

# Amusewiki: a year of development

Marco Pessotto (melmothX)

August 24-26, 2016, Cluj-Napoca

# About me: Marco Pessotto

**PAUSE id** MELMOTHX

**IRC** melmothX on Freenode and [irc.perl.org](http://irc.perl.org)

**E-mail** [melmothx@gmail.com](mailto:melmothx@gmail.com)

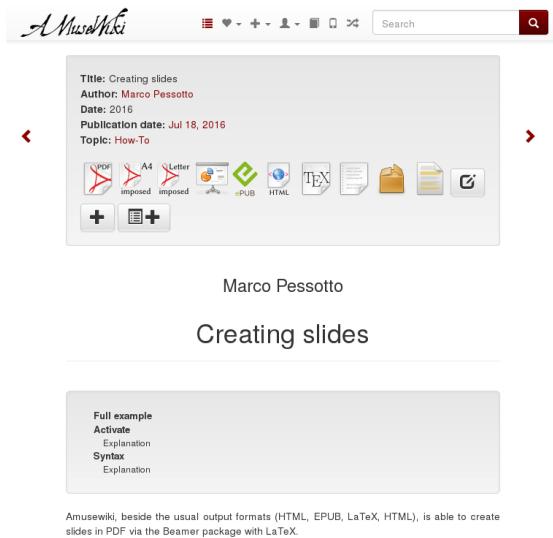
**Homepage and documentation** <https://amusewiki.org>

**IRC channel** <irc://chat.freenode.net/#amusewiki>

**GitHub** <https://github.com/melmothx>

**Background** Humanities (history and translations)

# Appearance



The screenshot shows the Amusewiki interface. At the top left is the logo "A MuseWiki". To its right is a navigation bar with icons for home, heart, plus, user, list, mobile, and refresh. Further right is a search box with a magnifying glass icon. The main content area features a slide card with the following information:

- Title:** Creating slides
- Author:** Marco Pessotto
- Date:** 2016
- Publication date:** Jul 18, 2016
- Topic:** How-To

Below the text are icons for various output formats: PDF, A4 imposed, Letter imposed, Beamer presentation, ePUB, HTML, LaTeX, and a folder icon. There are also two plus signs in boxes. Below the slide card, the author's name "Marco Pessotto" and the title "Creating slides" are displayed. A section titled "Full example" contains sub-sections "Activate" and "Syntax", each with an "Explanation" link. At the bottom, a paragraph states: "Amusewiki, beside the usual output formats (HTML, EPUB, LaTeX, HTML), is able to create slides in PDF via the Beamer package with LaTeX."

# 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 Freenode 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
<b>Title, author, date...</b>	<input checked="" type="checkbox"/>
Creating slides	<input type="checkbox"/>
<b>Full example</b>	<input type="checkbox"/>
<b>Activate</b>	<input type="checkbox"/>
Explanation	<input type="checkbox"/>
<b>Syntax</b>	<input type="checkbox"/>
Explanation	<input type="checkbox"/>

# Prepare books for printing with the bookbuilder

Pg 0004

6677.pdf Thu Aug 11 11:11:10 2016 page 4 #1/4

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 "Format".

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.

4

Pg 0004

6677.pdf Thu Aug 11 11:11:10 2016 page 4 #1/4

6677.pdf Thu Aug 11 11:11:10 2016 page 9 #1/9

Pg 0009

## 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

9

Pg 0004

6677.pdf Thu Aug 11 11:11:10 2016 page 4 #1/4

6677.pdf Thu Aug 11 11:11:10 2016 page 9 #1/9

Pg 0009



# 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

Build it

Update

# 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:

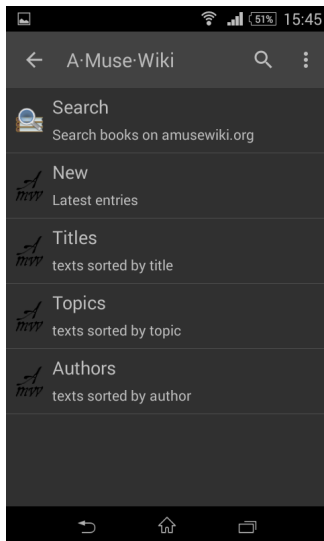
<https://amusewiki.org/help/opds>

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

## Latest entries

Test — test autori [sq]

Jul 26, 2016

xx [hr]

Jul 24, 2016

Prova [it]

This is a teaser, you can put a short description of the text here



[Read the whole text...](#)

Jun 25, 2016

### Archive by month / 2016

July 2016	2
June 2016	33
May 2016	26
April 2016	25
March 2016	16
February 2016	25
January 2016	33

📄 [46]	📄 [14]	publikacije [36]
📄 [73]	📄 [10]	Srbija [22]
📄 [18]	📄 [19]	

## 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!