

Statische Webseiten mit GoHugo

Treffen FSFE Gruppe Bonn
22.2.2015



Carsten Schönert

Was ist Hugo?

- Ein multiplattform fähiger, extrem flexibler HTML Generator
- Geschrieben in GoLang (Go) 2013 von [Steve Francia](#)
- Vermutlich aktuell der schnellste Generator
- Konfiguration und Steuerung per YAML, JSON und TOML
- Live Vorschau und Editiermodus
- Mächtiger Go Template Mechanismus
- Ideal für Blogs und statische Webseiten
- FOSS! Unterstützt Aliase, Kategorien und Permalink, ca. 50 Themes vorhanden

Installation

- Extrem einfache Installation da nur ein Binary
- Download des Plattform spezifischen Binarys oder Pakets unter: <https://github.com/spf13/hugo/releases>
- Hugo ist in [Debian Testing](#)! Backport?
- Binary muss im Pfad liegen (eher spezifisch Windows)
- Eventuell zusätzliche Tools nötig wie Git, asciidoc, etc., abhängig von der Art der gewünschten Benutzung

Vorteile/Nachteile

- Vorteile
 - Einfaches Binary
 - Plattformunabhängig
 - Klare Trennung von Content und Design
 - Extrem schnell, Liveeditiermodus
 - Zahlreiche Themes vorhanden
- Nachteile
 - Einfaches statisch gelinktes Binary (Plugins!)
 - Kenntnisse Go Template Mechanismus

Erste Benutzung

- Zentrale Steuerung über ein Kommando plus Optionen

Usage:

```
hugo [flags]
hugo [command]
```

Available Commands:

```
server      A high performance webserver
version     Print the version number of Hugo
config      Print the site configuration
check       Check content in the source directory
benchmark   Benchmark hugo by building a site a number of times.
convert     Convert your content to different formats
new         Create new content for your site
list        Listing out various types of content
undraft     Undraft changes the content's draft status from 'True' to 'False'
import      Import your site from others.
gen         A collection of several useful generators.
```

Ein neues leeres Projekt

- Neues Hugo Projekt anlegen

```
hugo new site ~/website/projecthugo/
```

```
cd ~/website/projecthugo/
```

- Was erstellt Hugo und wozu?

•		
—	archetypes	< Templates zur Erstellung von Dateien in content/
—	config.toml	< zentrale Konfigdatei
—	content	< die einzelnen Teilwebseiten in Markdown, AsciiDoc, etc.
—	data	< <i>optionale weitere Daten zum Erstellen</i>
—	layouts	< nötige Templates zur HTML Erstellung
—	static	< benötigter statischer Content, z.B. CSS, JS, Pics
—	themes	< Themes soweit nötig

- Neue Seite im Projekt anlegen

```
hugo new about.md
```

```
.../content/about.md created
```

Content bearbeiten

- Hugo erstellt basierend aus Vorgabetemplates (sofern vorhanden) leere Contentdateien

```
$ cat content/about.md
+++
date = "2016-02-08T10:09:31+01:00"
draft = true
title = "about"

+++
```

- Zahlreiche weitere Variablen möglich
→ <http://gohugo.io/overview/configuration/>

Content bearbeiten

- Content im Markdown Syntax hinzufügen

```
$ editor content/about.md
+++
date = "2016-02-08T10:09:31+01:00"
draft = true
title = "About"
url = „/about“
description = „Über die FSFE Gruppe Bonn“

+++

## Wer sind wir?
Wir sind eigentlich eine lustige Gruppe!

Lorem ipsum ...
```

- Bei Bedarf weitere Seiten hinzufügen

Was benötigen wir zusätzlich?

- Hugo bietet zahlreiche Themes → <http://themes.gohugo.io/>
- Sind primär für Blogposts erstellt, können aber einfach modifiziert werden
- Tieferes Wissen über den Go Template Mechanismus nötig
- Verständnis wie Hugo intern arbeitet
- Zum Starten ohne Themes reichen wenige Templates

_default und andere Templates

- Hugo benötigt Templates zum Erstellen des HTML Codes
- Die default (allgemein gültigen) Templates werden in `layout/_default/` abgelegt
- Typischer Weise benötigt man Templates für die einzelnen Seiten (*single.html*) und Listen (*list.html*), meistens noch für Zusammenfassungen (*summary.html*) und die Auflistungen (*li.html*)
- Partielle (`partial` genannt) Templates für Header und Footer
- Ein Template für `index.html` und `404.html` usw.
- **Templates sind der Schlüssel für die Webseiten!**

index.html Template

- Template wird unter `layout/index.html` abgelegt

```
$ editor layout/index.html
{{ partial "header.html" . }}
<h1>What's up</h1>
```

```
<p>Was gibt es neues? <a href="/">Neuigkeiten</a>.</p>
<p>Aktuelle Blog Posts: <a href="/">Hier geht's zum
Blog</a>.</p>
```

```
<p>Und was sonst noch hier stehen sollte.</p>
{{ partial "footer.html" . }}
```

partial Templates (Header)

