

## Contributor Guide!

TrueOG is not just a server *for* the community, it is a server *by* the community. There are two types of contributions:

### 1. Libre

Submitting code and/or maps to TrueOG under the terms of “The Unlicense” on the primary git Repository.

### 2. Licensed

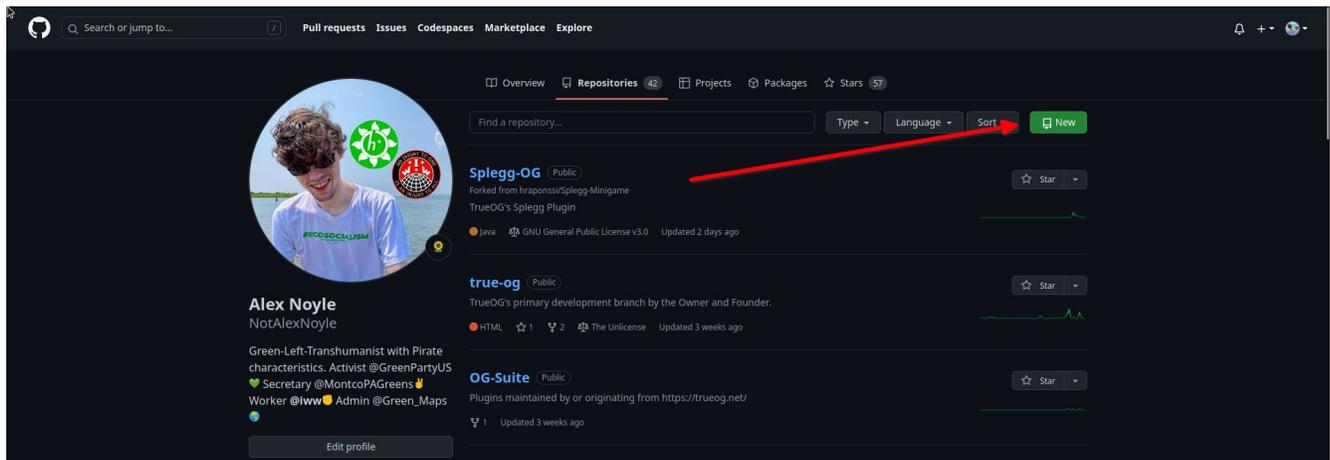
Submitting code and/or maps to TrueOG downstream under a license that is as permissive or more permissive than the GPLv3.

If your “Libre” contribution is accepted, you will be immediately entitled to developer rank. Licensed contributions will be evaluated on a case-by-case basis.

Future Admins, who will run TrueOG as a worker coop, will be selected from the pool of active developers.

# Step 1: Create a git repository.

1. If your contribution stands alone, log in to GitHub, and browse to your profile's repository tab. Click the Green "New" button to create a repository. If you are only modifying TrueOG's main repository, use the "fork" button at <https://github.com/true-og/true-og/> instead.



2. Fill out the form. Pay special attention to the fields with red arrows.

3. If your contribution is a new plugin for the "OG Suite", its name should end in "-OG".

4. Choose a license that is as free or more free than the GPLv3.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \*  / Repository name \*

Great repository names are short and memorable. Need inspiration? How about [sturdy-rotary-phone?](#)

Description (optional)

Public Anyone on the internet can see this repository. You choose who can commit.

Private You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

Add a README file This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

Choose a license

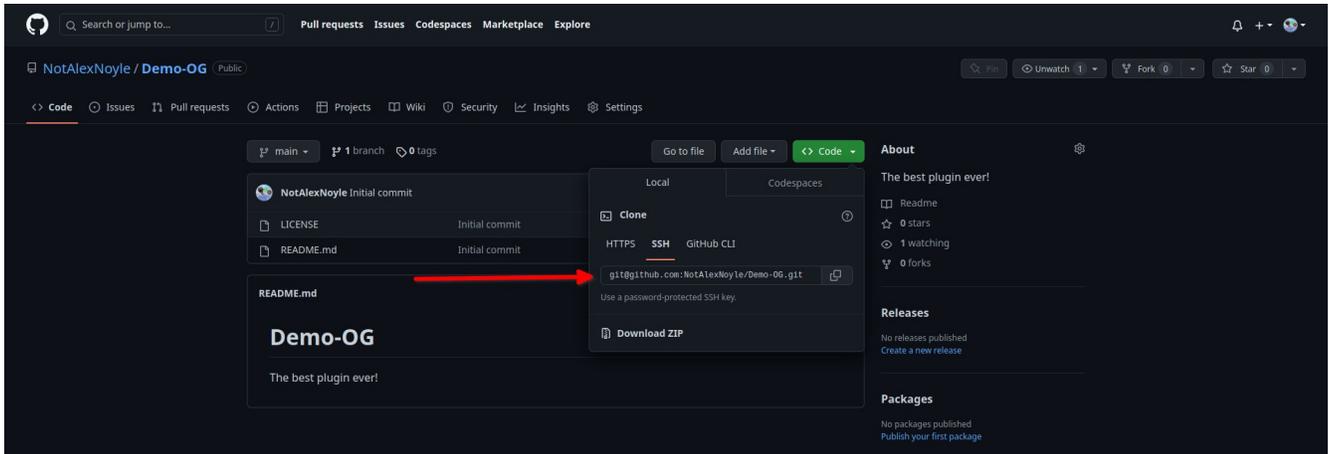
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set `main` as the default branch. Change the default name in your [settings](#).

