



# Numrich Albrecht Klumpp

## Redaktionsanleitung und Technische Dokumentation

Jan-Henrik Hempel

(Stand: 11.01.2016)

### Kontakt:

Jan-Henrik Hempel  
Schlatkow 6  
17390 Schmatzin

Mobil 0175.1661003  
E-Mail [jhh@j2h.com](mailto:jhh@j2h.com)  
Internet [www.j2h.com](http://www.j2h.com)

# 1 Redaktionsanleitung

## 1.1 Überblick

Die Website funktioniert grundsätzlich wie folgt:

Für das Rendering wird das System „Jekyll“ verwendet. Dieses generiert anhand von HTML-Templates und Inhalten im Markdown-Format eine komplette statische Website.

Sich häufig ändernde Inhalte (hier Projekte und Team-Fotos) werden in Markdown-Dateien abgelegt – das sind einfache Textdateien mit einigen Formatierungen).

Dazugehörige Bilder werden als Dateien abgelegt und in den Markdown-Inhalten referenziert.

Die Markdown-Dateien liegen in den Ordnern die „\_posts“ heißen, konkret unter „/projekte/\_posts/“ und „/buero/team/\_posts/“. Die Bilder sind in den Ordnern „projekte“ und „buero“ abgelegt.

Andere Inhalte stehen in den HTML-Templates.

## 1.2 Erstellen oder Verändern eines Projektes

### 1.2.1 Projektdateien

Die Projekte werden als Markdown Dateien mit YAML Frontmatter Daten als .md Dateien im Ordner „/projekte/\_posts/“ abgelegt.

Der Dateiname von „Posts“ muss immer YYYY-MM-TT-[Name].md heißen.

Das ist bei dem System einfach so, siehe auch:

<http://jekyllrb.com/docs/posts/#the-posts-folder>

*Einen wirklichen Zweck hat der Name allerdings nicht, da sowohl Datum als auch Link in der Markdown Datei stehen - vermutlich ist es ein historisches Relikt.*

Die Projektbilder (wie auch andere Bilder) werden einfach in Ordnern abgelegt – bei den Projekten am besten in Unterordnern pro Projekt unter „projekte“, wobei der Ordner den selben Namen wie die URL des Projektes hat. Die Originalbilder sollten in einer maximalen Breite von 2500px und als minimal komprimiertes (bspw. 98%) JPG abgespeichert werden.

### 1.2.2 Projektdaten und Metadaten

Metadaten sind Inhalte, welche nicht direkt der angezeigte Text sind, sondern Informationen zum Dokument enthalten. Die Metadaten werden in YAML Front Matter geschrieben (<http://jekyllrb.com/docs/frontmatter/>), d.h. diese sehen so aus:

---

layout: post

title: Projektname

---

### Hier ein vollständiges Beispiel:

---

layout: post

categories:

- projekt
- startseite

date: 2014-04-01

published: true

title: Rathaus Königs Wusterhausen

shorttitle: Rathaus Königs Wusterhausen

permalink: /projekte/RKW\_KoenigsWusterhausen/

tags:

- Schule
- Wettbewerb

homepage: /projekte/CHIC/NAK\_CHIC\_RGB\_0776-ORIG\_kl.jpg

news: Das Charlottenburger Innovations-Centrum - CHIC - wurde am 17.09.2015 feierlich eröffnet.

images:

- /projekte/RKW\_KoenigsWusterhausen/01.jpg
- /projekte/RKW\_KoenigsWusterhausen/02.jpg
- /projekte/RKW\_KoenigsWusterhausen/03.jpg
- /projekte/RKW\_KoenigsWusterhausen/04.jpg

captions:

- Mensa für das Bildungszentrum der Finanzverwaltung des Landes
- 02 In context of the running paid content
- 03 Lorem Ipsum
- 04 Lorem Ipsum

