

GOLLUM: Un wiki basé sur git

Table of Contents

- [GOLLUM: Un wiki basé sur git](#)
 - [Description](#)
 - [Configurations requise](#)
 - [Installation](#)
 - [Marges](#)
 - [Fonctionnement](#)
 - [Exécution depuis le source](#)
 - [Exécution avec Rack](#)
 - [Exécution en tant que service](#)
 - [Configuration](#)
 - [Fichier de configuration](#)

Description

Gollum est un wiki construit sur le système de gestion de contrôle de source Git. Un wiki de Gollum est simplement un dépôt git de nature spécifique:

- Le contenu d'un référentiel Gollum est éditable par l'homme. Les pages sont des fichiers texte uniques qui peuvent être organisés dans des répertoires comme bon vous semble, à condition qu'ils aient une extension de fichier reconnue. D'autres contenus peuvent également être inclus, par exemple des images, des PDF et des en-têtes / pieds de page.
- Pages Gollum:
- Peut être écrit dans une variété de [langages de marquage] (# balises).
- Peut être édité avec votre éditeur de système préféré ou IDE ou avec l'interface Web intégrée.
- Peut être affiché dans toutes les versions et peut facilement être restauré.
- Gollum prend en charge des fonctionnalités avancées telles que les diagrammes UML, les macros, les métadonnées et plus.

Gollum est petit et léger, mais parvient à mettre en œuvre tout ce qu'on peut raisonnablement souhaiter d'un wiki: liens entre les pages, formatage de texte de base, création, modification et suppression de page, recherche de texte intégral et un grand nombre d'options de formatage de texte.

Configurations requise

Système d'exploitation	Rubis	Adaptateurs	Pris en charge
Unix / Linux-like	Ruby (IRM) 2.1.0+	tous sauf [R]Git (https://github.com/repotag/rjgit)	oui

Système d'exploitation	Rubis	Adaptateurs	Pris en charge
Unix / Linux-like	[JRuby] (https://github.com/jruby/jruby) (compatible 1.9.3 +)	[R]Git (https://github.com/repotag/rjgit)	oui
Windows	Ruby (IRM) 2.1.0+	tous sauf [R]Git (https://github.com/repotag/rjgit)	non
Windows	[JRuby] (https://github.com/jruby/jruby) (compatible 1.9.3 +)	[R]Git (https://github.com/repotag/rjgit)	presque

Installation

Varie selon le système d'exploitation, le gestionnaire de paquets et l'installation de Ruby. En règle générale, il faut d'abord installer Ruby, puis Gollum.

1. Il est préférable d'installer Ruby via [RVM] (<https://rvm.io/>) ou un gestionnaire de package de choix.
2. Il est préférable d'installer Gollum via RubyGems:

```
[sudo] gem install gollum
```



Quelle que soit l'implémentation de Ruby utilisée, Gollum est fourni avec l'adaptateur git par défaut approprié. Donc, la procédure d'installation ci-dessus est commune à la fois pour MRI et JRuby.

Marges

Gollum est actuellement livré avec le support des balises suivantes:

- [Markdown] (<http://daringfireball.net/projects/markdown/syntax>)
- [RDoc] (<http://rdoc.sourceforge.net/>)

Étant donné que toutes les balises sont rendues par la gemme [github-markup] (<https://github.com/github/markup>), on peut facilement ajouter une prise en charge pour d'autres balises par une installation supplémentaire:

- [AsciiDoc] (<http://asciidoctor.org/docs/asciidoc-syntax-quick-reference/>) - [sudo] gem install asciidoctor
- [Creole] (<http://www.wikicreole.org/wiki/CheatSheet>) - [sudo] gem install creole
- [MediaWiki] (<http://www.mediawiki.org/wiki/Help:Formatting>) - [sudo] gem install wikicloth
- [Org] (<http://orgmode.org/worg/dev/org-syntax.html>) - [sudo] gem install org-ruby
- [Pod] (<http://perldoc.perl.org/perlpod.html>) - requiert Perl >= 5.10 (la commande perl doit être disponible sur votre ligne de commande)
 - Les versions inférieures devraient installer Pod :: Simple depuis CPAN.

