

Programming language detection



- [Built-in language detection](#)
 - [A short explanation of categories](#)
- [Adjusting the language detection](#)
 - [Overriding the language](#)
 - [Overriding the category](#)
 - [Overriding detection](#)

Forgejo tries to detect the languages used in a repository, for each file, to use this information in a number of different ways. The most prominent use is for the display of the language statistics line at the top of the repository's home view, but the same information is also used to exclude generated and vendored files from diffs.



This comes with sensible defaults that should work for most cases, but it is not without faults or compromises, and there are scenarios where the detection needs a little guidance. Further below, we will explain how it can be influenced, but let's look at the defaults first!

Built-in language detection

Whenever the contents of a repository change, Forgejo will look at the files, and try to put them into one of the following categories: "Programming language", "Markup language", "Documentation", "Dotfile", "Configuration", "Generated", "Vendored". The sorting into categories is done by the [go-enry](#) package, please consult its documentation for the specifics. The library is *mostly* compatible with [linguist](#), which is used by some other forges. The statistics are calculated based on the main branch of the repository, only update when the main branch changes, and the update may take some time.

For the repository language statistics, with the default configuration, only those files are considered that match the "Programming language" or "Markup" language categories, everything else is ignored. When viewing a diff, all files are shown, but "Generated" files are collapsed by default, and "Vendored" files are marked as such.

A short explanation of categories

While some of the categories are rather straightforward, a little explanation about them does not hurt, especially if you are considering reclassifying files into another category.

Vendored files are any file you have checked into your repository that you didn't write, but imported from elsewhere. These may include dependencies you want a local copy of, at a specific version - such as JavaScript libraries or Go packages. These may inflate your project's language stats, and may even cause it to be labeled as another language. Marking these files as vendored makes it possible to ignore them for statistics.

Generated files are - as the name implies - files that are generated, but still checked into the repository for one reason or another. These may include minified JavaScript, compiled CoffeeScript, various lock files, and so on. Similar to vendored files, you usually do not want these to show up in language stats. They're - unlike vendored files - hidden by default when viewing diffs.

Documentation are just as the name says, documentation. This includes files written in Markdown, AsciiDoc, Org Mode, and a number of others.

Configuration files are typically used to configure software, and as such, the category is not included in language statistics. Languages like JSON, TOML, YAML, SQL, and XML - among other things - are considered documentation by default.

Dotfiles are files whose name starts with a dot, which by convention, suggests they should be hidden, and as such, they are excluded from language statistics.

Programming languages and **Markup languages** are more or less self-explanatory. The former category includes languages like C, Go, Rust, JavaScript, and many, many others. Markup languages are CSS, HTML, Jinja templates, Jupyter Notebooks, and numerous other formats.

Please consult the [enry](#) or [linguist](#) documentation for more details.

Adjusting the language detection

Sometimes the programming language of a file is not recognized properly, or it is miscategorized. Forgejo provides a mechanism where the language detection can be told about the language of a file, and its category can be adjusted as well. The same mechanism can also force a file to be considered for language statistics, regardless of its category - or the opposite, too: to tell Forgejo

never to consider it.

The way to do this is via a `.gitattributes` file. This file has a simple syntax where each line is made up of a pattern, followed by a space separated list of attributes. There are many attributes that git itself supports, but we're only going to talk about the custom attributes for language detection. All of these have a `linguist-` prefix, as that is where they originate from.

Overriding the language

In case Forgejo does not correctly recognize a file's language, you can use the `linguist-language` attribute to override its detection. The language names are case-insensitive, and may be specified using an alias. Spaces within language names must be replaced with hyphens.

An example showing these overrides:

```
# Reclassify `.pl` files as Prolog
*.pl    linguist-language=Prolog

# Whitespace in language names must be replaced with hyphens
*.glyphs linguist-language=OpenStep-Property-List

# Language names are case-insensitive, and may be specified using an alias.
# All of the following three lines are equivalent:
*.es    linguist-language=js
*.es    linguist-language=JS
*.es    linguist-language=JavaScript
```

Overriding the category

It is possible to mark files matching a pattern as "Documentation", "Generated", or "Vendored". This can be useful when the automatic detection of these fail, or if you want to reclassify files for some other reason. It is also possible to reclassify files that would be considered either of these, as not being "Documentation", "Generated", or "Vendored".

To achieve this, the `linguist-documentation`, `linguist-generated`, and `linguist-vendored` attributes can be used. All three of these are boolean, you can set them simply by listing the attribute name. You can also set an explicit value by setting them to `true` or `false`. Prefixing the attribute with a minus sign is the same as setting it to `false`.

It's best to illustrate this with a few examples!

```
# Do not consider Markdown files documentation anymore!
# Note: Both forms here are equivalent.
*.md -linguist-documentation
*.md  linguist-documentation=false

# Consider files in `dist/` generated.
# Both forms here are equivalent.
/dist/**/* linguist-generated
/dist/**/* linguist-generated=true

# Do not categorize `cpplint.py` as vendored.
cpplint.py -linguist-vendored
cpplint.py  linguist-vendored=false
```

Reclassifying a file will result in Forgejo proceeding with language detection. That may still result in the file not being considered for statistics. Take a look at this example:

```
*.nib -linguist-generated
*.nib  linguist-language=Markdown
```

This will classify `*.nib` files as non-generated, and as Markdown. However, Markdown is considered documentation, which is, by default, excluded from language statistics.

Overriding detection

In cases where a file should be considered for language statistics, regardless of its category, the `linguist-detectable` attribute can be used. The same attribute can be used to hide a file from the language statistics, without reclassifying it into a category that would otherwise be hidden.

For a repository whose primary contents are documentation in Markdown format, the following override would make Forgejo consider the Markdown files for the language statistics:

```
*.md linguist-detectable
```

Similarly, to hide a file from the language statistics:

```
config/app.js -linguist-detectable
```

The above will not consider the app's configuration - in JSON, but with a `.js` extension - for language statistics.

