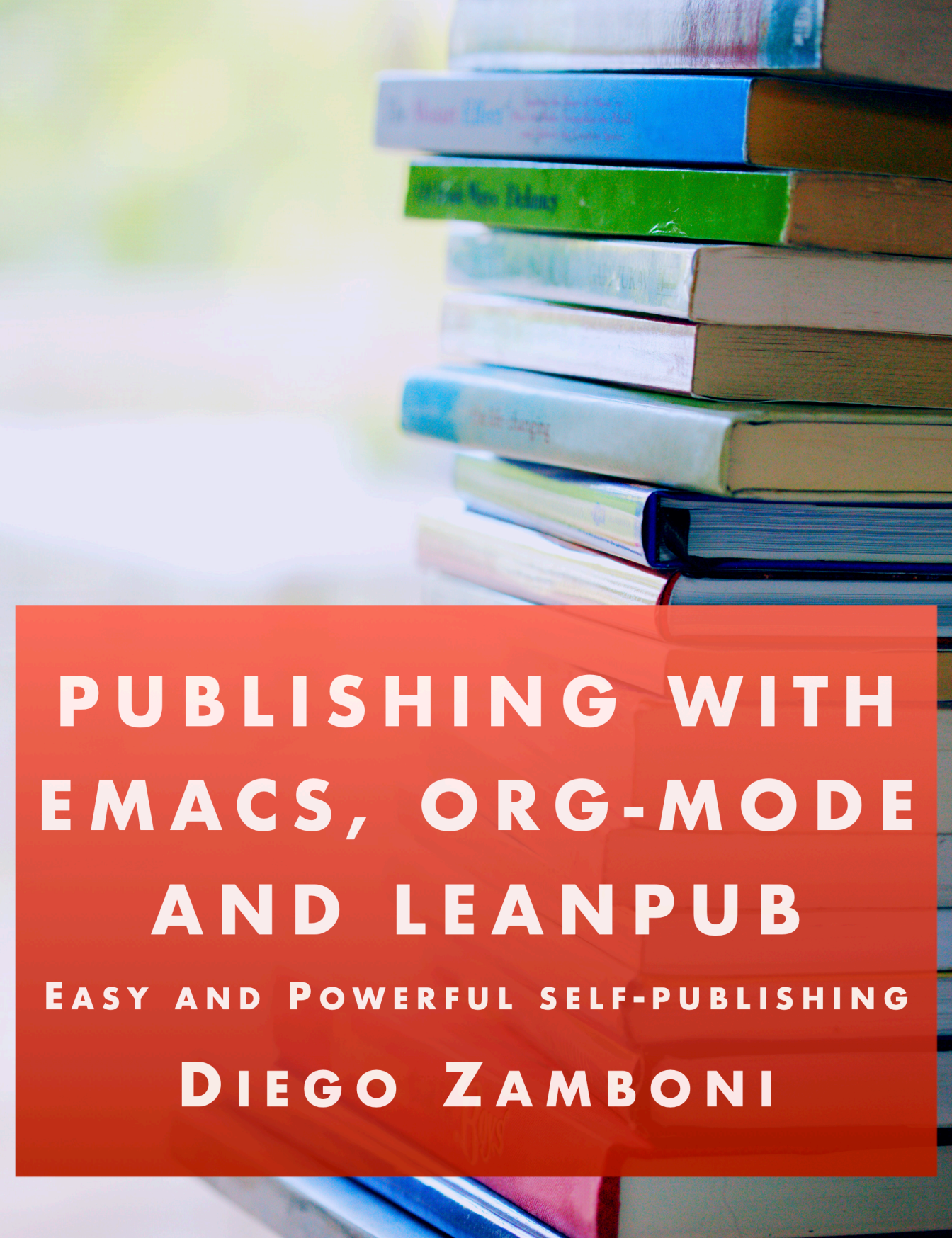


PUBLISHING WITH EMACS, ORG-MODE AND LEANPUB

Easy and Powerful self-publishing

Diego Zamboni

Publishing with Emacs, Org-mode and Leanpub

Easy and Powerful self-publishing

Diego Zamboni

This book is for sale at http://leanpub.com/emacs-org-leanpub

This version was published on 2021-07-16

Also By Diego Zamboni

Learning Hammerspoon

Learning CFEngine

Utilerías de Unix

Literate Configuration

Contents

1. Preface to the early release

What you are reading is an early release of this book, which means its contents is not yet complete, and you may find errors or omissions in it. Still, I hope you find it useful. I would really appreciate your feedback! Please let me know what you liked, what you didn't like, and what additional topics you would like me to cover. Please send me a message through the *Email the Author* page and let me know what you think.

Release notes

December 2020

This second release includes many improvements and new content, including:

- A whole new chapter *The workflow*, which covers detailed techniques for writing, exporting, previewing and publishing your book.
- A new section *Code block execution and output processing*, which shows how you can have code within your document which is executed on the fly, and which you can use to generate content within your book.
- Two new appendices, the first one containing the source code for the diagram shown in *The process*, and the second one containing samples of all the different block types supported by `ox-leanpub`.

- Many other changes, including new instructions on configuring Emacs Doom, innumerable wording, structure and clarity improvements, and a lot more.

With this release, the book is no longer free. If you find it valuable, I would really appreciate your support.

Hope you enjoy it!

June 2020

