

# Hugo

Christian Külker

2024-07-12

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Documentation</b>	<b>2</b>
<b>3</b>	<b>Installation Hugo 0.64.1 Debian 10</b>	<b>3</b>
3.1	Dependencies	3
3.2	Fetch Source Code	3
3.3	Hugo and Go Environment	3
3.4	Build Building Hugo	4
<b>4</b>	<b>{FAILED} Build Building Hugo Extended [Debian 10 Buster]</b>	<b>4</b>
<b>5</b>	<b>Extended Pre-compiled Hugo [Debian 10 Buster]</b>	<b>5</b>
<b>6</b>	<b>Testing docsy Theme</b>	<b>6</b>
<b>7</b>	<b>Testing Docsy Theme Example (Faster)</b>	<b>7</b>
7.1	[theme/docsy] How To Change The List Format In Sections?	7
<b>8</b>	<b>Source Code Latest Version</b>	<b>8</b>
8.1	Get The Source	8
8.2	Building Hugo [Debian Buster]	9
<b>9</b>	<b>Configuration</b>	<b>9</b>
9.1	boiler plate	9
9.2	Themes	9
9.3	Add New Page (Add Content)	10
<b>10</b>	<b>Hugo Front Matter</b>	<b>10</b>
10.1	Hugo Markup	10

<b>11 Usage</b> . . . . .	<b>12</b>
11.1 Server Development Site . . . . .	12
11.2 Publish Content . . . . .	13
11.3 Add Home Page . . . . .	13
<b>12 Simple Deploy</b> . . . . .	<b>13</b>
<b>13 Example Pages</b> . . . . .	<b>13</b>
<b>14 URLs</b> . . . . .	<b>14</b>
<b>15 Using Hugo 0.64.1</b> . . . . .	<b>14</b>
<b>16 Installing Techdoc Theme</b> . . . . .	<b>14</b>
<b>17 Installing A Theme Inside A Git Project (The Content) [NOT TESTED]</b> . . . . .	<b>15</b>
<b>18 Translating A Theme (i18n)</b> . . . . .	<b>15</b>
<b>19 Problems</b> . . . . .	<b>15</b>
<b>20 Output Formats</b> . . . . .	<b>15</b>
<b>21 History</b> . . . . .	<b>16</b>
<b>22 Disclaimer of Warranty</b> . . . . .	<b>16</b>
<b>23 Limitation of Liability</b> . . . . .	<b>16</b>

## 1 Introduction

Written in Go, Hugo is an open source static website generator (SSG) available under the Apache License 2.0. Hugo supports TOML, YAML, and JSON data file types, Markdown and HTML content files, and uses short-codes to add rich content. Other notable features include taxonomies, multi-language mode, image processing, custom output formats, HTML/CSS/JS minification, and support for Sass SCSS workflows.

## 2 Documentation

- <https://xyproto.github.io/splash/docs/all.html>
- <https://xyproto.github.io/splash/docs/longer/all.html>

## 3 Installation Hugo 0.64.1 Debian 10

This installation was tested on Debian 10 Buster. It clones a specific version from the upstream git repository.

### 3.1 Dependencies

- minimal `golang` 1.11 is needed

```
aptitude install golang
```

### 3.2 Fetch Source Code

```
export URL=https://github.com/gohugoio/hugo.git
export VER=0.64.1
mkdir ~/g/github.com/gohugoio/hugo-0.64.1
cd ~/g/github.com/gohugoio/
git clone --branch v$VER --single-branch --depth 1 $URL hugo-$VER
```

This fetches just 72 MB instead of 177 MB for a complete clone. However this clone is in a `detached HEAD` mode ; without tagging, no commits can be made, which should not be a problem.

