

目 录

致谢

1. 简介

1.1 变更日志

1.2 常见问题

1.3 故障排查

2. 下载安装

2.1 注册 Windows 服务

2.2 配置与运行

2.3 二进制安装

2.4 包管理安装

2.5 源码安装

3. 版本升级

3.1 二进制升级

3.2 源码升级

4. 功能介绍

4.1 自定义模板

4.2 Web 钩子

4.3 多国语言支持

4.4 授权认证

5. 高级用法

5.1 配置文件手册

5.2 公告与高阶指南

6. 捐赠我们

致谢

当前文档《Gogs中文文档》由进击的皇虫使用书栈(BookStack.CN)进行构建，生成于2018-01-21。

书栈(BookStack.CN)仅提供文档编写、整理、归类等功能，以及对文档内容的生成和导出工具。

文档内容由网友们编写和整理，书栈(BookStack.CN)难以确认文档内容知识点是否错漏。如果您在阅读文档获取知识的时候，发现文档内容有不恰当的地方，请向我们反馈，让我们共同携手，将知识准确、高效且有效地传递给每一个人。

同时，如果您在日常生活、工作和学习中遇到有价值有营养的知识文档，欢迎分享到书栈(BookStack.CN)，为知识的传承献上您的一份力量！

如果当前文档生成时间太久，请到书栈(BookStack.CN)获取最新的文档，以跟上知识更新换代的步伐。

文档地址：http://www.bookstack.cn/books/gogs_zh

书栈官网：<http://www.bookstack.cn>

书栈开源：<https://github.com/TruthHun>

分享，让知识传承更久远！感谢知识的创造者，感谢知识的分享者，也感谢每一位阅读到此处的读者，因为我们都将成为知识的传承者。

1. 简介

- [什么是 Gogs?](#)
 - [开发目的](#)
 - [开源组件](#)

什么是 Gogs?

Gogs 是一款极易搭建的自助 Git 服务。

开发目的

Gogs 的目标是打造一个最简单、最快速和最轻松的方式搭建自助 Git 服务。使用 Go 语言开发使得 Gogs 能够通过独立的二进制分发，并且支持 Go 语言支持的所有平台，包括 Linux、Mac OS X、Windows 以及 ARM 平台。

开源组件

- Web 框架: [Macaron](#)
- UI 组件:
 - [Semantic UI](#)
 - [GitHub Octicons](#)
 - [Font Awesome](#)
- 前端插件:
 - [DropzoneJS](#)
 - [highlight.js](#)
 - [clipboard.js](#)
 - [emojify.js](#)
 - [jQuery Date Time Picker](#)
 - [jQuery MiniColors](#)
 - [CodeMirror](#)
 - [notebook.js](#)
 - [marked](#)
- ORM: [Xorm](#)
- 数据库驱动:
 - [github.com/go-sql-driver/mysql](#)
 - [github.com/lib/pq](#)
 - [github.com/mattn/go-sqlite3](#)
 - [github.com/denisenkom/go-mssql](#)

- 以及其它所有 Go 语言的第三方包依赖。

1.1 变更日志

- [变更日志](#)
 - [0.11.34 @ 2017-11-22](#)
 - Bug 修复
 - [0.11.33 @ 2017-11-19](#)
 - Bug 修复
 - 新增特性
 - 功能改进
 - 其它变更
 - [0.11.29 @ 2017-08-15](#)
 - Bug 修复
 - [0.11.19 @ 2017-06-10](#)
 - Bug 修复
 - 新增特性
 - 功能改进
 - [0.11.4 @ 2017-04-05](#)
 - Bug 修复
 - [0.11 @ 2017-04-03](#)
 - Bug 修复
 - [0.11 RC @ 2017-03-27](#)
 - Bug 修复
 - 新增特性
 - 功能改进

变更日志

0.11.34 @ 2017-11-22

Bug 修复

- 回退：跨仓库合并请求无法正常工作 [#4890](#)

0.11.33 @ 2017-11-19

Bug 修复

- 部分安全修复
- 合并请求后发送的 Web 钩子推送内容包含错误的提交 ID [#4442](#)

- HTML 标签 go-import 未响应正确的值 [#4832](#)

新增特性

- 添加钉钉 Web 钩子支持 [#4773](#)
- 支持合并请求前先进行衍合操作 [#4798](#)

功能改进

- 在 LDAP BindDN 中支持使用 ‘%s’ 作为用户名占位符 [#2526](#)
- 允许通过环境变量指定 Docker 容器内 git 用户的 UID [#3520](#)
- 添加仓库设置以便在检查合并请求冲突时忽略空白符的差异 [#4834](#)

其它变更

- 添加新语种支持：斯洛伐克语

0.11.29 @ 2017-08-15

Bug 修复

- 如果仓库曾经为公开的，则变为私有后相关活动信息未被设为私有 [#4414](#)
- Web 钩子不接受 IPv6 URL [#4428](#)
- 通过代码提交关闭工单后没有发送邮件提醒 [#4430](#)
- 探索页面分页不正确 [#4441](#)
- `/api/v1/repos/search` 返回空值 [#4522](#)
- 创建合并请求完成后发生错误 [#4572](#)

0.11.19 @ 2017-06-10

Bug 修复

- 无法使用 go get 子包 [#1878](#)
- 非首次使用 LDAP 登录无法更新用户为管理员 [#2855](#)
- 使用 PAM 登录时发生错误 [#4216](#)
- PostgreSQL 恢复备份后出现错误 `unique constraint violation` [#4357](#)
- IPython notebook 的图片无法显示 [#4366](#)
- 编辑文件预览时无法正确处理图片相对链接 [#4368](#)
- 提交历史页面无法渲染 Emoji [#4439](#)
- 查看包含文件权限更改的单个提交时 CPU 异常高 [#4475](#)
- 无法修改协作者的权限 [#4512](#)

新增特性

- 支持两步验证登录 [#945](#)
- 支持 LDAP 登录时验证组成员身份 [#4398](#)

功能改进

- 安装页面检查 SMTP 地址中是否包含端口号 [#2243](#)
- 镜像仓库没有拉取新代码提交时不更新最后更新时间 [#4341](#)
- 支持 IPython Notebook 形式的 README [#4367](#)
- 支持自定义 TLS 相关配置 [#4450](#)

0.11.4 @ 2017-04-05

Bug 修复

- 在进行 HTTP/HTTPS 推送和拉取时客户端没有提示用户输入凭据
- 镜像用户凭据没有进行 URL 编码 [#4014](#)
- 仓库禁用合并请求时，发起合并请求按钮仍旧在分支页面显示 [#4377](#)
- 用户在安装页面发生验证错误时应用会崩溃 [#4383](#)

0.11 @ 2017-04-03

Bug 修复

- 编辑个人信息发生验证错误时丢失内容 [#1123](#)
- 组织控制面板显示错误的仓库统计 [#4351](#)
- 从 0.10 之前的版本自动迁移失败 [#4355](#)
- 允许公开访问工单的私有仓库没有正确处理匿名访问 [#4359](#)

0.11 RC @ 2017-03-27

Bug 修复

- 不正确的会话文件权限 [#3363](#)
- API：仓库对象的权限字段返回无效的值 [#4309](#)
- 非本地用户无法更新个人设置 [#4313](#)
- 工单索引不存在会显示随机工单 [#4315](#)
- 无法从二级派生仓库发起合并请求 [#4324](#)
- 无法更新保护分支白名单 [#4333](#)
- 派生、迁移和镜像仓库的体积没有正确更新 [#4336](#)

新增特性

- 支持私有仓库工单的公开访问 #649
- 支持私有仓库 Wiki 的公开访问 #2157
- 支持重新推送 Web 钩子的历史记录 #2187
- API: 同步镜像仓库 #2235
- 支持在控制面板中加载更多的活动内容 #2511
- 支持设置指定当前用户时候登录的 Cookie 值 #2885
- 支持备份及恢复 #2924
- 支持空仓库的工单和 Wiki 操作 #4104
- 提交历史页面自定义每页显示的数量 #4230
- 用户设置页面添加管理仓库面板 #4277

功能改进

- 探索页面显示私有但有访问权限的仓库 #3088
- 允许在安装页面选择使用终端模式的日志 #3119
- 允许配置 XORM 的日志器 #3183
- 添加有关 HTML 渲染模式的配置选项 #3608
- Web 钩子推送事件添加有关文件被添加、删除和修改的信息 #3897
- 允许设置控制面板每页显示的活动数量 #4247

更早的变更日志可以在 [GitHub](#) 上找到。

1.2 常见问题

- [常见问题](#)

- [部署应用](#)

- [如果我已经有其它应用使用端口 3000 了怎么办?](#)
- [如何使用 NGINX 的反向代理?](#)
 - [配置子路径](#)
 - [为什么上传大文件时总是发生错误?](#)
 - [如果我用的是 Apache 2 怎么配置子路径?](#)
 - [有没有用 lighttpd 配置子路径的例子?](#)
- [如何使用 HTTPS?](#)
- [如何使用离线模式?](#)
- [如何使用自定义 robots.txt?](#)
- [如何以守护进程形式运行?](#)
- [Systemd 服务](#)

- [管理权限](#)

- [如何成为管理员?](#)

- [仓库管理](#)

- [如何给予用户 Git 钩子权限?](#)

- [其它](#)

- [如何获取 Gogs 当前版本?](#)

常见问题

部署应用

[如果我已经有其它应用使用端口 3000 了怎么办?](#)

您可以在您第一次启动 Gogs 的时候通过以下方式修改默认的监听端口：

```
1. ./gogs web -port 3001
```

该方法同样会在您填写安装页面时生效，因此请选择一个您希望以后一直给 Gogs 使用的端口号。

如何使用 NGINX 的反向代理?