This is the initial release of the book! I would love to hear your thoughts. There is still much missing but the core of the book is there, and you can probably already use it to get started writing a book and publishing it on Leanpub using Emacs and Org-mode.

2. Introduction

Publishing your words has never been easier than it is today. Traditionally, publishing a book was a long and arduous process, in which you had to go through multiple gatekeepers to have a chance to see your words on paper. Today, publishing a book has become considerably easier through self publishing. There are many tools and publishers that allow you to get started for little or no money. Still, **getting started** can be confusing, and that is what this book is about.

In this book, I will show you the workflow and tools I use to publish my books. This toolset is geared towards technical writing, which is what I do, but can just as well be used for any other style. The three main tools are:

- The GNU Emacs editor together with Org mode and the ox-leanpub exporter for writing, editing and exporting your text;
- GitHub to store your book files.
- Leanpub for typesetting, publishing and selling your work.

The following are optional, but highly recommended for making the workflow more automated and consistent:

- GitHub Actions, CircleCI or some other CI/CD tool, for automating the typesetting, previewing and publication process;
- Hammerspoon (if you are using a Mac) for monitoring book builds.

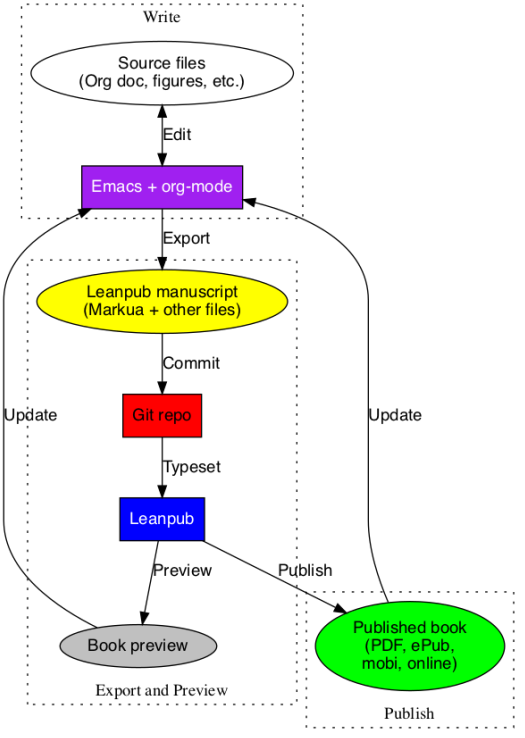
To illustrate the process and provide you with a starting point, the source repository for this book is available in GitHub. I am

populating the repository as I write this book, so you can follow the progress there. In the repository you can see all the “behind the scenes” text and configuration, and follow along as I describe them to you.

