# Making Access More Seamless with Access Broker Browser Extensions

## Introduction

In the previous two chapters, we introduced some of the issues users face when trying to access resources via institutional access; in this chapter, we cover one of the major classes of solutions to this problem—providing access via browser extensions. These browser extensions, which are sometimes called access broker browser extensions, have become very popular in recent years,[1] and most academic libraries officially support them via subscriptions. Some popular examples include Lean Library, LibKey Nomad, EndNote Click (formerly Kopernio), and free alternatives like Google Scholar Button and Lazy Scholar.

*Lean Library*
https://www.leanlibrary.com

*LibKey Nomad*
https://thirdiron.com/products/libkey-nomad

*EndNote Click*
https://click.endnote.com

*Lazy Scholar*
http://www.lazyscholar.org

## The General Concept of Access Broker Browser Extensions

As seen in chapter 2, one of the major barriers to achieving access occurs when the user is off campus and trying to check for access to a resource when not starting on a library-controlled website. Let us use the example from chapter 2 again, where a user tries to access an article directly from JSTOR. Perhaps they are given the direct link in an e-mail or a blog post and as a result, they land on the page, but the system is not able to easily identify who the user is or whether they have the rights to access the article. As we explained in the section Off-Campus Problem: IP Recognition and Proxy Servers in chapter 2, because access to the licensed resource is usually granted via IP recognition and the users are off campus, they will not have the right IP to be granted access (see figure 3.1).

Let us assume for the moment that the user does not use any of the SAML methods, which are covered in chapter 4. In most cases, to get around this problem, the user would need to waste time going back to the library home page to look for the article. Sadly, the user may often give up instead of trying to access it directly through the library web page.

The user experience in this scenario improves greatly and even has a changed outcome when the user has a particular kind of browser extension installed. The improvement happens because the browser extension can easily integrate into their workflow and, with a single click (or even automatically), provide the user access to the full-text article without making the user jump through hoops. The browser extension can accomplish this using a variety of methods in the background: for example, by appending the EZproxy stem to the page the user is on so that access is granted via the proxy or by guiding the user to sign on via SAML methods. The browser extension is not limited to just helping the user gain access to the article on the platform or site, but it can also provide a solution to the appropriate copy problem (see the section Appropriate Copy Problem Explained in chapter 2) by using the metadata of the requested article to direct the user to appropriate copies that may not be on JSTOR.

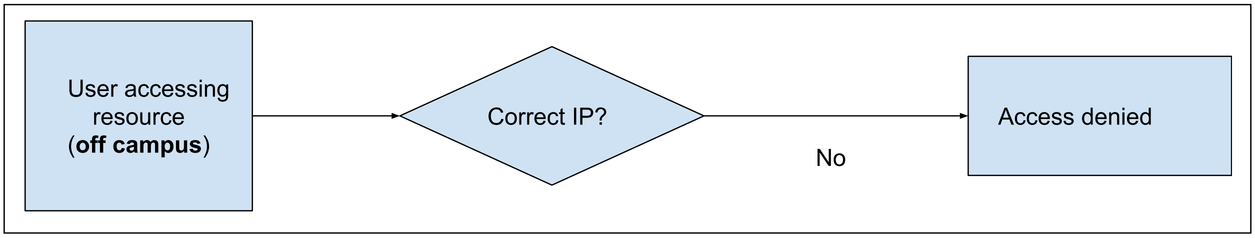


**Figure 3.1**
Prevented from accessing library resources when off campus

Regardless of the method employed, what is important here is that the browser extension (once installed) is ever-present in the user's browser, providing a convenient point for helping the user gain access.

## Proxy Bookmarklets, LibX Toolbars, and Unpaywall: Some Predecessors

The idea of libraries using browser extensions to assist users is not a new idea. During the heyday of Library 2.0 in the early and mid-2000s, librarians experimented with custom toolbars such as conduit toolbars, LibX toolbars, Greasemonkey scripts, and other similar ideas. These were tools that users installed on their browsers that would provide various library-related functionality to assist them on the web pages they were on.

For instance, the LibX toolbar, when integrated with your library catalog service, would convert DOIs and ISBNs on web pages into clickable links that, when clicked, would do a search of your system.[2] If set up correctly with the library's Summon discovery service, the user would receive an overlay with the results from the Summon discovery service displayed right on the web page when they hovered their mouse over such links! However, custom toolbars quickly became associated with malware, adware, and spyware, and they soon fell out of favor everywhere. In this section, we will discuss two more of these predecessor tools—the proxy bookmarklet and Unpaywall, which first introduced some of the functionality found in the current generation of access broker tools.

### Proxy Bookmarklet

The original inventor of the concept of the proxy bookmarklet tool is unknown, but by 2010, variants of this tool with varying names could be found mentioned on many academic library sites. Figure 3.2 provides an example of the instructions given to set up the proxy bookmarklet for the author's institution.

