

现代 Neovim 折腾指南

什么是 Neovim

Neovim 是 Vim 的一个 fork, 在 Vim 支持多线程的补丁被拒绝后, Neovim 项目于 2014 年启动。


- Neovim 全面兼容 Vim9.0 之前的配置, 只不过配置文件放置的位置不同
 - Neovim: `~/.config/nvim/init.[lua|vim]`
 - Vim: `~/.vim/vimrc` 或者 `~/.vimrc`
- Neovim 内置对 Lsp 协议 (语法补全) 和 Treesitter (语法高亮) 的支持
- Neovim 可以使用 lua 脚本语言进行配置, 而 Vim 只能使用 vimL

ps: 在之后所提到的命令, 如无特殊说明, Neovim 和 Vim 均适用

Neovim 插件系统介绍

加载插件实际上在干什么？

- 加载 Neovim 的 脚本文件
 - set runtimepath 会列出 Neovim 下查找脚本的路径
 - 在运行时，Neovim 自动加载这些路径下的 *.vim, *.lua 文件



The screenshot shows a terminal window with the following text:

```
  
runtimepath=~/.vim,~/.vim/pack/foo/start/test,~/.vim/pack/foo/start/surround,/usr/share/vim/vimfiles,/usr/share/vim/vi  
n91,/usr/share/vim/vimfiles/after,~/.vim/after  
Press ENTER or type command to continue
```

如何加载插件

- Vim8.0 之后提供原生的加载第三方插件的功能
 - ~/.vim/pack/[foobar]/start: 自动加载
 - ~/.vim/pack/[foobar]/opt: 按需加载

需要手动 clone, upgrade 😊, 不方便

- 使用包管理器
 - vim-plug 适用于 Vim 和 Neovim
 - packer.nvim 仅适用于 Neovim (unmaintained)
 - lazy.nvim 仅适用于 Neovim (Packer 作者亲情推荐)

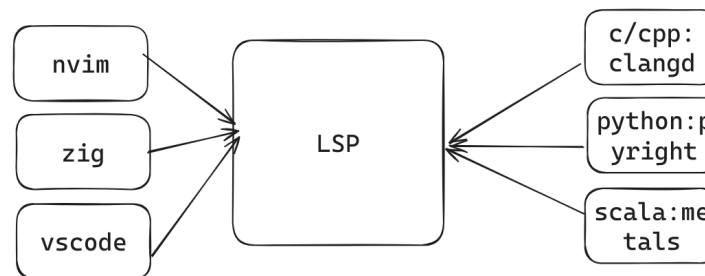
代码补全

代码补全基础: LSP 介绍

语言服务器协议 (LSP) 是一种基于 JSON-RPC 的开放协议，用于源代码

LSP 最初是微软为Vscode开发的，现在已经成为开放标准

- 在编辑器和编程语言之间加了一层抽象, 一个不恰当的比方
 - 编辑器对应着NEMU, NPC
 - 各种语言的语言服务器对应着AM在不同架构下的实现
 - LSP对应AM的统一的API
- 这样, 只要编辑器支持 LSP 协议，那么就可以复用各种语言服务器所带来的高级特性，比如重命名，linting 这类功能。



Neovim 下进行代码补全

使用插件提供的 LSP 支持

- [coc.nvim](#)(不支持 Metals):
 - 基于NodeJS 的适用于Vim8, Neovim 的Vim 智能补全插件, 配置风格类似Vscode
 - 虽然coc 是 vim 的一个插件, 但是他也有自己的插件系统

使用 Neovim 内置的 LSP

- [nvim-lspconfig](#), [mason.nvim](#), [mason-lspconfig.nvim](#) (不支持 metals):
 - nvim-lspconfig 用于配置 Neovim LSP
 - mason 用于管理 语言服务器
 - mason-lspconfig 用于兼容 nvim-lspconfig 和 mason
- [nvim-metals](#): (可以和上面的插件并存) 单独为 metals 提供服务

