GitHub Docs

Get started / Getting started with Git / Ignoring files

Ignoring files

In this article

Configuring ignored files for a single repository

Configuring ignored files for all repositories on your computer

Excluding local files without creating a .gitignore file

Further Reading

You can configure Git to ignore files you don't want to check in to GitHub Enterprise Server.

Mac Windows Linux

Configuring ignored files for a single repository

You can create a `.gitignore` file in your repository's root directory to tell Git which files and directories to ignore when you make a commit. To share the ignore rules with other users who clone the repository, commit the `.gitignore` file in to your repository.

GitHub maintains an official list of recommended `.gitignore` files for many popular operating systems, environments, and languages in the "github/gitignore" public repository. You can also use gitignore.io to create a `.gitignore` file for your operating system, programming language, or IDE. For more information, see "github/gitignore" and the "gitignore.io" site.

1. Open TerminalTerminalGit Bash.
2. Navigate to the location of your Git repository.
3. Create a `.gitignore` file for your repository.

   ```
   touch .gitignore
   ```

   If the command succeeds, there will be no output.

For an example `.gitignore` file, see "Some common .gitignore configurations" in the Octocat repository.

If you want to ignore a file that is already checked in, you must untrack the file before you add a rule to ignore it. From your terminal, untrack the file.

```
git rm --cached FILENAME
```

Configuring ignored files for all repositories on your computer

You can tell Git to always ignore certain files or directories when you make a commit in any Git repository on your computer. For example, you could use this feature to ignore any temporary backup files that your text editor creates.

To always ignore a certain file or directory, add it to a file named `ignore` that's located inside the directory `~/.config/git`. By default, Git will ignore any files and directories that are listed in the global configuration file `~/.config/git/ignore`. If the `git` directory and `ignore` file don't exist yet, you may need to create them.

Excluding local files without creating a *.gitignore* file

If you don't want to create a `.gitignore` file to share with others, you can create rules that are not committed with the repository. You can use this technique for locally-generated files that you don't expect other users to generate, such as files created by your editor.

Use your favorite text editor to open the file called `.git/info/exclude` within the root of your Git repository. Any rule you add here will not be checked in, and will only ignore files for your local repository.

1. Open TerminalTerminalGit Bash.
2. Navigate to the location of your Git repository.
3. Using your favorite text editor, open the file `.git/info/exclude`.

Further Reading

- Ignoring files in the Git documentation
- .gitignore in the Git documentation
- A collection of useful *.gitignore* templates in the github/gitignore repository
- gitignore.io site