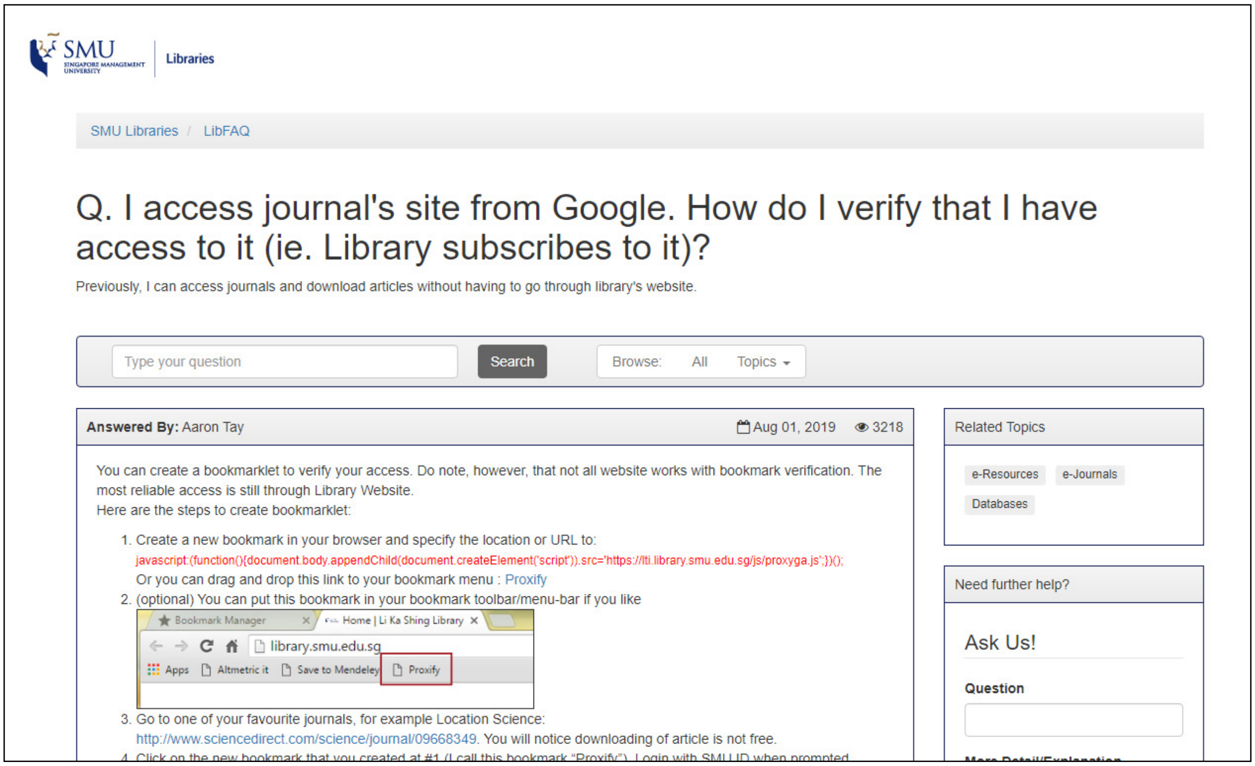
But what did the proxy bookmarklet do and how did it work? Simply put, when you clicked on the bookmarklet, it would use JavaScript to append the EZproxy string to the URL of the page you were on. As we saw in chapter 2, doing this would allow you to reroute your access via the proxy server, giving you access to the item (if your institution had access, of course; see this workflow in figure 3.3).

As I noted in a presentation in 2013, when heavily promoted, the proxy bookmarklet technique can be very popular.[3] However, there are quite a few drawbacks that make promoting it difficult. First, the concept of bookmarklets is not a mainstream idea, and installing them is not intuitive. Though most libraries create guides on how to use bookmarklets, in my experience, many students struggled to install it without individual guidance from a library worker.

Second, the proxy bookmarklet does not work automatically; users have to remember to click on it to activate it. As users are not given any indication when they can use the proxy bookmarklet to access resources, they will not know in advance if clicking on the bookmarklet will allow them access; a novice user might try using it on every web page, and this can lead to a lot of frustration if they click on it and find they have no access most of the time.

Third, as we have seen, the proxy bookmarklet works by appending the proxy to a URL, which gives users access to resources via EZproxy. As we saw in chapter 2, this will allow the platform to recognize that the user is from the appropriate institution with the corresponding access rights. For example, clicking on the proxy bookmarklet on the JSTOR home page would only authenticate the user to JSTOR via their institution. It is critical to note that the institution may not have licensed access to every piece of content on the JSTOR platform, so even after this process, the user might still be frustrated that they can't access this specific JSTOR article. This often leads to confusion and frustration for the user, as they have already signed on to the JSTOR platform and yet still do not have access to the content. As you will see later, an article-level approach might work better.

Last, the proxy bookmarklet alone does not handle the appropriate copy problem. This can cause users of the bookmarklet to think that they do not have access to the content when they may be able to access it via alternative platforms. This is particularly problematic



SMU Singapore Management University | Libraries

SMU Libraries / LibFAQ

Q. I access journal's site from Google. How do I verify that I have access to it (ie. Library subscribes to it)?

Previously, I can access journals and download articles without having to go through library's website.

