

This page outlines the steps that need to be done to create a new `golang.org/x` repository, in order for it to have the same properties as all existing `golang.org/x` repositories: - a `golang.org/x` redirect - automatic git mirroring from Gerrit to GitHub - automatic importing of GitHub PRs into Gerrit CLs

Steps

1. Create a new empty Gerrit repository at <https://go.googlesource.com>, complete with a description.
 - Create an initial commit with `AUTHORS`, `CONTRIBUTORS`, `LICENSE`, `PATENTS`, `CONTRIBUTING.md`, and `README.md` files and push it directly to the Gerrit repository. See example commit.
 - See the internal team instructions at `go/go-gerrit#new-repository` for how to create a repository.
2. Create a new empty GitHub repository at <https://github.com/golang> with the same name and description.
 - Turn off Wikis, Issues, Projects in repository settings.
 - On “Manage access” tab:
 - Add “golang org admins” team with Admin access.
 - Add “gophers” team with Write access.
 - Add “robots” team with Write access (can only be done by a maintainer of golang organization; ask someone else if you’re not).
 - Create “cla: yes” and “cla: no” labels, they need to exist so that @googlebot can automatically apply them. (Without a “cla: yes” label, PRs won’t be imported into Gerrit.)
3. Modify the `x/build/repos` package.
4. Update `x/website`’s version of `x/build` to include modified `x/build/repos` package.
5. Redeploy all affected commands (or ask an `x/build` owner to deploy if you’re not; the order shouldn’t matter):
 1. `x/build/cmd/gitmirror`
 2. `x/build/maintner/maintnerd`
 - Note that it’s expected for the new repo not to appear in maintner until first issue or PR is created (see #25744).
 3. `x/build/cmd/gerritbot`
 4. `x/build/cmd/coordinator`
 5. `x/website/cmd/golangorg`
6. You’re done.