# Contributing to LinuxGSM

?? Thank you for taking the time to contribute! ??

The following is a set of guidelines for contributing to LinuxGSM, which are hosted in the [GameServerManagers Organization](https://github.com/gameservermanagers) on GitHub. These are mostly guidelines, not rules. Use your best judgment, and feel free to propose changes to this document in a pull request.

## Table of Contents

[Contributing to LinuxGSM](#contributing-to-linuxgsm)

\* [Table of Contents](#table-of-contents)

\* [Code of Conduct](#code-of-conduct)

\* [? Bug/Enhancement Contributions ?](#bug-enhancement-contributions)

+ [?Reporting Bugs](#reporting-bugs)

- [Before Submitting A Bug Report](#before-submitting-a-bug-report)

- [How Do I Submit A (Good) Bug Report?](#how-do-i-submit-a--good--bug-report-)

+ [?Suggesting Features](#suggesting-features)

- [Before Submitting An Feature Suggestion](#before-submitting-an-feature-suggestion)

- [How Do I Submit A (Good) Feature Suggestion?](#how-do-i-submit-a--good--feature-suggestion-)

+ [? Game Server Requests](#game-server-requests)

- [Before Submitting a Game Server Request](#before-submitting-a-game-server-request)

- [How Do I Submit A (Good) Game Server Request?](#how-do-i-submit-a--good--game-server-request-)

+ [? Game Server Specific Issues](#game-server-specific-issues)

\* [? Code Contributions](#code-contributions)

+ [Pull Requests](#pull-requests)

- [Pull Request naming convention](#pull-request-naming-convention)

+ [Testing](#testing)

- [Pull Request Status Checks](#pull-request-status-checks)

- [Test Environment](#test-environment)

+ [:wine\_glass: Styleguides](#-wine-glass--styleguides)

- [Git Commit Messages](#git-commit-messages)

- [BASH Styleguide](#bash-styleguide)

\* [:blue\_book: Document Contributions](#-blue-book--document-contributions)

+ [Documentation Styleguide](#documentation-styleguide)

\* [Issue and Pull Request Labels](#issue-and-pull-request-labels)

## Code of Conduct

This project and everyone participating in it are governed by the [LinuxGSM Code of Conduct](https://github.com/GameServerManagers/linuxgsm/blob/master/CODE\_OF\_CONDUCT.md). By participating, you are expected to uphold this code. Please report unacceptable behaviour to [daniel.gibbs@linuxgsm.com](mailto:daniel.gibbs@linuxgsm.com).

## ? Bug/Enhancement Contributions ?

### ?Reporting Bugs

This section guides you through submitting a bug report for LinuxGSM. Following these guidelines help maintainers and the community understand your report ?, reproduce the behaviour?, and find any related reports ?.

Before creating bug reports, please check [this list](https://github.com/GameServerManagers/linuxgsm/blob/master/CONTRIBUTING.md#before-submitting-a-bug-report) as you might find out that you don’t need to create one. When you are creating a bug report, please [include as many details as possible](https://github.com/GameServerManagers/linuxgsm/blob/master/CONTRIBUTING.md#how-do-i-submit-a-good-bug-report). Fill out [the required template]([https://github.com/GameServerManagers/LinuxGSM/issues/new/choose](https://github.com/GameServerManagers/LinuxGSM/issues/new/choose)), the information it asks for helps us resolve issues faster.

#### Before Submitting A Bug Report

\* \*\*Check the [documentation](https://docs.linuxgsm.com).\*\* You might be able to find the cause of the problem and fix things yourself.

\* \*\*Check that the problem is not related to\*\* [\*\*support page\*\*](https://linuxgsm/com/support) for links to other support options.

\* \*\*Check the\*\* [\*\*support page\*\*](https://linuxgsm/com/support) for links to other support options.

\* \*\*Perform a\*\* [\*\*cursory search\*\*](https://github.com/search?q=org:GameServerManagers%20type:issues&type=Issues) to see if the problem has already been reported. If it has \*\*and the issue is still open\*\*, add a comment to the existing issue and give it a thumbs up instead of opening a new one.

#### How Do I Submit A (Good) Bug Report?

Bugs are tracked as [GitHub issues](https://guides.github.com/features/issues/). Create an issue and provide the following information by filling in [the issues form](https://github.com/GameServerManagers/LinuxGSM/issues/new/choose).

\* \*\*Use a clear and descriptive title\*\* for the issue to identify the problem.