- [ReStructuredText] (<http://docutils.sourceforge.net/docs/ref/rst/restructuredtext.html>) - requiert `python> = 2` (la commande `python2` doit être disponible sur votre ligne de commande)
 - Il faut également installer `docutils` pour Python 2. La procédure d'installation peut varier, encore une fois, selon le système d'exploitation et le gestionnaire de paquets.
- [Textile] (<http://redcloth.org/hobix.com/textile/quick.html>) - `[sudo] gem install RedCloth`

Par défaut, Gollum est livré avec le gem **kramdown** pour rendre Markdown. Cependant, on peut utiliser n'importe quel [Rendu de Markdown pris en charge par github-markup] (<https://github.com/github/markup/blob/master/lib/github/markup/markdown.rb>). Ce qu'il faut retenir, c'est que le premier moteur de rendu installé dans la liste sera utilisé. Ainsi, par exemple, **redcarpet** ne sera PAS utilisé si `github / markdown` est installé.

Fonctionnement

Simplement:

1. Accéder au référentiel git (wiki) via la ligne de commande.
2. Exécuter: `gollum`.
3. Ouvrir l'url `http://localhost:4567` dans le navigateur.

Cela démarrera un serveur Web (WEBrick) exécutant Gollum avec une interface Web, où on pourra voir et éditer le wiki.

Certaines limitations sont imposées par GitHub:

1. `gollum-lib` autorise les noms de page à avoir des barres obliques, qui se comportent comme prévu. Malheureusement, ni GitHub ni l'interface "officielle" de Gollum basée à Sinatra ne les supportent. Si ou souhaite organiser les pages du wiki en sous dossiers il faut "aplatir" le sous-répertoire git en convertissant les répertoires en nouvelles branches.



2. Gollum prend en charge plusieurs formats de balisage. Lors de la création d'une nouvelle page, on peut choisir le balisage cible via une liste déroulante. Ceci détermine directement l'extension de fichier de la nouvelle page (extension principale pour le balisage donné). De plus, Gollum a un ensemble prédéfini d'extensions pour chaque balise, pour lesquels il n'est pas possible de modifier le moteur de rendu (les fichiers `.txt` seront toujours traités en tant que Raw Txt même si ont les joints au format Markdow). Pour ces extensions on ne peut lier l'extension rationnelle à un autre ensemble d'extensions il faut donc modifier l'extension du fichier.

Exécution depuis le source

1. `git clone https://github.com/gollum/gollum`
2. `cd gollum`
3. `[sudo] bundle install` (peut ne pas toujours être nécessaire).
4. `bundle exec bin/gollum *` Comme cela, gollum suppose que le wiki cible (référentiel git) est le référentiel du projet lui-même. Si ce n'est pas le cas, exécuter `bundle exec`

```
bin/gollum <chemin d'accès au wiki>.
```

5. Ouvrir <http://localhost:4567> dans le navigateur.

Exécution avec Rack

On peut également exécuter gollum avec n'importe quel serveur compatible avec rack en plaçant ce fichier **config.ru** dans le référentiel wiki. Cela permet d'utiliser n'importe quel middleware Rack comme Rack::Auth, OmniAuth, etc.

```
#!/usr/bin/env ruby
besoin de 'rubygems'
nécessite 'gollum / app'

#!/usr/bin/env ruby
require 'rubygems'
require 'gollum/app'

gollum_path = File.expand_path(File.dirname(__FILE__)) # CHANGE THIS TO
POINT TO YOUR OWN WIKI REPO
wiki_options = {:universal_toc => false}
Precious::App.set(:gollum_path, gollum_path)
Precious::App.set(:default_markup, :markdown) # set your favorite markup
language
Precious::App.set(:wiki_options, wiki_options)
run Precious::App
```

Le middleware Rack peut transmettre les détails de l'auteur à Gollum dans un hachage de la session sous la clé **gollum.author**.

Les options disponibles pour le hachage: wiki_options correspondent généralement aux arguments utilisés lors de l'exécution de gollum à partir de la ligne de commande, à quelques exceptions près. Ci-dessous une liste des clés que vous pouvez utiliser pour définir les options de gollum lors de son exécution en tant qu'application Rack.

```
wiki_options = {}
wiki_options[:css] = true # Equivalent to --css
wiki_options[:js] = true # Equivalent to --js
wiki_options[:template_dir] = path # Equivalent to --template-dir
wiki_options[:page_file_dir] = path # Equivalent to --page-file-dir
wiki_options[:gollum_path] = path # Equivalent to ARGV
wiki_options[:ref] = ref ## Equivalent to --ref
wiki_options[:repo_is_bare] = true # Equivalent to --bare
wiki_options[:allow_editing] = false # # Equivalent to --no-edit
wiki_options[:live_preview] = true # Equivalent to --live-preview
wiki_options[:allow_uploads] = true # Equivalent to --allow-uploads
wiki_options[:per_page_uploads] = true # When :allow_uploads is set, store
uploads under a directory named after the page, as when using --allow-
uploads page
```

```
wiki_options[:mathjax] = true # Equivalent to --mathjax
wiki_options[:mathjax_config] = source # Equivalent to --mathjax-config
wiki_options[:user_icons] = source # Equivalent to --user-icons
wiki_options[:show_all] = true # Equivalent to --show-all
wiki_options[:collapse_tree] = true # Equivalent to --collapse-tree
wiki_options[:h1_title] = true # Equivalent to --h1-title
```