ⓘ You are creating a public repository in your personal account.

# Step 2: Clone your repository.

1. Locate the SSH URL under the green “Code” button.



*If you don't have ssh authentication set up through GitHub, follow this guide:*

<https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

2. Open a terminal with git installed and configured. Enter the command:

```
git clone git@github.com:NotAlexNoyle/Demo-OG.git
```

Replace the URL with your own.

*If git is not installed and configured, do so for your platform here: <https://git-scm.com/>*

3. Change directories to your repository: `cd my-repo-name/`

## Step 3: Commit your changes.

1. Add your new files or changes to the folder, and then run the command **git status**

```
admin@TrueOG-Devuan-Server:~/Downloads/Demo-OG$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        assets/

no changes added to commit (use "git add" and/or "git commit -a")
```

2. Add your changes to a new commit with **git add your-files**  
Then run **git status** again to make sure your repository looks how you expect it to.

```
admin@TrueOG-Devuan-Server:~/Downloads/Demo-OG$ git add README.md
admin@TrueOG-Devuan-Server:~/Downloads/Demo-OG$ git add assets/
admin@TrueOG-Devuan-Server:~/Downloads/Demo-OG$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md
        new file:   assets/space.png

admin@TrueOG-Devuan-Server:~/Downloads/Demo-OG$ _
```

## Step 4: Push your changes.

1. Commit your changes using `git commit -m "Message describing what you added or changed"`

2. Type `git push` to sync your local changes with the remote GitHub repository.

```
admin@TrueOG-Devuan-Server:~/Downloads/Demo-OG$ git commit -m "Add demo assets"
[main 3f231fb] Add demo assets
 2 files changed, 3 insertions(+)
 create mode 100644 assets/space.png
admin@TrueOG-Devuan-Server:~/Downloads/Demo-OG$ git push
Enter passphrase for key '/home/admin/.ssh/id_rsa':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 32 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 1.39 MiB | 2.51 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:NotAlexNoyle/Demo-OG.git
 3556e26..3f231fb  main -> main
admin@TrueOG-Devuan-Server:~/Downloads/Demo-OG$
```

## Step 5: Add your contribution to a TrueOG branch.

*Ignore this step if you just pushed your changes directly to a fork of TrueOG's primary repository.*

1. All contributions that stand alone should be submitted to TrueOG as git submodules. This includes public domain contributions. First, fork and clone TrueOG's primary repository locally if you haven't done so already, as described in Step 2.

2. Initialize the existing submodules with **git submodule update --init --recursive**

3. **cd** to the relevant directory in the tree, then add your standalone contribution as a submodule using **git submodule add https://github.com/NotAlexNoyle/Demo-OG**

Replace the URL with your own.

4. Add, commit, and push your changes to your TrueOG fork as described in Steps 3 and 4.

```
admin@TrueOG-Devuan-Server:~/Downloads/true-og$ git push
Enter passphrase for key '/home/admin/.ssh/id_rsa':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 32 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 269 bytes | 269.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:NotAlexNoyle/true-og/
 57ef26c..e0425ea  main -> main
```

*If your submodule resides inside of another submodule, you will have to run this process in the container submodule before running it a second time in the project root.*