There are as many possible variations to this workflow as users out there. If you have any suggestions or tips that you think should be mentioned, please let me know!

The process

The high-level workflow for publishing a book using Emacs, org-mode and Leanpub looks like this:



In this setup, the only files you manipulates directly are your source files. For your text, you use Emacs with org-mode. In many cases, you can also generate figures from within Org mode, if they can be programmatically produced. Emacs exports your Org files into the structure and format required by Leanpub, and you commit the result to a Git repository (Leanpub at the moment supports GitHub and Bitbucket). From there, Leanpub picks up the files and produces a preview of your book, which you can review to make further changes. When you want to publish your work, you instruct Leanpub to do so, and Leanpub takes care of producing the end result and publishing it for the word to see.

One of Org-mode's amazing features is its ability to evaluate code within the source file and include the code, its output or both in the resulting document. For example, the graph you see above is produced by evaluating the code from Appendix A: Workflow diagram source code using the `dot` command from the Graphviz package. Whenever I need to update the diagram, I type the changes directly in the document and press `C-c C-c` within the code block. Org-mode automatically renders the diagram and updates the figure in the document.

Leanpub

Leanpub is a powerful platform for self-publishing. It allows you to create books and courses and to publish, sell and update them directly through the Leanpub portal. It supports a comprehensive API which allows you to automate most operations, tough you can of course do them via the web UI if you prefer.

Books and courses are written in a specialized markup language called Markua, which is a derivative of Markdown but with some additional features that make it easier to publish longer works.

Leanpub also supports Leanpub Flavored Markdown, which is an older version of its markup language. Markua is the recommended format, and the one we use in this book.

I like Leanpub not only for the technical aspect of it (self-publishing, text-based markup, the API) but also because of the company behind it. My interactions with them have always been met with helpful and constructive responses, and you can tell they genuinely care about building the best possible publishing platform.

If you are not convinced yet, read Why Leanpub.

Emacs and org-mode

Emacs is a powerful open-source editor which runs on any platform, and offers infinite extensibility through its ability to be programmed using the embedded Emacs LISP programming language. If you are not familiar with Emacs, check out the Guided Tour of Emacs to get started.

The setup described in this book is based on Org mode, an incredibly powerful markup language and set of tools that allows keeping notes, tracking tasks, and writing documents. One of its key features is the concept of *Exporters*: the same Org document can be exported in a multitude of formats using any of the built-in exporters or a wide variety of community-developed exporters.

One of these exporters is `ox-leanpub`, which handles the conversion of Org markup to Leanpub's Markua or Markdown formats, and also splitting and structuring the files as expected by Leanpub. `Ox-leanpub` allows you to write your book or course entirely in Org mode, and completely manages the production of the files needed by Leanpub to render your material. This is the exporter we will use in this book.

Org mode is extensive and powerful, and I invite you to peruse its documentation but also to be patient — Org is best learned

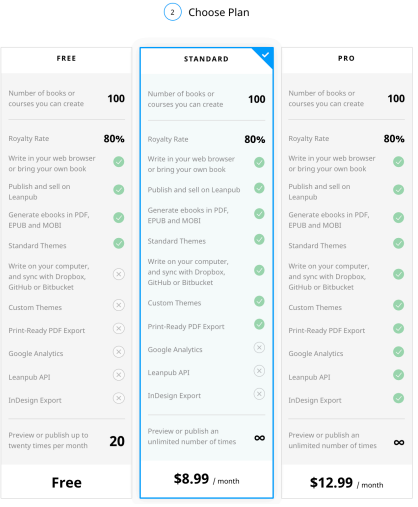
gradually and over time. Discover the parts that interest you the most first, and go from there. To get started with writing using Org, check out Getting started with Org-mode in the Org website to get an overview of the most useful markup constructs. For including source code in your documentation, I recommend reading the Working with Source Code section of the Org manual, as well as Howard Abrams' Introduction to Literate Programming.