L'option **-base-path** n'a pas d'équivalent dans le hachage des options. Cela est dû au fait que le chemin de base est implémenté en tant que middleware, que l'on doit ajouter manuellement à config.ru. Voici le code pour le middleware de bin/gollum:



```
require 'rack'

class MapGollum
  def initialize base_path
    @mg = Rack::Builder.new do
      map '/' do
        run Proc.new { [302, { 'Location' =>
"/#{base_path}" }, []] }
      end
      map "/#{base_path}" do
        run Precious::App
      end
    end
  end

  def call(env)
    @mg.call(env)
  end
end

# Rack::Handler does not work with Ctrl + C. Use Rack::Server
instead.
Rack::Server.new(:app=> MapGollum.new('wiki'), :Port =>
options['port'], :Host => options['bind']).start
```

Lors de l'utilisation de cette configuration, la ligne d'exécution Precious::App de l'exemple config.ru en haut de cette page devient superflue.

Exécution en tant que service

On peut utiliser Gollum en tant que service système:

- Créer un fichier **gollum.service**.
- Ajouter une ligne ExecStart=.... Cette ligne spécifie l'emplacement de l'exécutable gollum et des arguments pour l'exécuter. Par exemple: ExecStart=/usr/local/bin/gollum --

```
show-all "/home/someuser/wiki/"
```

- Démarrer le service: `[sudo] systemctl start gollum.service`

On peut exécuter Gollum en tant que service système, mais sous un utilisateur particulier (root par défaut):

- Ajouter une ligne `user=%i` au fichier `gollum.service` et renommer le fichier en `gollum@.service`.
- Modifier la ligne `ExecStart=/usr/local/bin/gollum --show-all "/home/someuser/wiki/"` to `ExecStart=/usr/local/bin/gollum --show-all "/home/%i/wiki/"`.
- Démarrer le service: `[sudo]systemctl start gollum@<utilisateur-cible>.service`, où `<utilisateur-cible>` est l'utilisateur sous lequel exécuter Gollum.

Il existe une alternative à la configuration précédente: exécuter le service dans une session utilisateur. Pour y parvenir:

- Effectuer les deux premières étapes de la création d'un service système (voir ci-dessus).
- Créer le dossier `mkdir -p ~/.config/systemd/user/` et déplacer le fichier `gollum.service` à cet emplacement. Modifier `gollum.service` si nécessaire.
- Démarrer le service utilisateur via `[sudo] systemctl --user start gollum`.

Les anciennes versions de Debian (et les distributions dérivées telles que Ubuntu) utilisent toujours un système init de type SysV, qui utilise des scripts de shell en tant que fichiers de service. `contrib/sysv-debian/` contient un exemple d'un tel fichier de service.

Pour l'installer:



- Copier `contrib/sysv-debian/init.d/gollum` vers `/etc/init.d/` sur le système.
- Adapter les variables `GOLLUM_*` :
 - **GOLLUM_USER** définit l'utilisateur sur lequel le processus gollum sera exécuté comme,
 - **GOLLUM_BASE** définit le chemin d'accès au référentiel contenant le contenu du wiki, et
 - **GOLLUM_OPTS** permet de définir des paramètres supplémentaires passés à gollum sur la ligne de commande.
- Rendre le script exécutable s'il ne l'est pas déjà: `chmod +x /etc/init.d/gollum`
- En fonction du système, utiliser `service gollum start` ou `/etc/init.d/gollum start` pour démarrer Gollum (Le script comprend les paramètres `start`, `stop`, `restart` et `status`).

