

Table of Contents

book	1.1
windows_cmd	1.2
external	1.3
Gitbook Tutorial	1.4
Gitbook Automation	1.4.1
Gadget	1.5
Gitbook Automation	1.5.1

Gitbook Online Website link

REF LINK: <https://hmlaiac.github.io/book/>

Abstract

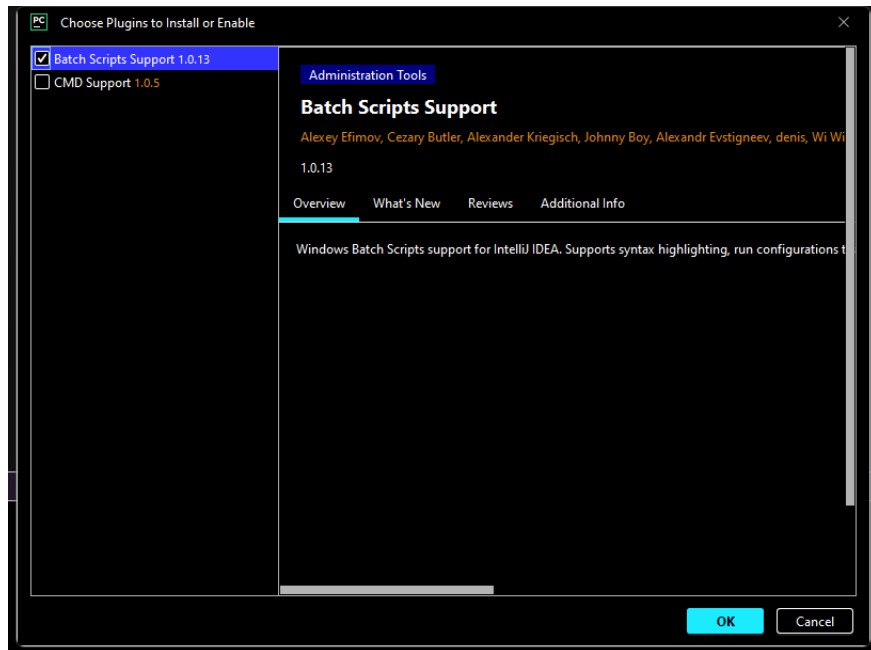
This is a collection of notes on various topics. The notes are written in markdown format, usually called "README.md". There is a github page in this project, which is used to show the content of the notes. Users can search the content with the specific key word. For example, if you want to search the content of "git", you can type "git" in the search. After getting desired result, you can find the project repository according to the name of folder. Then you can start your own development and implementation quickly and efficiently.

Introduction

In my personal experience, I have found that knowledge in my brain are no longer to exist after a period of time. Therefore, I have decided to write down all the knowledge in a new repository named "book". Forgetting some important information is a painful experience, because you need to spend extra time to do researches and studies that you have done before. As a result, a new documentation system is required to collect all notes in my computer. For small project, I will put down the folder under "root" directly. For large project, I will create a new repository to store all the notes. In this way, I can quickly find resources I need in the future.

Install plugins for windows bat/cmd

To help you to manage bat files, JetBrains suggest you to install plugins.



A. Basic setting commands

Change command prompt title

```
title <title>
```

Print out text

```
echo <text>
```

```
C:\Users\Admin\Documents\book\windows_cmd>echo Hello World
Hello World
```

System variables

```
echo %1
echo %2
```

```
PS C:\Users\Admin\Documents\book> windows_cmd/hello.bat peter marry

C:\Users\Admin\Documents\book>title Hello World

C:\Users\Admin\Documents\book>echo Hello World
Hello World

C:\Users\Admin\Documents\book>echo peter
peter

C:\Users\Admin\Documents\book>echo marry
marry
PS C:\Users\Admin\Documents\book>
```

After running command, we observe that command line run in bat/cmd file is linear. It means that it will run one by one.

B. Run with some logics

In this example, we are going to create a new bat file named "logic.bat" to include some logical operations.

Create a variable

```
set /A variable-name=value
```

- variable-name is the name of the variable you want to set.
- value is the value which needs to be set against the variable.
- /A – This switch is used if the value needs to be numeric in nature.

Logic operations

In windows bat/cmd, we can't use symbolic operators like '>', '<' or '==' to build the logic.