---

### Detaillierte Erläuterungen zu den Attributen:

layout	Sollte immer „post“ sein
categories	Als Aufzählung, es können eine oder mehrere Kategorien angegeben werden. Es gibt: <ul style="list-style-type: none"><li>- projekt (ein Projekt)</li><li>- startseite (ein Projekt, welches auf der Startseite erscheinen soll)</li><li>- team (ein Team-Foto auf der Büro-Seite)</li></ul>

date	Datum im Format YYYY-MM-TT, also bspw. 2014-04-01 Das Datum wird für die Sortierung verwendet, also auf der Startseite, auf der Projektübersicht und bei den Teamfotos.
published	Sollte immer „true“ sein. Wenn nicht, dann ist diese Seite nicht online zu sehen.
title	Titel des Projektes oder des Teamfotos
shorttitle	Kurztitel eines Projektes. Wird auf der Projektübersichtsseite angezeigt.
permalink	Der URL Pfad zur Projektseite. Hier sollte immer erst „projekte/“ stehen und dahinter ein kurzer aber sprechender Name des Projektes in URL-Form, also ohne Freizeichen, Umlaute usw. Bspw.: /projekte/CHIC/
tags	Aufzählung von Begriffen – hieraus werden die Filterkategorien auf der Projektübersicht gebildet. Ein Projekt kann ein oder mehrere Tags haben.
homepage	Optional. Wenn das Projekt auf der Startseite angezeigt werden soll, dann kann es hier ein spezielles Bild nur für die Startseite erhalten. Hier wird der Bildpfad angegeben, wo es liegt. Die Bilder sollten möglichst immer in einem Ordner „projekte/“ liegen und dann in einem Order der genauso wie der Permalink heißt. Bspw.: - /projekte/CHIC/01.jpg
projectpage	Wie „homepage“, allerdings für die Projektübersichtsseite.
news	Optional. Wenn das Projekt auf der Startseite angezeigt werden soll, dann kann hier ein News-Text eingefügt werden. Dieser legt sich dann mit einem weißen Kasten über das Bild.
images	Eine Aufzählung von Bildern, welche auf der Projektdetailseite im Karussell angezeigt werden sollen. Hier wird jeweils der Bildpfad angegeben, wo es liegt. Die Bilder sollten möglichst immer in einem Ordner „projekte/“ liegen und dann in einem Order der genauso wie der Permalink heißt. Bspw.: - /projekte/CHIC/01.jpg - /projekte/CHIC/02.jpg - ...
captions	Hier kommt analog zu „images“ eine Liste an Bildunterschriften rein. Diese werden jeweils passend zum Bild in „images“ angezeigt.

### 1.2.3 Projektdetails

In den Projektdetails wird zuerst immer der Titel aus den Metadaten („title“) als Überschrift angezeigt.

Anschließend kann einfach als Markdown (also Plain Text) eine Beschreibung eingesetzt werden.

Die zusätzlichen Projektdetails werden als Tabelle dargestellt und entsprechend auch als Tabelle im Markdown-Format gepflegt:

Adresse		Schloßstraße 4, 15711 Königs Wusterhausen
Bauherr		Stadt Königs Wusterhausen

BGF		2.300 m <sup>2</sup>
Bausumme		KGR 200-700, 7.000.000 €
Fertigstellung		April 2014
Projektleiter		Grant Kelly
Mitarbeiter		Arturo Panichi
Projektfotos		Nina Straßgüt1

Entsprechend sind so auch Verlinkungen möglich:

Projektfotos | [Nina Straßgüt1] (<http://www.ninastrg.de/>)

### 1.2.4 Bilder aktualisieren

Wenn man ein bereits vorhandenes Bild aktualisieren möchte, muss man aufpassen bzw. immer einen neuen Dateinamen vergeben und nicht die alte Datei nur überschreiben.

