com/doi/abs/10.3233/SW-210428. doi:10.3233/SW-210428.