

## Haketilo - Feature #48

### Load default\_setting.json using XMLHttpRequest

07/06/2021 12:12 PM - koszko

<b>Status:</b> Rejected	<b>Start date:</b> 07/06/2021
<b>Priority:</b> Normal	<b>Due date:</b>
<b>Assignee:</b>	<b>% Done:</b> 0%
<b>Category:</b>	<b>Estimated time:</b> 0.00 hour
<b>Target version:</b>	
<b>Description</b> I believe XMLHttpRequest can also be used to fetch extension's own files. After fetching the default settings file, we could use JSON.parse() on it. That would reduce the complexity of our build script	

#### History

##### #1 - 07/07/2021 12:09 AM - jahoti

(Responding here as the other issue is now resolved.)

Btw, I am considering maintaining old build.sh alongside the new build.html.  
Plus, IMHO, changing to use XMLHttpRequest would be a good idea in case of the JS build system, too

That makes sense now- I had been assuming the intention was to use XMLHttpRequest in the build process somehow, which (unless I'm again mistaken) is not the case?

##### #2 - 07/24/2021 12:38 AM - jahoti

Preferrably preceded by: <https://hachettebugs.koszko.org/issues/59>

##### #3 - 07/24/2021 08:47 AM - koszko

Depends. We can make this issue obsolete and completely move bundled settings to the repo.

Or, after moving most of stuff to the repo we can select some of it we consider the most useful and of highest quality and bundle it in addition to it being available from there

##### #4 - 07/25/2021 08:58 AM - jahoti

Definitely the latter; moving fixes to the repository first would only be in order to have the build script(s) download fixes from there, rather than including the code in default\_setting.json.

On the other hand, that could prevent offline builds; IDK.

##### #5 - 07/26/2021 08:44 AM - koszko

On the other hand, that could prevent offline builds; IDK.

Not necessarily.

I think reference to how Arch packaging works might be useful here. Arch PKGBUILD script used to create packages allows the packager to program more or less the following stages:

1. fetching of sources (done declaratively, by providing the relevant URLs).
2. validating of sources (also declaratively, by providing checksums or allowed PGP keys)
3. configuration
4. build
5. installation to a directory

The contents of that directory then ends up in the actual package.

Packaging for other distros follows a similar pattern, although Arch seems to have the least "magic" added around it. Actually, Debian source packages are tarballs with everything already downloaded (like after steps 1 and 2 that were already performed by a packager) and possibly patched for dfsg. So, the fact we need to download some more sources doesn't mean the build itself has to be online. We can still make releases in the form of tarballs with everything already downloaded.

Also, the practice of linking one git repo from another could be useful here.

## #6 - 07/26/2021 09:22 AM - jahoti

Not necessarily.

I think reference to how Arch packaging works might be useful here. Arch PKGBUILD script used to create packages allows the packager to program more or less the following stages:

- fetching of sources (done declaratively, by providing the relevant URLs).
- validating of sources (also declaratively, by providing checksums or allowed PGP keys)
- configuration
- build
- installation to a directory

The contents of that directory then ends up in the actual package.

Packaging for other distros follows a similar pattern, although Arch seems to have the least "magic" added around it. Actually, Debian source packages are tarballs with everything already downloaded (like after steps 1 and 2 that were already performed by a packager) and possibly patched for dfsg. So, the fact we need to download some more sources doesn't mean the build itself has to be online. We can still make releases in the form of tarballs with everything already downloaded.

Ah- I see! Such a system seems sensible, and if we package for distros it'll inevitably occur in some form or another anyway.

Also, the practice of linking one git repo from another could be useful here.

How do you mean?

**#7 - 07/28/2021 12:18 PM - koszko**

Also, the practice of linking one git repo from another could be useful here.

How do you mean?

<https://git-scm.com/book/en/v2/Git-Tools-Submodules>

**#8 - 11/27/2021 01:20 PM - koszko**

- *Parent task deleted (#44)*

**#9 - 12/07/2021 09:31 AM - koszko**

- *Status changed from New to Rejected*

We're moving to IndexedDB. Default settings will now be accessed from database initialization function, not from background/main.js. IndexedDB initialization happens in a transaction which does not allow arbitrary asynchronous operations to be performed. This makes asynchronous XHR not applicable. We'd have to either use synchronous XHR or put the default settings in a file. In this case I'd rather do the second. I will be moving the default settings to exports\_init.js. In the future we might try boosting the imports system with a directive like IMPORTJSON which could make the build system simpler