在 `nginx.conf` 文件中，将下面的 `server` 部分增加至 `http` 分区内并重载配置：

```
1. server {
2.     listen 80;
```

```

3.     server_name git.crystalnetwork.us;
4.
5.     location / {
6.         proxy_pass http://localhost:3000;
7.     }
8. }
```

配置子路径

如果您想要通过域名的子路径来访问 Gogs 实例，可以将 NGINX 的配置修改为以下形式（特别注意后缀 `/`）：

```

1. server {
2.     listen 80;
3.     server_name git.crystalnetwork.us;
4.
5.     location /gogs/ {
6.         proxy_pass http://localhost:3000/;
7.     }
8. }
```

然后在配置文件中设置 `[server] ROOT_URL = http://git.crystalnetwork.us/gogs/`。

为什么上传大文件时总是发生错误？

想要了解如何让 NGINX 支持大文件上传的讨论，可以查看 [此贴](#)。一般 NGINX 会返回 `413` 错误，在配置文件中加入以下内容可以解决该问题：

```
1. client_max_body_size 50m;
```

如果我用的是 Apache 2 怎么配置子路径？

可以尝试使用下面的配置模板：

```

1. <VirtualHost *:443>
2.     ...
3.     <Proxy *>
4.         Order allow,deny
5.         Allow from all
6.     </Proxy>
7.
8.     ProxyPass /git http://127.0.0.1:3000/
9.     ProxyPassReverse /git http://127.0.0.1:3000/
10.    </VirtualHost>
```

有没有用 lighttpd 配置子路径的例子？

可以尝试使用下面的配置模板：

```

1. server.modules += ( "mod_proxy_backend_http" )
2. $HTTP[url] =~ "^/gogs" {
3.     proxy-core.protocol = "http"
4.     proxy-core.backends = ( "localhost:3000" )
5.     proxy-core.rewrite-request = (
6.         "_uri" => ( "^/gogs/?(.*)" => "/$1" ),
7.         "Host" => ( ".*" => "localhost:3000" ),
8.     )
9. }
```

如何使用 HTTPS?

在 `custom/conf/app.ini` 文件中修改下列配置选项（以下仅为示例）：

```

1. [server]
2. PROTOCOL = https
3. ROOT_URL = https://try.gogs.io/
4. CERT_FILE = custom/https/cert.pem
5. KEY_FILE = custom/https/key.pem
```

如果您想要使用自签名的 HTTPS，则可以使用下列命令来生成所需文件（需要使用构建标签 `cert` 或直接从官方下载二进制）：

```
1. $ ./gogs cert -ca=true -duration=8760h0m0s -host=myhost.example.com
```

如何使用离线模式？

如果您需要将 Gogs 运行于内网环境下，只需将 `custom/conf/app.ini` 文件中的配置选项 `server -> OFFLINE_MODE` 修改为 `true` 即可。

如何使用自定义 robots.txt？

在 `custom` 目录下创建 `robots.txt` 文件即可。

如何以守护进程形式运行？

Gogs 拥有一些由第三方提供的脚本来支持以守护进程形式运行：

- [init.d/centos](#)
- [init.d/debian](#)
- 下小节中的 Systemd 服务

Systemd 服务

在 GitHub 上的 Gogs 仓库有一个 [systemd 服务模版文件](#)，您需要做出一定的修改才能够使用它：

1. 更新 `User`、`Group`、`WorkingDirectory`、`ExecStart` 和 `Environment` 为相对应的值。其中 `WorkingDirectory` 为您的 Gogs 实际安装路径根目录。
2. [可选] 如果您 Gogs 安装示例使用 `MySQL/MariaDB`、`PostgreSQL`、`Redis` 或 `memcached`，请去掉相应 `After` 属性的注释。

当您完成修改后，请将文件保存至 `/etc/systemd/system/gogs.service`，然后通过 `sudo systemctl enable gogs` 命令激活，最后执行 `sudo systemctl start gogs`。

您可以通过 `sudo systemctl status gogs -l` 或 `sudo journalctl -b -u gogs` 命令检查 Gogs 的运行状态。

管理权限

如何成为管理员？

1. 当用户注册时 `ID = 1` 则会自动成为管理员，无需邮箱验证。
2. 默认管理员登录 [Admin -> Users](#) 面板并设置其它人员为管理员。
3. 通过安装页面注册的用户会自动成为管理员。

仓库管理

如何给予用户 Git 钩子权限？

由于这是一个可能损坏您系统的高级权限，您必须在用户管理面板（`/admin/users/:userid`）为您完全信任的用户单独开启。

其它

如何获取 Gogs 当前版本？

纯文本形式的 Gogs 版本存储在文件 `templates/.VERSION` 中。

1.3 故障排查

- 故障排查
 - 安装
 - SSH
 - SSH 链接挂起 60 秒
 - Update 钩子指向错误的二进制路径
 - 推送成功但控制面板没有活动更新
 - 可能原因
 - 解决方案
- Git
 - 公钥使用冲突
 - Gogs 无法调用 ‘git’ 命令
- Cache
- 数据库
 - 使用错误的 MySQL 存储引擎
 - 过时的 MySQL 密码设置
 - 连接到错误的 SQLite3 数据库
- 邮件服务
- Windows
- 其它

故障排查

安装

- 错误描述: `../gosrc/src/github.com/gogits/gogs/cmd/cert.go:79: undefined: elliptic.P224`
- 可能原因: Go 语言的 RHEL/CentOS 官方发行版中因专利问题移除了相关加密算法的代码
- 解决方案: 从 golang.org/dl 下载安装 Go 语言

SSH

SSH 链接挂起 60 秒

- 可能原因: Gogs 会在每次 SSH Push 完成之后请求自身的 Web 服务, 您的防火墙或服务器提供商未允许该操作

Update 钩子指向错误的二进制路径

- 可能原因：您升级 Gogs 后将其移动到了和之前安装位置不同的目录
- 解决方案：到管理员控制面板（/admin）执行以下操作：
 - 重新生成 '.ssh/authorized_keys' 文件
 - 重新同步所有仓库的 pre-receive、update 和 post-receive 钩子

推送成功但控制面板没有活动更新

可能原因

在使用 SSH 推送时，Gogs 依赖通过执行钩子脚本（Hook Script）来更新仓库和最近活动。但出于某种原因，执行脚本的权限被拒绝了，尤其是在使用挂载（Mount）设备时。

解决方案

- 请确保挂载设备时没有启用 noexec 选项，必要时需要明确指定 exec 选项。
- 如果使用 vfat （或者 cifs）类型的挂载，请确保 uid、gid 和 fmask 选项允许运行 Gogs 的用户在挂载设备上执行脚本。
- 如果使用网络类型的挂载，请确保服务器（NFS 或 Samba）没有禁止远程文件系统的执行脚本权限。

Git

公钥使用冲突

- 错误描述：
 - fatal: 'XX/XX.git' does not appear to be a git repository
 - 推送代码提交后依旧显示为空仓库
- 可能原因：~/.ssh/authorized_keys 文件中存在重复的 SSH 密钥，可能是由于您曾经或正在通过同一个系统用户使用 GitLab。
- 解决方案：删除除了属于 Gogs 自动添加以外的所有密钥。

Gogs 无法调用 ‘git’ 命令

- 错误描述：repo.NewRepoContext(fail to set git user.email):
- 可能原因：该错误会发生在 Windows 安装 Git Bash 时未启用 cmd 选项。
- 解决方案：重装并启用 cmd 选项。

Cache

- 错误描述：cache: unknown adaptername "memcache" (forgotten import?)
- 可能原因：为了减少不必要的导入，您需要使用构建 tags 来启用某个缓存适配器。
- 解决方案：

- 下载: `go get -tags memcache github.com/gogits/gogs`
- 构建: `go build -tags memcache`
- 如果要启用 `redis` 也是一样的步骤。

数据库

使用错误的 MySQL 存储引擎

- 错误描述: `Error 1071: Specified key was too long; max key length is 1000 bytes`
- 可能原因: 这是由于数据库引擎为 MyISAM 导致的。
- 解决方案: 在使用 `config/mysql.sql` 创建完数据库后, 进入数据库然后执行:

```
1. use gogs;
2. set global storage_engine=INNODB;
```

最后, 访问 <http://localhost:3000/install> 即可 (感谢 [@linc01n](#))。

过时的 MySQL 密码设置

- 错误描述: `Database setting is not correct: This server only supports the insecure old password authentication. If you still want to use it, please add 'allowOldPasswords=1' to your DSN. See also https://github.com/go-sql-driver/mysql/wiki/old_passwords`
- 可能原因: 只更新了 [@localhost](#) 的密码, 但 @% 用户依旧使用旧密码
- 解决方案: [GitHub 讨论](#)

连接到错误的 SQLite3 数据库

- 错误描述: 推送仓库显示仓库所有者 (Owner) 未注册或 `user` 数据表不存在。
- 可能原因: 您以服务形式启动 Gogs 的时候, Gogs 可能会使用和您预期所不同的 SQLite3 数据库文件。
- 解决方案: 使用绝对路径配置 SQLite3 数据库文件路径。

邮件服务

- 错误描述: Gmail 发送返回 Error 534: `Please log in via your web browser and then try again`
- 可能原因: 这是因为 Google 不信任您的服务器导致的。
- 解决方案:
 - 访问 <https://accounts.google.com> 并登录。
 - 访问 <https://accounts.google.com/DisplayUnlockCaptcha> 单击 `continue`。
 - 重试发送。

Windows

- 错误描述：

```

1. 2014/09/18 15:04:40 [repo.go:115 CreatePost()] [E] CreatePost: initRepository: initRepository(git clone):
    cygwin warning:
2. MS-DOS style path detected: C:\Users\user\gogs-repositories\unos\test3.git/.git
3. Preferred POSIX equivalent is: /cygdrive/c/Users/user/gogs-repositories/unos/test3.git/.git
4. CYGWIN environment variable option "nodosfilewarning" turns off this warning.
5. Consult the user's guide for more details about POSIX paths:
6. http://cygwin.com/cygwin-ug-net/using.html#using-pathnames
7. Cloning into 'C:\Users\user\AppData\Local\Temp\484264900'...
8. fatal: '/cygdrive/d/svnroot/research/gogs/C:\Users\user\gogs-repositories\unos\test3.git' does not appear
   to be a git repository
9. fatal: Could not read from remote repository.