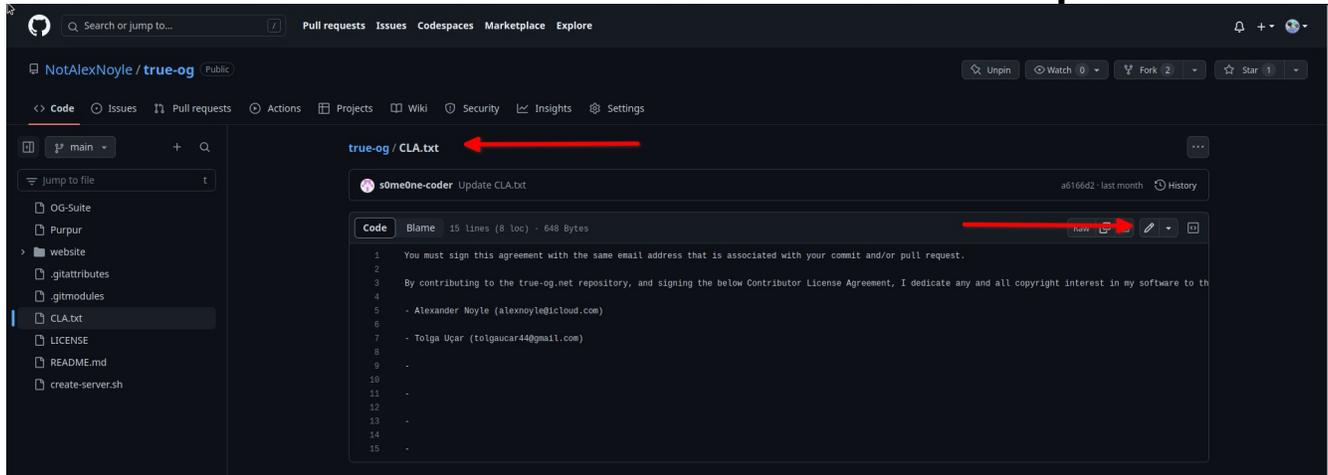
5. If you get a prompt for HTTPS credentials, or an error about permissions, set the origin url to your TrueOG fork with **git remote set-url origin git@github.com:Yourusername/true-og/**

*This requires SSH to be set up for GitHub, as described in Step 2.*

# Step 6: Sign the Contributor License Agreement (CLA)

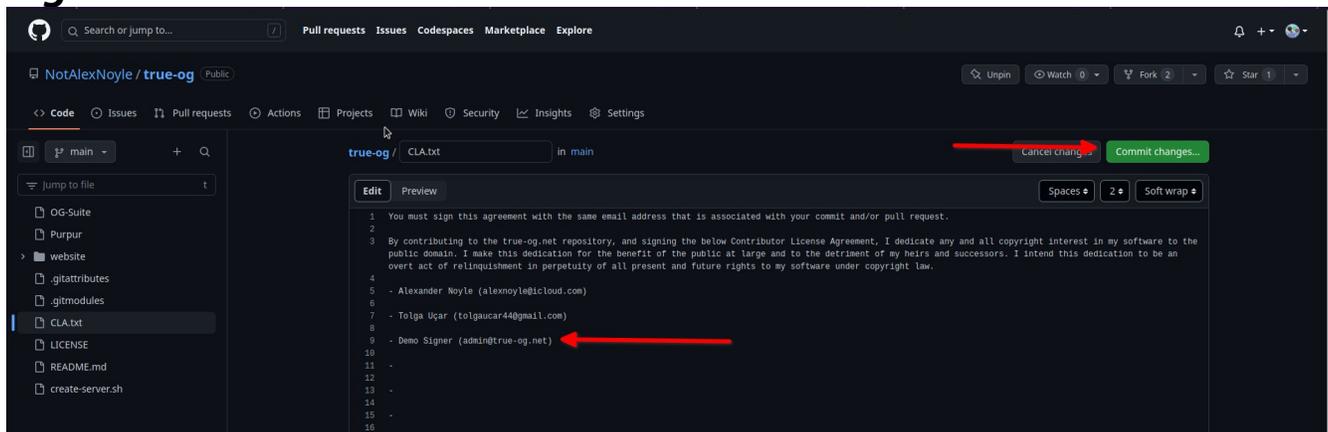
*If you are submitting a licensed contribution, ignore this step.*

1. Use GitHub to browse to the file CLA.txt in your fork of TrueOG. Click the “Edit” button that looks like a pencil.



2. Read the terms of the license agreement carefully, and then sign your name and email address in the same format as the others if you agree.

3. Click the green “Commit changes...” button in the upper-right.



4. Commit the changes to your fork using the pop-up menu. Make your commit message "Sign CLA". Select the same email address in the drop down menu as you used to sign the document.