- partial Template `layout/partial/header.html`

```
$ editor layout/partial/header.html
<!DOCTYPE html>
<html>
<head>

<meta name='viewport' content='width=device-width, initial-scale=1.0'>
<meta name="description" content="FSFE Gruppe Bonn">
<link rel="stylesheet" type="text/css" href="/css/mini.css">

<title>{{ .Title }} - fsfe-bonn.org</title>

</head>

<body>
<a href="/" class="h1logo"><h1>FSFE Gruppe Bonn</h1></a>
<p><a href="/">Home</a> | <a href="/post">Blog</a> | <a
href="/about">About</a> | <a href="/contact">Contact</a></p>
```

partial Templates (Footer)

- partial Template

```
layout/partial/footer.html
```

```
$ editor layout/partial/footer.html
```

```
<br />
```

```
<p>©;2016 FSFE Gruppe Bonn | Erstellt durch  
<a href="http://gohugo.io">Hugo</a></p>
```

```
</body>
```

```
</html>
```

_default single.html Template

- single.html Template

```
layout/_default/single.html
```

```
$ editor layout/_default/single.html  
{{ partial "header.html" . }}
```

```
<h1>{{ .Title }}</h1>
```

```
{{ .Content }}
```

```
{{ partial "footer.html" . }}
```

_default li{st}.html Template

- list.html Template

layout/_default/list.html

```
$ editor layout/_default/list.html
{{ partial "header.html" . }}
<h1>{{ .Title }}</h1>
<ul id="list">
    {{ range .Data.Pages }}
        {{ .Render "li" }}
    {{ end }}
</ul>
{{ partial "footer.html" . }}
```

- li.html Template layout/_default/li.html

```
$ editor layout/_default/li.html
<li>
  {{ .Date.Format "2006-01-02" }}
  | <a href="{{ .Permalink }}">{{ .Title }}</a>
</li>
```

Webseite testen

- statische Daten wie CSS und Bilder müssen noch unter `static/` abgelegt werden
- Hugo bietet einen Live Editiermodus und einen eigenen Webserver
- Veränderungen werden erkannt, geparst und Content neu erstellt
- Pitfall! Erstellte neue `*.md` Dateien enthalten „`draft = true`“, Hugo muss explizit mitgeteilt werden auch Draft Seiten zu erstellen

Webseite testen

- Webservice von Hugo starten

```
$ hugo server -v -w [--buildDrafts]
INFO: 2016/02/08 11:16:59 hugo.go:385: Using config file:
../minimalhugosite/config.toml
WARN: 2016/02/08 11:16:59 hugo.go:451: Unable to find Theme Static
Directory:
INFO: 2016/02/08 11:16:59 hugo.go:501: syncing static files to /
INFO: 2016/02/08 11:16:59 site.go:1274: found taxonomies:
map[string]string{"tag":"tags", "category":"categories"}
WARN: 2016/02/08 11:16:59 site.go:1994: Unable to locate layout for 404
page: [404.html]
WARN: 2016/02/08 11:16:59 site.go:1970: "404.html" is rendered empty
0 draft content
0 future content
5 pages created
0 non-page files copied
0 paginator pages created
0 tags created
0 categories created
in 22 ms <<<<<
Watching for changes in ../minimalhugosite/{data,content,layouts,static}
Serving pages from memory
Web Server is available at http://localhost:1313/ (bind address 127.0.0.1)
Press Ctrl+C to stop
```

Content generieren

- Hugo erstellt den Content im Verzeichnis `public/`
- Generierung durch Aufruf ohne weitere Parameter

```
$ hugo
0 draft content
0 future content
5 pages created <<<<
0 non-page files copied
0 paginator pages created
0 tags created
0 categories created
in 23 ms
$ ls public/
404.html  about  contact  css  images  index.html
index.xml  post  robots.txt  sitemap.xml
```

Fortführende Informationen

- Hugo Webseite <http://gohugo.io/overview/introduction/>
- <http://zackofalltrades.com/notes/2014/05/hugo-from-scratch/>
- <https://npf.io/2014/08/hugo-is-awesome/>
- <http://www.mariosanchez.org/post/minimalhugo/>
- <http://blog.srackham.com/posts/porting-my-blog-with-hugo/>
- Vergleich von Jekyll, Middelman, Roots und Hugo
- Go Template Mechanismus: GoLang GoHugo
- Source KiCad Webseite: <https://github.com/KiCad/kicad-website>
- Bootstrap CSS: <http://startbootstrap.com/>
- Hugo Themes <https://github.com/spf13/hugoThemes>

Eigene Seite FSFE Bonn?

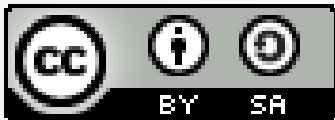
- Ist sicherlich machbar!
- Ist es die Mühe wert?
- Wer kennt sich ausreichend gut mit CSS und Stylesheets aus?
- Wer hat Zeit bzw. kann mitarbeiten?
- Welcher Content?

That's it!

Danke für die Aufmerksamkeit!

Carsten Schönert Februar 2016

c.schoenert [at] t-online [dot] de



This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.