```

- 可能原因：您在系统中安装了另一种命令行工具，并且与系统默认的 CMD 有不同路径风格。
- 解决方案：请您尝试用默认的 CMD 启动 Gogs。

- 错误描述：Resource interpreted as Stylesheet but transferred with MIME type application/x-css
- 可能原因：Windows 注册表中错误的值
- 解决方案：在注册表中找到 HKEY_CLASSES_ROOT 下的 .css，并将其 Content Type 的值修改为 text/css。

其它

- 错误描述：页面响应非常慢，但是显示在页面底部的时间正常（低于 100 毫秒）。
- 可能原因：这可能是由于 Nginx 尝试通过 IPv6 的方式解析 IPv4 的地址。
- 解决方案：显式的使用 127.0.0.1 作为主机名而不是 localhost。

- 错误描述：Error 1062: Duplicate entry 'Unknown-Mac' for key 'UQE_public_key_name'
- 可能原因：这是由于遗留代码导致的，public_key 表之前含有唯一索引 UQE_public_key_name。
- 解决方案：您只需删除它即可。

- 错误描述：GLIBC_2.14 not found
- 解决方案：尝试 sudo apt-get -t testing install libc6-dev。

- 错误描述：[Macaron] PANIC: session(start): mkdir data: permission denied
- 可能原因：Gogs 会在程序所在目录创建 data 子目录。
- 解决方案：确保 Gogs 具有在相同目录创建子目录的权限。

2. 下载安装

- [下载安装](#)
 - [环境要求](#)
 - [安装数据库](#)
 - [安装其它要求](#)
 - [Mac OS X](#)
 - [Debian/Ubuntu](#)
 - [Windows](#)
 - [安装 Gogs](#)

下载安装

环境要求

- 数据库（选择以下一项）：
 - [MySQL](#): 版本 $\geq 5.5.3$
 - [PostgreSQL](#)
 - [MSSQL](#)
 - [TiDB](#) (实验性支持，使用 MySQL 协议连接)
 - 或者 [什么都不安装](#) 直接使用 SQLite3
- [git](#) (bash)：
 - 服务端和客户端均需版本 $\geq 1.7.1$
 - Windows 系统建议使用最新版
- SSH 服务器：
 - 如果您只使用 **HTTP/HTTPS** 的话请忽略此项
 - 如果您选择在 Windows 系统使用内置 SSH 服务器，请确保添加 [ssh-keygen](#) 到您的 [%PATH%](#) 环境变量中
 - 推荐 Windows 系统使用 [Cygwin OpenSSH](#) 或 [Copssh](#)

安装数据库

请根据您的选择进行安装：

- [MySQL](#) (引擎: INNODB)
- [PostgreSQL](#)

注意事项 您可以使用 [etc/mysql.sql](#) 来自动创建名为 [gogs](#) 的数据库。如果您选择手动创建，请务必将编码设置为 [utf8mb4](#)。

安装其它要求

Mac OS X

假设您已经安装 [Homebrew](#):

```
1. $ brew update  
2. $ brew install git
```

Debian/Ubuntu

```
1. $ sudo apt-get update  
2. $ sudo apt-get install git
```

Windows

[下载并安装 Git](#)

安装 Gogs

- [二进制安装](#)
- [源码安装](#)

2.1 注册 Windows 服务

- [注册为 Windows 服务运行](#)
 - [前提要求](#)
 - [Use Builtin Functionality](#)
 - [Use NSSM](#)

注册为 Windows 服务运行

前提要求

想要使 Gogs 通过 Windows 服务的方式运行，必须满足以下两个条件：

1. 使用 `miniwinsvc` 构建标签获得内置 Windows 服务支持。
2. 不使用 `miniwinsvc` 构建标签并通过 [NSSM](#) 注册为服务。

在注册成为服务之前，需要确保给予 Gogs 二进制相应目录的读写权限，包括存放仓库的根目录
`([repository] ROOT)`。

修改 `C:\Gogs\custom\conf\app.ini` 文件的相应信息：

```
1. RUN_USER = COMPUTERNAME$
```

通过上面的配置将 Gogs 的运行用户设置为本地系统用户。`COMPUTERNAME` 的值可以通过命令 `echo %COMPUTERNAME%` 获得，如果该命令的返回值为 `USER-PC` 则使用 `RUN_USER = USER-PC$`：

```
1. [server]
2. DOMAIN = gogs
3. PROTOCOL = http
4. HTTP_ADDR = 127.0.1.1
5. HTTP_PORT = 80
6. OFFLINE_MODE = true
7. ROOT_URL = http://gogs/
```

Moves Gogs' listen port to 80 on the local interface subnet, and tells the Gogs HTTPd that its virtual host is the "gogs" domain.

This lets you forgo including a port number when browsing, and prevents collision with other localhost services.

The IP address can be anything in the range `127.0.0.2 - 127.254.254.254` , so long as it's unique to Gogs.

To complete that network route, open Notepad.exe as administrator and include the following in

```
C:\Windows\System32\drivers\etc\hosts :
```

```
1. # Gogs local HTTPd
2. 127.0.1.1      gogs
```

The hosts file entry will guarantee that all requests to the “gogs” domain are captured by your localhost interface.

In a web browser, generally, `gogs/` in the address bar is sufficient to trigger that route.

Use Builtin Functionality

Open a command prompt (`cmd.exe`) as an Administrator. Run the following command:

```
1. C:\> sc create gogs start= auto binPath= "\"C:\gogs\gogs.exe\" web --config
\"C:\gogs\custom\conf\app.ini\""
```

Ensure there is a space after each `=`. You can choose to add additional Arguments to further modify your service, or modify it manually in the service management console.

To start the service run following command:

```
1. C:\> net start gogs
```

You should see following output:

```
1. The gogs service is starting.
2. The gogs service was started successfully.
```

Use NSSM

Get the `nssm.exe` needed for your machine (32 or 64 bit; they’re packaged together in Iain’s zip file), and place it in a directory that is in (or will be added to) your `%PATH%` environment variable.

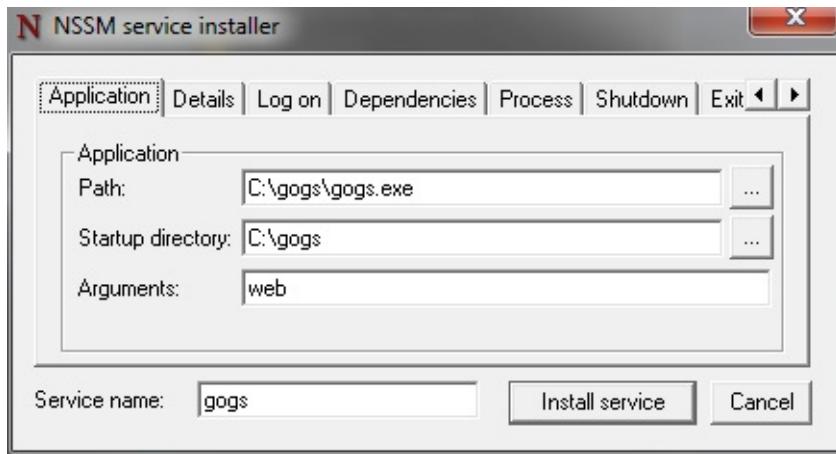
Open a command line as administrator and do following command to configure Gogs as a service:

```
1. C:\>nssm install gogs
```

“NSSM service installer” will appear. Configure it as follows:

Application tab:

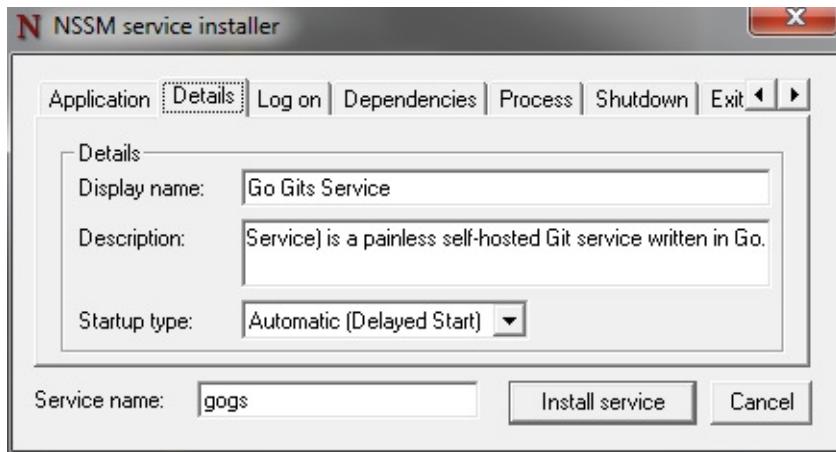
- Path: C:\gogs\gogs.exe
- Startup directory: C:\gogs
- Arguments: web



Details tab:

- Display name: Gogs
- Description: A painless self-hosted Git service.
- Startup type: Automatic (Delayed Start)

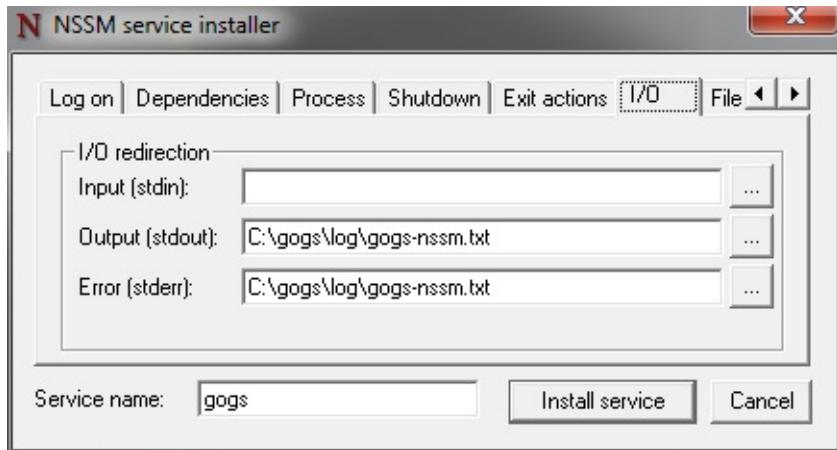
Note that we've chosen [delayed start](#), so that the service will not impact the early boot time. Gogs will start two minutes after the non-delayed services.



I/O tab:

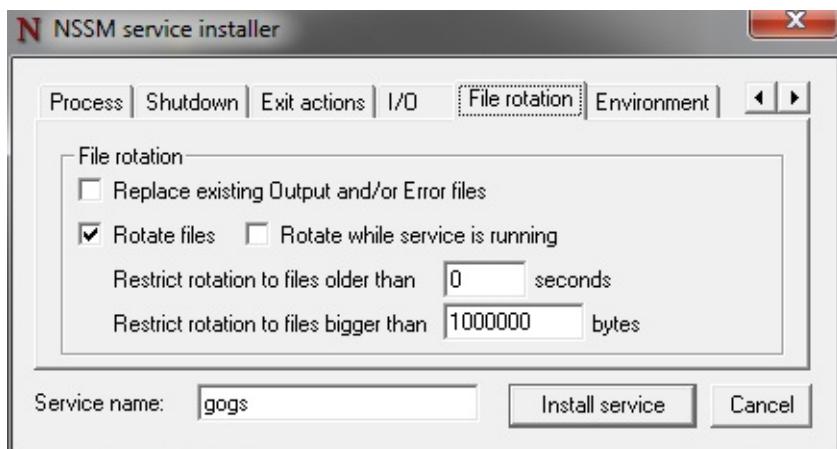
- Output (stdout): C:\gogs\log\gogs-nssm.txt
- Error (stderr): C:\gogs\log\gogs-nssm.txt

That will capture all text output that you would normally receive from Gogs on the command line console, and log it to that file instead.



