

Haketilo - Feature #6

[Roadmap 34] make it possible to automatically download page's served scripts and save them

07/01/2021 11:41 AM - koszko

Status:	New	Start date:	07/01/2021
Priority:	Low	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			
Description			
<p>Of course, running the same nonfree scripts but served locally would not be a significant improvement. This feature will, however, make it easy to modify the set of scripts provided by website - useful, if some of those scripts are already free, as is often the case.</p> <p>Please keep in mind that we won't be able to guarantee that such locally-served scripts would work properly in 100% of cases. A couple of cases where things are expected to break:</p> <ul style="list-style-type: none">• scripts that rely on being executed synchronously with the page load (e.g. those that use document.write() APIs) are still going to break• scripts that are added dynamically to the DOM (as the "parent" scripts that adds them will be unaware that a nonce has to be added to comply with Haketilo-injected CSP rule)• on-demand generated scripts that are meant to contain data that is different in subsequent page visits			
Roadmap			
Related issues:			
Blocked by Haketilo - Feature #7: [Roadmap 34][Milestone] find some convenien...		New	07/01/2021

History

#1 - 12/12/2021 12:55 AM - jacobk

Hello, I just made an account here. I've been looking for an extension like this for a while, so thanks for making this!

In the past I've used a combination of LibreJS, NoScript to block nonfree/unwanted JavaScript, and various extensions (Hachette, Greasemonkey, LocalCDN, etc.) to replace functionality on sites that depend on JavaScript to function. In many cases, I whitelist scripts that I know are free (such as the scripts on archive.org) in LibreJS, but often when websites update I have to go through and whitelist all the scripts again. So, I had an idea to make an extension that contains the non-LibreJS-accepted scripts at archive.org and sends them when they are blocked in LibreJS. Unfortunately, I couldn't get it to work, and I think because extension scripts run in a different context than page scripts (and I was trying to combine extension scripts and page scripts together), which I think is the same issue with Greasemonkey mentioned in the wiki here (I assume Greasemonkey's context problem doesn't apply to Haketilo, since it's mentioned as a comparison.).

Now, with Haketilo, as a test, I tried downloading to 2 scripts served at <https://accounts.fedoraproject.org> and putting them in a bag, and then telling Haketilo to execute them on <https://accounts.fedoraproject.org>. Unfortunately, I get an error "Uncaught TypeError: p is undefined" in the console. So, the question is: why? There are no "onclick" or similar elements in the HTML (unless I'm missing something which is quite possible, but I think LibreJS would show it), which I think is what issue #7 refers to. (I know accounts.fedoraproject.org is already supported by a (much lighter) script in the official repository, but I am curious as to why the original scripts don't work.) I think this is relevant to this issue since automatically saving/installing JavaScript on a page would be most useful if the scripts can run as is.

#2 - 12/13/2021 08:42 AM - koszko

- Description updated

jacobk wrote:

Hello, I just made an account here. I've been looking for an extension like this for a while, so thanks for making this!

Hi Jacob! Thanks for joining :)

In the past I've used a combination of LibreJS, NoScript to block nonfree/unwanted JavaScript, and various extensions (Hachette, Greasemonkey, LocalCDN, etc.) to replace functionality on sites that depend on JavaScript to function. In many cases, I whitelist scripts that I know are free (such as the scripts on archive.org) in LibreJS, but often when websites update I have to go through and whitelist all the scripts again. So, I had an idea to make an extension that contains the non-LibreJS-accepted scripts at archive.org and sends them when they are blocked in LibreJS. Unfortunately, I couldn't get it to work, and I think because extension scripts run in a different context than page scripts (and I was trying to combine extension scripts and page scripts together), which I think is the same issue with Greasemonkey mentioned in the wiki here
(I assume Greasemonkey's context problem doesn't apply to Haketilo, since it's mentioned as a comparison.).

Correct. We went great lengths to inject local scripts into the actual page and to prevent page's CSP rules from blocking those scripts.

Now, with Haketilo, as a test, I tried downloading to 2 scripts served at <https://accounts.fedoraproject.org> and putting them in a bag, and then telling Haketilo to execute them on <https://accounts.fedoraproject.org>. Unfortunately, I get an error "Uncaught TypeError: p is undefined" in the console. So, the question is: why? There are no "onclick" or similar elements in the HTML (unless I'm missing something which is quite possible, but I think LibreJS would show it), which I think is what issue #7 refers to.

I tried reproducing your issue and the exact error happens if I try to load Bootstrap before JQuery. If I reverse the order, it works properly (actually, this alternative fix would be worth adding to the list of official scripts!).

The thing is, Haketilo injects scripts in the same order they appear within a bag. In this case one of Fedora Accounts' scripts (Bootstrap) depends on the other one (JQuery) and thus should be placed after it in the bag.

Currently in Haketilo settings page there's yet no facility to conveniently reorder bag's scripts, so you'll just need to remove Bootstrap from the bag and re-add it (it will then appear after JQuery).

I think this is relevant to this issue since automatically saving/installing JavaScript on a page would be most useful if the scripts can run as is.

True. I updated this task's description to mention a few cases where we already know this won't work properly. Despite these, I expect a big subset of websites to not have such issues.

Btw, I am sad to inform that this feature is not of very high priority. I a (not-yet-public) Memorandum of Understanding document we'll be signing with NLnet it landed as an optional feature that's not part of the work our current monetary grant covers... You could still contribute the code yourself if you want ;)

#3 - 02/24/2022 01:12 PM - koszko

- Subject changed from *make it possible to automatically download page's served scripts and save them to [Roadmap34][Milestone]* make it possible to automatically download page's served scripts and save them

- Description updated

#4 - 02/24/2022 01:12 PM - koszko

- Subject changed from *[Roadmap34][Milestone]* make it possible to automatically download page's served scripts and save them to *[Roadmap 34][Milestone]* make it possible to automatically download page's served scripts and save them

#5 - 02/24/2022 01:17 PM - koszko

- Subject changed from *[Roadmap 34][Milestone]* make it possible to automatically download page's served scripts and save them to *[Roadmap 34]* make it possible to automatically download page's served scripts and save them

- Description updated

#6 - 02/24/2022 01:18 PM - koszko

- Blocked by Feature #7: *[Roadmap 34][Milestone]* find some convenient way to automatically re-add intrinsic javascript added