Commit changes

Commit Message

Sign CLA

Add an optional extended description..

alexnoyle@icloud.com

Choose which email address to associate with this commit

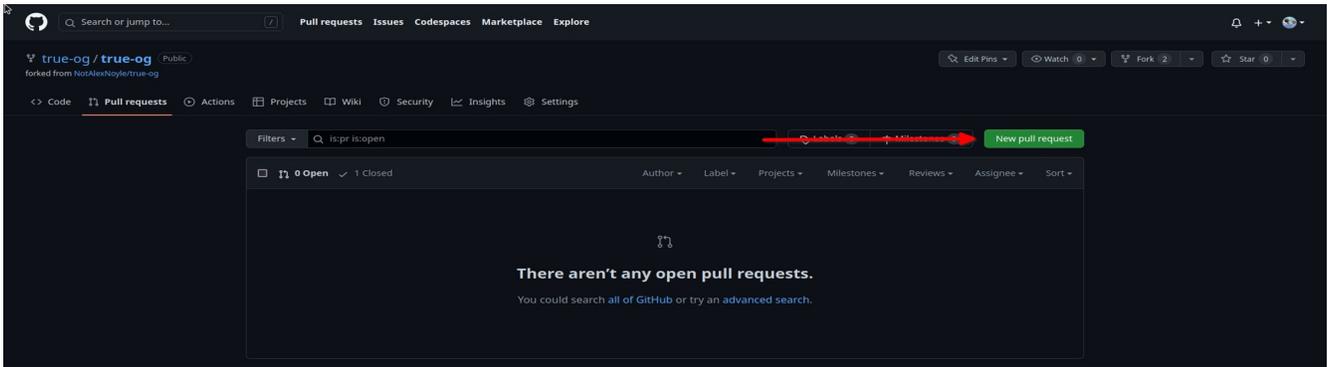
Commit directly to the main branch.

Create a new branch for this commit and start a pull request.  
[Learn more about pull requests.](#)