3. Setting up

Signing up for Leanpub

If you don’t have a Leanpub account yet, you can create it by visiting https://leanpub.com/ and clicking on the “Sign Up” link on the homepage. Note that you can also automatically create your account when you create your first book, as described in Creating the book on Leanpub.

Note that among Leanpub’s pricing plans, only the Standard and Pro plans support the “Write on your computer, and sync with Dropbox, GitHub or Bitbucket” option, which is what we focus on in this book. If you want to enable webhooks and other automation aspects, you will need a Pro plan, which supports the “Leanpub API” option. You need to choose your pricing plan when you create your first book, but you can change it later at any time.



You need a Standard or Pro plan to follow along the instructions in this book.

Emacs and org-mode configuration

The centerpiece of your local setup is the Emacs editor with Org-mode and the `ox-leanpub` exporter. There are many different ways of configuring Emacs. In this chapter we will look at setting up a basic configuration by hand, and also an example of a more advanced configuration using Doom Emacs, a popular "Emacs distro" which comes with a set of ready-to-use configuration settings.

Manual Emacs configuration

Emacs configuration is read from `~/.emacs.d/init.el`. Let's review the minimum configuration you need to follow along this book.

We show here a bare-minimum example, but Emacs configuration is an art, and there is a lot of things you can configure to improve the usability, functionality and aesthetics of your setup. For a realistic working configuration example, check out my full Emacs configuration. If you are an experienced Emacs user and have your configuration set up already, please check at least the Publishing to Leanpub section!

Make sure you have at least Emacs 26.1 installed, this is the minimum needed by some of the packages you use.

First, we need to set up the Emacs package system, which enables you to easily install packages from various repositories. Add the following lines to your `init.el` file to declare the package repositories to use:

```
(customize-set-variable
 'package-archives
 '(("marmalade" . "https://marmalade-repo.org/packages/")
   ("melpa"     . "https://melpa.org/packages/")
   ("elpa"      .  "https://elpa.gnu.org/packages/")))
```

Then we initialize the package system and refresh the list of packages.

```
(package-initialize)

(when (not package-archive-contents)
  (package-refresh-contents))
```

I highly recommend using the use-package library to manage the packages in your config, since it allows easy, self-contained and declarative installation and configuration of packages. Since `use-package` is not bundled with Emacs, the first thing we do is install and load it by hand. All other packages are then declaratively installed and configured with `use-package`.

```
(when (not (package-installed-p 'use-package))
  (package-install 'use-package))
(require 'use-package)
```

Using `use-package` we can load the `org` package. This is included with Emacs.

```
(use-package org)
```

Finally, we declare `ox-leanpub`. In this case, `use-package` installs the package thanks to the `:ensure t` declaration, and it loads it only after `org` has been loaded.

```
(use-package ox-leanpub
  :ensure t
  :after org)
```

Doom Emacs configuration

Doom Emacs is one of a few "Emacs Distros" that provide configuration frameworks for more easily utilizing the multiple features of Emacs. The Doom Emacs base configuration takes care of package management, performance tuning and reasonable defaults for a number of settings, allowing you to simply select and configure additional functionality you need.

To enable Org mode and the `ox-leanpub` exporter in Doom Emacs, you need to do first enable the `org` module by making sure the following line in your `~/.doom.d/init.el` file is uncommented (it already is in the default Doom installation):

```
org               ; organize your plain life in plain text
```

You can also enable additional options for the `org` module, whose descriptions you can see in the module documentation. These are the options I use:

```
(org +pretty +journal +hugo +roam +pandoc +present)
```

To install the `ox-leanpub` exporter, you need to add the following line to your `~/.doom.d/packages.el` file:

```
(package! ox-leanpub)
```

And load the package by adding the following to `~/.doom.d/config.el`:

```
(use-package! ox-leanpub
  :after org)
```

Once you have made these changes, you need to run `doom sync` from your terminal to make sure all the necessary modules are installed, and then restart Emacs.

