NAME

gitmodules − defining submodule properties

SYNOPSIS

$GIT_WORK_DIR/.gitmodules

DESCRIPTION

The .gitmodules file, located in the top−level directory of a git working tree, is a text file with a syntax matching the requirements of **git-config**(1).

The file contains one subsection per submodule, and the subsection value is the name of the submodule. Each submodule section also contains the following required keys:

submodule.<name>.path
: Defines the path, relative to the top−level directory of the git working tree, where the submodule is expected to be checked out. The path name must not end with a /. All submodule paths must be unique within the .gitmodules file.

submodule.<name>.url
: Defines an url from where the submodule repository can be cloned.

submodule.<name>.update
: Defines what to do when the submodule is updated by the superproject. If *checkout* (the default), the new commit specified in the superproject will be checked out in the submodule on a detached HEAD. If *rebase*, the current branch of the submodule will be rebased onto the commit specified in the superproject. If *merge*, the commit specified in the superproject will be merged into the current branch in the submodule. This config option is overridden if *git submodule update* is given the *−−merge* or *−−rebase* options.

EXAMPLES

Consider the following .gitmodules file:

```
[submodule "libfoo"]
        path = include/foo
        url = git://foo.com/git/lib.git

[submodule "libbar"]
        path = include/bar
        url = git://bar.com/git/lib.git
```

This defines two submodules, libfoo and libbar. These are expected to be checked out in the paths *include/foo* and *include/bar*, and for both submodules an url is specified which can be used for cloning the submodules.

SEE ALSO

git-submodule(1) **git-config**(1)

DOCUMENTATION

Documentation by Lars Hjemli <**hjemli@gmail.com**[1]>

GIT

Part of the **git**(1) suite

NOTES

1. hjemli@gmail.com
   mailto:hjemli@gmail.com