### 3.3 Hugo and Go Environment

As the self compiled version of hugo, the `PATH` should be set. This is a suggestion for an environment, located at `/srv/env/hugo-0.64.1`.

```
export ENV_HUGO_NS=hugo-0.64.1
export ENV_HUGO_ROOT=$HOME/g/github.com/gohugoio/hugo-0.64.1
export PATH=$ENV_HUGO_ROOT:$PATH
```

If Go is installed via the package manager, no special environment is needed. In this example a self compiled version of Go was used. The following is a suggestion for the Go environment, located at `/srv/env/go-v1.13.7`. Two variables are important: `GOROOT` and `GOPATH`.

```
export ENV_NODE_NS=go-v1.13.7
export GOROOT=$HOME/g/github.com/golang/go1.13.7
export GOPATH=/srv/go
export PATH=$GOPATH/bin:$GOROOT/bin:$PATH
```

### 3.4 Build Building Hugo

The build time of `hugo` depends on the state of download. If everything is downloaded it takes about 2 minutes. So this would **not** work on an isolated machine without network connection.

```
cd hugo-0.64.1
source /srv/env/go-v1.13.7
go version
go version go1.13.7 linux/amd64
go build

./hugo version
Hugo Static Site Generator v0.64.1 linux/amd64 BuildDate: unknown
```

## 4 {FAILED} Build Building Hugo Extended [Debian 10 Buster]

The extended version of hugo is needed for some themes to support TOCSS.

```
cd hugo-0.64.1
source /srv/env/go-v1.13.7
go version
go version go1.13.7 linux/amd64
go build --tags extended
# github.com/wellington/go-libsass/libs
/tmp/pkg/mod/github.com/wellington/go-libsass@v0.9.3-0.20181113175235-
↳ c6364420\
6701/libs/sass_number.go:8:9: undefined: SassNumber
/tmp/pkg/mod/github.com/wellington/go-libsass@v0.9.3-0.20181113175235-
↳ c6364420\
6701/libs/sass_number.go:151:10: undefined: SassNumber
/tmp/pkg/mod/github.com/wellington/go-libsass@v0.9.3-0.20181113175235-
↳ c6364420\
6701/libs/sass_number.go:153:9: undefined: SassNumber
/tmp/pkg/mod/github.com/wellington/go-libsass@v0.9.3-0.20181113175235-
↳ c6364420\
6701/libs/sass_number.go:157:10: undefined: SassNumber
/tmp/pkg/mod/github.com/wellington/go-libsass@v0.9.3-0.20181113175235-
↳ c6364420\
6701/libs/sass_number.go:159:9: undefined: SassNumber
/tmp/pkg/mod/github.com/wellington/go-libsass@v0.9.3-0.20181113175235-
↳ c6364420\
```

```
6701/libs/sass_number.go:164:10: undefined: SassNumber
/tmp/pkg/mod/github.com/wellington/go-libsass@v0.9.3-0.20181113175235-
↳ c6364420\
6701/libs/sass_number.go:166:9: undefined: SassNumber
/tmp/pkg/mod/github.com/wellington/go-libsass@v0.9.3-0.20181113175235-
↳ c6364420\
6701/libs/sass_number.go:170:10: undefined: SassNumber
/tmp/pkg/mod/github.com/wellington/go-libsass@v0.9.3-0.20181113175235-
↳ c6364420\
6701/libs/sass_number.go:175:28: undefined: SassNumber
/tmp/pkg/mod/github.com/wellington/go-libsass@v0.9.3-0.20181113175235-
↳ c6364420\
6701/libs/sass_number.go:186:24: undefined: SassNumber
/tmp/pkg/mod/github.com/wellington/go-libsass@v0.9.3-0.20181113175235-
↳ c6364420\
6701/libs/sass_number.go:166:9: too many errors
```

Build downloads 480MB and takes 1 minute for compile time

## 5 Extended Pre-compiled Hugo [Debian 10 Buster]

```
cd /srv/src
wget
↳ https://github.com/gohugoio/hugo/releases/download/v0.64.1/hugo_extended\
_0.64.1_Linux-64bit.tar.gz
cd /srv/build
mkdir hugo_extended_0.64.1
cd hugo_extended_0.64.1
tar xvzf ../../src/hugo_extended_0.64.1_Linux-64bit.tar.gz

source /srv/env/go-v1.13.7
source /srv/env/hugo-extended-0.64.1
hugo version
Hugo Static Site Generator v0.64.1-C327E75D/extended linux/amd64
↳ BuildDate:\
2020-02-09T20:51:43Z
```

