

# Hugo

- [Externe Links in Hugo automatisch in einem neuen Tab öffnen](#)

# Externe Links in Hugo automatisch in einem neuen Tab öffnen

Nachdem ich monatelang externe Links in Hugo mühsam manuell konfiguriert habe, kam ich zu dem Entschluss, dass es einen besseren Weg geben muss. Und den gibt es! Hier ist die Lösung, die für mich funktioniert hat.

Einige Themes bieten hierfür eine native Unterstützung an; in diesem Fall reicht ein Blick in die Dokumentation und das Hinzufügen einer Zeile in deiner `hugo.yaml` oder `hugo.toml`. Wenn du jedoch ein Theme verwendest, das diese Funktion nicht standardmäßig unterstützt, ist die folgende Methode ideal für dich.

---

## Schritt 1 – Konfigurationsdatei aktualisieren

---

Zuerst musst du einen Block zu deiner Hugo-Konfigurationsdatei hinzufügen. Je nach Setup heißt diese Datei meist `hugo.yaml` oder `hugo.toml`.

### YAML-Version:

```
markup:
  goldmark:
    renderer:
      unsafe: true
  parser:
    attribute:
      block: true
      title: true
```

Füge entweder den `markup`-Block neu hinzu oder aktualisiere deinen bestehenden Block mit diesen Einstellungen.

### Was bewirkt dieser Code?

- Hugo nutzt standardmäßig **Goldmark** als Markdown-Parser.

- `unsafe: true` erlaubt es dem Renderer, rohes HTML einzubeziehen, das du in deinen Templates schreibst.
- Die `attribute`-Einstellungen ermöglichen es Hugo, HTML-Attribute in Links korrekt zu verarbeiten.

## TOML-Version:

```
[markup]
  [markup.goldmark]
    [markup.goldmark.renderer]
      unsafe = true
  [markup.goldmark.parser]
    [markup.goldmark.parser.attribute]
      block = true
      title = true
```

“

**Tip:** Erstelle immer ein Backup deiner Konfigurationsdatei. Da diese Dateien sehr empfindlich auf Leerzeichen und Einrückungen reagieren, empfehle ich dir einen Editor wie Visual Studio Code.

## Schritt 2 – Ein Render-Hook-Template erstellen

Das klingt komplizierter, als es ist. Ein Render-Hook sagt Hugo vereinfacht: „Wann immer du auf einen Markdown-Link stößt, verwende dieses Template anstelle der Standard-Ausgabe.“

### Die Dateistruktur erstellen

Erstelle eine neue Datei namens `render-link.html` im Verzeichnis

`layouts/_default/_markup/`. Falls die Unterordner `_default` oder `_markup` bei dir noch nicht existieren, lege sie einfach an.

Deine Struktur sollte so aussehen:

```
deine-website/
??? layouts/
  ??? _default/
    ??? _markup/
```

```
??? render-link.html
```

## Den Code hinzufügen

Kopiere den folgenden Code in deine `render-link.html`:

```
{{- $link := .Destination -}}
{{- $isExternal := strings.HasPrefix $link "http" -}}
{{- if not $isExternal -}}
  {{- $isExternal = strings.HasPrefix $link "/" -}}
{{- end -}}

<a href="{{ .Destination | safeURL }}"
  {{- with .Title }} title="{{ . }}"{{ end }}
  {{- if $isExternal }} target="_blank" rel="noopener"{{ end -}}
>{{ .Text | safeHTML }}</a>
```

Dieser Code extrahiert die URL (`.Destination`) aus deinem Markdown-Link. Er prüft, ob die URL mit `http` beginnt – falls ja, wird sie als externer Link eingestuft. In diesem Fall (`if $isExternal`) wird das Attribut `target="_blank"` hinzugefügt.

Zusätzlich wird `rel="noopener"` eingefügt. Dies ist eine Sicherheitsmaßnahme (Best Practice), die deine Seite schützt, wenn externe Links in neuen Tabs geöffnet werden.

---

## Fazit

Wenn du nun deine Hugo-Seite in der Vorschau betrachtest, sollten sich interne Links im selben Tab öffnen, während externe Links automatisch in einem neuen Tab erscheinen.

Es gibt zwar alternative Lösungen (z. B. via JavaScript), aber diese Methode hat den großen Vorteil, dass sie bereits zum **Build-Zeitpunkt** und nicht erst im Browser des Nutzers ausgeführt wird. Das macht deine Website schneller und sorgt für eine bessere User Experience.