

STATIC WEBSITES

AM BEISPIEL HUGO

# AGENDA

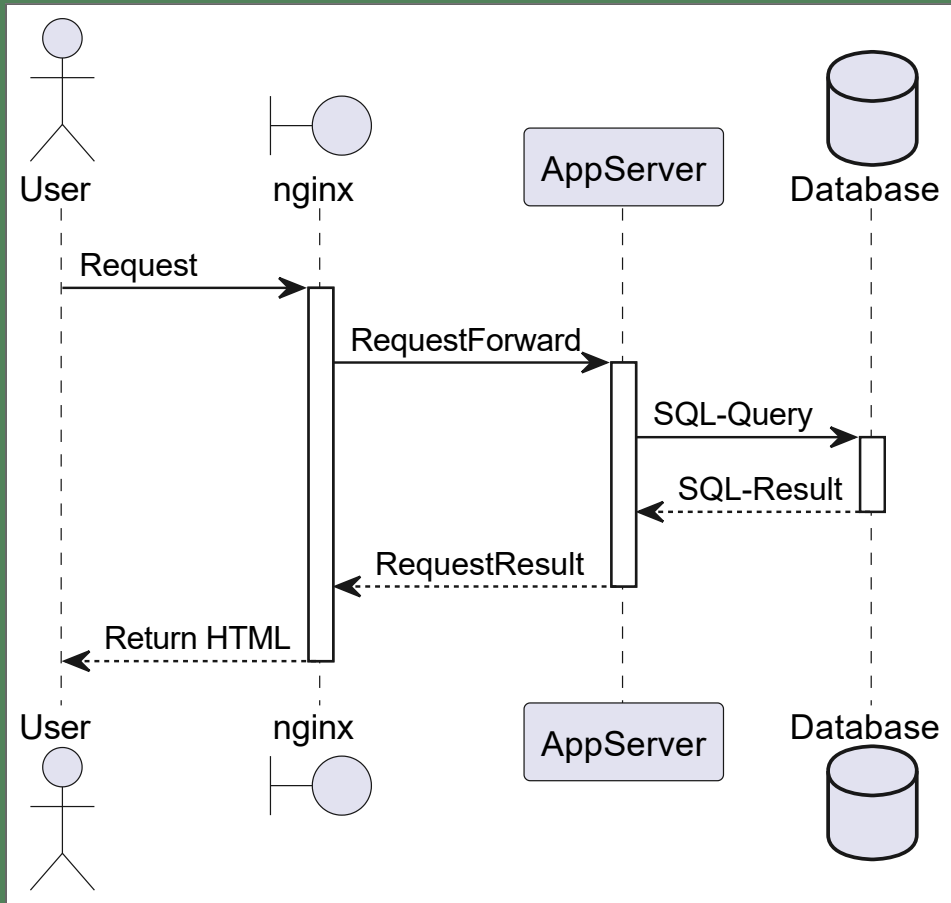
1. Motivation
2. Technische Daten
3. Use Case Szenario
4. Featureübersicht