Type your question | Search | Browse: All Topics

**Answered By:** Aaron Tay | Aug 01, 2019 | 3218

You can create a bookmarklet to verify your access. Do note, however, that not all website works with bookmark verification. The most reliable access is still through Library Website.
Here are the steps to create bookmarklet:

1. Create a new bookmark in your browser and specify the location or URL to:
javascript:(function(){document.body.appendChild(document.createElement('script')).src='https://lti.library.smu.edu.sg/js/proxyga.js';})();
Or you can drag and drop this link to your bookmark menu : Proxify
2. (optional) You can put this bookmark in your bookmark toolbar/menu-bar if you like
3. Go to one of your favourite journals, for example Location Science:
http://www.sciencedirect.com/science/journal/09668349. You will notice downloading of article is not free.
4. Click on the new bookmark that you created at #1 (I call this bookmark "Proxify"). Login with SMU ID when prompted

Related Topics
e-Resources
e-Journals
Databases

Need further help?
Ask Us!
Question
More Detail/Explanation

**Figure 3.2**
Instructions for setting up the proxy bookmarklet at the author's institution

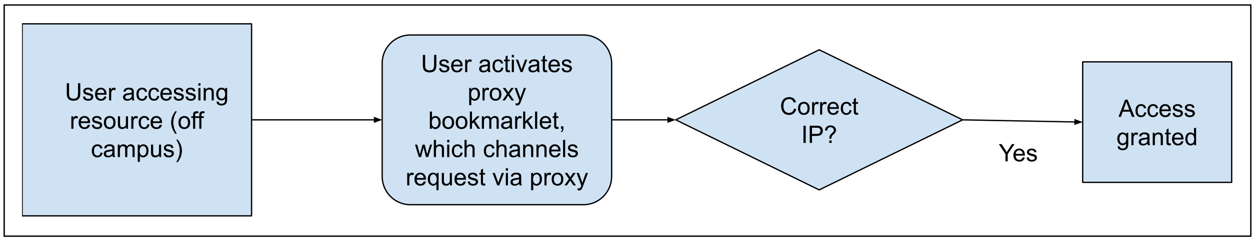

User accessing resource (off campus)
User activates proxy bookmarklet, which channels request via proxy
Correct IP?
Yes
Access granted

**Figure 3.3**
Granted access via proxy bookmarklet when accessing library resources off campus

for institutions that heavily provide access via aggregators like EBSCO or Ovid because such content might be missed. As you will see, the current generation of access broker browser extensions remedy some, if not most, of these problems.

## Unpaywall Browser Extension

Clearly, one obvious improvement would be to convert the proxy bookmarklet to a browser extension, which would make it easier to install, use, and update and to track usage. This is where browser extensions like Unpaywall come in.

First appearing in 2017, browser extensions such as Unpaywall, Open Access Button, and CORE Discovery are designed to direct users to open-access versions of articles. They work by examining the metadata of the web pages users are on for article identifiers (e.g., DOI and PMID). This metadata, or article identifiers, is then used to find open-access copies, and if one is available, the user is directed to the full-text resource.

*Open Access Button*
https://openaccessbutton.org

*CORE Discovery*
https://core.ac.uk/services/discovery

In other words, Unpaywall and other similar tools worked on an item- or article-level approach to identify the article needed and check if a full-text copy was available, *no matter where it was*. As already alluded to, this solves part of the appropriate copy problem.

**Table 3.1.** Current access broker browser extensions

| | Google Scholar Button | Lazy Scholar | Lean Library | LibKey Nomad | EndNote Click |
|---|---|---|---|---|---|
| Vendor | Google | Colby Vorland | Lean Library, a Sage publishing company | Third Iron | Clarivate |
| Business model | Free | Free | Subscription | Subscription | Freemium |
| Year launched | 2015 | 2013 | 2016 | 2019 | 2017 |
| Installations of Chrome extension as of Dec. 31, 2021 | 3 million+ | 10,000+ | 100,000+ | 200,000+ | 1 million+ |
| Authentication supported | Same as Google Scholar Library Links program | EZproxy | EZproxy, OpenAthens, Shibboleth | EZproxy, OpenAthens, Shibboleth | EZproxy, OpenAthens, Shibboleth |
| Setup required by library | None if you are already set up in Google Scholar | None | Some | Some, minimal if you have BrowZine set up | Minimal |
| 1 click to PDF | No | No | No | Yes | Yes |
| Supports DDS/ILL | No | No | Yes | Yes | Yes (with institutional bundle) |
| Some unique points | Free PDFs via Google Scholar | Recommendations, citation metrics | Supports e-book access<br>Workflow integration with Springshare LibGuides, EDS, Summon, Primo, and Scite<br>Widest support across different browsers | Integration with Retraction Watch Database<br>LibKey integration with Wikipedia and selected databases such as PubMed, Scopus, Web of Science | PDF cloud storage<br>COUNTER-compliant dashboard analytics (for institutions) |

Also, unlike the proxy bookmarklet, Unpaywall and similar extensions work automatically and are activated whenever a free copy is found, without any input from the user, avoiding all the issues mentioned in the discussion of the proxy bookmarklet. Of course, as popular as browser extensions like these are, they could only bring the user to open-access copies and do not completely solve the appropriate copy problem.