File rotation tab:

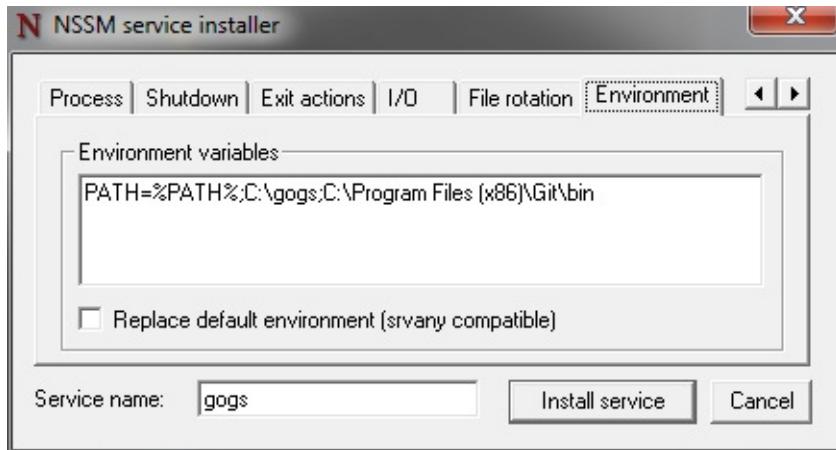
- Check: Rotate files
- Restrict rotation to files bigger than: 1000000 bytes



Environment tab:

- Environment variables: PATH=%PATH%;C:\gogs;C:\Program Files (x86)\Git\bin

That is a guarantee that both `gogs.exe` and `git.exe` will be on the Gogs service's path variable during runtime.



Click “Install service”, and you should be confirmed that it succeeded.

If it failed, refer back to the command line console you started, for the error message. When it succeeds, go to command line and do:

```
1. nssm start gogs
```

You should see:

```
1. gogs: START: The operation completed successfully.
```

Check that this is true by opening `C:\gogs\log\gogs-nssm.txt`. You should see Gogs’ start up procedures, ending with:

```
1. timestamp [I] Run Mode: Production
2. timestamp [I] Listen: http://127.0.1.1:80
```

Now point your web browser at `http://gogs/` and log in.

Gogs is running as a service, and will not need to be run manually, unless something goes wrong. NSSM will attempt to restart the service for you, if the Gogs server crashes.

When you need to restart it after `app.ini` changes, go to an administrator command line after the changes, and do:

```
1. nssm restart gogs
```

See the [configuration cheat sheet](#) for reference of the `custom\conf\app.ini` settings.

An example of the logging and error handling in action:

```

1. [log]
2. ROOT_PATH = C:\gogs\log

```

At the time of writing this line, it will cause the following to print in `C:\gogs\log\gogs-nssm.txt` :

```

1. timestamp [T] Custom path: C:/gogs/custom
2. timestamp [T] Log path: C:\gogs\log
3. timestamp [I] Gogs x.y.z
4. timestamp [log.go:294 Error()] [E] Fail to set logger(file): invalid character 'g' in string escape code

```

This is what was expected in `C:\custom\conf\app.ini` :

```
1. ROOT_PATH = C:/gogs/log
```

And this was the `nssm` interaction while solving it:

```

1. C:\>nssm restart gogs
2. gogs: STOP: The operation completed successfully.
3. gogs: Unexpected status SERVICE_PAUSED in response to START control.
4.
5. C:\>nssm start gogs
6. gogs: START: An instance of the service is already running.
7.
8. C:\>nssm stop gogs
9. gogs: STOP: The operation completed successfully.
10.
11. C:\>nssm start gogs
12. gogs: Unexpected status SERVICE_PAUSED in response to START control.
13.
14. C:\>nssm restart gogs
15. gogs: STOP: The operation completed successfully.
16. gogs: START: The operation completed successfully.

```

2.2 配置与运行

- 配置与运行
 - 配置文件
 - 默认配置文件
 - 自定义配置文件
 - 为什么要这么做?
 - 运行 Gogs 服务
 - 开发者模式
 - 部署模式

配置与运行

配置文件

默认配置文件

默认配置都保存在 `conf/app.ini`，您 永远不需要 编辑它。该文件从 `v0.6.0` 版本开始被嵌入到二进制中。

自定义配置文件

那么，在不允许修改默认配置文件 `conf/app.ini` 的情况下，怎么才能自定义配置呢？很简单，只要创建 `custom/conf/app.ini` 就可以！在 `custom/conf/app.ini` 文件中修改相应选项的值即可。

例如，需要改变仓库根目录的路径：

```
1. [repository]  
2. ROOT = /home/jiahuchen/gogs-repositories
```

当然，您也可以修改数据库配置：

```
1. [database]  
2. PASSWD = root
```

为什么要这么做？

乍一看，这么做有些复杂，但是这么做可以有效地保护您的自定义配置不被破坏：

- 从二进制安装的用户，可以直接替换二进制及其它文件而不至于重新编写自定义配置。
- 从源码安装的用户，可以避免由于版本管理系统导致的文件修改冲突。

运行 Gogs 服务

开发者模式

- 您需要在 `custom/conf/app.ini` 文件中将选项 `security -> INSTALL_LOCK` 的值设置为 `true`。
- 您可以使用超能的 `make` 命令：

```
1. $ make
2. $ ./gogs web
```

部署模式

脚本均放置在 `scripts` 目录，但请在仓库根目录执行它们

- Gogs 支持多种方式的启动：
 - 普通：只需执行 `./gogs web`
 - 守护进程：详见 `scripts` 文件夹
- 然后访问 `/install` 来完成首次运行的配置工作

2.3 二进制安装

- [二进制安装](#)
 - [备注](#)
 - [如何使用下载好的压缩包？](#)
 - [如何通过二进制升级？](#)
 - [0.11.34 @ 2017-11-22](#)
 - [0.11.29 @ 2017-08-15](#)
 - [0.10.18 @ 2017-03-14](#)
 - [v0.9.141 @ 2017-02-11](#)

二进制安装

目前只提供最近发布的小版本二进制下载，更多版本下载请前往 [GitHub](#) 查看。

所有的版本都支持 **MySQL**、**PostgreSQL**、**MSSQL** 和 **TiDB**（使用 MySQL 协议）作为数据库，并且均使用构建标签（build tags）`cert` 进行构建。需要注意的是，不同的版本的支持状态有所不同，请根据实际的 Gogs 提示进行操作。

备注

- `mws` 表示提供内置 Windows 服务支持，如果您使用 NSSM 请使用另外一个版本。

如何使用下载好的压缩包？

1. 解压压缩包。
2. 使用命令 `cd` 进入到刚刚创建的目录。
3. 执行命令 `./gogs web`，然后，就没有然后了。

如何通过二进制升级？

1. 下载最新版的压缩包。
2. 删除当前的 `templates` 目录。
3. 解压压缩包并将所有内容复制粘贴到相应（当前）的位置。

0.11.34 @ 2017-11-22

系统名称	系统类型	SQLite	PAM	下载
------	------	--------	-----	----

系统名称	系统类型	SQLite	PAM	下载 (GitHub)		
Linux	386			HTTPS: ZIP \	TAR.GZ - CDN: ZIP \	TAR.GZ
Linux	amd64			HTTPS: ZIP \	TAR.GZ - CDN: ZIP \	TAR.GZ
Linux	armv5			HTTPS: ZIP - CDN: ZIP		
Raspberry Pi	v2&v3 / armv6			HTTPS: ZIP - CDN: ZIP		
Windows	386			HTTPS: ZIP \	ZIP w/ mws - CDN: ZIP \	ZIP w/mws
Windows	amd64			HTTPS: ZIP \	ZIP w/ mws - CDN: ZIP \	ZIP w/ mws
Mac OS	amd64			HTTPS: ZIP - CDN: ZIP		

0.11.29 @ 2017-08-15

系统名称	系统类型	SQLite	PAM	下载 (GitHub)		
Linux	386			HTTPS: ZIP \	TAR.GZ - CDN: ZIP \	TAR.GZ
Linux	amd64			HTTPS: ZIP \	TAR.GZ - CDN: ZIP \	TAR.GZ
Linux	armv5			HTTPS: ZIP - CDN: ZIP		
Raspberry Pi	v2 / armv6			HTTPS: ZIP - CDN: ZIP		
Windows	386			HTTPS: ZIP \	ZIP w/ mws - CDN: ZIP \	ZIP w/mws
Windows	amd64			HTTPS: ZIP \	ZIP w/ mws - CDN: ZIP \	ZIP w/ mws
Mac OS	amd64			HTTPS: ZIP - CDN: ZIP		

0.10.18 @ 2017-03-14

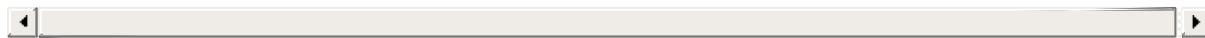
系统名称	系统类型	SQLite	PAM	下载 (GitHub)		
Linux	386			HTTPS: ZIP \	TAR.GZ - CDN: ZIP \	TAR.GZ
Linux	amd64			HTTPS: ZIP \	TAR.GZ - CDN: ZIP \	TAR.GZ
Linux	armv5			HTTPS: ZIP - CDN: ZIP		

Pi	armv6			CDN: ZIP		
Windows	386			HTTPS: ZIP \	ZIP w/ mws - CDN: ZIP \	ZIP w/mws
Windows	amd64			HTTPS: ZIP \	ZIP w/ mws - CDN: ZIP \	ZIP w/ mws
Mac OS	amd64			HTTPS: ZIP - CDN: ZIP		

v0.9.141 @ 2017-02-11

系统名称	系统类型	SQLite	PAM	下载 (GitHub)		
Linux	386			HTTPS: ZIP \	TAR.GZ - CDN: ZIP \	TAR.GZ
Linux	amd64			HTTPS: ZIP \	TAR.GZ - CDN: ZIP \	TAR.GZ
Linux	armv5			HTTPS: ZIP - CDN: ZIP		
Linux	armv6			HTTPS: ZIP - CDN: ZIP		
Raspberry Pi	v2			HTTPS: ZIP - CDN: ZIP		
Windows	386			HTTPS: ZIP \	ZIP w/ mws - CDN: ZIP \	ZIP w/mws
Windows	amd64			HTTPS: ZIP \	ZIP w/ mws - CDN: ZIP \	ZIP w/ mws
Mac OS	amd64			HTTPS: ZIP - CDN: ZIP		

安装完成后可继续参照 [配置与运行]。



2.4 包管理安装

- 包管理安装
 - [Packager.io](#)
 - [Arch Linux](#)

包管理安装

以下均为第三方提供，后果自负！

Packager.io

- 当前支持 Ubuntu 12.04 + 14.04、CentOS 6 + 7 和 Debian 7 + 8 版本。
- 可以从 [packager.io](#) 获取到相应包管理资源（该工具的自定义配置文件有时会放置在 `/etc/default/gogs`）。

