Robustifying Links With Zotero

|  |  |
| --- | --- |
| **Martin Klein** | **Shawn M. Jones** |
| *Los Alamos National Laboratory**USA**mklein@lanl.gov*[*https://orcid.org/0000-0003-0130-2097*](https://orcid.org/0000-0003-0130-2097) | *Los Alamos National Laboratory**USA**smjones@lanl.gov*[*https://orcid.org/0000-0002-4372-870X*](https://orcid.org/0000-0002-4372-870X) |
|  |  |

**Abstract – Referencing resources on the web has become an integral part of our digital scholarship. However, the long-term availability and accessibility of these resources has rarely been the focus of significant research and development efforts. In this paper we introduce the Zotero Robust Links extension, a tool that helps authors create archival copies of referenced web resources and offers robustified HTML code that can easily be copied into manuscripts. Our goal with this extension is to provide a tool that helps authors contribute to the integrity of the scholarly record.**

**Keywords – Reference Rot, Web-based Scholarly Communication, Zotero, Web Archiving**

**Conference Topics – creative solutions to shared challenges in daily practice; insight-oriented institutional and personal progress towards digital preservation goals**

# Introduction

References in scholarly articles are essential to provide supporting arguments, complementary information, and pointers to related work. Increasingly, these references have come in the form of HTTP links to resources on the web. A large body of research, for example [1-5], has provided ample evidence that the web is a very dynamic space with resources frequently being created, deleted, and moved. While missing resources and the often encountered infamous “404 - Page not found” error are a detriment to a user’s browsing experience, the issue is just as severe in the realm of scholarly communication. Broken references in scholarly articles are a hindrance to reproducibility and a risk to the integrity of the scholarly record. In previous work [6], we have shown the increasing use of links to web resources and quantified the ratio of links that are subject to link rot (scenario where the link does not work anymore, often resulting in a 404 error). Additionally, we have demonstrated that, over time, a large percentage of links are subject to content drift (scenario where the content of a linked page has changed significantly) [7]. As part of the Hiberlink Project[[1]](#footnote-1), we coined the phrase “reference rot” as the combination of link rot and content drift.

Persistent identifiers (PIDs)[[2]](#footnote-2) such as Digital Object Identifiers (DOIs)[[3]](#footnote-3) have been introduced to address this problem. Jones et al. [8] provided an in-depth analysis as to why DOIs insufficiently address the issue of reference rot in scholarly communications. Klein et al. [9] highlighted inconsistencies with resolving DOIs. Van de Sompel et al. [10] quantified the observed lack of use of DOIs. Aside from these shortcomings, we have shown that authors of scholarly papers frequently reference web resources that do not have a PID assigned [6], such as blog posts, videos, and, more recently, source code [11].

Fortunately, the web archiving landscape offers services that support the long-term availability and accessibility of web resources. Specifically, the Internet Archive’s “Save Page Now” (SPN), archive.today, perma.cc, and megalodon.jp are examples of proactive web archiving services that allow authors to create archival copies of the web resource they intend to reference. Our prior work [8] notes that some authors attempt to address the reference rot issue by first creating an archival copy. These authors then use that archival copy’s URI in their reference rather than using the URI of the original live web resource. A popular example is the Internet Archive’s link rot bot for Wikipedia[[4]](#footnote-4). However, this approach relies on the permanent existence and availability of one web archive, the one at which the archival copy was created - a scenario in which one link rot problem was merely replaced with another, as many types of disasters, technical, financial, and political, can befall individual web archives.

Analyzing this problem space motivated our creation of the “Robust Links” concept [12], which consists of two main steps:

1. proactive creation of an archival snapshot of the web resource to be referenced and
2. robustifying the reference by enhancing the HTML link with defined attributes.

The first step creates a version of the resource that is representative of what the author intends to reference and the second provides a number of fallback mechanisms in case the resource on the live web is not accessible (link rot), its content has changed (content drift), or its archival copy is unavailable from the one archive it was created in. For a more detailed overview of the Robust Links concept and its HTML attributes we refer to the Robust Links specification[[5]](#footnote-5).

To further the adoption of the Robust Links concept, we implemented a web service[[6]](#footnote-6) that supports users in conveniently executing the two steps outlined above [8]. The web service builds upon an API we designed and implemented[[7]](#footnote-7) to also allow for creating robustified links in bulk.

While the Robust Links web service targets individual scholars and authors of web-based manuscripts, it represents yet another system that needs to be included in the research life cycle and in the suite of tools researchers utilize in their daily work. We noticed from discussions with our target audience that there is a strong desire to incorporate such preservation and robust linking approaches into existing and widely used tools. Zotero, the popular open source reference manager, had been mentioned as an example of such tools[[8]](#footnote-8).

In this paper we introduce our Zotero Robust Links extension. We walk through its functionality and intended use and discuss some aspects of improvements planned for future work. With this paper and the introduced software extension, we hope to further raise awareness about the problem of reference rot in scholarly communication and highlight one possible path to address the issue - by robustifying links with Zotero.

