

Seek Assistance

- [Chatroom](#)
- [Issue tracker](#)

“ [!note] Note When asking for support, it may be a good idea to have the following available so that the person helping has all the info they need:

1. Your `app.ini` (with any sensitive data scrubbed as necessary).
2. The Forgejo logs, and any other appropriate log files for the situation.
 - When using systemd, use `journalctl --lines 1000 --unit forgejo` to collect logs.
 - When using docker, use `docker logs --tail 1000 <forgejo-container>` to collect logs.
 - By default, the logs are outputted to console. If you need to collect logs from files, you could copy the following config into your `app.ini` (remove all other `[log]` sections), then you can find the `*.log` files in Forgejo's log directory (default: `%(FORGEJO_WORK_DIR)/log`).

```
; To show all SQL logs, you can also set LOG_SQL=true in the [database] section
[log]
LEVEL=debug
MODE=console,file
ROUTER=console,file
XORM=console,file
ENABLE_XORM_LOG=true
FILE_NAME=forgejo.log
[log.file.router]
FILE_NAME=router.log
[log.file.xorm]
FILE_NAME=xorm.log
```

3. Any error messages you are seeing.
4. If you meet slow/hanging/deadlock problems, please report the stack trace when the problem occurs:
 1. Enable pprof in `app.ini` and restart Forgejo.

```
[server]
ENABLE_PPROF = true
```

2. Trigger the bug, and when Forgejo gets stuck, use curl or browser to visit:
`http://127.0.0.1:6060/debug/pprof/goroutine?debug=1` (IP must be `127.0.0.1` and port must be `6060`).
 3. If you are using Docker, please use `docker exec -it <container-name> curl "http://127.0.0.1:6060/debug/pprof/goroutine?debug=1"`.
 4. Report the output (the stack trace doesn't contain sensitive data).
-

#1

cocobook-sync 2026-10-28 16:41:23 UTC

cocobook-sync 2026-10-28 16:41:23 UTC