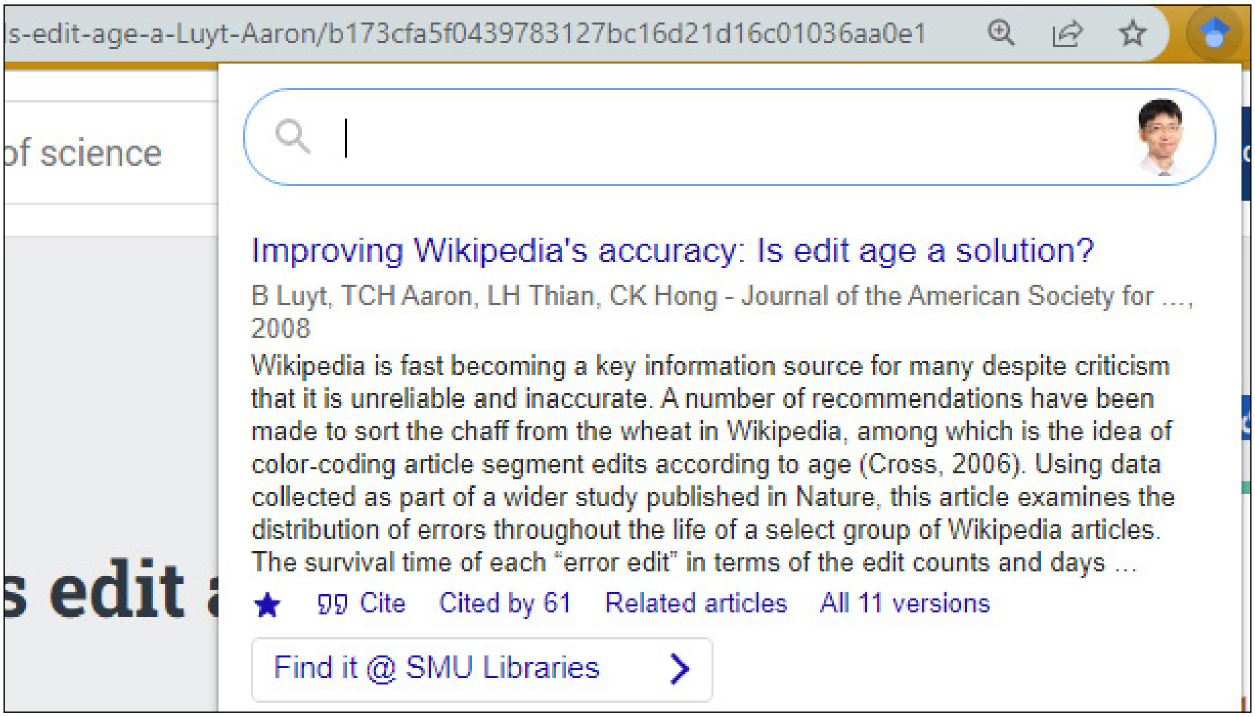
The access broker browser extensions described in the next section attempt to solve the issue by using the same idea, but instead of just trying to find open-access copies, they also try to find institutionally accessible copies. In fact, the modern access broker browser extension typically works like this for access to journal articles:

1. It automatically looks for article identifiers or other metadata of the item on the page the user is on.
2. If an identifier is found, it checks if the user has access to the item via institutional access and, if so, pops up a badge with a link to the full-text copies.
3. If an identifier isn't available, the user is directed to open-access copies if any are available.
4. If all these methods fail, the user is typically given additional options via the library link resolver, which usually include interlibrary loan or document delivery service (ILL/DDS).

Depending on the access broker in question and the setup, the authentication and authorization process for each user may occur differently, which is something we will discuss further in each product's section. Note that such an article-level approach works only for access to full-text journal articles but may not help with access to online resources such as abstracting and indexing databases or newspaper databases. Some access brokers, such as Lean Library, do provide alternative ways to support access.

## Current Access Broker Browser Extensions

Today it is common for academic libraries to subscribe to one or more commercial access broker extensions while providing support for other free versions. The market for such products is developing very quickly. Table 3.1 provides a brief look at current options available. For a more robust list, please refer to my comparison page of access broker browser extensions, which is actively updated.



s-edit-age-a-Luyt-Aaron/b173cfa5f0439783127bc16d21d16c01036aa0e1

f science

s edit a

Improving Wikipedia's accuracy: Is edit age a solution?

B Luyt, TCH Aaron, LH Thian, CK Hong - Journal of the American Society for …, 2008

Wikipedia is fast becoming a key information source for many despite criticism that it is unreliable and inaccurate. A number of recommendations have been made to sort the chaff from the wheat in Wikipedia, among which is the idea of color-coding article segment edits according to age (Cross, 2006). Using data collected as part of a wider study published in Nature, this article examines the distribution of errors throughout the life of a select group of Wikipedia articles. The survival time of each "error edit" in terms of the edit counts and days …

★ 99 Cite Cited by 61 Related articles All 11 versions

Find it @ SMU Libraries >

**Figure 3.4**
An example of Google Scholar Button pop-up when the user clicks on the extension

*Comparison of Access Broker Browser Extensions*
https://musingsaboutlibrarianship.blogspot.com/2019/07/a-comparison-of-6-access-broker-browser.html

### Google Scholar Button

- **Pros:** Free. Minimal setup if already in Google Library Links program. Google Scholar is known to be one of the most capable tools for finding open-access articles, and this carries over to the Google Scholar Button extension.
- **Cons:** User has to click to activate. There is no support for ILL/DDS when an article is not available.

Looking for an easy-to-use and free browser extension? Google Scholar Button fits the bill! First launched by Google in 2015, this easy-to-use tool is by far the most popular access broker browser extension on the list with over three million installations! There is a possibility your researcher might even already be using it.

How does it work and how do you set it up? The only requirement for the Google Scholar Button to work with your institution's holdings is that your library is set up for Google's Library Links program.[4] It is highly likely that your library has already done so. Often, libraries set up a monthly update to upload library holdings to Google Scholar to participate in the program. Assuming the library is set up for Library Links, researchers can turn on this feature for their institution by setting up Google Scholar in the Library Links settings. This process will result in links to full text available via their institution appearing next to results in Google Scholar.

