International FAIR Digital Objects Implementation Summit 2024
Full Papers
https://doi.org/10.52825/ocp.v5i.1411

Published: 18 Mar. 2025

PID-Component
A Web Component for Visualizing (not Only) FAIR Digital Objects

Maximilian Inckmann[1,*], Andreas Pfeil[1], and Thomas Jejkal[1]

[1]Karlsruhe Institute of Technology, Karlsruhe, DE

[*]Correspondence: Maximilian Inckmann, maximilian.inckmann@kit.edu

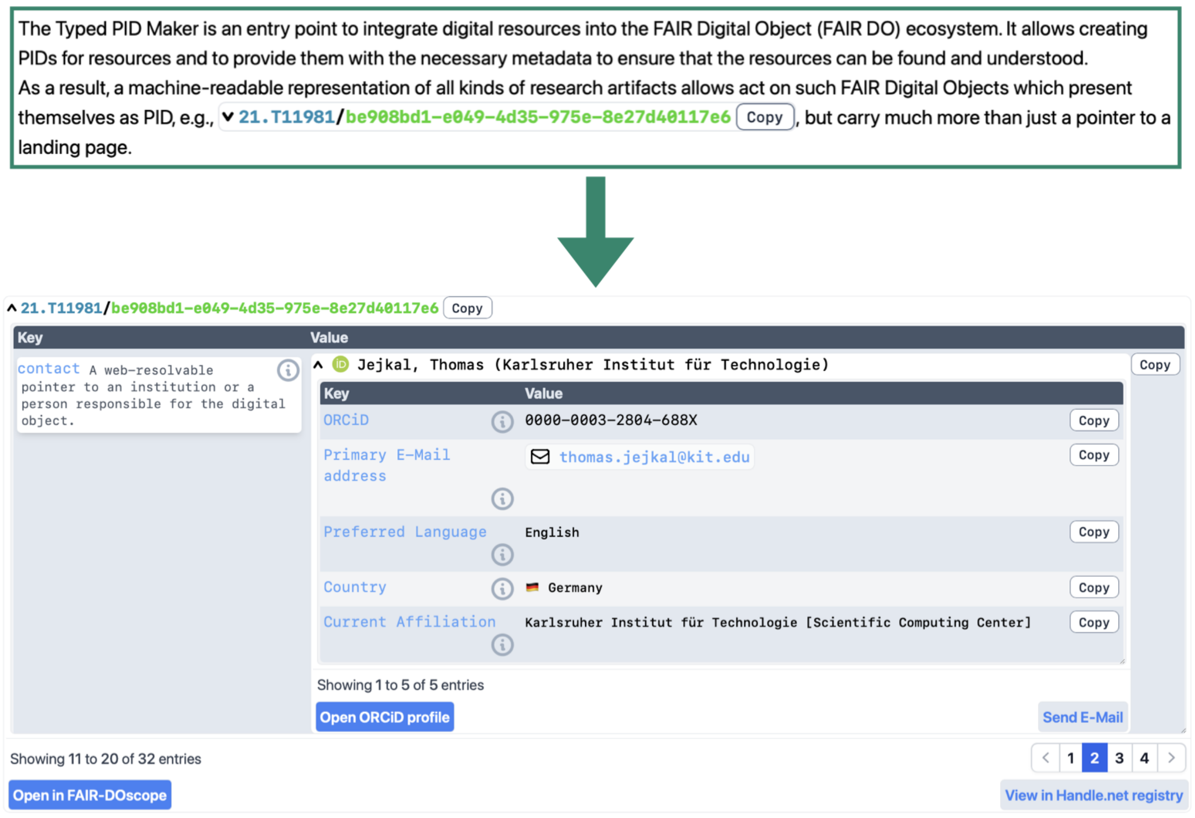
Abstract. The FAIR Digital Objects concept is primarily designed for machine readability, machine interpretability and machine actionability, which results in a highly complex and interconnected infrastructure. To improve the integration, usability, and acceptance of FAIR Digital Objects (FAIR-DOs) - especially among non-technical users - it is therefore important to provide user-friendly and straightforward solutions. This work presents the pid-component, a reusable and highly extensible web component for framework-agnostic visualization of a variety of inputs, including Persistent Identifiers (PIDs), Data Types, ORCiDs, and others. For users, the pid-component provides a clean, uniform, and minimalistic interface for consuming FAIR-DOs. For developers, it enables easy integration into their platforms with minimal effort. This is achieved through a highly modular design with a compact form factor, aligning with the height of a single regular text line, and a minimalistic design approach focused on usability. The pid-component is based on a series of W3C web standards and can be explored through an interactive playground, enabling developers to grasp the functionalities of the pid-component.

