Modules

Mark Woodhall

19-10-2024

Index `Org`

Mimis is made up of modules that can be enabled and setup.

Generally, the enable operation of a module will register required plugins and the setup operation will make sure they are in the right state to be used.

Every effort is made to try to keep these operations distict and free of terrible side effects, since when modules are wrapped as packages they may be enabled and setup multiple times.

Options

Sets many defaults for the editor.

Which Key

Discoverable keybindings. Bound to nvim.g.mapleader by default. Whichkey is a ”core” module used by many other modules in order to register bindings.

Colors

Manages colorschemes. Catppuccin and Tokyonight are available.

Keymaps

Sets up useful keybindings that don't rely on other modules or third party plugins.

Projects

Used to setup project related plugins. For example, vim-rooter which can be used to determine the root directory ”project” for the current file.

Git

Sets up plugins related to git source control. This currently includes vim-fugitive and neogit, when calling setup it is possible to specify which one to enable.

For example:

```
(package.setup {:modules.git [:fugitive]})
```

Quick fix

Sets up nvim-bqf, nvim better quick fix.

Status line

Sets up and configures lualine statusline plugin with a minimal set of blocks. Matches colors for default set of mimis themes.

Telescope

Sets up telescope with a consistent look and feel based on the emacs ivy style.

Various options can be enabled when setting up.

For example, you can configure bindings to be created for lsp, projects, finder, git, and buffers, like so:

```
:modules.telescope [:lsp :projects :finder :git :buffers]
```

Treesitter

Sets up nvim-treesitter for supplied languages.

Paredit

Sets up kovisoft/paredit for required languages, primarily of the lisp family.

Surround

Sets up tpope/vim-surround.

Lsp

Sets up a range of plugins related to lsp and completion. nvim-cmp is used for completion and nvim-lspconfig to setup various lsp servers.

For now, lsp and completion are coupled together, but this may change in the future. The configuration of nvim-cmp and lsp default capabilities make this a little awkward to do right now.

Repl

Repl is a module that doesn't rely on any other plugins. It uses neovims floating windows to setup jobs for running repls.

These jobs can then have data sent to them and evaluated. This could be considered somewhat like vim-slime.

Clojure, fennel, and Janet modules all make use of this.

The repl module makes a best guess at what process to start based on the language and/or project system in use for the current file and/or buffer.

Clojure

The clojure module relies on |mimis-modules-projects| and |mimis-modules-treesitter| in order to connect and send expressions to |mimis-modules-repl|.

It also sets up a range of keybindings that support the clojure reloaded workflow.

These bindings can be discovered with |mimis-modules-whichkey|.

Fennel

The fennel module relies on |mimis-modules-projects| and |mimis-modules-treesitter| in order to connect and send expressions to |mimis-modules-repl|.

It also sets up a range of keybindings for interacting with the repl.

These bindings can be discovered with |mimis-modules-whichkey|.

Janet

The janet module relies on |mimis-modules-projects| and |mimis-modules-treesitter| in order to connect and send expressions to |mimis-modules-repl|.

It also sets up a range of keybindings for interacting with the repl.

These bindings can be discovered with |mimis-modules-whichkeyZ.

SQL

The sql module relies on |mimis-modules-treesitter| and |mimis-modules-lsp|, it sets up tpope/vim-dadbod, kristijanhusak/vim-dadbod-ui, and kristijanhusak/vim-dadbod-completion

vim-dadbod-completion is set up as an nvim-cmp completion source.

Cmdline

Common

The common cmdline module makes use of tpope/vim-eunuch to provide commands like :Remove and also provides other user commands, that will run an executable using :terminal and provide the output as a bottom panel. An example of these commands include :Tail.

Aws

As per the `common` module but wraps various aws cli commands.

Npm

As per the `common` module but wraps various aws cli commands.

Docker

As per the `common` module but wraps various docker commands.