This feature works well, and it is often taken for granted by the millions of researchers who use Google Scholar. But one issue is that this feature works only on Google Scholar, so it will not help researchers who land on any other page. This is where the Google Scholar browser extension comes in. By installing it, you can bring this feature along with you. When you land on any article landing page for an article you cannot access, all you need to do is to click on the Google Scholar Button icon and the browser extension will attempt to

- check if there is an article title on the page you are in,
- send the title over to search Google Scholar, and
- if one or more articles match in Google Scholar, it will pop up and display the usual article-level linking made available by the Library Links programs.

On top of the links provided by the Google Scholar Library Links program, the user will also be shown the free PDFs found by Google Scholar. This helps supplement the copies available via institutional access (see figure 3.4 on p. 25 for an example). While other browser extensions have the same feature by utilizing external services such as Unpaywall, CORE OA Discovery, and so on, Google Scholar Button is often capable of finding additional free-to-read copies, using Google Scholar's own excellent open-access-finding capability.

Compared to the other access broker browser extensions, Google Scholar Button is missing two major features. First, this is the only access broker browser extension that requires the user to remember to click to activate the full-text-finding feature, while others do it automatically on every relevant page. Second, almost all the other browser extensions provide an option to a user (typically a route to ILL/DDS) if no institutional access is available, but this is not possible when using the Google Scholar Button.

## Lazy Scholar

- **Pros:** Free. Many features, such as checking for comments on PubPeer, PDF extraction of information such as population, intervention, outcomes, funding, references. Pulls citation data at article and journal levels.
- **Cons:** Overcomplicated extension and user interface. Supported by private individual.

Lazy Scholar is unique in that it is created and supported by a private individual—Colby Vorland, currently a postdoc at Indiana University. It is provided at no charge. One of the earliest browser extensions in this class of products, the earliest version of Lazy Scholar worked in a similar manner to the Google Scholar Button and scraped links from the Google Scholar Library Links program to provide access. The current version provides an added access option, similar to the way EndNote Click works.

Today, Lazy Scholar has a multitude of features, including

- recommended papers
- extraction of metrics (altmetrics, CiteScore)
- checking PubPeer comments,
- two institutional full-text options, including "automatic institutional full-text access"
- autosaving and auto-renaming PDFs
- PDF extraction features, including extraction of references, outline of sections, conflict of interest and funders, and more

While Lazy Scholar is free to use, the fact that it is maintained by one individual and is not open source might lead to concerns about sustainability. Even if this was not an issue, Lazy Scholar is clearly a work of passion, and the interface probably suffers due to the variety of features included.

## Lean Library

- **Pros:** Works with e-books. Integration with various common library platforms, including library discovery services (Summon, Primo, EBSCO Discovery Service), Springshare LibGuides, and more. LibAssist function allows librarians to leave customizable messages based on domain or URL user visits.
- **Cons:** Additional functionality means more setup costs. No direct PDF linking. The additional features that help with added discovery of content might be overkill for users who know what they want and seek only delivery options.

Traditionally, many library products began as an idea in the library itself, and Lean Library was no different. Johan Tilstra, a library technologist working in Utrecht University Library, came up with the idea of a browser extension, then dubbed UU Easy Access browser extension, which was eventually spun off as an incubator start-up, Lean Library.[5] In 2018, Sage acquired Lean Library.

We met Utrecht University earlier in chapter 1 and mentioned its pitch about "thinking the unthinkable" and prioritizing delivery instead of discovery. Lean Library was one of the dividends of this thinking.[6] The early version consisted of three different modules, with the Easy Access module being the heart of the extension. This can be seen as an improvement over the classic proxy bookmarklet by addressing the main drawback of that tool. The problem with the classic proxy bookmarklet is that it requires the user to remember to click to apply the proxy on the web page the user is on. Another related issue is that even when the user remembers to do so, they might be greeted with an error when they try to go on sites to which they have no access. As mentioned earlier, while clicking the bookmarklet will append the EZproxy string to the URL, and hence channel access via the proxy server, the proxy server will do so only for a list of authorized domains (e.g., jstor.org, ScienceDirect.com). If this is tried on a site that isn't on the authorized list, the user will see an error that is often an uninformative pop-up. This can be confusing and a big turnoff.

Lean Library's improvement is that the extension will automatically offer to proxy the site the user is on when appropriate. How does the extension know when it is appropriate? The library defines a list of domains that is stored in the extension.

Lean Library has expanded to include other useful functions bundled in different modules. Some are by now standard in most access broker browser extensions, including article-level linking to open-access papers, using services such as CORE Discovery, and support of link resolvers and ILL/DDS. However, some are unique to Lean Library.

For example, the LibAssist module allows librarians to leave custom messages that pop up when users visit certain domains or even URLs. One could set up a message to inform users, while they may not have access to market reports on a particular website, users can search databases such as Passport or Statista for similar content. Or the library could pop up a message when users are on a specific database such as Scopus asking them for their feedback via a survey link to solicit feedback for renewal.

Another unique feature of Lean Library is that it is the only access broker product on the market that provides linking to e-books, though this feature isn't as mature as for journal articles. In 2022, Lean Library is promoting a new product called Lean Library Futures and Lean Library's Workflow for LibGuides, which can integrate with Springshare LibGuides and FAQs as well as three major library discovery services—Summon, EBSCO Discovery Service, and Primo.