- EQU - equal
- NEQ - not equal
- LSS - less than
- LEQ - less or equal
- GTR - greater than
- GEQ - greater or equal

example:

```
set /A a = 1, b = 2
rem run some logics

if %a% equ %b% (echo yes) else (echo no)
if %a% neq %b% (echo yes) else (echo no)
if %a% lss %b% (echo yes) else (echo no)
if %a% leq %b% (echo yes) else (echo no)
if %a% gtr %b% (echo yes) else (echo no)
if %a% geq %b% (echo yes) else (echo no)
```

Other operators

& operator

The & operator is used to execute commands in sequence, as if they were on the same line, separated by spaces. The commands will be executed one by one. You can write down this command in command prompt directly, but it runs successfully in bat/cmd file.

```
echo hello & echo world
```

&& operator

The && operator is used to execute commands in sequence, but the commands behind "&&" will be executed only if the previous command is executed successfully.

```
find "ok\" c:\test.txt && echo SUCCESS
```

|| operator

The || operator is used to execute commands in sequence, but the commands behind "||" will be executed only if the previous command occurs error.

```
find "ok\" c:\test.txt || echo FAIL
```

```
C:\Users\Admin\Documents\book>echo hello & echo world
hello
world
```

> operator

The > operator is used to redirect the output of a command to a file. If the file already exists, it will be overwritten.

In this example, we create a new file named "test.txt" and write down "hello" in it.

```
echo hello > windows_cmd/test.txt
```

Result:

```
1 | hello
2 |
```

>> operator

The >> operator is used to redirect the output of a command to a file. If the file already exists, the new data will be appended to the end of the file.

In this example, we append "world" to the end of "test.txt".

```
echo world >> windows_cmd/test.txt
```

```
1 | hello
2 | world
3 |
```

< operator

The < operator is used to redirect the input of a command to a file.

In this example, we read the content of "test.txt" and print it out.

```
clip < windows_cmd/test.txt
```

6. 管道操作符 |

clip 将内容复制到剪切板

```
1 | clip < 1.txt
```

dir 组合 *clip* 的用法示例

```
1 | dir | clip
```

start 可以打开文件夹, 文件, 网址等

```
1 | start /max 1.txt
2 | start /min 1.txt
```

Problems: Can we automate gitbook build and push to github?

References

[Tencent Docs](#)

Book List

A. Gitbook Installation

Install Gitbook with npm, npm is a package manager for JavaScript, it is installed with Node.js.

A1. Prerequisites

Windows command prompt

You must use the windows cmd to install gitbook, otherwise, you will get unexpected error.

Node.js install version 10.4.1

You need to install nvm for windows to manage the version of Node.js. [nvm for windows](#)

```
nvm install 10.4.1  
nvm use 10.4.1
```

Install gitbook-cli instead of gitbook

You may find this error if install gitbook directly:

```
PS C:\Users\Admin\Documents\book\gitbook> npx gitbook -v  
You need to install "gitbook-cli" to have access to the gitbook command anywhere on your system.  
If you've installed this package globally, you need to uninstall it.  
>> Run "npm uninstall -g gitbook" then "npm install -g gitbook-cli"
```

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

Fetch gitbook with default / specific version

```
1-gitbook fetch {version}
```

Validate your installation

```
1-gitbook -V
```

Finally, you are able see the version of gitbook:

```
PS C:\Users\Admin\Documents\book> gitbook -V  
CLI version: 2.3.2  
GitBook version: 3.2.3
```

B. Create First Gitbook

B1. Create "SUMMARY.md" to include all "README.md" required to be shown in the book

```
# Summary

* [Gitbook Introduction](README.md)
```

B2. Initialize gitbook

```
```shell
gitbook init
```

### B3. Generate static website

```
gitbook serve ./{book_name}
```

### B4. Out the directory of static website to target directory

```
$ gitbook build --output=/tmp/gitbook
```

```
$ ls /tmp/gitbook/
```

### B5. Generate PDF

Install Calibre to generate PDF, [download the application here](#)

```
gitbook pdf {book_name}
```

## C. Publish your book to github with a static website

## C1. Create page folder in your project

In your root directory of your project, create a folder named "docs"