Keywords: FAIR Digital Objects, Visualization, Usability, Framework-Agnostic Web-Component, Extensibility

PID-Component

The realization of machine actionability through machine interpretability and machine readability of diverse data resources is the primary design objective of FAIR Digital Objects (FAIR-DOs). Despite this machine-oriented design, providing a consistent and user-friendly interface for human interaction remains essential. This approach is key to enhancing their integration, usability, and acceptance in both current and future research data management platforms. Until now, persistent identifiers (PIDs) have primarily served as primitive pointers to custom landing pages, which are the actual carriers of information ready to be consumed by users. With the emergence of FAIR-DOs, the potential of PIDs to carry contextual information and semantic links in their records was introduced, yet their full potential has remained unrealized due to a lack of effective tooling. Consequently, users often remain skeptical about FAIR-DOs because they appear abstract and intangible.

To address these challenges, solutions must be straightforward for both users and machines that wish to consume FAIR-DOs, as well as for developers tasked with integrating them into existing platforms. For improved human understanding of abstract FAIR-DOs, effective methods for visualization and exploration are necessary. Web components can contribute to this goal, as they can be integrated into almost any web application and presented and operated in the same manner as standard HTML elements. Furthermore, web components are built on a collection of W3C standards, ensuring functionality across all well-established browsers.


The Typed PID Maker is an entry point to integrate digital resources into the FAIR Digital Object (FAIR DO) ecosystem. It allows creating PIDs for resources and to provide them with the necessary metadata to ensure that the resources can be found and understood.
As a result, a machine-readable representation of all kinds of research artifacts allows act on such FAIR Digital Objects which present themselves as PID, e.g., 21.T11981/be908bd1-e049-4d35-975e-8e27d40117e6 Copy, but carry much more than just a pointer to a landing page.

21.T11981/be908bd1-e049-4d35-975e-8e27d40117e6 Copy

Key	Value
contact A web-resolvable pointer to an institution or a person responsible for the digital object.	Jejkal, Thomas (Karlsruher Institut für Technologie) Copy

Key	Value
ORCiD	0000-0003-2804-688X Copy
Primary E-Mail address	thomas.jejkal@kit.edu Copy
Preferred Language	English Copy
Country	Germany Copy
Current Affiliation	Karlsruher Institut für Technologie [Scientific Computing Center] Copy

Showing 1 to 5 of 5 entries

Showing 11 to 20 of 32 entries

Figure 1. *pid-component rendering of a FAIR Digital Object with resolved data types (left column) and input-related rendering of values (right column). Depending on the input, values can be further expanded to show further details, e.g., ORCiD record information.*

The pid-component, as illustrated in Figure 1, is designed around principles of user-friendliness, usability, minimalism, and extensibility, addressing the aforementioned challenges while adhering to Web Component standards. It displays highly diverse information in a uniform, explorable, and minimalistic manner to the user while maintaining a compact form factor. The pid-component can be seamlessly incorporated into regular text paragraphs (see the top of Figure 1), as its height aligns with that of the text. This design allows for smooth integration into existing web applications without disrupting the user interface, facilitating easier adoption and implementation by developers. Users can unfold the pid-component and explore further information, e.g., the content of a FAIR-DO, hierarchically (as shown at the bottom of Figure 1). The pid-component fetches all necessary information asynchronously from different sources (e.g., Handle.net, Data Type Registries, or ORCiD.org) and caches them locally in the user’s browser, speeding up the loading process by reducing network traffic.

The pid-component unlocks its full potential, especially when presenting complex FAIR-DOs. It allows for self-nesting, enabling intuitive displays of complex FAIR-DOs. Currently, it can resolve Handle PIDs with their records, evaluate data types to present essential information, and access ORCiD profiles. It renders the following inputs:

- **Handle PIDs and DOIs:** These are displayed with highlighted prefixes and suffixes for easy color-based identification, along with a button to open the respective PID in the FAIR-DOscope [1].
- **Data Type Information:** In FAIR-DOs values are described by their data type, which are stored in a Data Type Registry (DTR). The pid-component retrieves the name and description from the entry in the DTR, displaying them as readable text with a tooltip and linking them back to the DTR entry.
- **ORCiDs:** The component retrieves details through the ORCiD API [2], displaying the most important information in the compact preview. If a primary email address is available, a “Send E-Mail” button is shown for direct contact. Language and country information are rendered as localized descriptions, including matching flags.
- **Email Addresses:** Rendered as clickable `mailto` links, supporting comma-separated lists of email addresses as separate links.
- **Dates:** Rendered as formatted strings, with formatting based on the user’s localization settings in the browser.
- **URLs:** Rendered as clickable links.