Creating the book locally

All the steps that follow show the actual commands and operations I performed while setting up the book you are reading! You can find the sources for the current version of this book at https://github.com/zzamboni/emacs-org-leanpub.

The first step is to choose a short name or *slug* for your book. This is the URL identifier for your book in Leanpub, and it should also be the name of your git repository (this is not mandatory, but makes the automation easier). This book's slug is `emacs-org-leanpub`, so its Leanpub URL will be leanpub.com/emacs-org-leanpub.

Creating a git repository for your book

Once we have a slug, we create a new Git repository for the new book. Leanpub supports both GitHub and Bitbucket repositories. In these descriptions I use GitHub, but similar steps apply if you are using Bitbucket.

I use the command-line utility hub to interact with GitHub from the command line. You can of course do the corresponding operations through the GitHub web interface if you so prefer. If you use Bitbucket, you can use git-spindle to interact with it from the command line as well.

```
$ cd ~/Personal/writing
$ mkdir -p emacs-org-leanpub
$ cd emacs-org-leanpub
$ git init .
Initialized empty Git repository in
  /Users/taazadi1/Dropbox/Personal/writing/emacs-org-leanpub/.git/
```

Next, we create a new GitHub repository and connect it to our local repository:

```
$ cd ~/Personal/writing/emacs-org-leanpub
$ hub create
Updating origin
https://github.com/zzamboni/emacs-org-leanpub
$ git remote -v
origin      https://github.com/zzamboni/emacs-org-leanpub.git (fetch)
origin      https://github.com/zzamboni/emacs-org-leanpub.git (push)
```

Creating the book file

Now you can start writing your text inside the new repository. I usually write the main text in a file called `book.org` in the root directory of the repository.

To get you started, a basic skeleton for a book is the following:

```
#+startup: indent
#+tags: noexport sample frontmatter mainmatter backmatter
#+options: toc:nil tags:nil

#+title: Your book title
#+author: Your name

* Introduction
```

```
Some text

* Chapter 1

Some more text
```

For a more complex example, you can find the sources for this book at https://github.com/zzamboni/emacs-org-leanpub.

Once you have some text, you can simply commit and push the changes to your remote repository:

```
$ cd ~/Personal/writing/emacs-org-leanpub
$ git add book.org
$ git ci -m "Initial commit of the book"
[master (root-commit) 3e166f4] Initial commit of the book
 1 file changed, 230 insertions(+)
 create mode 100644 book.org
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 3.72 KiB | 3.72 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/zzamboni/emacs-org-leanpub.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

Your first book export

Now that you have the initial skeleton for your book, it's time to export it from Org to Leanpub's Markua format, from which

Leanpub can produce a rendered version of your book for you to preview.

For this, we use the `ox-leanpub` module which you installed on Emacs. Pressing `C-c C-e` will show you Org-mode's Export screen. Among other options, you should see the following:

```
[M] Export to Leanpub Markua
    [M] To temporary buffer       [m] To file
    [o] To file and open
    [b] Multifile: Whole book     [s] Multifile: Subset
```

Press `M b` to export the whole book in "Multifile format", which exports your book from the Org file and creates the structure and files needed by Leanpub to render the book. For example, for this book, the following files, directories and symlinks are created (the original source file is `book.org`, everything else is created from it, note that all images stored under `manuscript/resources/images` are omitted from this listing):

```
.
├── README.org
├── book.org
├── images -> manuscript/resources/images
└── manuscript
    ├── Book.txt
    ├── Subset.txt
    ├── automation.markua
    ├── backmatter.txt
    ├── colophon.markua
    ├── frontmatter.txt
    ├── images -> resources/images
    ├── introduction.markua
    ├── mainmatter.txt
    ├── preface-to-the-early-release.markua
    ├── resources
```

```
│   └── images
├── setting-up.markua
├── the-workflow.markua
└── tips-and-tricks.markua
```

This file tree is also generated automatically from within the book's source document, by using a simple shell command inside an Org-mode `src` block, like this:

```
#+name: book-tree
#+begin_src bash :results output :exports results :wrap example
  tree --noreport -L 3 -I 'covers|files'
#+end_src
```

In short, this is what the export operation does:

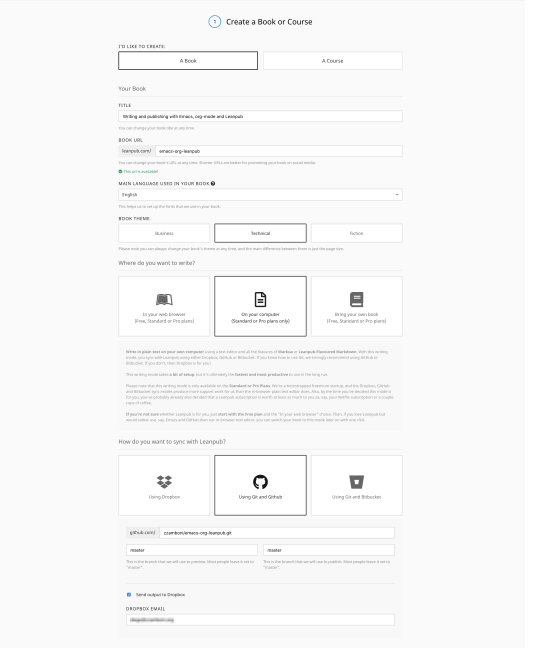
- Creates a `manuscript` folder if needed, under which all other files are stored.
  - A `resources/images` directory is created inside `manuscript`, as required by the Leanpub Markua exporter.
  - Symlinks to the `images` directory are created both from the top-level directory, and from the `manuscript` directory, to allow referencing the same image files both from the Org file and from the exported Markua files.
- Exports one `.markua` file for each top-level header (chapter) in your book.
- Creates the `Book.txt` file with the filenames corresponding to the chapters of your book.
  - Depending on the exporter settings, the `Subset.txt` and `Sample.txt` files may also be created.

Creating the book on Leanpub

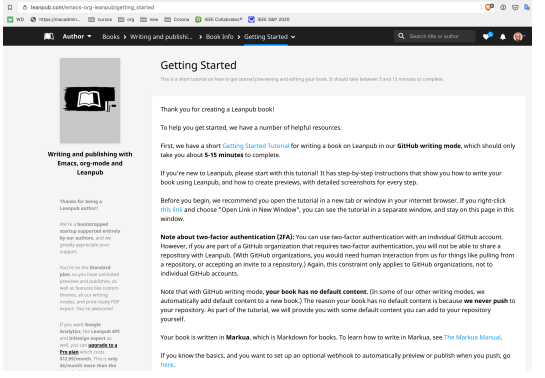
Now that you have the basics of a book, you need to create a new book in Leanpub and link it to your Git repository. Assuming you are signed into your Leanpub account, you can do this by visiting https://leanpub.com/create/book, and following the prompts. In particular, note the following:

Leanpub's pricing plans changed in October 2019, making it necessary to have a "Standard" or "Pro" plan to be able to write your book using Git integrations.

- The "On your computer" option under "Where do you want to write" is only available in the paid Leanpub plans. Choose the corresponding git option, and enter the path to your repository as created above.
- I suggest you use the same name for the "Book URL" and for your Git repository. This is not mandatory, but it makes some of the automation easier (particularly for integration with CI/CD systems).