Interestingly, a lot of these features focus on providing alternative discovery options as opposed to delivery options of what the user is currently looking for. For example, on a Google Scholar search page where you are searching for the phrase *social network*, you can mouse over a button generated by the extension and see an overlay of results for that same search from your library discovery service. This provides an alternative to what your Google Scholar search has offered.

This type of content integration works even for videos, so you could be on a YouTube page searching for *social networks* when Lean Library offers to show results from selected library video platforms such as Sage Video and JoVE. Even text on sites such as Wikipedia can be overlaid with definitions and taxonomies from sources such as Sage Research Methods or even Statista (a business database). Hovering your mouse over phrases or words will draw definitions or content from these sources. Finally, the new Lean Library Futures allows users to give feedback on licensed resources via Net Promoter Score and permits integration of badges and information from providers such as Altmetric and Scite. Lean Library provides a very comprehensive set of features, and overall it is quite a complete product.

In a way, these new features are the return to the Library 2.0 ideas of the early to mid-2000s, boasting all the bells and whistles that libraries can use to maintain a presence on users' browsers as they browse the web. This fact has two consequences. First, unlike the other access broker browser extensions, Lean Library has a lot more features to configure for the library setting it up. Second, from the user's point of view, if you are a senior researcher looking for a tool to quickly gain access to full-text papers (and offer DDS/ILL if a paper is not available), you might find all the additional popups and features distracting. As a result, this tool might be more suitable for students and less experienced searchers who need additional support beyond just streamlining delivery.

## LibKey Nomad

- **Pros:** Provides holistic deep linking technologies beyond browser extensions. Is the only extension besides EndNote Click to offer "one-click to PDF" technology whenever possible. Has good support of aggregators (EBSCO and ProQuest).
- **Cons:** No additional discovery feature. Supports access to only journal articles, not databases or other online resources.

While Lean Library provides a host of useful supplementary services, you might be more interested in a service that has a razor focus just on improving delivery. This is where Third Iron's LibKey Nomad comes in. LibKey Nomad is the newest of the browser extensions profiled in this chapter. For many academic librarians, Third Iron is well known, being the company behind the highly popular BrowZine product, which allows academic and medical libraries to offer a consistent web and mobile interface to users who want to browse subscription and open-access journals. To develop BrowZine, Third Iron developed its own proprietary linking technology now dubbed LibKey to link users directly to full text.

Third Iron's website states that it offers "expert system full-text linking . . . [that] intelligently deliver[s] one-click access to millions of PDFs and HTML articles." Subscribers to the Third Iron suite of services will be able to use LibKey services to "eliminate the confusing maze of clicks to deliver the experience users expect, save researchers valuable time, and reduce help desk and ILL requests."[7] Of course, librarians reading this text might wonder how LibKey differs from existing library link-resolver technology (including OpenURL).

What is important to note is that the LibKey infrastructure does not claim to be a complete link-resolver solution. It specializes in handling full-text requests with article identifiers (specifically DOIs and PMIDs) and acts as "a Link Resolver Accelerator."[8] The idea here is that instead of covering the full range of full-text requests (some of which will not have article identifiers), LibKey will sit in front of your normal link-resolver solution and resolve such requests with article identifiers only. If LibKey is unable to resolve the request for full text, the request will be passed on to the normal link-resolver solution. While linking via identifiers seems to be a relatively easy process for link resolvers, things get harder if the institution's access is via aggregators, and this is where Third Iron claims that its "dynamic link construction dramatically improves reliability of linking to other aggregated sources, minimizing common linking errors such as in press articles."[9] For example, unlike most of the other services on this list, it has its own proprietary linking to aggregators such as ProQuest and EBSCOhost (one-click PDF linking). As an added benefit, full-text articles requested via LibKey are checked for retractions via the Retraction Watch Database, and users will be warned if there is a hit.

The LibKey suite of services consists of

- LibKey Link
- LibKey Discovery
- LibKey.io
- LibKey Nomad

All four services utilize the LibKey linking technology in different contexts. LibKey Link functions as a link-resolver accelerator, which you can use in place of your usual link resolver in databases such as Scopus, Web of Science, and even PubMed (via Library LinkOut using Outside Tool[10]). As mentioned earlier, the service will try to resolve full-text DOI and PMID requests via LibKey first before passing on to the normal link resolver if necessary. LibKey Discovery is meant to be used in most major library discovery services, including EBSCO Discovery Service, Primo, Summon, and WorldCat.

LibKey.io is reminiscent of Sci-Hub in terms of functionality even though it is legal. Go to the website, enter a DOI or PMID, select an institution, and you will be brought to the full text if it is available. Doing so will set a cookie in your browser, so you will not have to select your institution again. Similar to LibKey Link, if LibKey.io fails to find full-text options, it will generate a library access option that usually goes to the library link resolver. Ultimately, to use LibKey.io for your institution, simply point your browser to this URL:

> https://libkey.io/libraries/{LibraryID}/{DOI or PMID}

So, for example, for my institution you need to go to

> https://libkey.io/libraries/646/10.1017/S104909
> 6511000199.

This URL works as well:

> https://libkey.io/{DOI or PMID}

However, in this instance, the user will be asked to select an institution, if they have not done so before. LibKey Linking is also natively supported by various databases and services. A short list includes Semantic Scholar, Scholarcy, CAB Direct, and the literature-mapping tool Inciteful.

