Amusewiki: a year of development

Marco Pessotto (melmothX)
Questions?

Contents

About me: Marco Pessotto 5
Appearance 6
Why would you want to use Amusewiki 7
Source file (Emacs Muse markup) 8
Architecture 9
Bookbuilder partial selection 10
Prepare books for printing with the bookbuilder 11
Slides and theme selections 12
OPDS 13
OPDS example 14
Blog mode 16
Blog example 17
Other notable improvements 18
Debian packages 19
Debian packages cons 20
Debian packages pros 21
Questions? 22
Debian packages pros

- Installation is straightforward
- Maintenance for the administrator is integrated in the regular apt-get routine
- Much smaller installation footprint (mostly because the texlive installation is optimized and shared)
- Improved security
  - code is installed and owned by root, not by the user running the application
  - debian is taking care of the security updates of the dependencies (not only the perl dependencies, but also the libraries used by LaTeX, cgit and other utilities)
- 0 downtime upgrades out of the box
- Predictability of locations and module versions (so gets more live testing)
Debian packages cons

- More burden on the developer
- Trivial changes require a package rebuild
- Needs to ship about 25 new packages (fonts and perl modules)
- Can’t use fresh code from the module used (e.g., newer Catalyst) without being too invasive.
- Debian people usually don’t like private repositories

About me: Marco Pessotto

PAUSE id MELMOTHX
IRC melmothX on Libera Chat and irc.perl.org
E-mail melmothx@gmail.com
Homepage and documentation https://amusewiki.org
IRC channel irc://irc.libera.chat/#amusewiki
GitHub https://github.com/melmothx
Background Humanities (history and translations)
Debian packages

Instead of running the app from the git tree on a dedicated home, with a local::lib tree or with a perl installed in the home, now you can visit http://packages.amusewiki.org/ and follow the instructions there (import the key, add the repository, and install the package).

Minimum requirement: Debian Jessie and Ubuntu 16.04 LTS.
https://github.com/melmothx/amusewiki-debian-packages
Other notable improvements

- SSL made free and automatic with Let’s Encrypt and Protocol::ACME
- Code highlight with highlight.js
- Automatic DB upgrades with DBIx::Class::DeploymentHandler
- Fine tune logging with Log::Log4perl and Log::Contextual
- Documentation on https://amusewiki.org

Why would you want to use Amusewiki

- built around texts, which are fully decoupled from the web application
- support for typesetting whole books (off-line editing and expressive markup)
- high-quality output (PDF with LaTeX quality and EPUB)
- flat file storage with Git
- create collections and reformat the PDF files for printing
- OPDS server for mobile users to deliver the texts straight into the apps
- Localized for English, German, Spanish, Finnish, French, Croatian, Italian, Macedonian, Russian, Albanian, Swedish
- production ready and comes with an extensive test suite
**About me: Marco Pessotto**

PAUSE id :: =MELMOTHX=
IRC :: =melmothX= on Libera Chat and =irc.perl.org=

Here the *the text* **start**.

** Blog example**
Blog mode

- Optional feature which can be activated in the admin console
- time-based sorting
- teasers
- decorative images
- tag cloud (accessible via ajax)
- monthly archives (accessible via ajax)
- form free layout elements (gives a chance to the admin to fully customize the site, in addition to local JS and CSS)
- bootswatch theme selection https://bootswatch.com/

Architecture

- Muse parser: Text::Amuse
- HTML import and cleanup: Text::Amuse::Preprocessor (with CLI)
- Compiler: Text::Amuse::Compile (with CLI)
- PDF cropmarks: PDF::Cropmarks (with CLI)
- PDF imposition: PDF::Imposition (with CLI)
- DBIx::Class and Xapian for archive indexing
- Background daemon for indexing and compilation
- Catalyst application on the frontend
## Bookbuilder partial selection

<table>
<thead>
<tr>
<th>Section</th>
<th>Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title, author, date...</td>
<td>✔️</td>
</tr>
<tr>
<td>Creating slides</td>
<td></td>
</tr>
<tr>
<td>Full example</td>
<td></td>
</tr>
<tr>
<td>Activate</td>
<td></td>
</tr>
<tr>
<td>Explanation</td>
<td></td>
</tr>
<tr>
<td>Syntax</td>
<td></td>
</tr>
<tr>
<td>Explanation</td>
<td></td>
</tr>
</tbody>
</table>

- **Search**: Search books on amusewiki.org

- **New**: Latest entries

- **Titles**: texts sorted by title

- **Topics**: texts sorted by topic

- **Authors**: texts sorted by author
OPDS example

Prepare books for printing with the bookbuilder

activate
- Amenable to provide aids
- Activate it in the editor to enable them
- Flag the text accordingly with *markers* you

explanation
- See above, I won’t repeat it here
Slides and theme selections

OPDS

- Open Publication Distribution System http://opds-spec.org/
- Atom feed with navigation and acquisition entries
- Supported by most EPUB Android readers (FBReader, Aldiko, Moon+ reader)
- Deliver the texts straight into the applications
- Module XML::OPDS
- Example and help: https://amusewiki.org/help/opds
- Transparent pagination to save bandwidth
- HTTP authorization supported for private sites
- Full text-search supported