Cloudinary (siehe unten) ist dafür da die Bilder zu skalieren und schnell auszuliefern. Dafür verwenden Sie ein CDN inkl. Caching, was sehr gut für die Performance der Seite ist. Die Herausforderung bei extensivem Caching ist es, Aktualisierungen vorzunehmen. Bei Cloudinary kann man per Kommando dort alle Bilder und Variante löschen. Allerdings hilft das nur bedingt, da deren Cache ein Zeitraum von 30 Tagen hat. Das ist grundsätzlich sehr gut. Für die Aktualisierung von Bildern heißt das aber, dass veränderte Bilder NICHT den selben Dateinamen haben sollten. Denn dann kann entsprechend auch kein Cache greifen und es werden die richtigen Bilder angezeigt. Wenn man dies mal vergessen sollte, erscheinen die Aktualisierungen dann nach ca. 30 Tagen.

## 1.3 Erstellen oder Verändern eines Team Fotos

Ein Team-Foto besteht aufgrund der wenigen anzuzeigenden Infos nur aus einigen wenigen Metainformationen:

---

```
layout: post
category: team
date: 2012-09-01
published: true
title: Venedig 2012
image: /buero/team/Team-in-Venedig-2012.png
---
```

layout	Sollte immer „post“ sein
categories	Die Kategorie ist „team“ (ein Team-Foto auf der Büro-Seite)
date	Datum im Format YYYY-MM-TT für die Sortierung. Die neusten erscheinen zuerst.

published	True, wenn das Bild online angezeigt werden soll.
titel	Der Titel, der auch neben dem Foto angezeigt wird.
image	Das Bild in Form des Bildpfades und möglich mit „buero/team/“ zu Beginn.

## 1.4 Verändern anderer Inhalte

Alle weiteren Inhalte sind in HTML in den Templates enthalten:

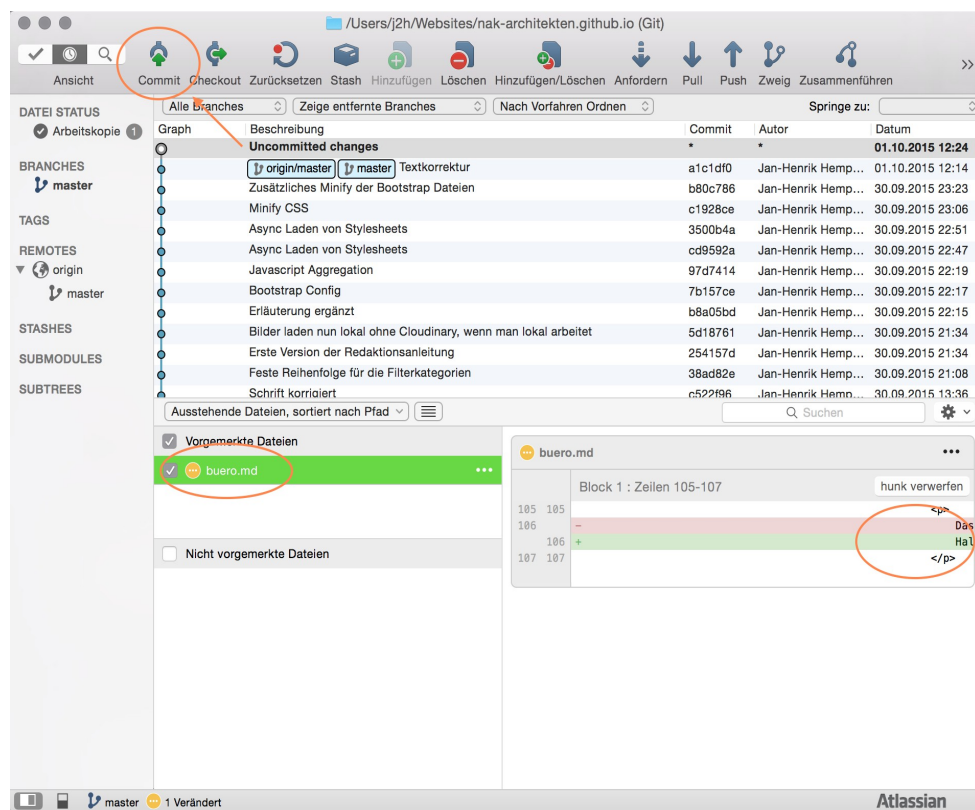
index.html → die Startseite

buero.md → die Büro Seite

projekte.md → die Projektübersichtsseite

## 1.5 Online-stellen von Änderungen

Änderungen müssen per **„commit“** in das Repository geschrieben und per **„push“** auf den Live-Server kopiert werden.