# WOZU NOCH STATISCHE WEBSITES?

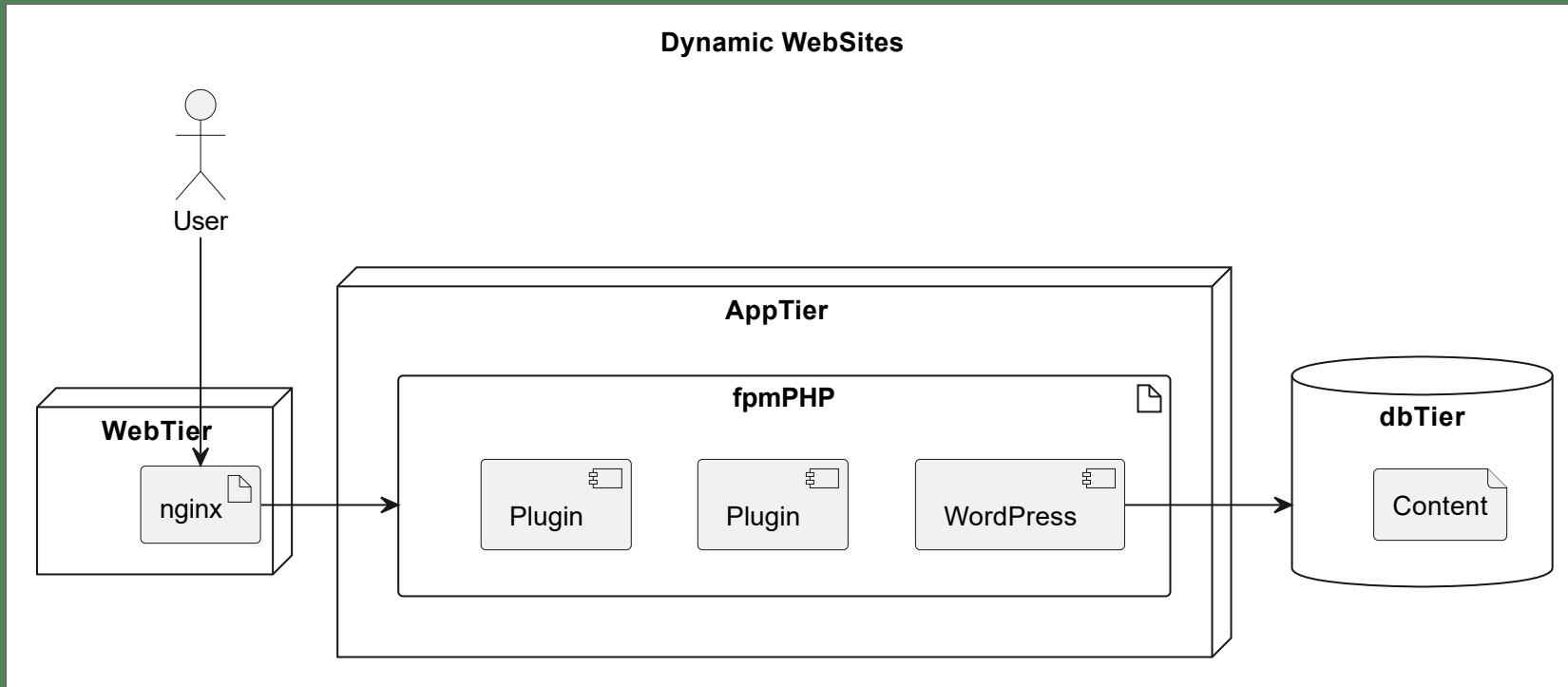
- Content Management Systeme (CMS) weit verbreitet
- Kostengünstig zu mieten
- Quasi Standard?

# DEFINITION

## STATISCH VS DYNAMISCH

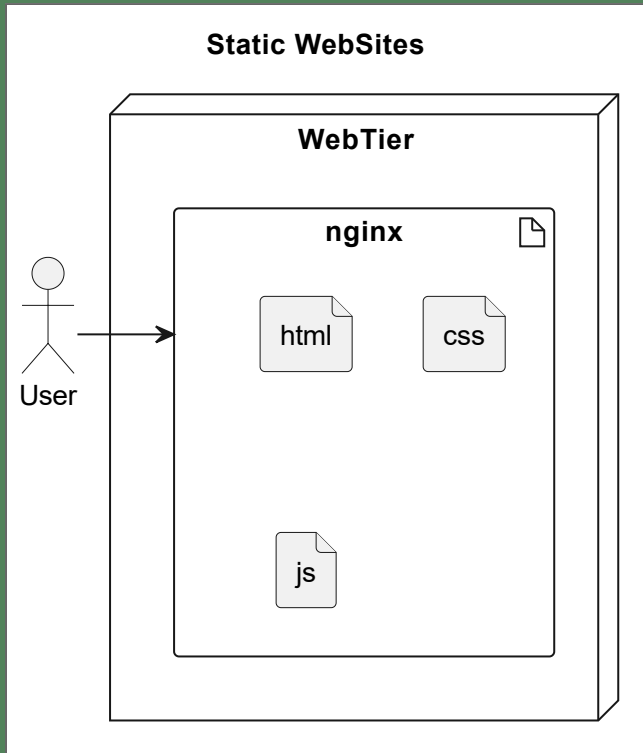


# DYNAMISCH



Die Antwort wird je Anfrage dynamisch generiert

# STATISCH



Die Antwort ist bereits generiert und wird nur noch ausgeliefert.