# Zotero and Robust Links

A detailed overview of Zotero’s functionality is beyond the scope of this paper but we refer to its documentation[[9]](#footnote-9) for a thorough overview. Generally speaking, Zotero comes as a desktop application, a web service, and a browser extension. Our extension is designed for the desktop application and we therefore focus our elaborations on this environment.

## Adding Items

Zotero allows the ingestion of an item by its identifier, specifically a DOI, ISBN, PMID, or arXiv ID. Upon pasting the identifier into the search bar, Zotero connects to the hosting platform as well as 3rd parties such as Crossref to gather all available metadata about the to be ingested item. It then automatically populates the corresponding metadata fields for the created record, visible in the Zotero “Item Pane”. For a typical journal article this includes authors, title, date, DOI, and the URI. As is common with many reference formats, Zotero populates the DOI field with the DOI PID string and not the HTTP DOI. Unfortunately, in these cases, the URL field is populated with a paper’s landing page (final link of the DOI resolution redirect chain) rather than its HTTP DOI URL. As some reference formats only support either a DOI or a URL, this behavior can lead Zotero users to unknowingly prevent DOI adoption.

Zotero also allows users to add new items manually to a collection. For example, to add a blog post, the user can click the “New Item” button and select “Blog Post” as the type of item to be added. It is then up to the user to manually add appropriate and corresponding metadata to the Item Pane, including the URI of the resource.

The Zotero browser extension offers a third way to add items to a collection. By displaying a resource in the browser and clicking the “Save to Zotero” button, a user can add the displayed resource to a selected Zotero collection. The ingestion process works automatically, very similar to the adding by identifier method, with all available metadata fields populated.

Figure 1. Zotero Collection (left), Items (center), Note (right)

## Robust Links in Zotero

By default, the Zotero Robust Links extension [13] automatically creates an archival copy of any item ingested via its identifier or via the browser extension, as long as Zotero had populated the URI field with a valid URI. It takes the URI and submits it to the Robust Links API, which, in turns submits the URI string to one of the web archiving services mentioned above. For manually created items, the Robust Links extension needs to be triggered manually. With a right mouse click on the item of interest, a user has the option to “Robustify this resource” and pick the default, any, or a specific web archive for the proactive archiving of the resource. In either event, the extension displays a notification with details regarding the URI and the web archive it is submitted to. The notification disappears after 5 seconds or after the user clicks on it.

## Robustified Links

Once the archiving process is complete, the extension again shows a notification on the screen with a message indicating success (or failure). This notification also disappears after 5 seconds or a mouse click. If an error occurs, the error message will remain and the user needs to click it to dismiss it.

Each item for which the extension has created a Robust Link now has a Zotero Note as a child node. The note, hierarchically aligned below the item, is named “Robust Link”. It contains the original URI of the resource, the URI of its created archival record as returned from the utilized web archive, and instructions on how to robustify the HTML link in a manuscript, satisfying the second step of creating a Robust Link mentioned above. From these instructions, a user can simply copy and paste the robustified HTML code into a manuscript.

Fig. 1 shows a screenshot of the Zotero main window. All collections are shown on the left in the “Collection Pane”. We have created a collection called “iPres2022” that contains all references from this paper. The items and their corresponding notes are shown in the center of Fig.1 and the content of one of the Robust Link notes is shown in the “Item Pane” to the right.

## Extension Configuration

The Zotero Robust Links extension comes with a number of default settings upon installation on a system. Via the “Tools” menu item, a user can access the configuration panel, displayed in Fig. 2. Recall that references added to Zotero will be automatically robustified if they contain URLs. The user can disable this functionality by toggling the corresponding setting on top of the panel. The extension’s default archive is “random”, which means the Robust Links API will randomly create a copy at either of the two currently available web archives (Internet Archive and archive.today). A user may prefer a specific archive, and thus can specify their preference. If this setting is left unchanged, the options of “default” and “any” web archive, mentioned in the manual processing above, are identical. The setting on the bottom of the configuration panel is aimed at advanced users. By default, the extension asks the Robust Links API to check if a submitted resource is already an archived version before creating a new one. With this setting, a user can ask the API to bypass this step and just create a new archival snapshot, potentially saving time.

## Exporting Robustified Links

Zotero supports the export of all or individual items in a collection. For example, a user can right-click on a collection in the “Collection Pane” and export the entire collection (all items in the collection). Alternatively, a user can select one or more items and export them with a right click. The export panel, shown in Fig. 3, prompts the user to pick a format. Behind each of these formats is a corresponding Zotero “Translator.” We maintain a special translator for Robust Links [13] that a user can install separately from our extension. This Robust Links translator allows Zotero to export items into an HTML file that contains each item’s corresponding robustified link. For demonstration purposes, we exported our entire iPres2022 collection containing all references of this paper, and made it publicly available[[10]](#footnote-10).