\* \*\*Complete the user story\*\* to give a summary of the issue.

\* \*\*Provide basic info\*\* to help us understand the context of the issue.

\* \*\*Provide further info\*\* to give specifics and more detail.

\* \*\*Give steps to reproduce\*\* the issue, allowing developers to follow steps that lead to the issue.

\* \*\*Explain what you expect\*\* to happen, so we know what you think should occur.

### ?Suggesting Features

This section guides you through submitting a feature suggestion for LinuxGSM, including completely new features and minor improvements to existing functionality. Following these guidelines help maintainers and the community understand your suggestion ? and find related suggestions ?.

#### Before Submitting An Feature Suggestion

\* \*\*Check the\*\* [\*\*documentation\*\*](https://docs.linuxgsm.com/%5D(https://docs.linuxgsm.com/)) to confirm that the enhancement doesn’t already exist.

\* \*\*Check your\*\* [\*\*LinuxGSM version\*\*](https://docs.linuxgsm.com/commands/update-lgsm)\*\*.\*\* A newer version of LinuxGSM may already have your enhancement.

\* \*\*Perform a\*\* [\*\*cursory search\*\*](https://github.com/search?q=org:GameServerManagers%20type:issues&type=Issues) to see if the enhancement has already been suggested. If it has \*\*and the enhancement is still open\*\*, add a comment to the existing issue and give it a thumbs up instead of opening a new one.

#### How Do I Submit A (Good) Feature Suggestion?

Features are tracked as [GitHub issues](https://guides.github.com/features/issues/). Create an issue and provide the following information by filling in [the issues form](https://github.com/GameServerManagers/LinuxGSM/issues/new/choose).

\* \*\*Use a clear and descriptive title\*\* for the issue to identify the problem.

\* \*\*Complete the user story\*\* to give a summary of the issue.

\* \*\*Provide basic info\*\* to help us understand the context of the enhancement.

\* \*\*Provide further info\*\* to give specifics and more detail.

\* \*\*Provide any further reading\*\* materials that might assist in developing the enhancement.

### ? Game Server Requests

This section guides you through submitting a game server request for LinuxGSM, Following these guidelines help maintainers and the community understand your game server request ?.

#### Before Submitting a Game Server Request

\* \*\*Check for existing\*\* [\*\*game server requests\*\*](https://github.com/GameServerManagers/LinuxGSM/labels/type%3A%20game%20server%20request) to see if the new game server has already been suggested. If it has \*\*and if the new game server is still open\*\*, give it a thumbs.

\* \*\*Check the game server is supported on Linux\*\*, this does not include Wine servers which we do not support.

#### How Do I Submit A (Good) Game Server Request?

\* The title should be as follows: \*\*[Server Request] Game Name\*\*

\* \*\*Provide Steam App ID\*\* if applicable

\* \*\*Supply any documentation/how-to guides\*\* for the game server.

### ? Game Server Specific Issues

LinuxGSM is a management script that acts as a wrapper around game servers. These game servers are developed by different game developers such as Valve, Epic and Facepunch to name a few.

LinuxGSM has no control over the development and limited knowledge of issues directly relating to the game servers themselves. The same also applies for any mods, add-ons, maps etc.

If there is an issue with a specific game server or mod the best action may be to contact the game/mod developers on there support forums. If it is unclear some community members should be able to help.

A [list](https://docs.linuxgsm.com/support/game-server) of known game developer forums is available on the [LinuxGSM docs](https://docs.linuxgsm.com/support/game-server).

## ? Code Contributions

### Pull Requests

The process described here has several goals:

\* Maintain LinuxGSM quality.

\* Fix problems that are important to users.

\* Engage the community in working toward the best possible LinuxGSM.

\* Enable a sustainable system for LinuxGSM maintainers to review contributions.

Please follow these steps to have your contribution considered by the maintainers:

1. Follow all check-list in [the template](https://github.com/GameServerManagers/LinuxGSM/blob/master/.github/pull\_request\_template.md)

2. Follow the [style guides](#styleguides)

3. After you submit your pull request, verify that all [status checks](https://help.github.com/articles/about-status-checks/) are passing

What if the status checks are failing? If a status check is failing, and you believe that the failure is unrelated to your change, please leave a comment on the pull request explaining why you believe the failure is unrelated. A maintainer will re-run the status check for you. If we conclude that the failure was a false positive, then we will open an issue to track that problem with our status check suite.

