

Write great documents



Thomas Lutz



Write great documents:

Thomas Lutz

[GNU General Public License (GPL) version 3](<https://www.gnu.org/licenses/gpl.html>)

Table of Contents

Introduction	1
Prerequisites	1
Usage	1
Configuration	2
Directory structure	2
Writing Docbook	3
Plantuml	4
Advanced example	4
Converter	5
Gitlab Wiki / Markdown / Pandoc	6
Usage without Docker	7
Maven	7
Note	8

List of Figures

1. uml	4
2. Sequence Diagram	4

Introduction

As a software engineer I love to work with source code that comes with Unit Tests and a well written documentation. Markdown is surely a good way to write your documents. But sometimes it comes in handy to have a shiny PDF to store offline, send via email or print in high resolution. For those cases Docbook (co-founder [O'Reilly Media](https://www.oreilly.com/) [https://www.oreilly.com/]) is a good solution because it supports different output formats like pdf, epub or html. My project [livingfire-docbook](https://gitlab.com/phoen1x/livingfire-docbook) [https://gitlab.com/phoen1x/livingfire-docbook] shows a easy way to get started with Docbook and alter this [example](https://gitlab.com/phoen1x/livingfire-docbook/raw/master/book.pdf) [https://gitlab.com/phoen1x/livingfire-docbook/raw/master/book.pdf] for your own needs. The software can also covert Markdown to Docbook using [pandoc](http://pandoc.org/) [http://pandoc.org/] and combine multiple Markdown files like a GitLab Wiki into one PDF. In addition to that you can write UML diagrams in [PlantUML](http://plantuml.com/) [http://plantuml.com/] and have them rendered into pictures in your book.

Prerequisites

Make sure you have a working [Docker](https://docs.docker.com/engine/installation/) [https://docs.docker.com/engine/installation/] and [docker-compose](https://docs.docker.com/compose/install/) [https://docs.docker.com/compose/install/] environment.

Usage

```
# download
git clone https://phoen1x@gitlab.com/phoen1x/livingfire-docbook.git
cd livingfire-docbook

# start project
docker-compose up -d

# docbook to pdf
docker-compose exec docbook /book/convertBook.sh
xdg-open book/target/docbkx/pdf/book.pdf

# import GitLab wiki
cd tmp/wiki
git init
git remote add origin https://gitlab.com/phoen1x/livingfire-docbook.wiki.git
git pull
rm -rf .git
cd ../../

# markdown to docbook
docker-compose exec docbook /book/convertPlantuml.sh
docker-compose exec docbook /book/convertWiki2Docbook.sh
xdg-open book/target/docbkx/pdf/book.pdf

# stop project
docker-compose down
cat book/target/docbkx/pdf/book.pdf > book.pdf
```

Configuration

Directory structure

```
book
|-- pom.xml
`-- src
    |-- main
    |   |-- documentation
    |   |   |-- docbook
    |   |   |   |-- book.xml
    |   |   |   |-- template_chapter.xml
    |   |   |   |-- chapters
    |   |   |       |-- chapter_1.xml
    |   |   |       |-- chapter_2.xml
    |   |   |       |-- chapter_n.xml
    |   |   |   |-- media
    |   |   |       |-- cover.png
    |   |   |       |-- logo.png
    |   |-- fonts
    |   |-- xsd
    |   |-- xslt
    |   |-- docbook.xsl
```

Probably everything you ever need to write a book is within the [livingfire-docbook/book/src/main/documentation/docbook](https://gitlab.com/phoen1x/livingfire-docbook/blob/master/book/src/main/documentation/docbook) [https://gitlab.com/phoen1x/livingfire-docbook/blob/master/book/src/main/documentation/docbook] directory. Here you can find the [book.xml](https://gitlab.com/phoen1x/livingfire-docbook/blob/master/book/src/main/documentation/docbook/book.xml) [https://gitlab.com/phoen1x/livingfire-docbook/blob/master/book/src/main/documentation/docbook/book.xml] and a templates for your own content. To tweak the style of your book alter [docbook.xsl](https://gitlab.com/phoen1x/livingfire-docbook/blob/master/book/src/main/documentation/xslt/docbook.xsl) [https://gitlab.com/phoen1x/livingfire-docbook/blob/master/book/src/main/documentation/xslt/docbook.xsl] and [pom.xml](https://gitlab.com/phoen1x/livingfire-docbook/blob/master/book/pom.xml) [https://gitlab.com/phoen1x/livingfire-docbook/blob/master/book/pom.xml].

Writing Docbook

You can use a Markdown Wiki like [Gollum](https://github.com/gollum/gollum/wiki) [https://github.com/gollum/gollum/wiki] to create your docbook chapters and then further refine the document using these links:

- [DocBook Cheat Sheet](https://workaround.org/docbook-reference/) [https://workaround.org/docbook-reference/]
- [DocBook Tutorial](http://www.vogella.com/tutorials/DocBook/article.html) [http://www.vogella.com/tutorials/DocBook/article.html]
- [DocBook Information](http://bikesutorrent.weebly.com/blog/generate-pdf-docbookdownload-free-software-programs-online) [http://bikesutorrent.weebly.com/blog/generate-pdf-docbookdownload-free-software-programs-online]
- [Page layout](http://www.sagehill.net/docbookxsl/PrintOutput.html) [http://www.sagehill.net/docbookxsl/PrintOutput.html]
- [Headers and Footers](http://www.sagehill.net/docbookxsl/PrintHeaders.html) [http://www.sagehill.net/docbookxsl/PrintHeaders.html]
- [Maven + Docbook](http://docbkx-tools.sourceforge.net/) [http://docbkx-tools.sourceforge.net/]

