
目錄

介绍	1.1
基本安装	1.2
Node.js安装	1.2.1
Gitbook安装	1.2.2
Gitbook命令行速览	1.2.3
图书项目结构	1.3
README.md 与 SUMMARY编写	1.3.1
目录初始化	1.3.2
图书输出	1.4
输出为静态网站	1.4.1
输出PDF	1.4.2
发布	1.5
发布到Github Pages	1.5.1
发布到公司文档服务器	1.5.2
结束	1.6

Introduction

Gitbook 使用教程 使用 Git 和 Markdown 制作精美在线电子书。

支持格式 GitBook 支持输出多种文档格式，如：

静态站点：GitBook 默认输出该种格式，生成的静态站点可直接托管搭载 Github Pages 服务上； PDF：需要安装 ebook-concert 依赖； eBook：需要安装 ebook-concert； 单 HTML 网页：支持将内容输出为单页的 HTML，不过一般用在将电子书格式转换为 PDF 或 eBook 的中间过程； JSON：一般用于电子书的调试或元数据提取。 Gitbook 项目地址 GitBook 项目官网：<http://www.gitbook.io> GitBook Github 地址：<https://github.com/GitbookIO/gitbook> toolchain

<https://toolchain.gitbook.com/> 本项目地址 仓库：<https://github.com/einverne/gitbook-tutorial> 在线阅读：<https://einverne.github.io/gitbook-tutorial>

基本安装 这一章主要讲 Gitbook 的环境及安装，以及命令行的使用概览。

本地安装 在本地安装 gitbook 命令非常简单，本地系统需要依赖

NodeJS v4.0 及以上 系统需要满足 Windows, Linux, Unix, or Mac OS X

Node.js是一个基于Chrome Javascript运行时建立的一个平台，用来方便的搭建快速的，易于扩展的网络应用。

Node.js借助事件驱动，非阻塞I/O模型变得轻量和高效，非常适合run across distributed devices的数据密集型实时应用。

Node.js的安装，请参考：

在 Linux 中安装 Nodejs

安装完成之后，可以通过下面的命令来验证一下Node.js是否安装成功。

```
$ node -v v0.10.33
```

Gitbook命令安装

Gitbook是使用NPM来进行安装的，可以在命令行中输入下面的命令进行安装：

```
$ npm install gitbook-cli -g
```

安装完成之后，你可以使用下面的命令来检验是否安装成功

```
$ gitbook -V  
CLI version: 2.3.2  
GitBook version: 3.2.3
```

如果你看到了上面类似的版本信息，则表示你已经安装成功了。

Gitbook命令行使用

Gitbook是一个命令行工具，使用方法：

初始化

初始化一本书

gitbook init 在使用 gitbook init 之后本地会生成两个文件 README.md 和 SUMMARY.md ，这两个文件都是必须的，一个为介绍，一个为目录结构。

编辑电子书

首先，GitBook 使用 SUMMARY.md 文件组织整个内容的目录，比如可以新建 Faq.md 文件，来记录常见问题，并在 SUMMARY.md 文件中添加目录。

Summary

- 简介
- 常见问题

本地预览

当内容书写完毕后，可以在终端中输入如下命令，实现实时预览

gitbook serve gitbook serve ./{book_name} gitbook serve 命令实际会先调用 gitbook build 编译书籍，完成后打开 web 服务器，默认监听本地 4000 端口，在浏览器打开 <http://localhost:4000> 即可浏览电子书。

发布电子书

gitbook build gitbook build ./{book_name} --output=.{outputFolde} gitbook build ./ --log=debug --debug 当电子书内容制作好之后，可以使用如下命令来生成 HTML 静态网页版电子书。该命令会在当前文件夹中生成 _book 文件夹，这个文件夹中的内容就是静态网页版电子书。

