

## Site fixes - Site script request/donation #101

### [site fix] box.com

01/19/2022 02:30 AM - jacobk

<b>Status:</b>	Closed	<b>Start date:</b>	01/18/2022
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>Description</b>			
<p>Hello again! I made a simple partial fix for downloading files from app.box.com. It was made in the span of only a couple of days. It's somewhat messy, it doesn't support files marked as not downloadable, and it doesn't support folders within folders, but I wanted to go ahead and share what I have, since this could be useful to someone already. Is this the right place to post the script for inclusion in the Haketilo repository? I have attached the script as a text file.</p> <p>I might tweak the script a bit as I'm using it, but I probably won't add support for files not marked as downloadable because it looks somewhat complicated and I don't personally have a need for it, and I definitely won't add support for folders within folders until I actually see another example of this (pretty sure I've seen one before, but not one that I still have access to).</p>			

### History

#### #1 - 01/20/2022 08:37 AM - koszko

jacobk wrote:

Hello again! I made a simple partial fix for downloading files from app.box.com. It was made in the span of only a couple of days. It's somewhat messy, it doesn't support files marked as not downloadable, and it doesn't support folders within folders, but I wanted to go ahead and share what I have, since this could be useful to someone already. Is this the right place to post the script for inclusion in the Haketilo repository? I have attached the script as a text file.

Hi again and thanks for the fix! It is indeed the right place :)

I'll test it and I'll also ask Nick to look at it (RMS requested us to have all contributed fixes reviewed by at least 2 Haketilo devs before including them)

#### EDIT

As to the "A" license - it should be considered "obsolete". It turned out there're some technical problems with it. Plus RMS expressed concerns about it creating a bad image (relevant if we're going to succeed in joining GNU).

While I myself compromised with RMS to instead use just GPL and attach an "I won't sue" promise to my extension code, there're no special license requirements for inclusion of a script in the repository (except the basic need for it to be free software ☐☐). Whether you use plain GLPv3+, GPLv3+ with exceptions, MPL, Expat, CC0, etc. we'll still gladly accept your fixes :)

#### #2 - 01/21/2022 02:41 PM - 0gitnick

Hi.

I read the script and manually confirmed it contains no malicious code. I also installed and tested it on my machine and confirmed it works for the test cases provided. I approve adding it to the repository.

Thanks for contributing! :)

**#3 - 01/22/2022 02:51 AM - koszko**

- *File box-fix.js added*

Ogitnick wrote:

Hi.

I read the script and manually confirmed it contains no malicious code.

Same here!

Jacobk, I will add the Box fix to the repository today.

Also, I couldn't resist and hacked the downloading of preview-only files... It felt much like REing Odysee - the entire page is a JS-driven app that relies on it to merely display anything, making a fix involves tracking tokens through network logs and recreating the logic. Fortunately, now I know JS better than I did when making the Odysee fix ^^

The enhanced script is attached. Besides presenting a working download button regardless of item being officially downloadable, it also displays some additional metadata. I didn't bother making it pretty - the data just gets dumped to the page as JSON. If you want, you can enhance the script to display that data in some more elegant way. This might include - as you suggested in the comment - giving a prominent message when the file is not officially meant to be downloadable (right now that information is buried somewhere in the JSON that is dumped...).

Btw, the metadata fields suggest that files might exist which are not even previewable. I didn't account for that in the script because it's not the case with any of the Box links we got

**#4 - 01/22/2022 02:51 AM - koszko**

- *% Done changed from 0 to 90*

**#5 - 01/22/2022 11:25 AM - koszko**

- *Status changed from New to Closed*

- *% Done changed from 90 to 100*

Now served through Hydrilla:

<https://hachette-hydrilla.org/>

<https://api-demo.hachette-hydrilla.org/query?n=https://ucla.app.box.com/s/mv32q624ojihohzh8d0mhhj0b3xluzbz>

**#6 - 01/22/2022 11:19 PM - jacobk**

- File `box-fix.js` added

Thanks for enhancing the fix and adding it to the repository!

I'm not sure how best to communicate the `can_download` permission to the user. It would be easy if we were showing the preview in the browser rather than downloading them as a normal file, but it would be silly to hide the download button when that's the only way the user can even see the file.

I did modify the script to show the name of the file as a heading. In some cases (maybe only private links), there's some other info like the name of the uploader, the date uploaded, and the date the link expires, but I don't know where that is and I haven't looked into it at all yet.

For folders, I tried to replace `XMLHttpRequest` with `fetch` in order to stop the script from downloading every file in the folder before the user clicks on anything (This is due to transparent redirects.), but `'redirect: "manual"'` doesn't allow me to get the location value so I can get the final URL, so `fetch` didn't actually help. For some reason though, a `GET` request to the URL I was sending a `POST` request to also works, so now instead of sending a `POST` request to URL A which redirects to URL B, I just put URL A on the button, and clicking it downloads the file. The "right" way to do it might just be to use an event listener on the button, but then it wouldn't be as easy for the user to get the download URL (which maybe doesn't actually matter).

I have attached the updated script. There's still a lot of functionality that could be added, of course, but for most people this should work fine.

Also, how do you type multi-line comments? Do you add the stars and word-wrapping manually or does your text editor handle it automatically somehow?

**#7 - 01/24/2022 12:09 PM - koszko**

jacobk wrote:

Thanks for enhancing the fix and adding it to the repository!

You're welcome :)

I took your modifications and changed the code a bit further. Here it is (format importable from Haketilo 0.1 settings page):

[https://hachette-hydrilla.org/https://ucla.app.box.com/s/\\_\\_\\_\\_\\_.json](https://hachette-hydrilla.org/https://ucla.app.box.com/s/_____.json)

I'm not sure how best to communicate the `can_download` permission to the user. It would be easy if we were showing the preview in the browser rather than downloading them as a normal file, but it would be silly to hide the download button when that's the only way the user can even see the file.