Arch Linux

- 基于源码构建，作者 [@fanningert](#):
 - `master` 分支: [AUR GitHub](#)
 - `develop` 分支: [AUR GitHub](#)
 - 版本: [AUR GitHub](#)

能可以到 [Arch Linux Wiki entry](#) 查看完整说明。

2.5 源码安装

- 源码安装
 - 安装依赖
 - 基本依赖
 - 第三方包
 - 安装 Go 语言
 - 下载
 - 设置环境
 - 安装 Gogs
 - 构建 `develop` 分支版本
 - 测试安装
 - 使用标签构建

源码安装

安装依赖

基本依赖

- Go 语言：版本 ≥ 1.6

第三方包

请通过 [gopmfile](#) 查看完整的第三方包依赖列表。一般情况下只有在您为 Gogs 制作构建包的时候才会用到。

安装 Go 语言

如果您的系统已经安装要求版本的 Go 语言，可以跳过此小节。

下载

您可以通过以下方式安装 Go 语言到 `/home/git/local/go` 目录：

```
1. sudo su - git
2. cd ~
3. # create a folder to install 'go'
```

```

4. mkdir local
5. # Download go (change go$VERSION.$OS-$ARCH.tar.gz to the latest release)
6. wget https://storage.googleapis.com/golang/go$VERSION.$OS-$ARCH.tar.gz
7. # expand it to ~/local
8. tar -C /home/git/local -xzf go$VERSION.$OS-$ARCH.tar.gz

```

设置环境

请设置和您系统环境对应的路径：

```

1. sudo su -
2. cd ~
3. echo 'export GOROOT=$HOME/local/go' >> $HOME/.bashrc
4. echo 'export GOPATH=$HOME/go' >> $HOME/.bashrc
5. echo 'export PATH=$PATH:$GOROOT/bin:$GOPATH/bin' >> $HOME/.bashrc
6. source $HOME/.bashrc

```

安装 Gogs

常用的安装方式：

```

1. # 下载并安装依赖
2. $ go get -u github.com/gogits/gogs
3.
4. # 构建主程序
5. $ cd $GOPATH/src/github.com/gogits/gogs
6. $ go build

```

构建 `develop` 分支版本

如果您想要安装 `develop` (或其它) 分支版本，则可以通过以下命令：

```

1. $ mkdir -p $GOPATH/src/github.com/gogits
2. $ cd $GOPATH/src/github.com/gogits
3.
4. # 请确保没有使用 "https://github.com/gogits/gogs.git"
5. $ git clone --depth=1 -b develop https://github.com/gogits/gogs
6. $ cd gogs
7. $ go build

```

测试安装

您可以通过以下方式检查 Gogs 是否可以正常工作：

```
1. cd $GOPATH/src/github.com/gogits/gogs  
2. ./gogs web
```

如果您没有发现任何错误信息，则可以使用 `Ctrl-C` 来终止运行。

使用标签构建

Gogs 默认并没有支持一些功能，这些功能需要在构建时明确使用构建标签（[build tags](#)）来支持。

目前使用标签构建的功能如下：

- `sqlite3`：SQLite3 数据库支持
- `pam`：PAM 授权认证支持
- `cert`：生成自定义证书支持
- `miniwinsvc`：Windows 服务内置支持（或者您可以使用 NSSM 来创建服务）

例如，您需要支持以上所有功能，则需要先删除 `$GOPATH/pkg/{GOOS_GOARCH}/github.com/gogits/gogs` 目录，然后执行以下命令：

```
1. $ go get -u -tags "sqlite pam cert" github.com/gogits/gogs  
2. $ cd $GOPATH/src/github.com/gogits/gogs  
3. $ go build -tags "sqlite pam cert"
```

安装完成后可继续参照 [配置与运行]。

3. 版本升级

- [版本升级](#)

版本升级

3.1 二进制升级

- [从二进制升级](#)

从二进制升级

相关下载可以从 [二进制安装](#) 页面查看。

首先，确认当前安装的位置：

```
1. # 默认位置在 git 用户下的家目录
2. $ sudo su - git
3. $ cd ~
4. $ pwd
5. /home/git
6. $ ls
7. gogs gogs-repositories
```

然后将当前目录移动到另一个临时的位置，但不是删除！

```
1. $ mv gogs gogs_old
```

下载并解压新的二进制：

```
1. # 请根据系统和类型获取相应的二进制版本
2. $ wget https://dl.gogs.io/$VERSION/$OS_$ARCH.tar.gz
3. $ tar -zxf $OS_$ARCH.tar.gz
4. $ ls
5. gogs gogs_old gogs-repositories $OS_$ARCH.tar.gz
```

复制 `custom`、`data` 和 `log` 目录到新解压的目录中：

```
1. $ cp -R gogs_old/custom gogs
2. $ cp -R gogs_old/data gogs
3. $ cp -R gogs_old/log gogs
```

最后，运行并打开浏览器进行测试：

```
1. $ cd gogs
2. $ ./gogs web
```

给自己点一个赞！

3.2 源码升级

- [从源码升级](#)

从源码升级

升级 Gogs 的一般步骤：

```
1. # 更新源码以及依赖
2. $ go get -u github.com/gogits/gogs
3.
4. $ cd $GOPATH/src/github.com/gogits/gogs
5.
6. # 移除旧的二进制
7. $ rm gogs
8.
9. # 或将旧的二进制进行备份
10. $ mv gogs gogs.$(date +%Y-%m-%d).old
11.
12. # 重新构建 Gogs
13. $ go build
```

其它操作：

- 构建 `develop` 分支版本
- 测试安装
- 使用标签构建

4. 功能介绍

- [功能介绍](#)

功能介绍

4.1 自定义模板

- [自定义模板](#)
 - [添加自定义 CSS 文件](#)

自定义模板

您可以不需要修改仓库源码就能注入自定义头部和尾部内容，这对添加分析代码和自定义静态资源非常有用。

了解更多有关 [注入自定义头部和尾部](#)。

添加自定义 CSS 文件

这里展示如何为您的 Gogs 实例添加自定义 CSS 文件，目录和文件名都是为了方便演示，您可以把文件放在任何能够通过网络访问的目录。

1. 在 `public/css` 目录下创建一个名为 `custom.css` 的文件
2. 向文件中添加一些 CSS 规则
3. 编辑 `custom/templates/inject/head tmpl` 文件并添加一行内容 `<link rel="stylesheet" href="/css/custom.css">`
4. 重启 Gogs
5. 后续对自定义 CSS 文件的编辑不需要重启 Gogs

4.2 Web 钩子

- Web 钩子
 - 事件信息

Web 钩子

Gogs 支持针对仓库事件的 Web 钩子服务，您可以在仓库的设置相关页面中找到（`/:username/:reponame/settings/hooks`）。所有的事件推送均为 POST 请求，目前支持 Gogs 和 Slack 两种格式的内容。

事件信息

以下为 Gogs 向 Payload URL 发送的事件信息示例：

1. X-Gogs-Delivery: f6266f16-1bf3-46a5-9ea4-602e06ead473
2. X-Gogs-Event: push
3. X-Gogs-Signature: 1921679ed6274399b6514721056337f6913b6ff1cb35a24d340e983745d637f1

```
1. {
2.     "ref": "refs/heads/develop",
3.     "before": "28e1879d029cb852e4844d9c718537df08844e03",
4.     "after": "bffeb74224043ba2feb48d137756c8a9331c449a",
5.     "compare_url":
6.         "http://localhost:3000/unknwon/webhooks/compare/28e1879d029cb852e4844d9c718537df08844e03...bffeb74224043ba2",
7.     "commits": [
8.         {
9.             "id": "bffeb74224043ba2feb48d137756c8a9331c449a",
10.            "message": "!@#%\\u003e\\u003e\\u003e\\u003e\\u003c\\u003c\\u003c\\u003e\\u003e\\u003e\\u003e\\n",
11.            "url": "http://localhost:3000/unknwon/webhooks/commit/bffeb74224043ba2feb48d137756c8a9331c449a",
12.            "author": {
13.                "name": "Unknwon",
14.                "email": "u@gogs.io",
15.                "username": "unknwon"
16.            },
17.            "committer": {
18.                "name": "Unknwon",
19.                "email": "u@gogs.io",
20.                "username": "unknwon"
21.            },
22.            "timestamp": "2017-03-13T13:52:11-04:00"
23.        }
24.    ],
25. }
```

```
24. "repository": {
25.     "id": 140,
26.     "owner": {
27.         "id": 1,
28.         "login": "unknwon",
29.         "full_name": "Unknwon",
30.         "email": "u@gogs.io",
31.         "avatar_url": "https://secure.gravatar.com/avatar/d8b2871cdac01b57bbda23716cc03b96",
32.         "username": "unknwon"
33.     },
34.     "name": "webhooks",
35.     "full_name": "unknwon/webhooks",
36.     "description": "",
37.     "private": false,
38.     "fork": false,
39.     "html_url": "http://localhost:3000/unknwon/webhooks",
40.     "ssh_url": "ssh://unknwon@localhost:2222/unknwon/webhooks.git",
41.     "clone_url": "http://localhost:3000/unknwon/webhooks.git",
42.     "website": "",
43.     "stars_count": 0,
44.     "forks_count": 1,
45.     "watchers_count": 1,
46.     "open_issues_count": 7,
47.     "default_branch": "master",
48.     "created_at": "2017-02-26T04:29:06-05:00",
49.     "updated_at": "2017-03-13T13:51:58-04:00"
50. },
51. "pusher": {
52.     "id": 1,
53.     "login": "unknwon",
54.     "full_name": "Unknwon",
55.     "email": "u@gogs.io",
56.     "avatar_url": "https://secure.gravatar.com/avatar/d8b2871cdac01b57bbda23716cc03b96",
57.     "username": "unknwon"
58. },
59. "sender": {
60.     "id": 1,
61.     "login": "unknwon",
62.     "full_name": "Unknwon",
63.     "email": "u@gogs.io",
64.     "avatar_url": "https://secure.gravatar.com/avatar/d8b2871cdac01b57bbda23716cc03b96",
65.     "username": "unknwon"
66. }
67. }
```


4.3 多国语言支持

- 本地化 Gogs 应用
 - 贡献翻译
 - 自定义本地化文件

本地化 Gogs 应用

Gogs 项目从 `v0.5.0` 版本开始支持应用的本地化，而您只需要鼠标单击就可实现即时语言切换。

在 `custom/conf/app.ini` 文件中启用它们（默认启用全部）：

```
1. [i18n]
2. LANGS = en-US,zh-CN,zh-HK,de-DE,fr-FR,nl-NL,lv-LV,ru-RU,ja-JP,es-ES,pt-BR,pl-PL,bg-BG,it-IT
3. NAMES = English,简体中文,繁體中文,Deutsch,Français,Nederlands,Latviešu,Русский,日本語,Español,Português do Brasil,Polski,български,Italiano
```

