COPIM •

Using PubPub for scholarly output: Import, Collaboration, Citations, and Zotero

Tobias Steiner

Published on: Jul 17, 2020

DOI: 10.21428/785a6451.f72bab42

License: [Creative Commons Attribution 4.0 International License \(CC-BY 4.0\)](https://creativecommons.org/licenses/by/4.0/)

Here at COPIM, we are more and more getting into the groove of using PubPub for all sorts of purposes, and the team is developing a growing body of [reports](#) around the variety of [workshops that have already taken place](#).

Next to these concise overviews, COPIM's [Work Packages](#) are now engaging in using PubPub for writing more long-form research reports. In that context, while referencing other scholarly works is established best practice in all fields of academia, the actual juggling of footnotes, citations and references can be an art form in itself. □

With that particular perspective in mind, I wanted to share our experiences using PubPub as a platform to create scholarly output that includes citations, reference lists, and footnotes in conjunction with [Zotero](#), an open source reference management tool.

The following segments will be dealing with issues that came up during our early stages in working with PubPub, including questions around

- importing pre-existing research documents,
- dealing with in-text citations from those imported documents,
- using reference management tools such as Zotero in conjunction with PubPub,
- collaborating with colleagues, and keeping track of changes

One-Minute Intro: The Pub Editor

I think it is safe to say that PubPub boasts a very intuitive text editor, with a majority of the text formatting features being quite self-explanatory. If you're looking for more detailed info on the editor's basic functionality, the [PubPub team has you covered](#).

Fig. 1: Text formatting with the Pub editor: Tool ribbon and pop-up elements.

Using these different pop-up options, you can quickly add **bold** and *italic* text, have

- bullet points / an unordered list

as well as

1. numbered /
2. ordered list items,

insert links to other content, include a

citation paragraph

and even add lowercase or uppercase segments, have ~~striketrough-text~~ and add code that is styled differently from the main text.

I won't go into much detail on how to add tables and insert different forms of audiovisual content through the Media option; just to note that you can insert files into the document by having them hosted locally through an internal uploader (video, audio, image, other document files), while also having the option to embed external content directly from YouTube, Soundcloud, Vimeo, Codepen, or a generic embed to include other external content¹.

Preparations: Things worth considering before actually starting on a longer writing process

Maybe first things first: those of us accustomed to using reference management tools such as EndNote, Citavi, or Zotero in direct conjunction with the text editors of our choice might initially be a bit frustrated to learn that the comfort of directly calling up your reference manager for a given citation doesn't (yet?) work in the PubPub environment.

In the following, we will use [Zotero](#) as a test case reference management system, because it perfectly [aligns with our vision of scholar-led non-profit open source infrastructure](#).

Fig. 2: Zotero client view of the [ScholarLed / COPIM library](#).

Note: Having a Zotero library to support your writing process is not only a means to stay organized and have all your source data ready and sorted at your fingertips. Working with Zotero, you can also have the bibliographic dataset available to share with [other members of your research group](#) or even [publicly on the internet](#) in the spirit of #opendata and #openculture.

Given that you have an existing draft version of a report ready to be imported into your new Pub, the best preparation is

1. to quickly check if all references used in the document are listed in your Zotero library,
2. and then unlink the draft document from Zotero (see “Unlink Citations” in the [Zotero manual](#)). Doing so removes any field codes that won't be of use in PubPub anyway.

Caveat: Handling of citation data in PubPub - structured data, DOIs, and BibTeX

PubPub's approach to citations is quite extraordinary, because it has the use of linked, semantically-structured data at the basis of its concept. Such an approach would in theory allow you to use e.g. a simple DOI (or other persistent identifiers/[PIDs](#) — next to DOIs, PubPub takes BibTeX, wikidata URLs, and bibjson) as input for a given citation, and PubPub's rendering engine would then automagically look up the reference details behind that DOI and be able to render a full citation from that -

simply because these kind of PIDs usually carry all the metadata necessary to fill in the gaps (author, date, title, ...) that we as authors would otherwise collate manually.

Now, while in theory this is a beautiful idea and setup, and does neatly work with shorter text, we found that at the time of writing, this can lead to erroneous behaviour of in-text renderings when you have a document with, say, 50+ citations / references (our use case counted 154 in-text citations).

Next to that, the sometimes insufficient quality of metadata returned through these DOIs meant that we had to revisit bits and pieces in a given citation.

Also, many of the sources we have been referencing — including reports, documentation, press releases, and other grey literature — do not have DOIs assigned to them, which makes it difficult to have them included when solely relying on that kind of PID.

Thankfully, PubPub also offers other kinds of identifiers that can be used to get citation data into the system in a structured way. Our approach thus has been to go with **BibTeX** as input format to use for citations, because this has allowed us to control data quality as well as the range of resources to cite, and so far, this has been working flawlessly.

Getting citations into PubPub

Coming back to our earlier-described use case of a preexisting draft document, we then save that document as an Open Document or Word file, and use the import option that is displayed when you start up a new pub.

