Amusewiki: a year of development

Marco Pessotto (melmothX)

August 24-26, 2016, Cluj-Napoca

About me: Marco Pessotto

PAUSE id `MELMOTHX`

IRC `melmothX` on Libera Chat and `irc.perl.org`

E-mail `melmothx@gmail.com`

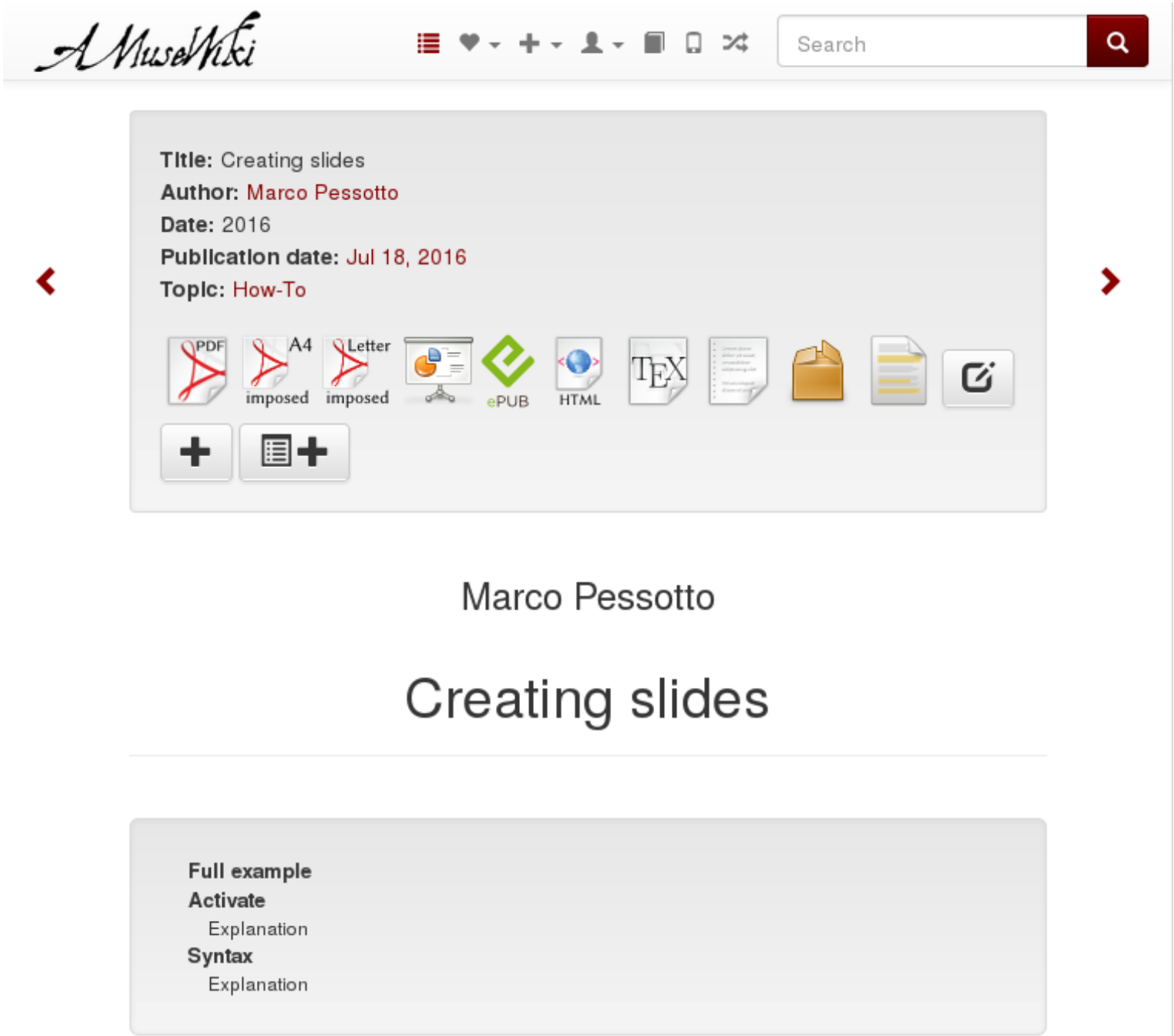
Homepage and documentation `https://amusewiki.org`

IRC channel `irc://irc.libera.chat/#amusewiki`

GitHub `https://github.com/melmothx`

Background Humanities (history and translations)

Appearance


MuseWiki
Search
Title: Creating slides
Author: Marco Pessotto
Date: 2016
Publication date: Jul 18, 2016
Topic: How-To
PDF
A4
imposed
Letter
imposed
ePUB
HTML
TEX
Marco Pessotto
Creating slides
Full example
Activate
Explanation
Syntax
Explanation

