Goal: Support tables in PDF

More precise: Figure out a good way to go forward with the #Technical Wishlist request “show tables in PDF”

Option 1: Generate LaTeX tables

Issue: fragile
would need whitelist
no css support
would need to special case infoboxes etc
long tables would require special packages, which are fragile in other ways

Option 2: Use browser rendering as OCG backend

- render html, then feed to browser / electron

Option 3: Use mwoffliner

Option 4: use services with browser based rendering without collection features

How we plan to proceed:

Hack 1:

- Option 4: Try to use the pdf render service mentioned by Gabriel to offer an option for article pdf downloading with tables (see https://phabricator.wikimedia.org/T134205)
- Figure out what would need to be done to get it working for books, too. Issues
    - Layout (but what exactly)?
        - Multi-column layout probably most important
    - Performance
- WMDE’s technical wishes team will aim at getting the option for article pdf downloading production ready and deployed as quickly as possible
- Depending on the outcome of step two, they might integrate this, too, or forward it to Community Tech, or maybe keep it as an option for the next hackathon?
-
- During the hackathon
  WMDE-Gabriel and Leszek work on step 1 (getting help from WMF-Gabriel and Scott

where necessary), WMF-Gabriel and Scott discuss performance and layout needs for book issues

How would this look like in the future?

WMDE asks the German community, if they would be happy with the solution (“This is the fastest solution that could be realized”). If yes (at least for single pages):
WMDE would build an extension to render pdfs “the new way” .

WMF-Gabriel’s Services team would maintain and run the server on their cluster.

- JS-Handling of javascript that has not been sanitized by us
    - Users might have any type of js included in their user pages that might be injected in pdfs.
    - Could we disable js or the user page for the pdf rendering? Or better: Render the page as an anonymous user would see it, then no user specific js code would be executed
- Make sure you can only pass in wikipedia sites
    - Petr from Services has a prototype pdf entry point spec for the REST API (en.wikipedia.org/api/rest_v1)
-

Hack 2:

- Revamp CScott’s tables in pdf patch. When rendering the pdf first try to render it with tables. If this doesn’t work, render it without tables, and add an info, that tables could not be rendered and were therefore omitted.
- Ideally get volunteers involved in improving more and more table types

Open questions:

- Do we want to replace the link “download as pdf” or offer the new option on the collection page?
- Can we remove the “See printable version” link if we do Hack 1?

Wikibooks support

It looks like most (all?) books in https://en.wikibooks.org/wiki/Wikibooks:Featured_books have a printable version already, which could serve as a stop-gap until the collection feature is supported natively.

Performance

- Wikibooks LaTeX: 140 pages, ~16s render time in labs

- Wikibooks C programming: 198 pages, ~12s render time in labs
- Wikibooks Control Systems: 147 papes, 25s render time
- Wikibooks Haskell: 440 pages, 36s render time
- Enwiki Barack Obama: 47 pages, 5s render time
- Book with loads of pages: Health Care (~1900 pages, static test copy): Does not render within timeout, but timeout triggers reliably & frees resources.

Emanuel from Kiwiks is generelly interested in our work, and everythign that we do to clean up css and html for printing