For developers looking to incorporate the pid-component, the only essential parameter required is the PID of a FAIR-DO or another compatible input value. The pid-component intelligently determines how to display the input without necessitating server-side hosting, as all processing occurs locally in the user’s browser. To access the pid-component, developers can utilize the JavaScript module through npm [3] or a CDN that offers npm packages, such as unpkg.com. For easy integration, the open-source codebase is available on GitHub, accompanied by an interactive playground where developers can explore the pid-component’s features and test different inputs. Releases are available on npmjs and GitHub Packages for convenient use.

The pid-component represents an advancement in the visualization of FAIR-DOs, providing developers with a robust tool that enhances the usability and accessibility of PIDs. Its design enables seamless integration into various web applications, such as repository search interfaces, making it a versatile solution for both technical and non-technical users.

One of the key features of the pid-component is its ability to operate entirely within the user’s browser, eliminating the need for server-side hosting. This local processing capability not only streamlines the integration process but also enhances performance by reducing network traffic. The component’s reliance on a simple input parameter — the PID of a FAIR-DO or any compatible value — ensures that developers can easily incorporate it into their existing platforms without extensive modifications.

As the adoption of FAIR Digital Objects continues to grow, the pid-component can bridge the gap between the rather abstract FAIR-DO concept and user-friendly interfaces. Future developments will focus on expanding input types and renderers, including support for ROR identifiers [4] and SPDX license URLs [5]. Enhancements will also address the visualization of entities from the ePIC Data Type Registry, improving the intuitiveness of complex FAIR-DO displays. To ensure the pid-component meets user needs, we welcome contributions from the community, particularly in developing new renderers and exploring diverse use cases.

In conclusion, the pid-component is well-positioned to enhance the FAIR Digital Objects ecosystem. Its user-friendly design and modularity enable ongoing development efforts and will drive its effectiveness in facilitating FAIR-DO visualization.

Data availability statement

This work is not based on specific data, since it aims at providing a useful visualization tool for a wide variety of possible input values. As this work represents a software development project, its data consists of the publicly available source code on GitHub (https://github.com/kit-data-manager/pid-component). Release packages are available via GitHub Packages (https://github.com/orgs/kit-data-manager/packages/npm/package/pid-component) and npm (https://www.npmjs.com/package/@kit-data-manager/pid-component).

Underlying and related material

The source code of the pid-component is available in a GitHub Repository: https://github.com/kit-data-manager/pid-component. Release packages are available via GitHub Packages (https://github.com/orgs/kit-data-manager/packages/npm/package/pid-component) and npm (https://www.npmjs.com/package/@kit-data-manager/pid-component). An interactive playground is available to explore the capabilities of the pid-component here: https://kit-data-manager.github.io/pid-component.

Author contributions

Maximilian Inckmann: Conceptualization, Methodology, Software Development, Writing, Visualization

Andreas Pfeil: Conceptualization, Methodology, Supervision, Guidance on Further Development

Thomas Jejkal: Conceptualization, Writing, Guidance on Further Development

Competing interests

The authors declare that they have no competing interests.

Funding

This research has been supported by the research program 'Engineering Digital Futures' of the Helmholtz Association of German Research Centers and the Helmholtz Metadata Collaboration Platform (HMC). We thank all contributors from the Research Data Alliance, the FAIR DO Forum, and HMC as well as potential users for fruitful discussions and feedback.

References

[1] T. Jejkal, “FAIR-DOscope - explore the facets of FAIR digital objects,” 2022, Publisher: Karlsruher Institut für Technologie (KIT). DOI: 10.5445/IR/1000151433. Accessed: Aug. 25, 2023. [Online]. Available: https://publikationen.bibliothek.kit.edu/1000151433.

[2] ORCiD.org. “ORCiD.org public API,” Accessed: Jun. 21, 2024. [Online]. Available: https://info.orcid.org/documentation/features/public-api/.

[3] M. Inckmann. “Npmjs.org - @kit-data-manager/pid-component,” Accessed: Jun. 21, 2024. [Online]. Available: https://www.npmjs.com/package/@kit-data-manager/pid-component.

[4] ror.org. “Research Organization Registry (ROR),” Accessed: Jul. 4, 2024. [Online]. Available: https://ror.org.

[5] spdx.org. “SPDX License List — Software Package Data Exchange (SPDX),” Accessed: Jul. 4, 2024. [Online]. Available: https://spdx.org/licenses/.