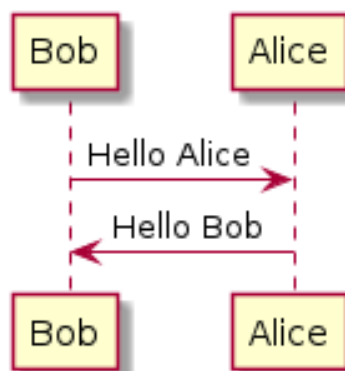
Plantuml

The project is also capable of converting [PlantUML](http://plantuml.com/) [http://plantuml.com/]

```
@startuml
Bob  -> Alice : Hello Alice
Alice -> Bob  : Hello Bob
@enduml
```

into an image

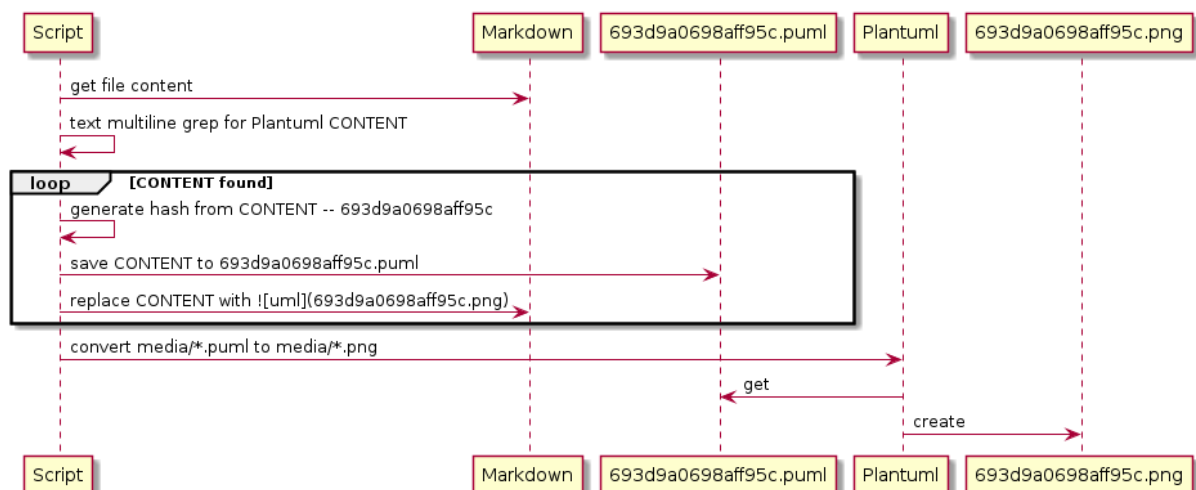
Figure 1. uml



Advanced example

```
@startuml
Script -> Markdown: get file content
Script -> Script : text multiline grep for Plantuml CONTENT
loop CONTENT found
    Script -> Script : generate hash from CONTENT -- 693d9a0698aff95c
    Script -> 693d9a0698aff95c.puml: save CONTENT to 693d9a0698aff95c.puml
    Script -> Markdown: replace CONTENT with ![uml](693d9a0698aff95c.png)
end
Script -> Plantuml: convert media/*.puml to media/*.png
Plantuml -> 693d9a0698aff95c.puml: get
Plantuml -> 693d9a0698aff95c.png: create
@enduml
```

Figure 2. Sequence Diagram



More information on Plantuml can be found at the [PlantUML website](http://plantuml.com/) [http://plantuml.com/].

Converter

The source code of my converter can be found [here](https://gitlab.com/phoen1x/livingfire-docbook/blob/master/book/src/main/java/de/livingfire/PlantumlConverter.java) [https://gitlab.com/phoen1x/livingfire-docbook/blob/master/book/src/main/java/de/livingfire/PlantumlConverter.java].

Gitlab Wiki / Markdown / Pandoc

To manually convert Markdown you need to install [pandoc](http://pandoc.org/) [http://pandoc.org/]

```
# apt-get install pandoc
pandoc --chapters -t docbook --output page.xml page.md
```

Usage without Docker

Maven

```
# install Java -- https://java.com/en/download/  
git clone https://phoenix@gitlab.com/phoenix/livingfire-docbook.git  
cd livingfire-docbook  
  
# Linux  
./mvnw clean site  
xdg-open target/docbkx/pdf/book.pdf  
  
# Windows  
mvnw.cmd clean site  
# Open target/docbkx/pdf/book.pdf in file browser
```

Note

This book was written in a [GitLab Wiki](https://gitlab.com/phoen1x/livingfire-docbook/wikis/home) [https://gitlab.com/phoen1x/livingfire-docbook/wikis/home]. Feel free to add your improvements via [Merge Request](https://gitlab.com/phoen1x/livingfire-docbook) [https://gitlab.com/phoen1x/livingfire-docbook] or open an [issue](https://gitlab.com/phoen1x/livingfire-docbook/issues) [https://gitlab.com/phoen1x/livingfire-docbook/issues] if you find a bug.