- I usually enable "Send output to Dropbox" to always have the latest rendered version of my books synchronized to my machine, but feel free to leave it disabled if you prefer.



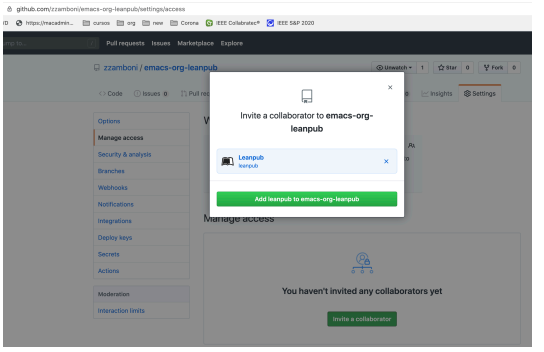
After you create the book, Leanpub shows you a “Getting Started” page which describes some additional steps you need to complete to finish setting up your book and its integration with your git repository.



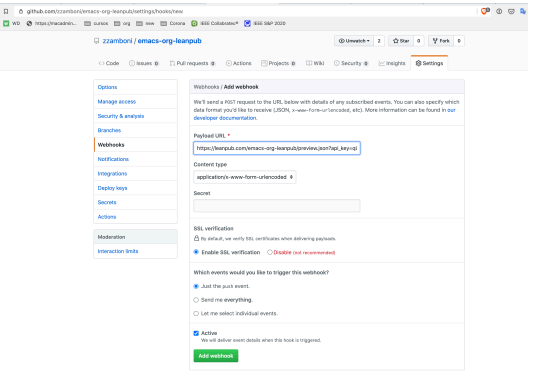
Make sure you follow these instructions, which include:

- Adding Leanpub as a collaborator to your GitHub or BitBucket

repository. This makes it possible for Leanpub to read the files from your repository to render the book.



- Adding a webhook to your repository to trigger an automatic preview of your book whenever you push new changes to your repository. This is an optional step, but one which makes it much easier to generate the book whenever you make changes. Not this this “hardcodes” the type of book generation (Preview or Publish) which happens when you push changes. For a more complex setup, see CI/CD for previewing and publishing.



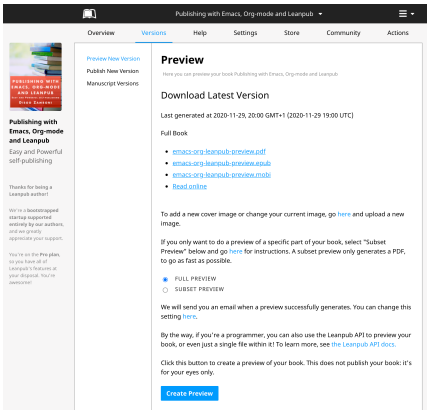
Leanpub’s “Edit Webhook” page includes your real Leanpub API key, which you should keep secret, as it enables access to all operations on your book.

Your first book preview

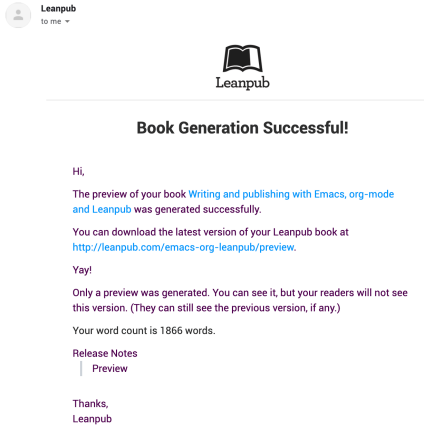
Once you have created your book on Leanpub and connected it to your Git repository, you are ready to produce your first preview. Follow the same steps shown in Your first book export to generate the Markua files from your Org file, and then commit and push the changes to your repository.