## 6 Testing docsy Theme

```
cd ~/scratch/hugo-0.64.1-new-site-docsy/themes

git clone --recursive

OR (after cloning)
cd docsy
git submodule update --init

OR (after cloning, and expecting nested stuff)
cd docsy
git submodule update --init --recursive

Submodule 'assets/vendor/Font-Awesome'
↳ (https://github.com/FortAwesome/Font-Aw\
esome.git) registered for path 'assets/vendor/Font-Awesome'
Submodule 'assets/vendor/bootstrap'
↳ (https://github.com/twbs/bootstrap.git) re\
gistered for path 'assets/vendor/bootstrap'
Cloning into
↳ '$HOME/scratch/hugo-0.64.1-new-site-docsy/themes/docsy/assets/v\
endor/Font-Awesome'...
Cloning into
↳ '$HOME/scratch/hugo-0.64.1-new-site-docsy/themes/docsy/assets/v\
endor/bootstrap'...
Submodule path 'assets/vendor/Font-Awesome': checked out
↳ '538a5bc4cf7989821d05\
2e3ff919e2bead07021c'
Submodule path 'assets/vendor/bootstrap': checked out
↳ '3b558734382ce58b51e5fc6\
76453bfd53bba9201'

source /srv/env/node-v12.5.0-linux-x64
# -D, --save-dev: Package will appear in your devDependencies.
npm install -D --save autoprefixer
...
+ autoprefixer@9.7.4
npm install -D --save postcss-cli
...
+ postcss-cli@7.1.0
```

If you omit npm installations, you will encounter this error.

```
cd ~/scratch/hugo-0.64.1-new-site-docsy
hugo
Building sites ... ERROR 2020/02/14 23:40:20 Transformation failed: SCSS
↳ process\
ing failed: file "stdin", line 6, col 1: File to import not found or
↳ unreadabl\
e: ../vendor/bootstrap/scss/bootstrap.
```

```
1 Building sites ...
2 Replace Autoprefixer browsers option to Browserslist config.
3 Use browserslist key in package.json or .browserslistrc file.
4
5 Using browsers option can cause errors. Browserslist config
6 can be used for Babel, Autoprefixer, postcss-normalize and other tools.
7
8 If you really need to use option, rename it to overrideBrowserslist.
9
10 Learn more at:
11 https://github.com/browserslist/browserslist#readme
12 https://twitter.com/browserslist
```

## 7 Testing Docsy Theme Example (Faster)

```
export URL=https://github.com/google/docsy-example.git
export DIR=hugo-0.64.1-new-site-docsy-example
git clone --recurse-submodules $URL $DIR
cd $DIR
du -sch *
66M total
source /srv/env/go-v1.13.7
source /srv/env/hugo-extended-0.64.1
source /srv/env/node-v12.5.0-linux-x64
npm install -D --save autoprefixer
npm install -D --save postcss-cli
```

### 7.1 [theme/docsy] How To Change The List Format In Sections?

Sections are Directories with a `_index.md` file that (if chosen) list the sub pages of that directory.

```
1 dev
2   _index.md
3   python.md
4   perl.md
```

Below the content of `_index.html` a list is shown. The format of list depends on the front matter:

```
1 ---
2   simple_list: true
3 ---
```

For `simple_list: true` a simple list (`ul, li`) is shown.

```
<ul>
  {{ range $pages }}
    {{ if eq .Parent $parent }}
      <li><a href="{{ .RelPermalink }}">{{- .Title -}}</a></li>
    {{ end }}
  {{ end }}
</ul>
```

Where as for `simple_list: false` a h5 description list is shown (that takes up more space). The `theme/docsy` template that do that is: `layouts/partials/section-index.html`.

```
<div class="entry">
  <h5>
    <a href="{{ .RelPermalink }}">{{- .Title -}}</a>
  </h5>
  <p>{{ .Description | markdownify }}</p>
</div>
```

## 8 Source Code Latest Version

### 8.1 Get The Source

```
cd ~/g/github.com/
mkdir gohugoio
cd gohugoio
git clone https://github.com/gohugoio/hugo.git
```



---

## 8.2 Building Hugo [Debian Buster]

takes some time, mega download, puts 884MB into ~/go !!!