I made the download button instead display "unofficial download (officially disallowed)" for those files.

I did modify the script to show the name of the file as a heading.

Great. Seeing there are many dynamically-created DOM nodes, I changed the code to instead use DOMParser.

In some cases (maybe only private links), there's some other info like the name of the uploader, the date uploaded, and the date the link expires, but I don't know where that is and I haven't looked into it at all yet.

The same API endpoint I used to get file metadata also serves those values. The proprietary client js would get the upload-related data in a separate request, but here I just added the relevant fields to also be fetched in the request we make for file metadata.

For folders, I tried to replace XMLHttpRequest with fetch in order to stop the script from downloading every file in the folder before the user clicks on anything (This is due to transparent redirects.), but 'redirect: "manual"' doesn't allow me to get the location value so I can get the final URL, so fetch didn't actually help. For some reason though, a GET request to the URL I was sending a POST request to also works, so now instead of sending a POST request to URL A which redirects to URL B, I just put URL A on the button, and clicking it downloads the file. The "right" way to do it might just be to use an event listener on the button, but then it wouldn't be as easy for the user to get the download URL (which maybe doesn't actually matter).

I also don't think it actually matters. For a user who clicked the button at least once, wouldn't the link be available through browser's downloads history? If so, we could replace <a>'s with a <form> that performs the POST.

Anyway, now that we know how to query box's API for a file given its id, a better way seems to be to move part of hack\_file() responsible for getting file's metadata (including the download URL) to a separate function and call that function for every file inside folder to get all their download URLs. This way, we'd also create working links for files in the folder that might be preview-only.

I have attached the updated script. There's still a lot of functionality that could be added, of course, but for most people this should work fine.

Indeed. At some point I'd like to split such fixes into 2 parts: a site-agnostic javascript UI library for displaying online disk contents and a site-specific javascript library for getting data from a site like app.box.com. With that, supporting yet another similar site would amount to REing its API, writing a script that queries it and plugging it into the UI library we'd already have.

Also, how do you type multi-line comments? Do you add the stars and word-wrapping manually or does your text editor handle it automatically somehow?

Shame to admit, I do it manually. I guess it should be easy to find a tool that does it automatically or even write my own snippet of emacs lisp code to do that. And I also guess I should do that to avoid embarrassment ☹☹

Responding to your comments in the code:

The above logic does not send a request\_token in the body, so I'm not sure why it works.

Perhaps the request token is only needed for some of the files?

A way to use POST without downloading everything in advance would be to use event listeners for the buttons, but then the original download folder button sends a GET request that gets 2 URLs in the response. 1 of those URLs downloads the file, and a POST request is sent after (or maybe while in some cases?) a file is downloaded, to let the server know how much is downloaded.

We could then show a button to allow informing the server about this download **when** the user wants that. Although I don't think that's really needed right now...

#### EDIT

I had no way to test if my changes didn't break the folder case. You know, a single typo and...

**#8 - 01/26/2022 11:58 PM - jacobk**

I tested your update with the folders I have access to, and everything seems to work fine. I switched out of the class section that was posting files in Box folders though, so it's possible that in the near future I won't be able to test how well folders work either. But, as long as the links still work for me I'll keep testing the fix on each update and if I have time I'll try to add download-via-preview and folder downloads (which will have to use the official download method I think, because simply looking at a folder preview doesn't download the folder).

**#9 - 01/27/2022 11:17 AM - koszko**

jacobk wrote:

I'll try to add download-via-preview

You mean showing a preview of files like PDFs when they are meant to be preview-only (as discussed earlier)? I guess we could achieve that by pulling pdf.js. But then I thought it'd be easier to just create an iframe and let the browser itself (which has the pdf viewer built in) render the document. Perhaps we could also treat other file formats (text/plain, svg, png, webm) in the same fashion since browser is naturally expected to know how to deal with them.

and folder downloads (which will have to use the official download method I think, because simply looking at a folder preview doesn't download the folder).

Here, we also seem to have 2 options. Either learn box's folder download API or pull a dependency like zip.js and construct the archive in JS :)

I guess we could start with the first option, though (and in later versions of Haketilo I'll add some special facility to handle libraries)

**#10 - 01/27/2022 09:23 PM - jacobk**

By download-via-preview, I meant downloading files from folders the way the preview downloads in order to allow downloading preview-only files, just like we currently download individual files. I wasn't yet considering actually showing the file in the browser (It could be a useful feature, but not a particularly important one, in my opinion.).

Regarding downloading folders, it would be good to have both methods of downloading, because I think there are cases where both could fail. In the case of downloading all of the files and then zipping them together, but in that case, the folder could be missing folders within folders, as well as other types of items we haven't seen yet (only until we can find an example of them, though). Downloading the folder the way the official client does it means we'll probably get everything, except files marked as not downloadable, perhaps. So, in order to account for both unexpected item types and items marked as not downloadable, I think it would be good to have both methods.

**#11 - 01/27/2022 10:21 PM - koszko**

jacobk wrote:

I wasn't yet considering actually showing the file in the browser (It could be a useful feature, but not a particularly important one, in my opinion.).

Agreed

Regarding downloading folders, it would be good to have both methods of downloading [...]

Your reasoning is convincing ☐☐

**#12 - 03/03/2022 05:22 PM - koszko**

- *Tracker changed from Support to Site script request/donation*

- *Project changed from Haketilo to Site fixes*

**Files**

---

box-fix.js	10.4 KB	01/19/2022	jacobk
box-fix.js	11.2 KB	01/22/2022	koszko
box-fix.js	12.2 KB	01/22/2022	jacobk