

GOLLUM: Syntaxe

Table of Contents

- [GOLLUM: Syntaxe](#)
 - [Titres de page](#)
 - [Titres personnalisés via CLI](#)
 - [Titres personnalisés via la syntaxe de métadonnées](#)
 - [Sous-pages](#)
 - [Assainissement de pages](#)
 - [Tags](#)
 - [Liens](#)
 - [Lien vers des ressources externes](#)
 - [Lien vers des fichiers internes \(pas d'images ou de pages\)](#)
 - [Lien vers des images internes](#)
 - [Lien vers des pages internes](#)
 - [Inclusion de pages](#)
 - [Balise Table des matières \(COT\)](#)
 - [Tags échappés](#)
 - [Code Blocks](#)
 - [Code Blocs clôturés \(style GitHub\)](#)
 - [Code Blocks clôturés \(style Kramdown\)](#)
 - [Syntaxe des blocs de code en surbrillance](#)
 - [Macros](#)
 - [Macros par défaut](#)
 - [Macros personnalisées](#)
 - [Diagrammes](#)
 - [Diagrammes de séquence](#)
 - [Diagrammes PlantUML](#)
 - [Mathématiques](#)
 - [Authentification](#)

Titres de page

Par défaut, le titre d'une page correspond à son chemin d'accès au fichier, par rapport à la racine du référentiel (ou au répertoire de fichier de la page), sans extension (par exemple, /mordor/Sauron).

Titres personnalisés via CLI

Si Gollum est lancé avec l'option **-h1-title**, le premier en-tête **<h1>** de la page sera utilisé. Ainsi, par exemple, cette page de démarrage:

```
# Ceci est mon titre  
# Ceci est le premier h1 qui sera affiché
```

sera rendu avec le titre **Ceci est le titre** de la page de titre. Le premier en-tête affiché dans le texte de la page sera **Ceci est le premier h1 qui sera affiché**.

Cette option remplace la syntaxe de métadonnées.

Titres personnalisés via la syntaxe de métadonnées

La syntaxe:

```
<!-- --- title: Le titre de ma page -->
```

Note: es métadonnées doivent être le premier morceau de texte de la page.

Sous-pages

Gollum permet d'ajouter un en-tête, une barre latérale et un pied de page aux pages. Ils seront très probablement partagés entre plusieurs pages. Les sous-pages affectent toutes les pages de leur répertoire et tous les sous-répertoires qui ne possèdent pas de sous-page.

- Les fichiers d'en-tête sont nommés **_Header.ext**, où ext est l'une des extensions de formats pris en charge.
- Les fichiers de la barre latérale sont nommés **_Sidebar.ext**, où ext est l'une des extensions de formats pris en charge.
- Les fichiers de pied de page sont nommés **_Footer.ext**, où ext est l'une des extensions de formats pris en charge.

Assainissement de pages

Pour des raisons de sécurité et de compatibilité, les pages Gollum peuvent ne pas contenir de code CSS personnalisé, JavaScript ni de balises HTML potentiellement dangereuses ou inconnues. Ils seront supprimés du HTML généré.

Tags

Les tags de Gollum ont été conçus pour ressembler aux tags d'autres balises, en particulier de MediaWiki. La syntaxe de base est la suivante:

```
[[tag]]
```

Certaines balises acceptent des attributs séparés par des symboles de canal. Certains attributs doivent précéder la balise et d'autres doivent la suivre:

```
[[prefix-attribute|tag]]  
[[tag|suffix-attribute]]
```

Liens

La balise link crée un lien cliquable vers une ressource. Il a deux formes:

```
[[linked-resource]] # le texte du lien créé sera égal à celui de la  
ressource  
[[link-text|linked-resource]] # le texte du lien créé sera link-text
```

Lien vers des ressources externes

Exemples:

```
[[http://example.com]]  
[[link-text|http://example.com/pdfs/gollum.pdf]]
```

Lien vers des fichiers internes (pas d'images ou de pages)

Identique aux liens vers de ressources externes, mais les URL doivent être des chemins:

- Soit un chemin relatif, par exemple docs / diagram.png. Les chemins relatifs pointent vers un fichier relatif à la page, où le lien est défini. Ils ne doivent pas être préfixés par une barre oblique.
- Ou un chemin absolu, par exemple /docs/diagram.png. Les chemins absolus pointent vers un fichier par rapport à la racine du référentiel Gollum. Ils doivent être précédés d'une barre oblique.

Lien vers des images internes

Identique aux liens vers fichiers internes, mais supporte également plusieurs attributs spéciaux:

[[image-url alt=text]]	Texte à afficher lorsque l'image n'est pas trouvée.
[[image-url frame]]	Indique à Gollum de placer l'image dans un , ce qui affichera une bordure autour de l'image.
[[image-url align=position]]	Demande à Gollum d'aligner l'image de la manière indiquée. La position peut être gauche, centre et droite. Par défaut: à gauche.
[[image-url float]]	Indique à Gollum de faire flotter l'image de manière à ce que le texte l'entoure. Cette option ne peut pas être utilisée avec l'option align=center. Lorsque la fonction flottante est activée et que l'alignement n'est pas spécifié, align=left est appliqué.
[[image-url height=valeur]]	Définit la hauteur maximale de l'image. Les valeurs doivent être spécifiées avec l'unité px ou em.