贡献翻译

到 [Crowdin](#) 注册一个用户然后填补未翻译的字段。

有时候您会发现您无法将一个句子准确地从英文翻译到您的语言，没关系，这很正常！只需要使用您的语言将中心思想表达出来就可以，而不需要斤斤计较。

如果您想要测试翻译好的本地化文件，但又不想涉及到 Git 历史的变动，您可以将本地化文件放至 `custom/conf/locale/<file>` 然后重启 Gogs。

自定义本地化文件

如果您对官方的本地化翻译不够满意，可以修改其中的部分字段并保存到 `custom/conf/locale_locale_<lang>.ini` 中，然后重启 Gogs。

4.4 授权认证

- [授权认证](#)
 - [LDAP](#)
 - [PAM](#)
 - [SMTP](#)
 - [Freeipa](#)

授权认证

LDAP

基于 BindDN 和 simple auth 的 LDAP 授权方式共享以下字段：

- 认证名称（必填）
 - A name to assign to the new method of authorization.
- 主机地址（必填）
 - The address where the LDAP server can be reached.
 - Example: `mydomain.com`
- 主机端口（必填）
 - The port to use when connecting to the server.
 - Example: `636`
- 启用 TLS 加密（可选）
 - Whether to use TLS when connecting to the LDAP server.
- 管理员过滤规则（可选）
 - An LDAP filter specifying if a user should be given administrator privileges. If a user accounts passes the filter, the user will be privileged as an administrator.
 - Example: `(objectClass=adminAccount)`
- 用户名属性（可选）
 - The attribute of the user's LDAP record containing the user name. Given attribute value will be used for new Gogs account user name after first

successful sign-in. Leave empty to use login name given on sign-in form.

- This is useful when supplied login name is matched against multiple attributes, but only single specific attribute should be used for Gogs account name, see “User Filter”.
- 例如: `uid`
- 名字属性 (可选)
 - The attribute of the user’s LDAP record containing the user’s first name. This will be used to populate their account information.
 - Example: `givenName`
- 姓氏属性 (可选)
 - The attribute of the user’s LDAP record containing the user’s surname. This will be used to populate their account information.
 - Example: `sn`
- 邮箱属性 (必填)
 - The attribute of the user’s LDAP record containing the user’s email address. This will be used to populate their account information.
 - Example: `mail`

基于 **BindDN** 需要填充以下字段:

- 绑定 DN (可选)
 - The DN to bind to the LDAP server with when searching for the user. This may be left blank to perform an anonymous search.
 - Example: `cn=Search,dc=mydomain,dc=com`
- 绑定密码 (可选)
 - The password for the Bind DN specified above, if any. *Note: The password is stored in plaintext at the server. As such, ensure that your Bind DN has as few privileges as possible.*
- 用户搜索基准 (必填)
 - The LDAP base at which user accounts will be searched for.
 - Example: `ou=Users,dc=mydomain,dc=com`
- 用户过滤规则 (必填)
 - An LDAP filter declaring how to find the user record that is attempting to

authenticate. The `%s` matching parameter will be substituted with login name given on sign-in form.

- Example: `(&(objectClass=posixAccount)(uid=%s))`
- To substitute more than once `%[1}s` should be used instead, eg. when matching supplied login name against multiple attributes such as user identifier, email or even phone number.
- Example: `(&(objectClass=Person)(|(uid=%[1}s)(mail=%[1}s)(mobile=%[1}s)))`

基于 **simple auth** 需要填充以下字段：

- User DN (必填)
 - A template to use as the user's DN. The `%s` matching parameter will be substituted with login name given on sign-in form.
 - Example: `cn=%s,ou=Users,dc=mydomain,dc=com`
 - Example: `uid=%s,ou=Users,dc=mydomain,dc=com`
- 用户过滤规则 (必填)
 - An LDAP filter declaring when a user should be allowed to log in. The `%s` matching parameter will be substituted with login name given on sign-in form.
 - Example: `(&(objectClass=posixAccount)(cn=%s))`
 - Example: `(&(objectClass=posixAccount)(uid=%s))`

验证组成员 需要填充以下字段：

- 组搜索基准 DN (可选)
 - The LDAP DN used for groups.
 - Example: `ou=group,dc=mydomain,dc=com`
- 组名称过滤 (可选)
 - An LDAP filter declaring how to find valid groups in the above DN.
 - Example: `(|(cn=gogs_users)(cn=admins))`
- 包含用户的组属性 (可选)
 - Which group LDAP attribute contains an array above user attribute names.
 - Example: `memberUID`
- 组内用户属性 (可选)
 - Which user LDAP attribute is listed in the group.
 - Example: `uid`

PAM

To configure this you just need to set the ‘PAM Service Name’ to a filename in `/etc/pam.d/`.

If you want it to work with normal Linux passwords, the user running Gogs must have read access to `/etc/shadow`.

SMTP

Gogs 支持通过指定邮箱服务器来对用户进行创建和认证，可以通过配置以下选项启用该功能：

- 认证名称（必填）
 - A name to assign to the new method of authorization.
- SMTP 认证类型（必填）
 - Type of authentication for use on your SMTP host, PLAIN or LOGIN.
- 主机地址（必填）
 - The address where the SMTP host can be reached.
 - Example: `smtp.mydomain.com`
- 端口号（必填）
 - The port to use when connecting to the server.
 - Example: `587`
- 域名白名单
 - Restrict what domains can log in if you’re using public SMTP host or SMTP host with multiple domains.
 - Example: `gogs.io,mydomain.com,mydomain2.com`
- 启用 TLS 加密
 - Enable TLS encryption on authentication.
- 忽略 TLS 验证
 - Disable TLS verify on authentication.
- This authentication is activate
 - Enable or disable this auth.

Freeipa

- In order to login to the Gogs using FreeIPA credentials, you need to create a bind account for Gogs to use:
- On the FreeIPA server, create a `gogs.ldif` file, replacing dc=example,dc=com with your DN, and providing an appropriately secure password:

```

1. dn: uid=gogs,cn=sysaccounts,cn=etc,dc=example,dc=com
2. changetype: add
3. objectclass: account
4. objectclass: simplesecurityobject
5. uid: gogs
6. userPassword: secure password
7. passwordExpirationTime: 20380119031407Z
8. nsIdleTimeout: 0

```

- Import the LDIF (change localhost to an IPA server if needed), you'll be prompted for your Directory Manager password:

```

1. ldapmodify -h localhost -p 389 -x -D \
2. "cn=Directory Manager" -W -f gogs.ldif

```

- Add an IPA group for gogs_users :

```
1. ipa group-add --desc="Gogs Users" gogs_users
```

- Note! if you get error about ipa credentials please run kinit admin and give your admin account password.
- Now login to the gogs as an Admin, click on “Authentication” under Admin Panel. Then click New LDAP Source and fill in the details, changing all where appropriate to your own domain as photo below:

- Dashboard
- Users
- Organizations
- Repositories
- Authentications
- Configuration
- System Notices
- Monitoring

Authentication Type LDAP (via BindDN)

Authentication Name * ipa

Security Protocol * StartTLS

Host * ipa01.example.com

Port * 389

Bind DN uid=gogs,cn=sysaccounts,cn=etc,dc=example,dc=com

Bind Password *****

Warning: This password is stored in plain text. Do not use a high privileged account.

User Search Base * cn=accounts,dc=example,dc=com

User Filter * (&(uid=%s)(memberOf=cn=gogs__users,cn=groups,cn=accounts,dc=example,dc=com))

Admin Filter

Username attribute Leave empty to use sign-in form field value for user name.

First name attribute

Surname attribute

Email attribute * mail

Fetch attributes in Bind DN context

Skip TLS Verify

This authentication is activated

Update Authentication Setting Delete This Authentication

5. 高级用法

- [高级用法](#)

高级用法

5.1 配置文件手册

- 配置文件手册
 - 概览
 - 服务器 (`server`)
 - 仓库 (`repository`)
 - 仓库 - 编辑器 (`repository.editor`)
 - 仓库 - 文件上传 (`repository.upload`)
 - 版本发布 - 附件 (`release.attachment`)
 - Markdown (`markdown`)
 - Smartypants (`smartypants`)
 - HTTP (`http`)
 - 数据库 (`database`)
 - 应用管理 (`admin`)
 - 安全 (`security`)
 - 服务 (`service`)
 - Web 钩子 (`webhook`)
 - 邮件 (`mailer`)
 - 缓存 (`cache`)
 - 会话 (`session`)
 - 图片 (`picture`)
 - 附件 (`attachment`)
 - 时间 (`time`)
 - 日志 (`log`)
 - 日志 - 控制台 (`log.console`)
 - 日志 - 文件 (`log.file`)
 - 日志 - Slack (`log.slack`)
 - Cron (`cron`)
 - Cron - 更新镜像 (`cron.update_mirrors`)
 - Cron - 仓库健康检查 (`cron.repo_health_check`)
 - Cron - 仓库统计检查 (`cron.check_repo_stats`)
 - Cron - 仓库归档清理 (`cron.repo_archive_cleanup`)
 - Git (`git`)
 - Git - 超时 (`git.timeout`)
 - UI (`ui`)
 - UI - Admin (`ui.admin`)
 - 其他 (`other`)

配置文件手册

本手册会详尽地描述有关 Gogs 配置文件的选项，帮助您更好地理解和使用 Gogs。

请记住，任何修改都是发生在 `custom/conf/app.ini` 自定义配置文件中，该文件的具体位置与您的设置有关。

完整的默认设置可以通过 `app.ini` 文件查看。如果您看到类似 `%(x)s` 字符，这是由 `ini` 提供的递归取值的特性。

概览

名称	描述
<code>APP_NAME</code>	应用名称，可以改成您的组织或公司名称
<code>RUN_USER</code>	运行应用的用户名，我们建议您使用 <code>git</code> ，但如果您在个人计算机上运行 Gogs，请修改为您的系统用户名。如果没有正确设置这个值，很可能导致您的应用崩溃
<code>RUN_MODE</code>	鉴于性能和其它考虑，建议在部署环境下修改为 <code>prod</code> 模式。在您完成安装操作时，该值也会被设置为 <code>prod</code>

服务器 (`server`)