使用 --log=debug --debug 可以用来调试，会打印出 stack trace。

查看帮助

```
$ gitbook -h
```

```
Usage: gitbook [options] [command]
```

```
Commands:
```

```
build [options] [source_dir] Build a gitbook from a directory
serve [options] [source_dir] Build then serve a gitbook from a directory
install [options] [source_dir] Install plugins for a book
```

```
pdf [options] [source_dir] Build a gitbook as a PDF
epub [options] [source_dir] Build a gitbook as a ePub book
mobi [options] [source_dir] Build a gitbook as a Mobi book
init [source_dir]      Create files and folders based on contents of SUMMARY.md
publish [source_dir]   Publish content to the associated gitbook.io book
git:remote [source_dir] [book_id] Adds a git remote to a book repository
```

Options:

```
-h, --help      output usage information
-V, --version   output the version number
```

图书项目结构

是Gitbook项目必备的两个文件，也就是一本最简单的gitbook也必须含有这两个文件，它们在一本Gitbook中具有不同的用处。

目录结构

Gitbook 使用简单的目录结构

```
.  
├── book.json  
├── README.md  
├── SUMMARY.md  
├── chapter-1/  
│   ├── README.md  
│   └── something.md  
└── chapter-2/  
    ├── README.md  
    └── something.md
```

简单的介绍下每个文件的作用

File 作用描述

book.json 记录电子书的配置（可选）
README.md 前言 或者 电子书的介绍 必须
SUMMARY.md 电子书的目录 可选，参考Pages
GLOSSARY.md 术语表，可选，参考 Glossary
多语言

当使用多语言时，为每一种语言建立一个子目录，然后需要额外根目录建立一个文件 LANGS.md

Languages

```
* [English](en/)  
* [French](fr/)  
* [Español](es/)
```

README.md 与 SUMMARY编写

使用语法 在Gitbook中所有文字的编写都使用Markdown语法。

README.md 这个文件相对于是一本Gitbook的简介，比如我们这本书的README.md：

Gitbook 使用入门

GitBook 是一个基于 Node.js 的命令行工具，可使用 Github/Git 和 Markdown 来制作精美的电子书。

本书将简单介绍如何安装、编写、生成、发布一本在线图书。SUMMARY.md 这个文件相对于是一本书的目录结构。比如我们这本书的SUMMARY.md：

- [介绍](#)
- [基本安装](#)
 - [Node.js安装](#)
 - [Gitbook安装](#)
 - [Gitbook命令行速览](#)
- [图书项目结构](#)
 - [README.md 与 SUMMARY编写](#)
 - [目录初始化](#)
- [图书输出](#)
 - [输出为静态网站](#)
 - [输出PDF](#)
- [发布](#)
 - [发布到Github Pages](#)
 - [发布到公司文档服务器](#)
- [结束](#) SUMMARY.md基本上是列表加链接的语法。链接中可以使用目录，也可以使用。

图书项目结构

是Gitbook项目必备的两个文件，也就是一本最简单的gitbook也必须含有这两个文件，它们在一本Gitbook中具有不同的用处。

目录结构

Gitbook 使用简单的目录结构

```
.
├── book.json
├── README.md
├── SUMMARY.md
└── chapter-1/
    ├── README.md
    └── something.md
└── chapter-2/
    ├── README.md
    └── something.md
```

简单的介绍下每个文件的作用

File	作用描述
book.json	存放书籍的基本信息
README.md	书籍的简介
SUMMARY.md	书籍的目录结构
chapter-1/	书籍的第一个章节
chapter-2/	书籍的第二个章节

```
book.json    记录电子书的配置 (可选)
README.md    前言 或者 电子书的介绍 必须
SUMMARY.md   电子书的目录 可选, 参考Pages
GLOSSARY.md  术语表 , 可选, 参考 Glossary
多语言
当使用多语言时, 为每一种语言建立一个子目录, 然后需要额外根目录建立一个文件 LANGS.md

# Languages

* [English](en/)
* [French](fr/)
* [Español](es/)
```