<code>[[image-url width=value]]</code>	Définit la largeur maximale de l'image. Les valeurs doivent être spécifiées avec l'unité px ou em.
--	--

Tous ces attributs peuvent être combinés en les séparant simplement avec des pipes :

<code>[[image-url frame alt=text]]</code>	Indique à Gollum de placer l'image dans un <code></code> et utilise le texte text comme légende.
---	--



Par défaut, le texte remplira tout l'espace autour de l'image. Pour contrôler le nombre de pages à afficher, utiliser cette balise pour arrêter et démarrer un nouveau bloc afin que du contenu supplémentaire ne soit pas rempli.

`[[_]]`

Lien vers des pages internes

La balise suivante créera un lien vers une autre page Gollum:

```
[[canonical-page-filename]]
```

Le nom de fichier canonique est le nom de fichier de la page, mais après avoir appliqué ces règles:

- L'extension est enlevée.
- Les espaces (U+0020) sont remplacés par des tirets (U+002D).
- Les barres obliques (U+002F) sont remplacées par des tirets (U+002D).

Exemples:

```
[[J. R. R. Tolkien]] fera référence à la page J.-R.-R.-Tolkien.ext
[[Movies / The Hobbit]] fera référence à la page Movies---The-Hobbit.ext
[[モルドール]] fera référence à la page モルドール.ext
```

La page référencée peut exister n'importe où dans le référentiel git. Gollum la recherchera et liera la première page de ce type qu'elle trouvera.

Inclusion de pages

Contrairement à la balise link, celle-ci est dédiée aux pages, a une syntaxe spéciale et inclut une page dans une autre, plutôt que de la lier:

```
[[include:canonical-page-filename]]
```

Balise Table des matières (COT)

Gollum fournit une balise spéciale pour incorporer une table des matières dans une page:

```
[[_TOC_]]
```

Ou avec une profondeur maximale

```
[[_TOC_|levels=3]]
```

Remarques:

- Il est sensible à la casse, utilisez donc toujours des majuscules.
- Il peut également être inséré dans des sous-pages.

Il existe également un paramètre global de la table des matières (désactivé par défaut) qui force la table des matières dans chaque page. Ajouter simplement la ligne suivante au fichier de configuration:

```
Precious::App.set(:wiki_options,{:universal_toc=>true})
```

Tags échappés

Dans le cas où une balise doit être affichée littéralement sur une page, la préfacier par un guillemet simple:

```
'[[tag]]'
```

Code Blocks

Code Blocs clôturés (style GitHub)

Naturellement, toutes les pages peuvent bénéficier de cette syntaxe. Exemple:

```
```\n\ndef foo\n  met 'bar'\nend\n```
```

Le bloc doit être enfermé avec trois **backticks**.

- Les backticks d'ouverture peuvent être indentés arbitrairement.
- Le contenu du bloc doit être mis en retrait au même niveau que les backtick d'ouverture. S'il est mis en retrait avec deux espaces supplémentaires ou un onglet, ils seront ignorés (cela rend les blocs plus faciles à lire en texte brut).
- Les backticks de fin doivent être mis en retrait au même niveau que les backticks d'ouverture.

## Code Blocks clôturés (style Kramdown)

Naturellement, toutes les pages peuvent bénéficier de cette syntaxe. Exemple:

```
~~~~~  
Voici du code.  
~~~~~
```

Le bloc doit être entouré de trois tildes ou plus (~).

## Syntaxe des blocs de code en surbrillance

C'est une extension des blocs de code de style GitHub. Naturellement, toutes les pages peuvent bénéficier de cette syntaxe. Exemple:

```
```ruby  
def foo  
  met 'bar'  
end  
```
```



Le langage/le balisage du code est défini à côté des backticks d'ouverture. Il existe une grande variété de langages/balises à utiliser, qui dépendent du surligneur de syntaxe actuellement utilisé dans Gollum:

- \* **Rouge**: C'est la valeur par défaut, sauf si Python et Pygments est installé sur le système. Ensuite, les pygments ont plus de priorité.
- \* **Pygments**: Nécessite Python 2.5+ (2.7.x recommandé).

De plus, on peut mettre en surbrillance un fichier à partir d'un référentiel. Naturellement, lorsque le fichier est mis à jour et que la page est actualisée dans le navigateur, l'extrait de code est également mis à jour:

```
```ruby: /lib/gollum/app.rb```
```

Dans ce cas, le fichier `/lib/gollum/app.rb` sera inclus dans la page et la syntaxe en surbrillance.

Macros

Outre les balises, Gollum fournit une autre couche de son propre balisage, les macros. Les principales différences sont les suivantes:

- Les balises sont toutes prédéfinies et ne peuvent pas être personnalisées. Les macros peuvent être personnalisées et vous pouvez même créer les vôtres.
- Syntaxe différente.