Last, we come to LibKey Nomad, the access broker browser extension. It is very similar to the other extensions we have already considered. Once installed, it sits quietly in the user's browser and will activate when the user lands on an appropriate page, showing a badge with a link to the full text if LibKey Nomad detects that full text is available. Like all the other services, it will also direct you to open-access copies if they are available.

If full text is not available either way, then it will show another option called Access Options, which typically links to the institution's link resolver. Installing the LibKey Nomad extension also brings additional functionality. For example, when a user is browsing Wikipedia, the references in the reference sections of the page will be overlaid with full-text links when possible, and this feature extends to other popular abstracting and indexing (A&I) databases such as Scopus, Web of Science, PubMed, and so on.

Third Iron's suite of products is very focused and enhances the reliability of your link resolvers. Setup is relatively easy, particularly if you have already set up the BrowZine product; otherwise, it provides support for most popular electronic resource management systems, such as Alma and Serial Solutions, to sync your holdings. It is important to note that the LibKey suite of products assists users with access to article content only. While this covers a lot of what users might be looking for, it will not help users who need help accessing non-journal content such as e-books. Nor will it help the user to sign on to non-full-text platforms and databases such as Scopus or Artstor, unlike Lean Library.

## EndNote Click (formerly Kopernio)

- **Pros:** Almost no setup required. Good one-click PDF linking.
- **Cons:** May not cover aggregators (e.g., EBSCO) well. Requires saving of username and password in extension. Extension also uses username and password on behalf of user, which may infringe some institutions' IT use policy.

Kopernio, which is now known as EndNote Click, was also an early entrant. It began as a start-up by Jan Reichelt and Ben Kaube in 2017 before eventually being acquired by Clarivate in April 2018.

When it launched, the product garnered a lot of press and publicity and became very popular in researcher circles. It was, and still is, offered as a freemium product from the very start, which explains how it racked up one million installations compared to

Lean Library's 100,000+ and Third Iron's 200,000+ installations of LibKey Nomad, which required an institutional subscription to fully benefit from their features.

EndNote Click's basic functionality should be very familiar by now. It's a browser extension that sits in the browser and will pop up a badge with the text "View PDF" (see figure 3.5 for an example). Clicking on it will link you to full text if it can find a copy available via your institution. Like the other browser extensions, it can also link you to open-access copies of papers.

For many institutions that have set it up, it will also offer additional options if no full-text option is available (including open-access copies). Typically, it will send users to the link-resolver options if it is unable to find full text.

EndNote Click was the first access broker browser extension to tout the advantage of "one click to PDF" functionality, and today it may still be the best at this functionality. As with LibKey Nomad, installing the extension will result in the extension overlaying popular websites such as Wikipedia and databases such as PubMed with buttons that are links to PDF files.

In terms of unique functionality, EndNote Click provides a "locker" that allows users to store in the cloud PDFs that they have downloaded via the extension. EndNote Click is the only access broker browser extension offered as a freemium product. Choosing to pay as an individual will increase cloud storage as well as providing additional functionality such as syncing to Dropbox; however, if you do not use this feature, EndNote Click is perfectly functional as a free service. EndNote Click leverages its position as a Clarivate company with integration with other Clarivate products, including EndNote and Web of Science, but otherwise it works similarly to other access broker extensions.

While this product feels very similar to other access brokers already introduced—particularly LibKey Nomad—from the back end EndNote Click works quite differently. For access broker browser extensions to work, the access broker usually needs to work with each institution. Whether it be the need to get a holdings file of what the institution has coverage of or the institution's OpenURL path, some setup needs to be done. In the case of LibKey Nomad and Lean Library, there is a need to send a list of your institution's holdings to the access broker vendors before the extension works. Even in the case of the Google Scholar Button or Lazy Scholar, where the feature appears to work automatically, these extensions work by leveraging the work institutions are already doing by working with Google Scholar in the Library Links program.

Yet many institutions may be surprised to find that despite their not working with EndNote Click directly, the extension may be already fully functional for their users even if their institutions do not provide coverage data. Indeed, unlike other services in this list, no setup is needed to indicate your holdings, which is a great time saver. How this works is described on the EndNote Click web page, but essentially, it will use your users' stored credentials in real time to check if they have access to the full text. The trade-off for this is that your users will need their user accounts and passwords saved in the application, which EndNote Click will use to check journal websites directly for full-text access. While Clarivate assures us that all user credentials are encrypted, stored in your local browser cache, and never "sent to EndNote Click and never leave your browser other than to be submitted to the verified institutional authentication service," some people might still have privacy concerns.[11]

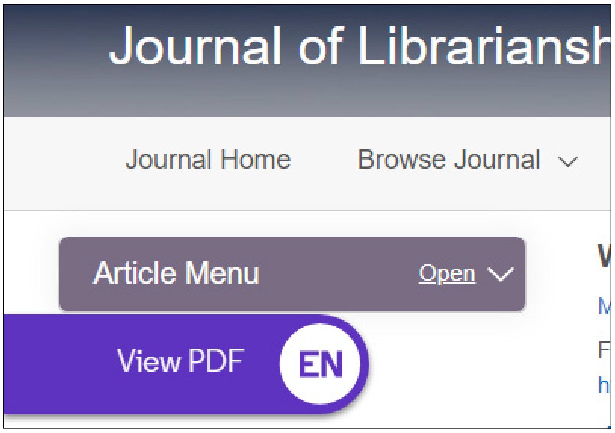


**Figure 3.5**
An example of EndNote Click's pop-up badge

One other disadvantage of EndNote Click is that it prioritizes full-text access from publishers' platforms and provides little aggregator support. As of the time of this writing (March 2022), it supports ProQuest as an aggregator but not EBSCOhost and may not be suitable for an institution that provides access to journal content largely via EBSCO only.