名称	描述
<code>PROTOCOL</code>	<code>http</code> 或 <code>https</code>
<code>DOMAIN</code>	服务器域名
<code>ROOT_URL</code>	公开的完整 URL 路径
<code>HTTP_ADDR</code>	应用 HTTP 监听地址
<code>HTTP_PORT</code>	应用 HTTP 监听端口号
<code>UNIX_SOCKET_PERMISSION</code>	Unix 套接字文件的权限
<code>LOCAL_ROOT_URL</code>	用于 Gogs 工作进程（如：SSH）回访应用的本地（DMZ）URL，一般情况下请保持默认值，除非您的 SSH 服务器节点与 HTTP 并不是同一个节点入口
<code>DISABLE_SSH</code>	当 SSH 功能不可用时可以禁用
<code>START_SSH_SERVER</code>	启用该选项来启动内置 SSH 服务器
<code>SSH_DOMAIN</code>	允许公用网络访问 SSH 的域名
<code>SSH_PORT</code>	SSH 端口号，如果不为 <code>22</code> 的话可以在此修改
<code>SSH_LISTEN_HOST</code>	内置 SSH 服务器监听的地址
<code>SSH_LISTEN_PORT</code>	内置 SSH 服务器监听的端口
<code>SSH_ROOT_PATH</code>	SSH 根目录，一般为 <code>~/.ssh</code> ，但必须填写为 <code>/home/git/.ssh</code>
<code>SSH_KEY_TEST_PATH</code>	用于测试 SSH 公钥的临时目录
<code>SSH_KEYGEN_PATH</code>	<code>ssh-keygen</code> 程序的路径，默认为 <code>ssh-keygen</code> 即通过系统路径查找
<code>MINIMUM_KEY_SIZE_CHECK</code>	指定不同类型的公钥的最小密钥大小

<code>OFFLINE_MODE</code>	D激活该选项来禁止从 CDN 获取静态资源，同时 Gravatar 服务也将被自动禁用
<code>DISABLE_ROUTER_LOG</code>	激活该选项来禁止打印路由日志
<code>CERT_FILE</code>	HTTPS 授权文件路径
<code>KEY_FILE</code>	HTTPS 的密钥文件路径
<code>STATIC_ROOT_PATH</code>	模板文件和静态文件的上级目录，默认为应用二进制所在的位置
<code>APP_DATA_PATH</code>	应用内部数据的存放目录
<code>ENABLE_GZIP</code>	激活该选项来启用应用级别 GZIP 支持
<code>LANDING_PAGE</code>	未登录用户的默认首页，可以是 <code>home</code> 或 <code>explore</code> (探索页)

仓库 (`repository`)

名称	描述
<code>ROOT</code>	用户仓库存储根目录，必须为绝对路径，默认为 <code>~/<user name>/gogs-repositories</code>
<code>SCRIPT_TYPE</code>	系统脚本类型，一般情况下均为 <code>bash</code> ，但有些用户反应只能使用 <code>sh</code>
<code>ANSI_CHARSET</code>	当遇到无法识别的字符集时使用的默认字符集
<code>FORCE_PRIVATE</code>	强制要求所有新建的仓库都是私有的
<code>MAX_CREATION_LIMIT</code>	全局默认的每个用户可创建仓库上限， <code>-1</code> 表示无限制
<code>PREFERRED_LICENSES</code>	建议用户首选的授权类型
<code>DISABLE_HTTP_GIT</code>	激活该选项来禁止用户通过 HTTP 对 Git 仓库进行交互操作，即用户只能通过 SSH 操作
<code>ENABLE_LOCAL_PATH_MIGRATION</code>	激活该选项来启用本地路径迁移仓库功能。启动后默认只有管理员可以使用，普通用户必须经由管理员授权

仓库 - 编辑器 (`repository.editor`)

名称	描述
<code>LINE_WRAP_EXTENSIONS</code>	需要显示为行包装的文件名后缀，通过逗号分隔。如果是无后缀名的文件，则单独放置一个逗号，例如： <code>.txt,</code>

仓库 - 文件上传 (`repository.upload`)

名称	描述
<code>ENABLED</code>	激活该选项来启用仓库文件上传功能
<code>TEMP_PATH</code>	文件上传的临时存放目录
<code>ALLOWED_TYPES</code>	允许上传的文件类型（例如： <code>"image/png"</code> ），留空表示允许上传任意类型的文件
<code>FILE_MAX_SIZE</code>	单个上传的文件的最大体积，以 MB 为单位

MAX_FILES	单次同时上传的最多文件个数
-----------	---------------

版本发布 - 附件 (`release.attachment`)

名称	描述	
ENABLED	激活该选项来启用版本发布附件功能	
PATH	存放附件的路径	
ALLOWED_TYPES	允许上传的 MIME 类型, 例如 “image/jpeg”	“image/png”, 使用 <code>/*/*</code> 允许所有类型的文件
MAX_SIZE	最大允许上传的附件体积, 单位为 MB, 例如 <code>32</code>	
MAX_FILES	最大允许一次性上传的附件个数, 例如 <code>10</code>	

Markdown (`markdown`)

名称	描述
ENABLE_HARD_LINE_BREAK	指示是否启用硬性换行扩展
CUSTOM_URL_SCHEMES	允许被解析为链接的自定义 URL 方案, 例如 <code>git</code> (用于 <code>git://</code>) 和 <code>magnet</code> (用于 <code>magnet://</code>)
FILE_EXTENSIONS	需要被渲染为 Markdown 格式的文件名后缀, 通过逗号分隔。如果是无后缀名的文件, 则单独放置一个逗号, 例如: <code>.markdown,</code>

Smartypants (`smartypants`)

名称	描述
ENABLED	指示是否启用 SmartyPants 扩展

HTTP (`http`)

名称	描述
ACCESS_CONTROL_ALLOW_ORIGIN	头信息 <code>Access-Control-Allow-Origin</code> 的自定义值, 默认为空, 即不响应此头信息

数据库 (`database`)

名称	描述
DB_TYPE	数据库类型, 可以是 <code>mysql</code> 、 <code>postgres</code> 、 <code>mssql</code> 或 <code>sqlite3</code>
HOST	数据库主机地址与端口
NAME	数据库名称

USER	数据库用户名
PASSWD	数据库用户密码
SSL_MODE	仅限 PostgreSQL 使用
PATH	仅限 SQLite3 使用，数据库文件路径

应用管理 (`admin`)

名称	描述
<code>DISABLE_REGULAR_ORG_CREATION</code>	激活该选项来禁止普通用户（非管理员）创建组织

安全 (`security`)

名称	描述
<code>INSTALL_LOCK</code>	用于指示是否允许访问安装页面（该页面可以设置管理员帐号，因此该选项非常重要）
<code>SECRET_KEY</code>	全局的加密密钥，务必修改该值以确保您的服务器安全（会在每次安装时自动生成随机字符串）
<code>LOGIN_REMEMBER_DAYS</code>	记住登录的天数
<code>COOKIE_USERNAME</code>	记录用户名的 Cookie 名称
<code>COOKIE_REMEMBER_NAME</code>	记录用户自动登录信息的 Cookie 名称
<code>REVERSE_PROXY_AUTHENTICATION_USER</code>	反向代理认证用户的 Header 字段名

服务 (`service`)

名称	描述
<code>ACTIVE_CODE_LIVE_MINUTES</code>	激活码的有效期，单位为分钟
<code>RESET_PASSWORD_CODE_LIVE_MINUTES</code>	重置密码的有效期，单位为分钟
<code>REGISTER_EMAIL_CONFIRM</code>	激活该选项来要求注册用户必须验证邮箱，要求已启用 <code>Mailer</code>
<code>DISABLE_REGISTRATION</code>	激活该选项来禁止用户注册功能，只能由管理员创建帐号
<code>SHOW_REGISTRATION_BUTTON</code>	用于指示是否显示注册按钮
<code>REQUIRE_SIGNIN_VIEW</code>	激活该选项来要求用户必须登录才能浏览任何页面
<code>ENABLE_CACHE_AVATAR</code>	激活该选项来缓存 Gravatar 的头像
<code>ENABLE_NOTIFY_MAIL</code>	激活该选项来发送通知邮件给关注者，例如创建 issue 时，要求已启用 <code>Mailer</code>
<code>ENABLE_REVERSE_PROXY_AUTHENTICATION</code>	激活该选项来开启反向代理用户认证，请从 https://github.com/gogits/gogs/issues/165 了解更多信息
<code>ENABLE_REVERSE_PROXY_AUTO_REGISTRATION</code>	激活该选项来开启反向代理用户认证的自动注册功能

<code>DISABLE_MINIMUM_KEY_SIZE_CHECK</code>	激活该选项来禁止检查响应类型的密钥最小长度
<code>ENABLE_CAPTCHA</code>	激活该选项以在用户注册时要求输入验证码

Web 钩子 (`webhook`)

名称	描述
<code>TYPES</code>	启动的 Web 钩子类型，可以是 <code>gogs</code> 、 <code>slack</code> 或 <code>discord</code>
<code>DELIVER_TIMEOUT</code>	发送通知的超时时间，以秒为单位
<code>SKIP_TLS_VERIFY</code>	指示是否允许向具有非信任证书的地址发送通知
<code>PAGING_NUM</code>	Web 钩子历史页面每页显示记录条数

邮件 (`mailer`)

名称	描述
<code>ENABLED</code>	启用该选项以激活邮件服务
<code>DISABLE_HELO</code>	禁用 HELO 操作
<code>HELO_HOSTNAME</code>	HELO 操作的自定义主机名
<code>HOST</code>	SMTP 主机地址与端口
<code>FROM</code>	邮箱的来自地址，遵循 RFC 5322 规范，可以是一个单纯的邮箱地址或者 "名字" < email@example.com > 的形式
<code>USER</code>	邮箱用户名
<code>PASSWD</code>	邮箱密码
<code>SKIP_VERIFY</code>	不验证自签发证书的有效性
<code>USE_PLAIN_TEXT</code>	使用 <code>text/plain</code> 作为邮件内容格式

备注：Gogs 仅支持使用 STARTTLS 的 SMTP 协议

缓存 (`cache`)

名称	描述
<code>ADAPTER</code>	缓存引擎适配器，可以为 <code>momery</code> 、 <code>redis</code> 或 <code>memcache</code> 。如果您使用 <code>redis</code> 或 <code>memcache</code> ，请确保使用 <code>-tags</code> 选项重新构建所有依赖，例如： <code>go build -tags='redis'</code>
<code>INTERVAL</code>	仅限内存缓存使用，GC 周期，单位为秒
<code>HOST</code>	仅限 <code>redis</code> 和 <code>memcache</code> 使用，主机地址和端口号
-	<code>Redis</code> : <code>network=tcp,addr=127.0.0.1:6379,password=macaron,db=0,pool_size=100,idle_time=300</code>
-	<code>Memache</code> : <code>127.0.0.1:9090;127.0.0.1:9091</code>

