The task is to update the Planet software/theme to allow selecting feeds, just as https://planet.gnome.org allows (at the bottom)

Before we continue thinking about this, I would like to hear about the origin of this task. Who asked for this?

Personally I find this feature completely useless. The fact that the feature on Planet GNOME has apparently been broken for a long time tells me that nobody is using it. It smells of a solution looking for a problem, really.

These planet-type applications aim to provide a curated experience of feed reading. A change like this would be a hacky move towards personalised feed management.

In the case that some real actual person wants this, I would not try to hack this on top of our existing planet, but look into Pluto and make a feature request for it: https://github.com/feedreader

If it's broken on Planet GNOME, I am indeed sceptical about replicating it on our system. However, I don't mind giving our Planet a revamp. Can we setup a Pluto test instance and have a look? If it at the same time improves the design and responsiveness of the page, that'd be a win!

Backend changes needed to build something similar to the GNOME solution:

1. The `<div class="channelgroup">` elements need an additional class name to associate them with the author. Could be just a lowercased author name with spaces replaced with underscores, examples: libreoffice_design_blog, official_tdf_blog
2. Daily cron job generating an authors.js file having an object like this containing all the authors:

```javascript
var authors = {
  "libreoffice_design_blog": "LibreOffice Design Blog",
  "official_tdf_blog": "Official TDF Blog",
};
```

I can prototype this offline already on the client side.

#5 - 2020-04-04 14:27 - Beluga Beluga

Scratch that number 2 from my previous comment. We already have the authors list visible in the sidebar, so there is no need for much gymnastics. This is enough:

The li elements should have classes corresponding to the blog authors like

```html
<li class="libreoffice_design_blog">
</li>
```

#6 - 2020-04-04 19:19 - Beluga Beluga

- File planet-filter.zip added

Attached is my static prototype. It includes a screenshot of how things look like, when you hide all authors on the page - just to demonstrate the fundamental hackiness of this approach.

No external libraries are used. The JS does not work in IE11, because I wanted to minimise the investment in this hackery.

If you want to test the cookie functionality, you can erase the cookie by giving this in your browser's dev tools console:

```javascript
document.cookie = "hidden=0;max-age=0";
```

The authors who happened to be featured on this planet snapshot:

"Andreas Kainz"
"CIB News"
"Collabora Community"
"LibreOffice QA Blog"
"Luboč Luňák"
"Marius Popa"
"Michael Meeks"
"Miklos Vajna"
"Official TDF Blog"
"Roman Kuznetsov"
"Tamás Bunth"
"Tamas Zolnai"
"Tomaž Vajngerl"

For Guilhem: I checked and there is actually no need to normalise unicode characters for use with HTML & CSS. Only NUL and space characters can not be used. It would have been a potential problem anyway, because possibly some author name could be in CJK form and normalising those naively would result in an empty string (with Python unicodedata.normalize at least).

So the only thing that needs to be done for the class names is to replace spaces with underscores.

#7 - 2020-04-16 01:43 - Guilhem Moulin

- Status changed from New to Feedback

For Guilhem: I checked and there is actually no need to normalise unicode characters for use with HTML & CSS. However AFAICT escaping is required before updating the cookie.

So the only thing that needs to be done for the class names is to replace spaces with underscores.

The template logic doesn't allow easy transliteration, but exposes an automatic variable "css-id" (which seems to be the lower-case feed name with hyphens instead of spaces) so I just used that. Applied some minor fixed and exposed the result to https://planet.documentfoundation.org/authorfilter/. Thanks!

#8 - 2020-04-16 08:55 - Beluga Beluga
Thanks for fixing my mistakes!

Regarding escape() and unescape(), apparently we should not be using them, but rather these:

It seems there are gotchas with all of them, though:

#9 - 2020-04-20 20:49 - Guilhem Moulin

Regarding escape() and unescape(), apparently we should not be using them, but rather these:

Thanks for the links (my JS-fu isn't stellar :-P), replaced encode() (resp. decode()) with encodeURIComponent() (resp. decodeURIComponent()) as its semantics seem appropriate for this use-case.

#10 - 2020-04-27 13:53 - Guilhem Moulin

- Status changed from Feedback to Closed
- % Done changed from 0 to 100

Applied to production, closing.

Files
planet-filter.zip 9.08 MB 2020-04-04 Beluga Beluga