=> <https://gohugo.io/getting-started/installing>

```
cd hugo
go build

./hugo version
Hugo Static Site Generator v0.61.0-DEV linux/amd64 BuildDate: unknown
```

## 9 Configuration

- [doc](#)

### 9.1 boiler plate

```
$HOME/g/github.com/gohugoio/hugo/hugo new site /tmp/new-site

Congratulations! Your new Hugo site is created in /tmp/new-site.

Just a few more steps and you're ready to go:

1. Download a theme into the same-named folder.
   Choose a theme from https://themes.gohugo.io/ or
   create your own with the "hugo new theme <THEMENAME>" command.
2. Perhaps you want to add some content. You can add single files
   with "hugo new <SECTIONNAME>/<FILENAME>.<FORMAT>".
3. Start the built-in live server via "hugo server".

Visit https://gohugo.io/ for quickstart guide and full documentation.
```

### 9.2 Themes

- [github](#)

```
cd new-site/theme
git clone https://github.com/thingsym/hugo-theme-techdoc.git
```

```
cd new-site
echo 'theme = "hugo-theme-techdoc"' >> config.toml
```

Consider:

As a test installation the `Mainroad` theme is used. Seems well supported by 26 contributors and has a lot of features and supports many browsers

- [github](#)
- [Example](#)

```
git clone https://github.com/vimux/mainroad git submodule add https://github.com/vimux/mainroad
```

Consider (beautiful)

- [github](#)
- [demo](#)

### 9.3 Add New Page (Add Content)

```
$HOME/g/github.com/gohugoio/hugo/hugo new cms/hugo.md
/tmp/new-site/content/cms/hugo.md created
vim /tmp/new-site/content/cms/hugo.md
```

## 10 Hugo Front Matter

See <https://gohugo.io/content-management/front-matter/>

### 10.1 Hugo Markup

- <https://www.ii.com/hugo-markup-languages/>
- <https://freecontent.manning.com/using-markup-languages-with-hugo/>
- <https://gohugo.io/getting-started/configuration-markup/>

#### 10.1.1 Frontmatter:

```
1 ---
2 markup: pandoc
3 ---
```

### 10.1.2 config.toml

```
[markup]
  defaultMarkdownHandler = "goldmark"
[markup.blackFriday]
  angledQuotes = false
  footnoteAnchorPrefix = ""
  footnoteReturnLinkContents = ""
  fractions = true
  hrefTargetBlank = false
  latexDashes = true
  nofollowLinks = false
  noreferrerLinks = false
  plainIDAnchors = true
  skipHTML = false
  smartDashes = true
  smartypants = true
  smartypantsQuotesNBSP = false
  taskLists = true
[markup.goldmark]
  [markup.goldmark.extensions]
    definitionList = true
    footnote = true
    linkify = true
    strikethrough = true
    table = true
    taskList = true
    typographer = true
  [markup.goldmark.parser]
    attribute = true
    autoHeadingID = true
    autoHeadingIDType = "github"
  [markup.goldmark.renderer]
    hardWraps = false
    unsafe = false
    xHTML = false
[markup.highlight]
  codeFences = true
  guessSyntax = false
  hl_lines = ""
  lineNoStart = 1
```

```
lineNos = false
lineNumbersInTable = true
noClasses = true
style = "monokai"
tabWidth = 4
[markup.tableOfContents]
endLevel = 3
ordered = false
startLevel = 2
```

## 11 Usage

### 11.1 Server Development Site

```
cd $HOME/scratch/hugo-new-site
$HOME/g/github.com/gohugoio/hugo/hugo server

Building sites ... WARN 2019/12/02 12:15:53 Page.Hugo is deprecated and will
↳ be
removed in a future release. Use the global hugo function.

WARN 2019/12/02 12:15:53 Page.URL is deprecated and will be removed in a
↳ future
release. Use .Permalink or .RelPermalink. If what you want is the front
↳ matter
URL value, use .Params.url

                | EN
+-----+-----+
Pages           | 7
Paginator pages | 0
Non-page files  | 0
Static files    | 5
Processed images | 0
Aliases         | 0
Sitemaps        | 1
Cleaned         | 0

Built in 31 ms
Watching for changes in
/tmp/new-site/{archetypes,content,data,layouts,static,themes}
```

```
Watching for config changes in /tmp/new-site/config.toml
Environment: "development"
Serving pages from memory
Running in Fast Render Mode. For full rebuilds on change: hugo server
--disableFastRender
Web Server is available at http://localhost:1313/ (bind address 127.0.0.1)
Press Ctrl+C to stop
```