```
mkdir docs
```

Copy and past the static website to the "docs" folder

## C2. Git hub Settings

In your project github repository, go to "Settings" -> "Pages" -> "Build and Deployments" -> "Branch: docs" -> "Save"

In my page you can see my [demo website here](#)

## Reference

Reference: [Gitbook Chinese](#)

# Problem Statement

- How to automate the gitbook initialization, build and deploy?

User Guide:

1. Folder name is case sensitive in this project, because the title of the book and chapter name will be generated from the folder name.
2. Book name: {1-book\_name}
3. Chapter name: {1.1-chapter\_name}

e.g. folder\_name {1-gitbook} --> 1. Gitbook

# Project Workflow

```
flowchart TD
 A[Input your directory] --> B[Input your confirmation]
 B --> C{Is is confirmed?}
 C -->|Yes| D[Initialize gitbook]
 C -->|No| A
 D --> E[Create SUMMARY.md]
 E --> F[Initialize gitbook]
 F --> G[Delete previous auto-generated files]
 G --> H[Generate static website]
 H --> I[Handle file conflicts]
```

# User Guide

## A. Prerequisites

- Windows 11
- Python 3.8.5 or above
- Gitbook (If you haven't installed gitbook, please refer to previous chapter to see the installation guide)
- Cablibre (Same as above)

## B. How to use

- Clone the from book/Gadget.
- Inside the "Gitbook Automation" should include "md\_to\_gitbook.py" and "gitbook\_auto\_package.bat" programs
- The programs helps you to collect "ReadME.md" file in the repository automatically

- Click the "gitbook\_auto\_package.bat" to run the script, make sure your current system satisfies the prerequisites.
- You can run the bat file with administrator permission to prevent some errors

```
C:\Windows\system32\cmd.e
input your directory:|
```

- Input your absolute directory, e.g. C:\Users\Admin\Documents\book Make sure your root folder name of the project should be renamed to **book**

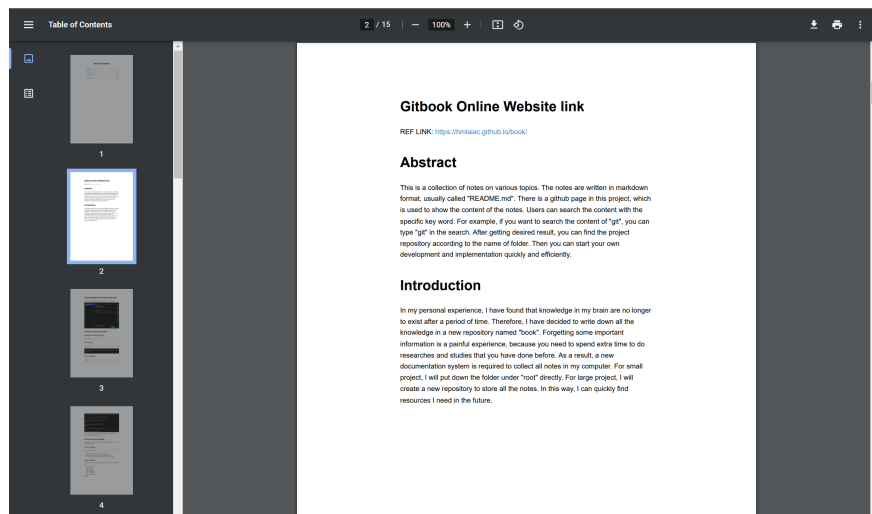
```
C:\Windows\system32\cmd.e
input your directory:C:\Users\Admin\Documents\book
```

- In the command prompt, type "yes" to make sure your input directory is correct. It may cost you a lot of time if generating a large book file.

```
C:\Windows\system32\cmd.e
input your directory:C:\Users\Admin\Documents\book
Please confirm the directory of gitbook
Your current directory is "C:\Users\Admin\Documents\book"
input (yes/no):yes
```

## C. Output Result

### C1. A pdf file named book.pdf



### C2. A static website repository

All files will be stored in the "docs" folder, which is a special folder to help you build the page of github. Follow the instructions from the previous chapter, you can display your book with a static website. If you understand the settings of github page and configure the page directory, commit and push this repository to github. You can find your book in a static website @Thomas smile :)