Working in parallel, we open the Zotero library client and either work on a citation-by-citation basis, using Zotero's [Quick Copy option to generate](#) a BibTeX entry of the reference we want to cite.

```
@misc{barnesCostOpenAccess2020,
  title = {The Cost of {{Open Access}} Books: A Publisher Writes},
  shorttitle = {The Cost of {{Open Access}} Books},
  author = {Barnes, Lucy and Gatti, Rupert},
  year = {2020},
  month = may,
  doi = {10.11647/OBP.0173.0143},
  abstract = {OBP lays out our costs and revenue for the last financial year (2018 - 2019), to add some numbers to d
  howpublished = {http://doi.org/10.11647/OBP.0173.0143},
  journal = {Open Book Publishers Blog},
  language = {en}
}
```

Fig. 3: BibTeX export sample from the [ScholarLed / COPIM Zotero library](#).

Alternatively, you can create a full BibTeX export of all entries in your Zotero bibliography and then use that as the basis for inserting citations in PubPub².

In the general settings tab of your Pub, you can define a citation style that the rendering engine will use for citations.

Fig. 4: Pub settings, segment “Citation Style”

Returning to the main text editor, you can use the “Citation” feature

Fig. 5: Text formatting tool ribbon, with the “Citation” tool highlighted.

to insert a source that you want to cite from. As mentioned earlier, you can use different kinds of automated input. My personal choice was to go with BibTeX.

Fig. 6: The “**Structured Data**”, “**Rich Text**” and “**Inline Format**” fields allow you to adjust the way a citation is displayed in the text.

Having copied your source’s BibTeX snippet from Zotero, you open the “Citation” dialogue (see Fig. 6) and paste that snippet into the “Structured Data” field. This will quickly get pre-rendered in the “Preview” field and allows you to see the actual citation output that will then be displayed in the main text body when you hover your pointer over the inline citation.

To modify how the citation is displayed in the main text, you can

- either go with the overall citation output format as predefined via your Pub settings (see Fig. 4)
- or override this via the “Custom” option, which allows you to adapt the citation anchor with customized text to include elements such as “see, cf., e.g., ...” like in this example: [see PubPub Help, “Editor”, 2018](#).

Using the “Custom” as well as the “Rich Text” options can prove particularly useful in cases when you get seemingly nonsensical previews from BibTeX input, e.g. when you have a source `Internet Archive (2020)`, and the system would then output this as `Archive, I. [...]`. For these cases, you can skip the BibTeX input and use the “Rich text” field to define how the citation itself will look like, and then go ahead and tweak the inline placeholder for this citation.

Note: Allowing for such a high level of freedom in editing your citations might seem a bit complicated at first, but actually is quite an elegant solution: enabling the user to define whatever they want to put in, the Pub editor is thus able to cater for a larger variety of different scholarly backgrounds and corresponding variety of citation practices.

Working on more long-form texts, you will quickly notice how the amount of citations keeps growing and growing. To get a full count and list of all citations used in your Pub, scroll down to the bottom of your text.

Fig. 7: Metadata field: Citations, License, and Comments

Here, you can find parts of the existing metadata available on your text that comprise info on Citations, License, and Comments. Clicking on each of the segments will display more detailed information. The Citations counter then reveals all citations used in the text, and clicking on the $\leftrightarrow$ symbol at the end of each entry lets you jump to the in-text position of that citation.

Footnotes

Input of footnotes is quite similar to citations; you can also include structured data à la BibTeX or DOIs, or use the Rich Text editor field to define custom text.

Using the Rich Text field, you can also add links to external content (see Fig. 8). Note that at the time of writing, combining footnote text plus an extra citation within the footnote doesn't seem to work, as you can't go full *Inception* (yet?) and open the citation dialogue within the footnote dialogue ☐

Fig. 8: Footnotes can take regular text, structured data input, or a combination of both. Note that in case of combining regular Rich Text with structured data, the structured data gets rendered first, then followed by text input.

Collaboration: working on a Pub with others

Collaborative editing is one of the key features of the Pub editor. For more details on how to collaboratively work on a given document, see the Knowledge Future Group's blog written by Heather Staines on [how KFG uses PubPub as a GoogleDoc alternative](#).

Fig. 9: Collaborative editing of a Pub (source: [PubPub](#))

Overall text rendering in PubPub is realized along the principles of version control that some might know from git-based workflows.

Fig. 10: schema of a git-based versioning process (source: [CodingAPI](#))

The benefit of this is that you can always revert to an earlier version of the text, and also track changes that are made by other collaborator invited to work on the text with you.

This feature is particularly useful if you want to continue working on a Pub in the sense of a living document. The base text can iteratively grow and evolve over time. Making use of versioning, you can then also “branch out” and use this for publishing a time-stamped iteration of that Pub, while the overall document can continue to evolve even after this particular branch has been released. For COPIM, this is a crucial element as we are working on a variety of reports that include scoping activities, and using such a living document approach will help us to create published, time-stamped output as well as allowing us to develop our research further over time.

Fig. 11: Pub History, displaying different versions of a draft or published version.

For more on what COPIM is working on, visit our PubPub community at copim.pubpub.org.

Photo by [Kelly Sikkema](#) on [Unsplash](#)

Footnotes

1. Note that PubPub adheres to common web etiquette standards, so third-party content that forbids [hotlinking](#) through its media settings won't be displayed in your Pub. ↵

2.

Note that Zotero's generic BibTeX export stores very detailed output of all the information available in Zotero — including where a given reference has stored its PDF file on your local harddisk (see the {file} element in your BibTeX export).

To ensure that only the data that is actually required gets included in the export, you might want to consider using the [Better BibTeX for Zotero](#) plugin. ↵

Citations

1. Pub Editor. (2018). *PubPub Help*. Retrieved from <https://help.pubpub.org/pub/pub-editor-reference> ↵