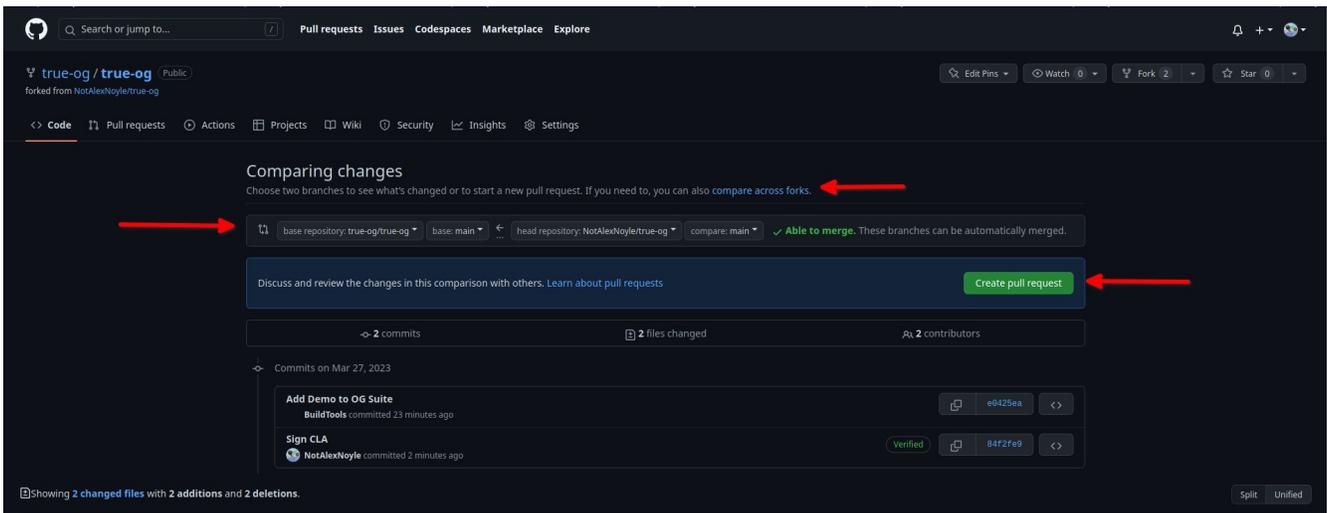
Cancel Commit changes

## ~~Step 7: Submit your contribution.~~

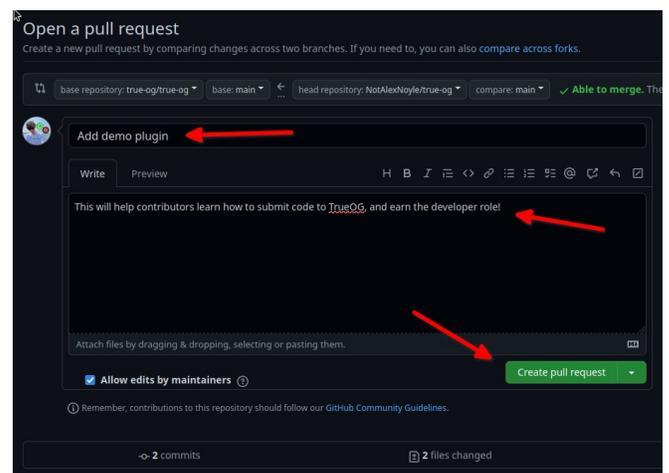
1. Browse to <https://github.com/true-og/true-og/pulls> and click the green "New pull request" button.



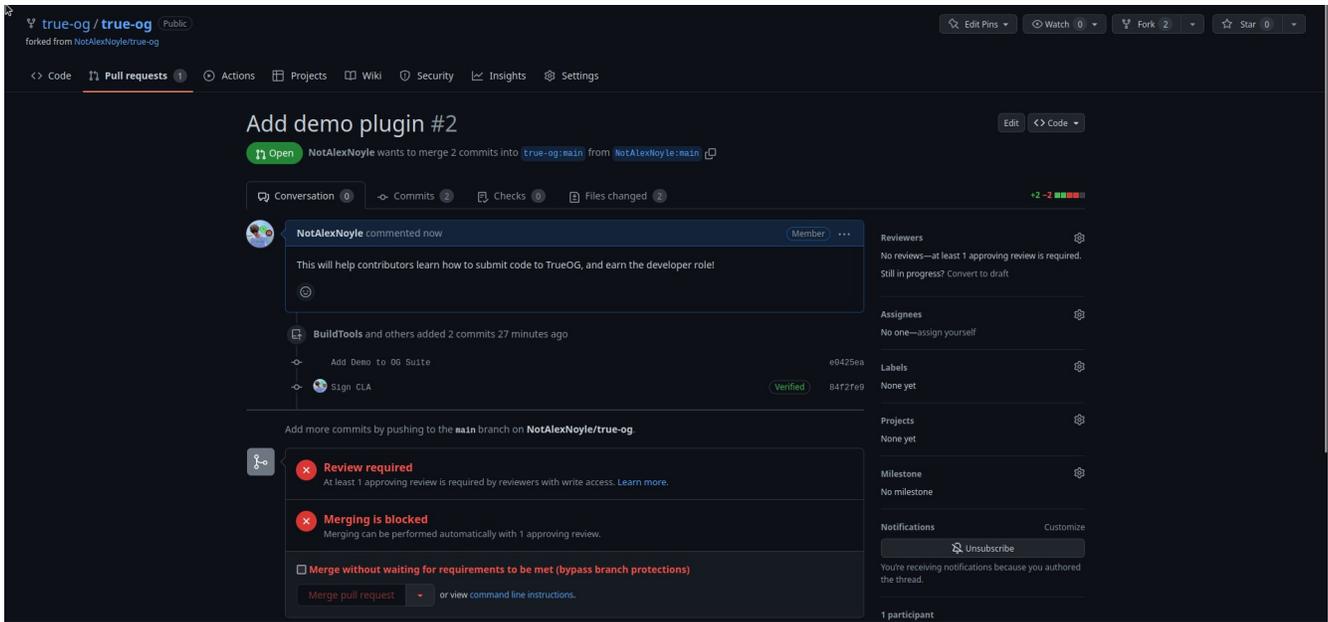
2. Select true-og/true-og as the base repository, and your fork as the head repository. Ensure there are no merge conflicts before submission. Then click the green “Create pull request” button.



3. Name your pull request after what it does, and describe why you think your changes would improve the server in the comment section. Keep “allow edits by maintainers” checked. Finally, click the green “Create pull request” button once again.



Your contribution has now been submitted!



If it is accepted, and it meets the criteria for a developer role, an Admin will be in touch to give you your rank in Discord and in-game. The developer role is represented by an orange lambda symbol, symbolizing computer science.

 [IWV] [Owner] NotAlexNoyle ❤️ λ > hi

TrueOG can't thank you enough for dedicating your time and labor to what we are building together.

A special thank you to those who chose to dedicate their contributions to the public domain. Together, via our collective actions, we can stop copyright tyranny.