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

Source file (Emacs Muse markup)

```
#title Amusewiki: a year of development
#author Marco Pessotto (melmothX)
#slides on
#lang en
#date August 24-26, 2016, Cluj-Napoca
#pubdate 2016-08-27

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

** About me: Marco Pessotto

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

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

Add the selected parts to the bookbuilder

Section	Select
Title, author, date...	☑
Creating slides	☐
Full example	☐
Activate	☐
Explanation	☐
Syntax	☐
Explanation	☐

Prepare books for printing with the bookbuilder

Amusewiki, beside the usual output formats (HTML, EPUB, LaTeX, HTML), is able to create slides in PDF via the Beamer package with LaTeX.

The slides feature is off by default, but you can easily activate it in the admin console with the checkbox under "Formats".

Now, given that you generally don't want slides for *all* your texts, you have to flag it accordingly. You can do so with the provided checkbox in the text creation page, or simply adding the header

```
#slides yes
```

in the editing screen (i.e., in the `.muse` file).

It's important to note that the slide material must be placed inside a section, no matter how deep.

Material at the beginning of the text, not belonging to any section, is just ignored.

You can exclude sections of the documents from the slides by placing a comment cookie in the excluded sections.

```
*** this section will be excluded

; no slides

Here goes the lengthy explanation...
```

This is useful if you want to create slides and handout from the same source (as you should).