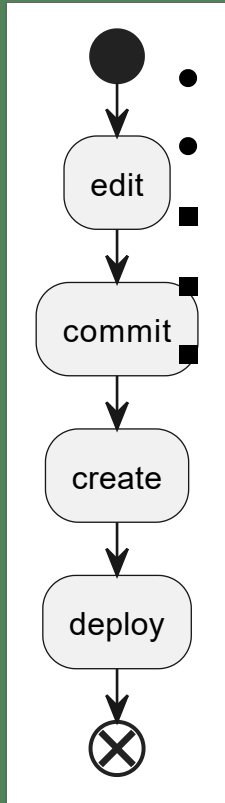
## VOR- & NACHTEILE

- Geschwindigkeit
- Fehleranfälligkeit
- Sicherheit
- Vendor-Lockin
- Nachhaltigkeit

# STATIC SITE GENERATORS

Stichworte sind

- SSGs
- JAMstack für
- Javascript
- API
- Markup





SSGS

AUSWAHL

1. Gatsby
2. Next.js
3. Hugo
4. Jekyll
5. Pelican

# AUSWAHLKRITERIEN

- Programmiersprache
- Zeitbudget zum Starten
- Persönliche Vorlieben

# HUGO

## ZAHLEN, DATEN, FAKTEN



- In Go geschrieben
- OpenSource, Apache Licence 2.0
- Entwicklung seit dem 5. Juli 2013
- Aktuelles Release 0.123.8 (7.3.2024)

## AUTOREN

## PROJEKTGRÜNDER

- Steve Francia, Gründe von Hugo
- Beteiligt bei diversen Projekten
  - Go
  - Docker
  - MongoDB



© <https://spf13.com/about/>

AUTOREN

LEAD DEVELOPER

Bjørn Erik Pedersen

Lead Developer seit Release v0.14, (~ Mai 2015)

© <https://bep.is/en/>,

<https://nn.wikipedia.org/wiki/Brukar:Bep>



# SZENARIO

## ALS BEISPIEL

- Vereinshomepage
- Information für Mitglieder und Interessierte
- Einfach zu verwalten
- Wenig Pflegeaufwand

## BEISPIELE

- **Cumulus Linux Docs**
- **<https://1password.com/>**
- **<https://docs.airship.com/>**





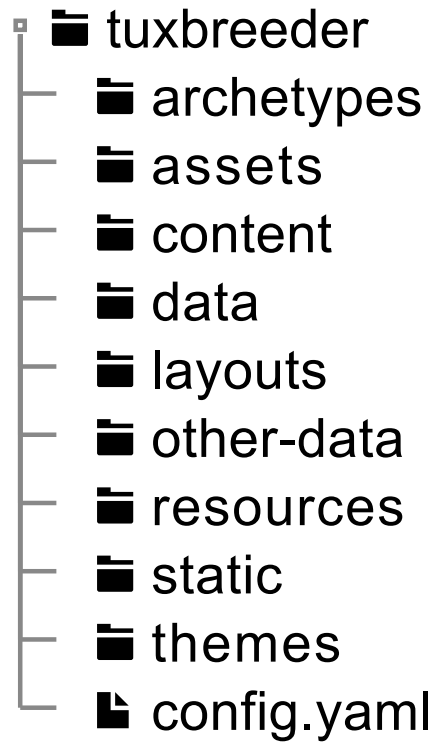
# ALLER ANFANG

1. Download des aktuellen Releases (am Besten extended Version)
2. Auspacken und loslegen

```
1 cd hugo  
2 ./hugo new site tuxbreeder
```

```
3 ./hugo new site tuxbreeder --format yaml
```

# VERZEICHNISSTRUKTUR



# GRUNDKONFIGURATION

Datei config.yaml

```
1 baseUrl: https://tuxbreeder.example/  
2 languageCode: de-de  
3 title: Tux Breeder Deutschland  
4 enableEmoji: true
```

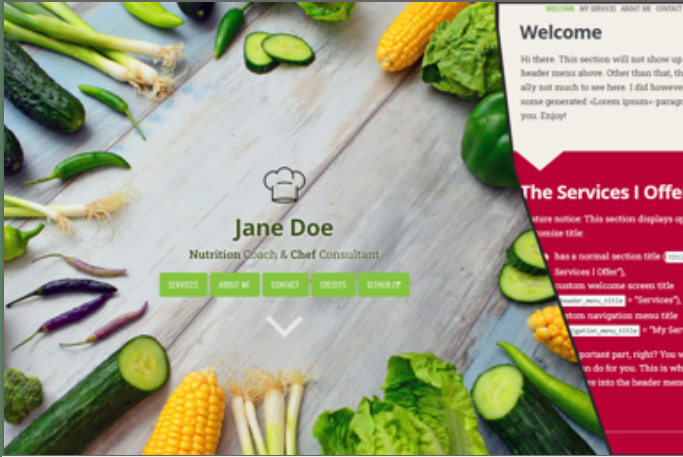
# THEMES

- Vordefiniertes Layout
- Austausch- und wiederverwendbar
- Konfigurierbar via config.yaml
- Verzeichnis im Internet: **<https://themes.gohugo.io/>**