会话 (`session`)

名称	描述
PROVIDER	Session 引擎提供者，可以是 <code>memory</code> 、 <code>file</code> 、 <code>redis</code> 或 <code>mysql</code>
PROVIDER_CONFIG	如果提供者为 <code>file</code> ，则为文件根目录；如果为其它提供者，则为主机地址和端口号
COOKIE_SECURE	激活该选项以要求所有 session 操作均通过 HTTPS
GC_INTERVAL_TIME	GC 周期，单位为秒

图片 (`picture`)

名称	描述
AVATAR_UPLOAD_PATH	存放用户上传头像的目录
GRAVATAR_SOURCE	可以是 <code>gravatar</code> 、 <code>duoshuo</code> 或任何 URL，例如： http://cn.gravatar.com/avatar/
DISABLE_GRAVATAR	激活该选项来仅使用本地头像
ENABLE_FEDERATED_AVATAR	激活该选项来启用 Federated 头像服务 (http://www.libravatar.org)，当 Gravatar 被禁用时此选项无法生效

附件 (`attachment`)

名称	描述
ENABLED	激活该选项以允许用户上传附件
PATH	存放附件的路径
ALLOWED_TYPES	允许上传的 MIME 类型，例如 “image/jpeg”
MAX_SIZE	最大允许上传的附件体积，单位为 MB，例如 4
MAX_FILES	最大允许一次性上传的附件个数，例如 5

时间 (`time`)

名称	描述
FORMAT	指定日期的输出格式，默认为 RFC1123，其它可选的格式为 ANSIC、UnixDate、RubyDate、RFC822、RFC822Z、RFC850、RFC1123、RFC1123Z、RFC3339、RFC3339Nano、Kitchen、Stamp、StampMilli、StampMicro 和 StampNano。访问 http://golang.org/pkg/time/#pkg-constants 查看详情

日志 (`log`)

名称	描述

<code>ROOT_PATH</code>	日志文件的根目录
<code>MODE</code>	日志记录模式, 默认为 <code>console</code> 。如果想要开启多模式, 请使用逗号分割, 例如: <code>"console, file"</code>
<code>LEVEL</code>	基本日志级别, 默认为 <code>Trace</code>

日志 - 控制台 (`log.console`)

名称	描述
<code>LEVEL</code>	控制台日志级别, 留空则继承父值

日志 - 文件 (`log.file`)

名称	描述
<code>LEVEL</code>	控制台日志级别, 留空则继承父值
<code>LOG_RORATE</code>	激活该选项以启用日志文件自转
<code>DAILY_ROTATE</code>	激活该选项以进行日常自转
<code>MAX_SIZE_SHIFT</code>	自转需要达到的最大文件体积, 使用位左移, 默认为 28 即 $1 << 28$, 表示 256MB
<code>MAX_LINES</code>	自转需要达到的最大文件行数, 默认为 1000000
<code>MAX_DAYS</code>	保留自转文件的最长期限, 默认为 7 天后删除

日志 - Slack (`log.slack`)

名称	描述
<code>LEVEL</code>	控制台日志级别, 留空则继承父值
<code>URL</code>	Slack Web 钩子 URL

Cron (`cron`)

名称	描述
<code>ENABLED</code>	激活该选项以允许周期性运行 Cron 任务
<code>RUN_AT_START</code>	激活该选项以允许在启动时执行 Cron 任务

Cron - 更新镜像 (`cron.update_mirrors`)

名称	描述
<code>SCHEDULE</code>	定时更新仓库镜像的 Cron 语法, 例如: <code>@every 1h</code>

Cron - 仓库健康检查 (`cron.repo_health_check`)

名称	描述
SCHEDULE	定时进行仓库健康检查的 Cron 语法, 例如: <code>@every 24h</code>
TIMEOUT	仓库健康检查超时的定义语法, 例如: <code>60s</code>
ARGS	<code>git fsck</code> 命令的参数, 例如: <code>--unreachable --tags</code>

Cron - 仓库统计检查 (`cron.check_repo_stats`)

名称	描述
RUN_AT_START	激活该选项以在启动时执行仓库统计检查
SCHEDULE	定时进行仓库统计检查的 Cron 语法, 例如: <code>@every 24h</code>

Cron - 仓库归档清理 (`cron.repo_archive_cleanup`)

名称	描述
RUN_AT_START	激活该选项以在启动时执行仓库归档清理
SCHEDULE	定时进行仓库归档清理的 Cron 语法, 例如: <code>@every 24h</code>
OLDER_THAN	仓库归档的文件有效期, 过期的归档将被清理, 例如: <code>24h</code>

Git (`git`)

名称	描述
DISABLE_DIFF_HIGHLIGHT	激活该选项以禁用行内差异高亮
MAX_GIT_DIFF_LINES	差异对比页面单个文件显示的最大行数
MAX_GIT_DIFF_LINE_CHARACTERS	差异对比页面单行显示的最大字符数
MAX_GIT_DIFF_FILES	差异对比页面文件显示的最多个数
GC_ARGS	<code>git gc</code> 命令的参数, 例如: <code>--aggressive --auto</code>

Git - 超时 (`git.timeout`)

名称	描述
MIGRATE	仓库迁移操作超时, 默认为 <code>600</code> 秒
MIRROR	仓库镜像同步操作超时, 默认为 <code>300</code> 秒
CLONE	仓库克隆操作超时, 默认为 <code>300</code> 秒
PULL	仓库拉取操作超时, 默认为 <code>300</code> 秒
GC	仓库垃圾回收操作超时, 默认为 <code>60</code> 秒

UI (`ui`)

名称	描述
EXPLORE_PAGING_NUM	探索页面每页显示仓库的数量
ISSUE_PAGING_NUM	每页显示工单（Issue）的数量（应用到所有以列表形式显示工单的页面）
FEED_MAX_COMMIT_NUM	一条最新活动中显示代码提交（Commit）的最大数量
THEME_COLOR_META_TAG	被用于 Android ≥ 5.0 版本“theme-color”标记的值，无效的值将被忽略并使用默认值（ 查看详情 ）
MAX_DISPLAY_FILE_SIZE	显示到页面的最大文件体积（Byte）

UI - Admin (`ui.admin`)

名称	描述
USER_PAGING_NUM	用户管理页面每页显示记录条数
REPO_PAGING_NUM	仓库管理页面每页显示记录条数
NOTICE_PAGING_NUM	系统提示管理页面每页显示记录条数
ORG_PAGING_NUM	组织管理页面每页显示记录条数

其他 (`other`)

名称	描述
SHOW_FOOTER_BRANDING	激活该选项以在页脚显示 Gogs 推广信息
SHOW_FOOTER_VERSION	激活该选项以在页脚显示 Gogs 版本信息
SHOW_FOOTER_TEMPLATE_LOAD_TIME	激活该选项以在页脚显示 Gogs 模板加载时间



5.2 公告与高阶指南

- [公告与高阶指南](#)
 - [公告博文](#)
 - [高阶指南](#)

公告与高阶指南

公告博文

- [v0.2](#) : [Gogs](#): 用二进制才是真正的部署
- [v0.3](#) : 使用 Gogs 搭建私有 Git 托管服务
- [v0.4](#) : 基于 Gogs 进行协同工作
- [v0.5](#) : 通过 Gogs 的组织功能管理团队

高阶指南

- [使用 OAuth 登录 Gogs](#)

6. 捐赠我们

- [捐赠我们](#)
 - 您的捐赠将被用于:
 - [捐赠列表](#) (按照捐赠时间倒序)

捐赠我们

在您的支持与鼓励下，**Gogs** 开发团队将会更加努力地开发出更好的产品！

支付宝捐赠:

Paypal 捐赠



您的捐赠将被用于:

- 持续和深入的开发
- 维护社区的运行稳定
- 租用更好的带宽
- 奖励团队的杰出贡献者
- 社区活动或讲座

捐赠列表（按照捐赠时间倒序）

捐赠时间	捐赠人	金额
2017-11-23	朱同学	¥20.00
2017-11-20	余同学	¥10.00
2017-10-28	寒子	¥6.66
2017-10-26	向同学	¥10.00
2017-10-21	向同学	¥200.00
2017-10-12	向同学	¥5.00
2017-10-05	向同学	¥5.00
2017-09-30	向同学	¥200.00
2017-09-14	适同学	¥101.00
2017-09-14	九梦	¥10.00
2017-09-06	朱同学	¥98.98
2017-08-17	向同学	¥5.00
2017-07-21	老韩	¥100.00
2017-05-26	Refactoring	¥100.00
2017-04-03	胜同学	¥100.00
2017-03-26	向同学	¥18.00
2017-03-21	陈同学	¥100.00
2017-03-16	godlike	¥200.00
2017-03-16	向同学	¥100.00
2017-02-21	智同学	¥100.00
2017-02-14	曾同学	¥100.00
2017-02-14	自富	¥200.00
2017-02-14	刘同学	¥100.00
2017-01-11	亚同学	¥3.00
2016-12-29	涛同学	¥15.00
2016-12-25	明同学	¥18.80
2016-10-18	倩同学	¥200.00
2016-10-16	nivk	¥400.00
2016-10-14	东威	¥9.99
2016-09-21	悟道人	¥50.00
2016-08-01	冒险王	¥100.00
2016-07-25	诸葛同学	¥50.00
2016-07-10	峰同学	¥100.00

2016-06-30	null	¥20.00
2016-06-24	细同学	¥20.00
2016-06-21	ZNing	¥6.88
2016-06-16	天涯同学	¥20.00
2016-06-11	军同学	¥200.00
2016-05-30	老K	¥5.00
2016-05-16	百同学	¥ 100.00
2016-05-04	军同学	¥ 100.00
2016-04-16	秦同学	¥50.00
2016-04-10	greatpie	¥6.66
2016-04-05	kuuyee	¥500.00
2016-03-23	崔同学	¥ 100.00
2016-03-19	李同学	¥ 100.00
2016-03-07	binzu	6.66
2016-02-05	huihoo	¥50.00
2016-02-02	鹏同学	¥100.00
2016-01-31	huihoo	¥100.00
2015-12-14	维同学	¥20.00
2015-11-28	无同学	¥100.00
2015-09-17	雪同学	¥100.00
2015-09-13	匿名	¥100.00
2015-09-11	霍同学	¥100.00
2015-08-20	马同学	¥100.00
2015-07-19	谭同学	¥100.00
2015-07-16	军同学	¥100.00
2014-05-05	吴同学	¥50.00
2014-05-02	李同学	¥8.88
2014-05-02	熊同学	¥9.99
2014-05-01	艾同学	¥0.50
2014-05-01	何同学	¥50.00
2014-05-01	codepapa	¥199.00 多出教程