New pages are drafts and are NOT displayed. To actually display them use `-D` option or `--buildDrafts` option!

```
$HOME/g/github.com/gohugoio/hugo/hugo server --enableGitInfo
↳ --i18n-warnings\
--path-warnings --templateMetrics --verbose --debug -D
```

## 11.2 Publish Content

```
$HOME/g/github.com/gohugoio/hugo/hugo
```

Will write data to `public`

## 11.3 Add Home Page

```
cd /tmp/new-site/content
vim _index.md
```

## 12 Simple Deploy

```
cd /tmp/new-site/
$HOME/g/github.com/gohugoio/hugo/hugo server
```

## 13 Example Pages

- [hichdev.com](https://hichdev.com) hugo + Material Theme

## 14 URLs

- [home](#)
- [github](#)

## 15 Using Hugo 0.64.1

```
source /srv/env/hugo-0.64.1
hugo version
Hugo Static Site Generator v0.64.1 linux/amd64 BuildDate: unknown
```

```
hugo new site $HOME/scratch/hugo-0.64.1-new-site
Congratulations! Your new Hugo site is created in
↳ $HOME/scratch/hugo-0.64.1-new-site.
```

Just a few more steps and you're ready to go:

1. Download a theme into the same-named folder.  
Choose a theme from <https://themes.gohugo.io/> or create your own with the "hugo new theme <THEMENAME>" command.
2. Perhaps you want to add some content. You can add single files with "hugo new <SECTIONNAME>/<FILENAME>.<FORMAT>".
3. Start the built-in live server via "hugo server".

Visit <https://gohugo.io/> for quickstart guide and full documentation.

## 16 Installing Techdoc Theme

```
mkdir -p ~/g/github.com/thingsym
cd ~/g/github.com/thingsym
git clone https://github.com/thingsym/hugo-theme-techdoc.git
cp -a ~/g/github.com/thingsym/hugo-theme-techdoc
↳ $HOME/scratch/hugo-0.64.1-new-site/themes
```

## 17 Installing A Theme Inside A Git Project (The Content) [NOT TESTED]

```
cd ~/scratch/hugo-new-site
git submodule add https://github.com/alex-shpak/hugo-book themes/book
```

## 18 Translating A Theme (i18n)

```
cd hugo-site
mkdir i18n
echo "[imprint]" >> i18n/en.toml
echo 'other = "Imprint"' >> i18n/en.toml
echo "[imprint]" >> i18n/de.toml
echo 'other = "Impressum"' >> i18n/de.toml
```

Then use it as `{{ T "imprint" }}`

## 19 Problems

Your rendered home page is blank: /index.html is zero-length

- Did you specify a theme on the command-line or in your “config.toml” file? (Current theme: ””)
- For more debugging information, run “hugo -v”

## 20 Output Formats

A

```
{{ range .AlternativeOutputFormats -}}
<link rel="{{ .Rel }}" type="{{ .MediaType.Type }}" href="{{ .Permalink |
  ↪ safeURL }}">
{{ end -}}
```

B

```
{{ .RelPermalink }} > /that-page/
{{ with .OutputFormats.Get "json" -}}
```

```
{{ .RelPermalink }} > /that-page/index.json  
{{- end }}
```

## 21 History

---

Version	Date	Notes
0.1.3	2024-07-12	Add How To Change The List Format In Sections
0.1.2	2024-07-11	Fix frontmatter (remove double 'tags' key), Format
0.1.1	2023-03-24	Improve writing, history
0.1.0	2020-02-12	Initial release

---

## 22 Disclaimer of Warranty

THERE IS NO WARRANTY FOR THIS INFORMATION, DOCUMENTS AND PROGRAMS, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE INFORMATION, DOCUMENT OR THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE INFORMATION, DOCUMENTS AND PROGRAMS IS WITH YOU. SHOULD THE INFORMATION, DOCUMENTS OR PROGRAMS PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

## 23 Limitation of Liability

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE INFORMATION, DOCUMENTS OR PROGRAMS AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE INFORMATION, DOCUMENTS OR PROGRAMS (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE INFORMATION, DOCUMENTS OR PROGRAMS TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.