# THEMES

## BEISPIELE





# Themes

# MENÜS

- Automatisch mittels sections

```
5  sectionPagesMenu: main
```

oder manuell:

```
25  menus:
26    main:
27      - name: Home
28        pageRef: /
29        weight: 10
30      - name: Aktuelles
31        pageRef: /blog
32        weight: 20
33      - name: Veranstaltungen
34        pageRef: /calendar
35        weight: 30
36      - name: Vorstand
37        pageRef: /management
38        weight: 40
```

# CONTENTORGANISATION

- Organisation in Sections und Leafs
- Baumstruktur unterhalb von content
- Markdown Dateien zur Beschreibung



# CONTENT

## MARKDOWN BEISPIEL

```
1      ---
2      draft: true
3      title: Neue Website online
4      params:
5          author: Max Mustermann
6      ---
7
8      # Überschrift
9
10     Hallo Liebe Leser,
11
12     Unsere neue Website ...
```

# CONTENT

## SECTIONS

- Verzeichnis ist eine Section wenn es,
  - weitere subsections enthält
  - \_index.md Datei
  - keine index.md Datei
- Sections erhalten ein URL Segment zB /calendar/january/
- Logische Vorgänger und Nachfolger
- Hugo verwendet zur Darstellung Auflistungen "list pages"
- Neuer Name: (page) branch bundle

# CONTENT

## NON-SECTION (LEAF)

- Ebenfalls ein Verzeichnis, aber ohne \_index.md
  - enthält index.md
  - keine logischen Nachfolger
  - Hugo verwendet zur Darstellung eine Seitendarstellung "single page"
- Neuer Name: (page) leaf bundle

# CONTENTTREE



# VORLAGEN

Verzeichnis: archetypes

- Vorlagen mit Funktionen zum automatischen Vorfürllen
- Vorlage wird Anhand des Sektionsnamen gewählt
- Sektionsname "blog" -> blog.md

```
1 hugo new content blog/my-next-post.md
```

# BEISPIEL

## VORLAGENDATEI

```
1      ---
2      date: '{{ .Date }}'
3      draft: true
4      title: '{{ replace .File.ContentBaseName `-` ` ` | title }}'
5      author:
6      ---
7
8      # Blogeintrag vom '{{ time.Now | time.Format "Mon 2.Jan 2006" }}'
9
10     Hier kommt der Eintrag hin.
```

# AUSPROBIEREN

## TESTSERVER

```
1 hugo server
2 ...
3 Running in Fast Render Mode. For full rebuilds on change:
4   hugo server --disableFastRender
5 Web Server is available at http://localhost:1313/ (bind address 127.0.0.1)
6 Press Ctrl+C to stop
```

# LAYOUT

Verzeichnis: layouts

- Abbild orientiert sich an den Sections
- 2 Typen list und single
- html Dateien mit integrierten Go-Templates
- Meist durch Theme Vorgeben



## DATEITYPEN

- list.html
- single.html
- baseof.html

# BEISPIEL

```
1      <!doctype html>
2      <html>
3      <head>
4          <meta charset="utf-8">
5          <title>{{ .Page.Title }}</title>
6          {{ $style := resources.Get "sass/main.scss" | resources.ToCSS |
resources.Minify }}
7          <link rel="stylesheet" href="{{ $style.Permalink }}">
8
9      </head>
10
11     <body>
12
13     {{ partial "nav.html" }}
14
15     {{ block "main" . }}
16     {{ end }}
17     </body>
18     </html>
```

# SPEZIALFORMATE

Hugo kann nicht nur HTML erzeugen

# REALLY SIMPLE SYNDICATION FEEDS

Standard-XML-Format um Leser über Aktualisierungen zu informieren - RSS-Feeds

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <feed xmlns="http://www.w3.org/2005/Atom">
3
4      <title>Example Feed</title>
5      <link href="http://example.org/" />
6      <updated>2003-12-13T18:30:02Z</updated>
7      <author>
8          <name>John Doe</name>
9      </author>
10     <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>
11
12     <entry>
13         <title>Atom-Powered Robots Run Amok</title>
14         <link href="http://example.org/2003/12/13/atom03" />
15         <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
16         <updated>2003-12-13T18:30:02Z</updated>
17         <summary>Some text.</summary>
18     </entry>
19
20 </feed>
```

# EINSATZ

Beim publizieren (siehe später) automatisch miterzeugt.

Man muss nur noch den Link im HTML Layout darauf einbauen.