```
$ cd ~/Personal/writing/emacs-org-leanpub
$ git add .
$ git ci -m 'Commit for first book Preview'
[master edc3c97] Commit for first book Preview
 7 files changed, 80 insertions(+), 9 deletions(-)
$ git push
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Writing objects: 100% (12/12), 1.32 MiB | 1.28 MiB/s, done.
To https://github.com/zzamboni/emacs-org-leanpub.git
   00a67a1..edc3c97  master -> master
```

If you configured the webhook in the previous steps, the `git push` will automatically trigger a book preview. If you did not, you can visit the Preview page of your book at *https://leanpub.com/your-book-id/preview* and clicking on the “Create Preview” button.



In any case, after a few minutes you will get an email from Leanpub telling you about the preview creation.

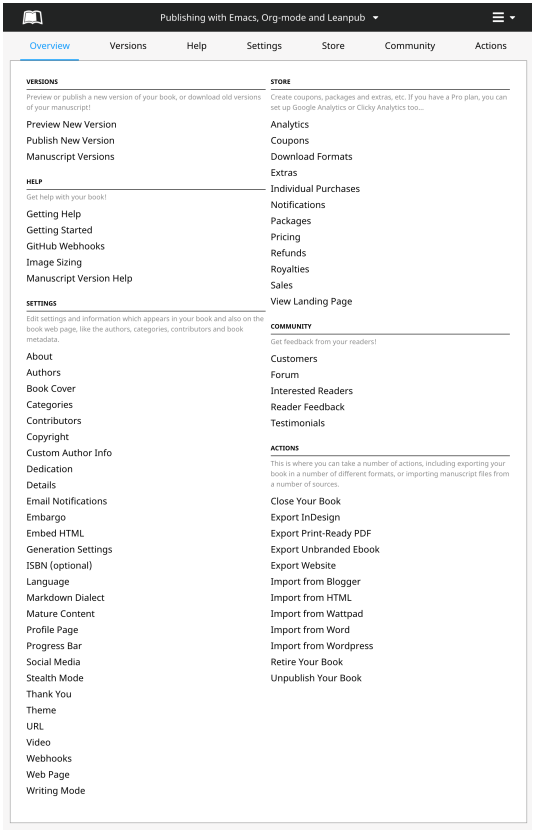


If you enabled Dropbox integration, the generated preview files of your book will be automatically added to your Dropbox folder, and you can also download them from the Preview page of your book.

Congratulations! You have executed the first end-to-end build of your book, and you can now hold in your hands (or in your PDF viewer, at least) its very first copy.

Additional book configuration

Leanpub offers you a great degree of control over all aspects of your books aspect and production. All of them are optional, so you can do them as you explore and feel more comfortable. I highly recommend that you explore your book's management menu in Leanpub.



Here are some of the common things that I like to configure right away:

Visual settings

Found in *Settings* / *Theme*. You can choose one of the ready-

made themes as chosen when you created the book (Business, Technical, Fiction) but I like to choose “Custom” and fine-tune the different parameters. You can customize the page size, fonts, line spacing, line numbering in code blocks, and many other things. I recommend you set at least the page size you want before creating a cover for your book, since it determines the size of the image you have to upload. But otherwise, feel free to experiment, preview your book with different settings and choose the look you like best.

Cover

Found in *Settings* / *Book Cover*. This is perhaps one of the most important visual aspects you’ll want to configure to make sure your book stands out. You can upload an image to use as your book cover, although the page also notes that if you do not upload an image, you can also set the cover of your book by storing an image called `title_page.png` or `title_page.jpg` inside your `images` folder. The size of the image depends on the page size of your book.

Title and subtitle

Found in the *Settings* / *Details* page. You can modify the title you specified during book creation, and optionally add a subtitle.

Description and teaser text

Found in *Settings* / *About*. You enter here multiple text description for your book, which appear in the book’s webpage in Leanpub.

Categories

Found in *Settings* / *Categories*. Allows you to define certain categories for your book, to make them easier to find by readers.

These are just some of the highlights - you should explore all the sections!

4. The workflow

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

Writing

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

Chapters and parts

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

Directives

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

Conditional book and sample inclusion

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

Block elements

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

Block captions

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

Code blocks

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

Leanpub attributes

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

Other Markua and LFM elements

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

Exporting and Previewing

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

Publishing

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

Tagging your book releases

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

5. TODO Automation

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

CI/CD for previewing and publishing

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

Basic concepts

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

Example: Using CircleCI

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

Example: Using GitHub Actions

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

Triggering and monitoring book builds

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

From the command line

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

Using Hammerspoon

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

6. DRAFT Tips and tricks

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

Headline export levels

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

org-special-ctrl-a/e/k

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

Visual configuration

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

Code block execution and output processing

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

7. Appendix A: Workflow diagram source code

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

8. Appendix B: Block types

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.

9. Colophon

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/emacs-org-leanpub.