Configuration

Gollum est livré avec les options de ligne de commande suivantes:

Option	Arguments	Description
-host	[HOST]	Spécifie le nom d'hôte ou l'adresse IP sur laquelle écouter. Par défaut: 0.0.0.0 . <Sup> 1 </ sup>
-port	[PORT]	Spécifie le port avec lequel lier Gollum. Par défaut: 4567 .
-config	[FICHIER]	Spécifie le chemin d'accès au fichier de configuration de Gollum.
-ref	[REF]	Spécifie la branche git à servir. Par défaut: master .
-adapter	[ADAPTATEUR]	Lance Gollum en utilisant un adaptateur git spécifique. Par défaut: grit . <Sup> 2 </ sup>
-bare	aucun	Indique à Gollum que le référentiel git doit être traité comme un élément bare. Cela n'est nécessaire que lorsqu'on utilise l'adaptateur grit par défaut.
-base-path	[CHEMIN]	Spécifie la partie principale de toutes les URL Gollum (informations de chemin d'accès). Définir ceci sur /wiki rendra le wiki accessible sous http://localhost:4567/wiki/ . Par défaut: / .
-page-file-dir	[CHEMIN]	Spécifie le sous-répertoire pour toutes les pages. Si défini, Gollum ne servira que les pages de ce répertoire et de ses sous-répertoires. Par défaut: racine du référentiel .
-css	aucun	Indique à Gollum d'injecter du CSS personnalisé dans chaque page. Utilise custom.css à partir de la racine du référentiel.
-js	aucun	Indique à Gollum d'injecter du JS personnalisé dans chaque page. Utilise custom.js à partir de la racine du référentiel.
-emoji	aucun	Analyse et interprète les tags emoji (par exemple: heart :).
-no-edit	aucun	Désactive la fonctionnalité d'édition de pages.
-live-preview	aucun	Active la fonctionnalité de prévisualisation en direct dans l'éditeur de page.
-no-live-preview	aucun	Désactive la fonctionnalité d'aperçu en direct dans l'éditeur de page.
-allow-uploads	[MODE]	Active les téléchargements de fichiers. Si défini sur dir , Gollum stockera tous les téléchargements dans le répertoire /uploads/ à la racine du référentiel. Si réglé sur page , Gollum stockera chaque téléchargement sur la page en cours d'édition.
-mathjax	aucun	Active MathJax (rend les équations mathématiques). Par défaut, utilise la configuration TeX-AMS-MML_HTMLorMML avec l'extension autoload-all .
-irb	aucun	Lance Gollum en "mode console", avec une [API prédéfinie] (https://github.com/gollum/gollum-lib/).
-h1-title	aucun	Indique à Gollum d'utiliser le premier <h1> comme titre de page.
-show-all	aucun	Indique à Gollum d'afficher également les fichiers dans la vue des fichiers. Par défaut, seules les pages valides sont affichées.
-collapse-tree	aucun	Indique à Gollum de réduire l'arborescence de fichiers lorsque la vue de fichier est ouverte. Par défaut, l'arborescence est développée.
-user-icons	[MODE]	Indique à Gollum d'utiliser des icônes d'utilisateur spécifiques pour l'affichage de l'historique. Peut être réglé sur «gravatar», «identicon» ou «aucun». Par défaut: none .
-mathjax-config	[FICHIER]	Spécifie le chemin d'accès à une configuration MathJax personnalisée. S'il n'est pas spécifié, utilise le fichier mathjax.config.js à partir de la racine du référentiel.
-template-dir	[CHEMIN]	Spécifie le répertoire du modèle de mustache personnalisé.
-help	aucun	Affiche la liste des options sur la ligne de commande.
-version	aucun	Affiche la version actuelle de Gollum.





1. L'adresse IP **0.0.0.0** permet l'accès à distance. Pour une utilisation en local, utiliser **127.0.0.1**.
2. Lorsque **-css** ou **-js** est utilisé, les fichiers correspondants doivent être enregistrés dans le référentiel git, sinon gollum retournera une redirection 302 vers la page de création.
3. Les fichiers peuvent être téléchargés simplement en les faisant glisser dans la zone de texte de l'éditeur (ceci est cependant exclusif à l'éditeur par défaut, pas à l'éditeur de prévisualisation en direct).

Fichier de configuration

Lorsque l'option **-config** est utilisée, certaines parties internes de Gollum peuvent être personnalisées.

Toutes les modifications fonctionnent à la fois pour le fichier de configuration de Gollum (**config.rb**) et le fichier de configuration de Rack (**config.ru**).

From:

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

Permanent link:

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

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