```
1 <link rel="alternate" type="application/rss+xml"
2     href="{{.Site.BaseURL }}/index.xml"
3     title="{{.Site.Title }}">
```

```
45 copyright: 2024 Tuxbreeder e.V. Alle Rechte vorbehalten
46
47 Author:
48     name: Webredakteur Tuxbreeder e.V.
49     email: webmaster@tuxbreeder.de
```

# DATENSTRUKTUR

Verzeichnis: data

- Strukturierte Daten wie yaml, json oder toml als Quellen
- Mittels Methods auch CSV-Dateien
- Verarbeitung in den Layouts
- Ideal für Listen gleichartiger Daten

# BEISPIEL

## SIEGER WETTRENNEN

```
1 champions:  
2   2023: King penguin  
3   2022: Gentoo penguin  
4   2021: Banded penguin
```

## Verwendung in Templates:

```
10 <dl>  
11   {{ range $k, $v := $.Site.Data.champs.champions }}  
12     <dt>{{ $k }}</dt>  
13     <dd>{{ $v }}</dd>  
14   </li>  
15   {{ end }}  
16 </dl>
```

# ERGEBNIS

**2021**

Banded penguin

**2022**

Gentoo penguin

**2023**

King penguin



# KALENDER

## CSV-DATEI

Verzeichnis beliebiges Verzeichnis

Funktion: getCSV Delimiter File

```
1  {{ $data := getCSV "|" "other-data/calendar.csv" }}
2
3  <table>
4  {{ range $data }}
5  <tr>
6      {{ range . }}
7          <td>{{ . }}</td>
8      {{ end }}
9  </tr>
10 {{ end }}
11 </table>
```

# KALENDER

## ICS

### Schritt 1: config.yaml

```
1 outputFormats:
2   calendar:
3     mediaType: text/calendar
4     protocol: http://
```

### Schritt 2: content/section-cal/\_index.md

```
1   ---
2   outputs:
3     - html
4     - calendar
5   ---
```

# ICS - FORTSETZUNG

## Schritt 3: layouts/section-cal/list.ics

```
1 BEGIN:VCALENDAR
2 VERSION:2.0
3 PRODID:Hugo/Calendar
4 METHOD:PUBLISH
5 {{ $data := getCSV "|" "other-data/calendar.csv" }}
6     BEGIN:VEVENT
7         SUMMARY:{{index $data 2 }}
8         DESCRIPTION:{{index $data 3 }}
9     END:VEVENT
10 {{end}}
11 END:VCALENDAR
```

SEO

SUCHMASCHINENOPTIMIERUNG

# SITEMAP.XML

- XML-Beschreibung einer Website für Suchmaschinen
- Konfiguration in der globalen config.yaml

```
1 sitemap:  
2   changeFreq: monthly  
3   filename: sitemap.xml  
4   priority: 0.5
```

# ROBOTS.TXT

```
1 enableRobotsTXT: true
```

# INTERNATIONALISIERUNG

Hinweis: Hugo ist kein Übersetzer!

Aber richtig Konfiguriert gibt es eine mehrsprachige Seite.

# KONFIGURATION

## VARIANTE 1

Datei: config.yaml

```
1 DefaultContentLanguage: de
2 languages:
3   de:
4     languageName: Deutsch
5     contentDir: content/de
6     weight: 1
7   en:
8     languageName: English
9     contentDir: content/en
10    weight: 2
```



# KONFIGURATION

## VARIANTE 2

Datei: config.yaml

```
1 DefaultContentLanguage: de
2 languages:
3   de:
4     languageName: Deutsch
5     weight: 1
6   en:
7     languageName: English
8     weight: 2
```

Statt dessen auf Dateiebene

# KONFIGURATION

## VARIANTE 2

```
1 info.en.md  
2 info.de.md
```

# PUBLIZIEREN

```
1 hugo
```

```
2 hugo --gc --minify
```

## WEITERE QUELLEN

- <https://hugocodex.org/about/>
- <https://github.com/gohugoio/>
- <https://gohugo.io/documentation/>

Vielen Dank  
für die Aufmerksamkeit  
Fragen?

Speaker notes