# Future Work And Concluding Remarks

Limitations to the extension remain. For example, Zotero allows for the creation of a bibliography of items with a chosen citation style, for example, that of the American Chemical Society. In the future we will analyze various citation styles and assess how to incorporate robustified links. Currently the extension only supports two web archives. We are negotiating the addition of other web archives, such as perma.cc, to increase the chances for each reference’s long-term availability. Lastly, Zotero Notes are currently the best way to convey robustified HTML. In future Zotero versions, it may be possible to actively edit metadata in the “Items Pane”, potentially offering better options to inform the user about the robustified links.

Figure 2. Configuration Panel.

Our goal is a tool for manuscript authors that can help them be better stewards of their references and scholarly articles at large. With our extension to the popular reference manager Zotero, we aim to meet the researchers where they are rather than creating yet another tool for them to learn. We hope that adoption of our extension will help robustify more links to support the integrity of the scholarly record.

# REFERENCES

1. Norman Matloff. 2005. Estimation of internet file-access/modification rates from indirect data. ACM Trans. Model. Comput. Simul. 15, 3 (July 2005), 233–253. DOI:<https://doi.org/10.1145/1103323.1103326>
2. Junghoo Cho and Hector Garcia-Molina. 2003. Estimating frequency of change. ACM Trans. Internet Technol. 3, 3 (August 2003), 256–290. DOI:<https://doi.org/10.1145/857166.857170>

Figure 3. Export panel.

1. Alexandros Ntoulas, Junghoo Cho, and Christopher Olston. 2004. What's new on the web? the evolution of the web from a search engine perspective. In Proceedings of the 13th international conference on World Wide Web (WWW '04). Association for Computing Machinery, New York, NY, USA, 1–12. DOI:<https://doi.org/10.1145/988672.988674>
2. Carrie Grimes, Daniel Ford. 2008. Estimation of Web Page Change Rates. <https://storage.googleapis.com/pub-tools-public-publication-data/pdf/34428.pdf>
3. Sanasam Ranbir Singh. 2007. Estimating the Rate of Web Page Updates. <https://www.aaai.org/Papers/IJCAI/2007/IJCAI07-462.pdf>
4. Klein M, Van de Sompel H, Sanderson R, Shankar H, Balakireva L, Zhou K, et al. (2014) Scholarly Context Not Found: One in Five Articles Suffers from Reference Rot. PLoS ONE 9(12): e115253.<https://doi.org/10.1371/journal.pone.0115253>
5. Jones SM, Van de Sompel H, Shankar H, Klein M, Tobin R, Grover C (2016) Scholarly Context Adrift: Three out of Four URI References Lead to Changed Content. PLoS ONE 11(12): e0167475.<https://doi.org/10.1371/journal.pone.0167475>
6. Jones, Shawn M., Martin Klein, and Herbert Van de Sompel. “Robustifying Links To Combat Reference Rot.” The Code4Lib Journal, no. 50 (February 10, 2021).<https://journal.code4lib.org/articles/15509>.
7. Klein, M., Balakireva, L. An extended analysis of the persistence of persistent identifiers of the scholarly web. Int J Digit Libr (2021).<https://doi.org/10.1007/s00799-021-00315-w>
8. Herbert Van de Sompel, Martin Klein, and Shawn M. Jones. 2016. Persistent URIs Must Be Used To Be Persistent. In Proceedings of the 25th International Conference Companion on World Wide Web. Geneva, CHE, 119–120. DOI:<https://doi.org/10.1145/2872518.2889352>
9.  One in Five arXiv Articles Reference GitHub. <https://ws-dl.blogspot.com/2022/02/2021-02-23-one-in-five-articles.html>
10. Van de Sompel H., Klein M., Shankar H. (2014) Towards Robust Hyperlinks for Web-Based Scholarly Communication. In: Watt S.M., Davenport J.H., Sexton A.P., Sojka P., Urban J. (eds) Intelligent Computer Mathematics. CICM 2014. Lecture Notes in Computer Science, vol 8543. Springer, Cham.<https://doi.org/10.1007/978-3-319-08434-3_2>
11. Jones, S., & Klein, M. (2021). Zotero Robust Links Extension: Create Robust Links from within Zotero (Version 2.0.0-20210419153723) [Computer software].<https://github.com/lanl/Zotero-Robust-Links-Extension>
1. <http://hiberlink.org/> [↑](#footnote-ref-1)
2. <https://www.dpconline.org/handbook/technical-solutions-and-tools/persistent-identifiers> [↑](#footnote-ref-2)
3. <https://www.doi.org/> [↑](#footnote-ref-3)
4. [https://en.wikipedia.org/wiki/Wikipedia:Link\_rot](https://en.wikipedia.org/wiki/Wikipedia%3ALink_rot) [↑](#footnote-ref-4)
5. <https://robustlinks.mementoweb.org/spec/> [↑](#footnote-ref-5)
6. <https://robustlinks.mementoweb.org/> [↑](#footnote-ref-6)
7. <https://robustlinks.mementoweb.org/api-docs/> [↑](#footnote-ref-7)
8. <https://trello.com/c/5Jk5bbxz> [↑](#footnote-ref-8)
9. <https://www.zotero.org/support/> [↑](#footnote-ref-9)
10. <https://robustlinks.mementoweb.org/zotero/iPres2022.html> [↑](#footnote-ref-10)