Finally, when you have the document ready, you can change the theme, the colors and the fonts adding it to the bookbuilder (using the button in the text's infobox) and asking for slides.

Activate

- Amusewiki can produce slides
- Activate it in the admin to enable them
- Flag the text accordingly with `#slides yes`

Explanation

See above, I won't repeat it here

Slides and theme selections

PDF | Slides | EPUB

Please select the desired fonts

Keep in mind that some character may be missing in the required font. Missing character will be reported.

Please choose the sans font

Computer Modern Sans Serif

Please choose the mono font

Computer Modern Typewriter Text

Please choose the font size

10 pt

Please setup the Beamer layout

Here you can find a preview of the Beamer themes and color themes

Please choose the Beamer theme

Boadilla

Please choose the Beamer color theme

wolverine

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`

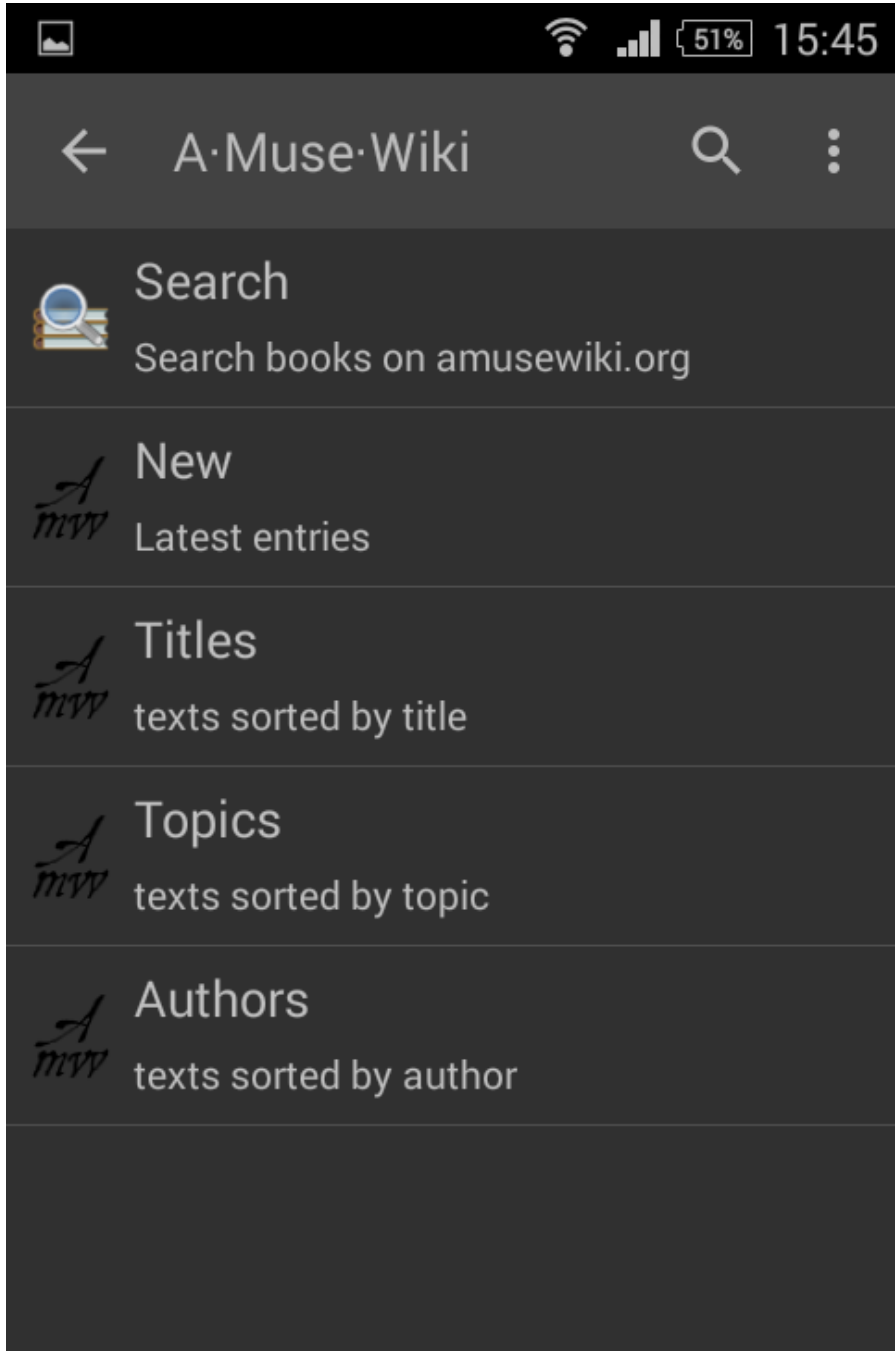
Example and help: `https://amusewiki.org/help/opds`

Transparent pagination to save bandwidth

HTTP authorization supported for private sites

Full text-search supported

OPDS example



Blog mode

Optional feature which can be activated in the admin console

time-based sorting

teasers

decorative images

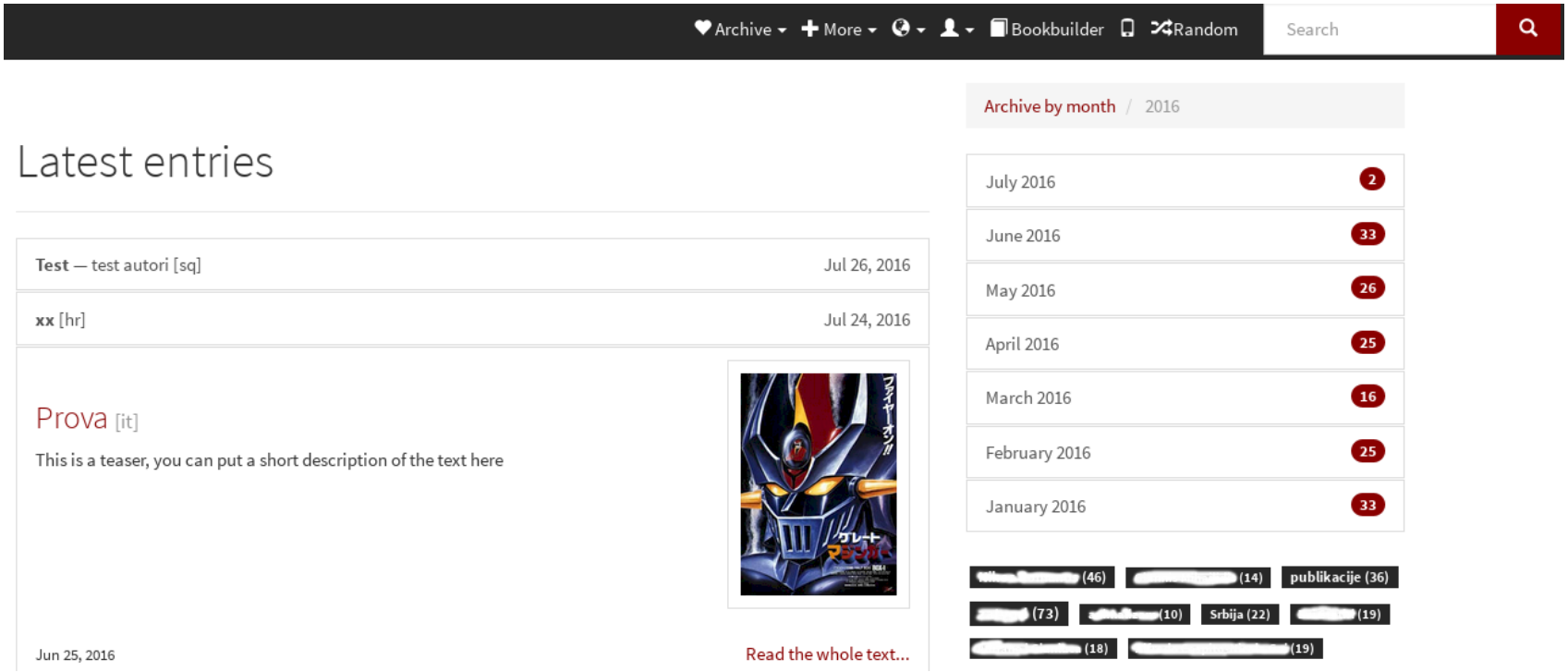
tag cloud (accessible via ajax)

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/`

Blog example



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`

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`

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

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)

Questions?

Thanks!