## 2 Technische Dokumentation

### 2.1 Hosting und Dateiverwaltung per GitHub Pages

GitHub Pages vereint eine Versionsverwaltung für Dateien, ein Hosting statischer Dateien und die Bereitstellung von Jekyll zur Generierung einer kompletten statischen Website. Für einfache Anwendungen gibt es einen kostenfreien Account.

Link: <https://pages.github.com/>

Repository: <https://github.com/nak-architekten/nak-architekten.github.io>

Account: [jhh+nak@j2h.com](mailto:jhh+nak@j2h.com)

Passwort: x[kpo76CdQkGfJx

Collaborators: <https://github.com/NAK-Filser>  
<https://github.com/j2hcom>

### 2.2 Generierung per Jekyll

Die Website ist mit HTML-Templates für Jekyll erstellt und erzeugt auf Basis der Templates und der Inhalte im Markdown-Format eine komplette statische Website.

Jekyll: <https://jekyllrb.com/>

Markdown: <http://kramdown.gettalong.org/syntax.html>

YAML: <http://jekyllrb.com/docs/frontmatter/>

### 2.3 Bildskalierung per Cloundinary

Für die optimale Skalierung der Bilder für den jeweiligen Zweck bzw. die beste Größe sowie das Beschneiden zum richtigen Seitenverhältnis wird Cloundinary verwendet.

Dabei werden die Originalbilder im Git Repository verwaltet und die „echten“ URLs als Quellreferenz an Cloundinary übergeben. So ist keinerlei Dateiverwaltung bei Cloundinary nötig.

Link: <http://cloudinary.com/>

Account: jhh+nak@j2h.com

Passwort: FroirP2bcMQL8\$x

Account ID: dpcao7gpf

URLs: \*.j2h.com

\*.nak-architekten.de

nak-architekten.github.io

### 2.4 Webtracking per Google Analytics

Tracking-ID: UA-416191-16



## 2.5 Webfonts per fonts.com

Account: von NAK

Integration:

```
<link type="text/css" rel="stylesheet"
href="http://fast.fonts.net/cssapi/78b7c7e4-d03d-4af2-8e78-
1a950f9eb329.css"/>
```





## 3 Lokale Kopie aufsetzen

### 3.1 Schritt 1 - Jekyll installieren

Siehe <https://jekyllrb.com/>

Per Terminal:

```
$ gem install jekyll
```

### 3.2 Schritt 2 - Website von GitHub holen

```
$ cd ~/Websites/
```

```
$ git clone git@github.com:nak-architekten/nak-architekten.github.io.git
```

### 3.3 Schritt 3 - Jekyll starten

In das Verzeichnis wechseln:

```
cd nak-architekten.github.io.git
```

```
$ jekyll serve --config _config.local.yml
```

Zur Erläuterung: Um auch lokal Bilder zu sehen und nicht die generelle Konfigurationsdatei zu verändern, sollte eine lokale Config verwendet werden:

### 3.4 Weitere Infos

<https://jekyllrb.com/docs/usage/>



## 4 Tipps zur Nutzung von Git

Git ist eine Versionsverwaltung, d.h. hier werden an einem zentralen Ort (das Repository) die Dateien der Website verwaltet und alle Änderungen können zu jeder Zeit nachvollzogen und auch rückgängig gemacht werden. Man kann so auch verschiedene Versionen (Branches) der Website erstellen, um bspw. Größere Änderungen vorzunehmen oder zu bestimmten Anlässen wieder auf eine frühere Version zurück zu springen.

Entsprechend ist die Verwendung etwas anders als bspw. bei einem FTP. Man kopiert nicht einfach Dateien auf den Server und löscht dort alte, sondern man checkt Änderungen ein (commit) und schickt diese dann zum Server (push).

Ein einfaches Tool mit grafischer Oberfläche ist bspw.

<https://www.sourcetreeapp.com/>