目录初始化

图书输出

目前为止，Gitbook 支持如下输出：

静态HTML，可以看作一个静态网站 PDF格式 eBook格式 单个HTML文件 JSON格式 我们这里着重说下如何输出静态的HTML和PDF文件。

输出为静态网站

你有两种方式输出一个静态的网站：

本地预览是自动生成

当你编辑好 gitbook 文档之后，你可以使用 gitbook 的命令来进行本地预览。

```
$ gitbook serve ./book_name
```

gitbook会启动一个4000端口用于预览。

比如，通过gitbook serve来预览本文档：

```
$ gitbook serve gitbook-howtouse
```

Press CTRL+C to quit ...

```
Live reload server started on port: 35729 Starting build ... Successfully built!
```

```
Starting server ... Serving book on http://localhost:4000 你可以你的浏览器中打开这个网址： http://localhost:4000
```

输出为PDF

输出为PDF文件，需要先安装gitbook pdf

\$ npm install gitbook-pdf -g 如果在安装gitbook-pdf时，觉得下载phantomjs包太慢的话，你可以到phantomjs的官方网站上去下载。

<http://phantomjs.org/>

这个包的安装方式，参考其官网的说明文档。

然后，用下面的命令就可以生成PDF文件了。

\$ gitbook pdf 如果，你已经在编写的gitbook当前目录，也可以使用相对路径。

\$ gitbook pdf . 然后，你就会发现，你的目录中多了一个名为book.pdf的文件。

发布

可以使用Github Pages服务将我们写的Gitbook发布到互联网上，前提是已经了解了Git、Github及Github Pages的使用。

发布到 Gitbook.com

Gitbook.com 是一个发布 gitbook 编写书籍的在线网站，提供公开和私有的托管服务，和 GitHub 一样，私有的 Gitbook 是需要付费使用的。但是公开的书籍是无上限的，使用在线的 Gitbook 可以让未接触过 Git 和 Markdown 的作者轻松的创建书籍，并且在线的编辑器可以让作者安心的只关心内容，而不必在意排版，并且支持多人协作编写。

在 Gitbook 上建立账户，新建项目之后，其实也是一个 Git 项目，如果对 Git 熟悉，将本地内容推送到远端即可。

发布到Github Pages

将静态网站直接发布到Github Pages

可以将编写好的.md文件通过Gitbook处理成静态网站，然后发布到Github Pages上。

将md文件与Github Pages静态文件存放在一个仓库中。md文件为master分支，而html文件为 gh-pages分支。

下面将介绍使用一个仓库托管源码，而使用 Travis 自动将静态网站发布到 gh-pages 分支中。这样就可以避免提交源码的同时，还需要同步一遍 gh-pages 分支。

domenic 制作了一个脚本，当 master 分支更新时，自动使用 CI Travis 拉取更新，然后和 gh-pages 分支做比较，如果有差异了，自动将 master 分支的修改提交到 gh-pages 分支。

使用项目的Pages服务

除了上面的直接发布静态文件到Github Pages的方法以外，还可以使用一个单独的项目的Github Pages功能。

创建仓库与分支

登陆到Github，创建一个新的仓库，名称我们就命名为gitbook-tutorial，这样我就得到一个gitbook-tutorial仓库。克隆仓库到本地： git clone git@github.com:/USER_NAME/gitbook-tutorial.git 创建一个新分支： git checkout -b gh-pages，注意，分支名必须为gh-pages。将分支push到仓库： git push -u origin gh-pages。切换到主分支： git checkout master。经过这一步处理，我们已经创建了gh-pages分支了，有了这个分支，Github会自动为你分配一个网址。

<http://USERNAME.github.io/gitbook-tutorial>

结语

Gitbook 初次尝试还是非常不错的，结合 Gitbook.com 网站能够快速托管自己的电子书。不管是用来自己做笔记还是真的用来写小说，或者正式的写作，大概是一个不错的代替方案。

关于 Markdown 的语法规则，可以参考 <http://einverne.github.io/markdown-style-guide/zh.html>

Gitbook 官方文档: <https://toolchain.gitbook.com/>