Macros par défaut

- **AllPages**
 - **Description:** Imprime une liste de toutes les pages du wiki.
 - **Syntaxe:** `<<AllPages()>>`
 - **Résultat (exemple):** `<ul id="pages"><li>AllPagesMacroPage</li></ul>`
- **GlobalTOC**
 - **Description:** Imprime une table des matières cliquable (toutes les pages du site)
 - **Syntaxe:** `<<GlobalTOC()>>` on peut également remplacer le titre: «GlobalTOC("Toutes les pages")»
- **Navigation**
 - **Description:** Imprime une table des matières cliquable (de toutes les pages sous le chemin indiqué ou sous le chemin de la page en cours par défaut)
 - **Exemples:**
 - `<<Navigation()>>`
 - `<<Navigation("My TOC","subdir",true)>>`
 - **Syntaxe:** `<<Navigation(title,subdir,full\_path)>>`
 - title: string, en-tête au-dessus de la table des matières imprimée. La valeur par défaut est "Naviguer dans la table des matières".
 - subdir: string, le sous-répertoire à lister. Par défaut, le chemin de la page en cours.
 - full_path: true ou false. Indique si le chemin complet de chaque page doit être affiché, au lieu du chemin relatif sous sous-répertoire. Par défaut: false.
- **Series**
 - **Description:** render Suivant: et Précédent: des liens permettant de naviguer dans une série de pages du même répertoire. On peut avoir une série de pages dans un sous-répertoire nommé par exemple. steps_1.md, steps_2.md, steps_3.md. Ajouter simplement `<Série ("steps_")` » (dans l'en-tête/le pied de page) pour générer des liens vers les étapes précédentes et suivantes. De cette manière, on peut facilement générer, par exemple, un manuel d'utilisation pas à pas.
 - **Usage:** «Serie("préfixe")» Cherche toutes les pages avec un nom commençant par "préfixe" et crée un lien vers les pages de la série qui précèdent et suivent immédiatement la page actuelle (par nom de page, triées par ordre alphanumérique).

Macros personnalisées

Chaque macro doit:

- inclure la classe Macro.
- indiquer la méthode de rendu.
- être enregistré dans Gollum. Donc:
 - soit démarrer Gollum avec l'option **-config** et placer la classe Macro personnalisée dans le fichier **config.rb**,
 - soit démarrer Gollum via Rack et placer la classe Macro personnalisée dans le fichier **config.ru**.

Diagrammes

Diagrammes de séquence

Gollum peut générer des diagrammes de séquence à partir de la source sur une page. Avec l'utilisation de **WebSequenceDiagrams**. Par exemple:

```
{}{}{}{blue-modern
  alice-> bob: test
  bob-> alice: test response
}{}{}{}
```

On peut remplacer le style blue-moderne par tout autre style pris en charge.

Diagrammes PlantUML

Gollum peut restituer les diagrammes PlantUML à partir des sources sur une page. en utilisant un serveur PlantUML autohébergé. Par exemple:

```
@startuml
Alice -> Bob: demande d'authentification
Bob -> Alice: Réponse d'authentification
Alice -> Bob: une autre demande d'authentification
Alice <- Bob: une autre réponse d'authentification
@enduml
```

On peut intégrer n'importe quel type de diagramme pris en charge par PlantUML: diagrammes de séquence, diagrammes de cas, diagrammes de classes, diagrammes d'activité, diagrammes de composants, diagrammes d'objets et même de diagrammes filaires.



Pour les diagrammes filaires, utiliser la balise **@startuml** suivie de la syntaxe du mot-clé **salt**. La nouvelle syntaxe **@salt** n'est pas prise en charge.



Il ne peut y avoir de lignes vides à l'intérieur du bloc **@startuml/@enduml**. PlantUML attend un seul paragraphe et l'ajout de lignes vides entre eux sera lu comme plusieurs paragraphes.

Mathématiques

Pour activer les expressions mathématiques dans Gollum, le démarrer avec l'option **-mathjax**.

Exemples:

- Maths en ligne:

```
\\(2^2\\)
```

- Afficher les mathématiques:

```
$$2^2$$  
\\[2^2\\]
```

Par défaut, Gollum utilise la configuration **TeX-AMS-MML_HTMLorMML** avec l'extension **autoload-all**. On peut également définir une configuration mathjax avec l'option **-mathjax-config**.



Lorsqu'on utilise le balisage en mode organisationnel, il faut encapsuler les blocs mathématiques d'affichage dans les blocs
`#+BEGIN\_HTML . . . #+END\_HTML.`

Authentification

Gollum ne prend pas en charge l'authentification d'utilisateur.

Un projet distinct existe, OmniGollum, qui ajoute OmniAuth pour Gollum. Avec OmniGollum, on peut ajouter de nombreux fournisseurs OAuth (Github, Google, etc.) pour effectuer l'authentification.

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=outils:gollum-syntaxe>

Last update: **2025/02/19 10:59**