While the prerequisites above must be satisfied before having your pull request reviewed, the reviewer(s) may ask you to complete additional design work, tests, or other changes before your pull request can be ultimately accepted.

#### Pull Request naming convention

When naming a pull request to ensure that it is following [Conventional Commits](https://www.conventionalcommits.org/) standards; as your pull request commits will be squashed, with the PR subject becoming the commit that is used for generating the [changelog](https://github.com/GameServerManagers/LinuxGSM/releases) for the next release.

The pull request subject line should always be able to complete the following sentence:

If applied, this commit will \_your subject line here\_

For example:

\* If applied, this commit will \*\*refactor subsystem X for readability\*\*

\* If applied, this commit will \*\*update getting started documentation\*\*

\* If applied, this commit will \*\*remove deprecated methods\*\*

\* If applied, this commit will \*\*release version 1.0.0\*\*

\* If applied, this commit will \*\*merge pull request #123 from user/branch\*\*

Notice how this doesn’t work for the other non-imperative forms:

\* If applied, this commit will \*\*fixed bug with Y\*\*

\* If applied, this commit will \*\*change the behaviour of X\*\*

\* If applied, this commit will \*\*more fixes for broken stuff\*\*

\* If applied, this commit will \*\*sweet new API methods\*\*

Below is an example of the subject line for a pull request:

\*\*feat(alerts): add slack support to alerts\*\*

\*\*fix(csgoserver): remove SteamCMD auth requirement 32-bit workaround\*\*

### Testing

#### Pull Request Status Checks

When a Pull Request is submitted, a series of status check tests are conducted. These tests will asses the code quality, complete CI tests etc. To get your PR merged these status checks must pass.

#### Test Environment

It is recommended that you have a testing environment available to test your code during development. To test your own code you must change some variables within the `linuxgsm.sh` file. This will force the use of your own code branch.

```bash

## GitHub Branch Select

# Allows for the use of different function files

# from a different repo and/or branch.

githubuser="GameServerManagers"

githubrepo="LinuxGSM"

githubbranch="master"

```

### :wine\_glass: Styleguides

#### Git Commit Messages

LinuxGSM uses the Conventional commits standard to allow other developers to get easy to understand, descriptive commit messages as you develop. While it is recommended that you use this standard for your commits, as your commits will eventually be squashed when your PR is merged following this standard is not strictly enforced for commits, however, it is recommended for more complex commits.

#### BASH Styleguide

LinuxGSM uses [ShellCheck](https://www.shellcheck.net/) to follow BASH best practices. It is recommended that you make use of linter tools for your text editor such as [linter-shellcheck](https://atom.io/packages/linter-shellcheck). LinuxGSM uses [Codacy](https://app.codacy.com/manual/GameServerManagers/LinuxGSM/dashboard) to analyse any Pull Requests to give you feedback on code standards.

LinuxGSM also has some of its style standards that should be followed. These are available in the [dev docs](https://dev-docs.linuxgsm.com/)

## :blue\_book: Document Contributions

As well as code contributions it is possible to contribute by writing and improving documentation. Documents contributions can be submitted similarly by submitting a Pull Request.

### Documentation Styleguide

LinuxGSM has various documentation available to assist [users](https://docs.linuxgsm.com) and [developers](dev-docs.linuxgsm.com). LinuxGSM primarily uses [GitBook](http://gitbook.com/) which uses the [Markdown](https://www.markdownguide.org/) document standard. LinuxGSM uses [Codacy](https://app.codacy.com/manual/GameServerManagers/LinuxGSM/dashboard) to analyse any Pull Requests to give you feedback on markup standards.

## Issue and Pull Request Labels

This section lists the labels we use to help us track and manage issues and pull requests.

[GitHub search](https://help.github.com/articles/searching-issues/) makes it easy to use labels for finding groups of issues or pull requests you're interested in. There are several categories of labels available:

\*\*command\*\* Labels

Highlights the LinuxGSM command the Issue/PR relates too.

\*\*info\*\* Labels

Labels to help pinpoint what the issue or PR relates too.

variants:

\* \_distro\_

\* \_engine\_

\* \_game\_

\* \_info\_

\*\*outcome\*\* Labels

Labels that identify why an issue was closed.

\*\*status\*\* Labels

Labels to update people on the status of the issue.

\*\*type\*\* Labels

Labels identifying the type of issue, such as a bug, feature, refactor etc.