Ultimately, though EndNote Click can provide access to full text without additional work by the institution, the product works better with official institutional involvement. For example, the institution can add additional options if no full text is found and also has access to an institutional dashboard that shows the pattern of downloads from its users of the extension. This helps with figuring out which titles are popular, how many of the PDFs that are downloaded are publisher-hosted versus OA alternatives, and more.

## Why Not Access Broker Browser Extensions?

While access broker browser extensions appear to be a great way to improve user access and delivery options, they have some drawbacks. The first is a practical one. Your users will not by default have the browser extension installed, and while some institutions, such as the University of Manchester, have achieved some degree of success with high penetration rates with their user base, this usually can't be achieved without a fair degree of marketing effort.[12] Working with your institution's IT department to see if there is some mass auto-deployment method available (e.g., via Microsoft Group Policy) might be advisable.[13]

The other objection to the widespread deployment of access broker browser extensions is concern about privacy issues that browser extensions might bring. One of the strongest cases against this class of products was made by the group known as RA21. This group has since been followed by a successor organization, SeamlessAccess, whose work we will discuss in detail in chapter 4. In "RA21 Position Statement on Access Brokers," the group, which also coined the phrase *access brokers*, took aim at some of these tools and argued that

- these tools may have potential security risks as some of them store users' institutional usernames and passwords;
- they often require the creation of individual accounts, leading to privacy risks; and
- they generally enable providers of these tools to track "end user behavior and reading habits across publisher sites potentially impacting privacy and research freedom."[14]

Ultimately, RA21 claims that such tools, while useful in the short term, do not actually fundamentally solve the issue. As we will see in the next chapter, RA21 aims to solve this problem by working on and implementing a long-term fundamental solution by building on federated identity management practices.

## Conclusion

In this chapter, we covered the history of access broker browser extensions and introduced five popular tools that institutions have been using to help users with access. Consisting of both free tools (Google Scholar and Lazy Scholar) and commercial tools (Lean Library, LibKey Nomad, and EndNote Click), they are must-have tools in the arsenal of academic libraries today. However, as noted in the last section, these tools are not a complete solution as they require our users to be aware of them and to install them, and they may create privacy and security risks. RA21, now succeeded by SeamlessAccess, argues that such tools are at best a temporary Band-Aid and that we are better off with solutions that improve the way we authenticate and authorize users to access resources. We shall turn our attention to that next.

## Notes

1. RA21: Resource Access for the 21st Century, "RA21 Position Statement on Access Brokers," August 23, 2018, https://ra21.org/what-is-ra21/ra21-position-statement-on-access-brokers/.
2. Galadriel Chilton and Joelle Thomas, "LibX: The Small but Mighty Button for E-Resource Discovery and Access," *Serials Librarian* 66, no. 1–4 (2014): 146–52, https://doi.org/10.1080/0361526X.2014.879019.
3. Aaron Tay, "Starting Research from Outside the Library Homepage: An Analysis of User Behavior from an Academic Library" (presentation, IFLA 2013 Satellite Meeting on Literacy and Reference Services, National Library Board, Singapore, August 2013).
4. "Library Support," Google Scholar Help, accessed October 5, 2021, https://scholar.google.com/intl/en/scholar/libraries.html.
5. Johan Tilstra, "Lean Library's Browser Extension: Seamless Delivery for Users," *Aaron Tay's Musings about Librarianship* (blog), May 30, 2017, http://musingsaboutlibrarianship.blogspot.com/2017/05/guest-post-lean-librarys-browser.html.
6. Simone Kortekaas and Bianca Kramer, "Thinking the Unthinkable: Doing Away with the Library Catalogue," *Insights* 27, no. 3 (November 3, 2014): 244–48, https://doi.org/10.1629/2048-7754.174.
7. Third Iron home page, https://thirdiron.com.
8. Third Iron, "LibKey Link Technical FAQ," accessed June 13, 2022, https://support.thirdiron.com/knowledgebase/articles/1947991-libkey-link-technical-faq.
9. Third Iron home page.
10. For more information, see M. Collins, "PubMed Labs Update: Library LinkOut using Outside Tool," *NLM Technical Bulletin,* no. 429 (July–August 2019), https://www.nlm.nih.gov/pubs/techbull/ja19/ja19_pubmed_labs_linkout.html.
11. "EndNote Click: How Does EndNote Click Store and Protect Institutional Credentials?," Clarivate Support, April, 9, 2021, https://support.clarivate.com/Endnote/s/article/EndNote-Click-How-does-EndNote-Click-store-and-protect-institutional-credentials.
12. Third Iron, "How the University of Manchester Library Reduced Support Ticket Volume by 25%," case study, 2020, https://thirdiron.com/wp-content/uploads/2020/02/University-of-Manchester-case-study.pdf.
13. Third Iron, "Libkey Nomad Group Policy Deployment (Edge)," accessed October 11, 2021, https://support.thirdiron.com/knowledgebase/articles/1970322-libkey-nomad-group-policy-deployment-edge.
14. RA21: Resource Access for the 21st Century, "RA21 Position Statement," para. 7.