配置分享

我的配置

基本移动

键位映射

^			^	
k			i	
< h	l >	-->	< j	l >
j			k	
v			v	

配置代码

```
noremap <silent> i k  
noremap <silent> j h  
noremap <silent> k j  
noremap <silent> z i
```

这里提供 vimL 的配置方式，lua 的配置方法请看代码仓库/文档

窗口管理

键位映射

- s + u/d/l/r: 向上/下/左/右 分屏
- <Space> + 方向键: 切换光标所在屏幕

配置代码

```
map sr :set splitright<CR>:vsplit<CR>
map sl :set nosplitright<CR>:vsplit<CR>
map su :set nosplitbelow<CR>:split<CR>
map sd :set splitbelow<CR>:split<CR>
noremap <SPACE><up>    <C-w>k
noremap <SPACE><left>   <C-w>h
noremap <SPACE><down>   <C-w>j
noremap <SPACE><right>  <C-w>l
```

Tab 管理

键位映射

- <space> + n 新建一个tab
- th 访问右侧的 tab
- tl 访问左侧的 tab

配置代码

```
map <leader>n :tabnew<CR>  
map tl :tabnext<CR>  
map th :tabprevious<CR>
```

查找

Telescope: 快速查找

- C-p 查找文件
- C-f 查找文件内容
- C-h 查找历史文件
- C-l 查找内置命令

```
-- lua/config/plugins/telescope.lua
local builtin = require('telescope.builtin')
vim.keymap.set("n", "<C-l>", builtin.commands, m)
vim.keymap.set('n', '<c-p>', builtin.find_files, m)
vim.keymap.set('n', '<c-h>', builtin.oldfiles, m)
-- ...
```

LSP

- gd 跳转到定义
- <leader>- 跳转到上一个诊断
- <leader>+ 跳转到下一个诊断
- <leader>h 查看文档
- <leader>rn 重命名

```
-- lua/config/plugins/lspconfig.lua
vim.keymap.set('n', '<leader>h', show_documentation, opts)
vim.keymap.set('n', 'gd', vim.lsp.buf.definition, opts)
vim.keymap.set('n', '<leader>rn', vim.lsp.buf.rename, opts)
vim.keymap.set('n', '<leader>- ', vim.diagnostic.goto_prev, opts)
vim.keymap.set('n', '<leader>= ', vim.diagnostic.goto_next, opts)
```

Metals 配置

- 下载 `courier`, 并将其添加到 PATH 中
- 可能需要配置 Java 代理

```
## ~/.bashrc  
export JAVA_OPTS="-Dhttps.proxyHost=127.0.0.1 -Dhttps.proxyPort=
```

- 安装 `nvim-metals` 插件, 并进行正确配置(查阅官方文档或我的配置仓库)
- 打开 `npc` 工作目录下的 `build.sc` 或 `*.scala` 文件
- 如果没有自动安装 `metals`, 可以手动执行 `:MetalsInstall`

想要 config from scratch?

人生苦短，还是用框架把

- [Lazyvim](#)
 - 没错，就是Lazy.nvim包管理器的作者维护的
- [AstroNvim](#)
 - 社区相当活跃
- [NvChad](#)
- [LunarVim](#)

这些都是由社区维护的 Nvim 配置，文档详细，你可以在这些框架的基础上再添加自己想要的功能

就想写个 chisel, 别整这么多有的没得

参考配置

支持 nvim-metals 插件的简洁配置

- 仅包含 lsp, 语法高亮, 和一个文件树插件
 - `minimum-configuration`

```
plugin/  
  catppuccin.lua --> 主题  
  completion.lua  
  lsp-config.lua  
  none-ls.lua  
  metals.lua      ---> lsp  
  neotree.lua     ---> 文件树 (`tt` 打开/关闭 文件树)  
  telescope.lua  ---> 查找  
  treesitter.lua ---> 语法高亮
```

彩蛋

小 tips

- % 可以跳转到另一个配对括号
- :term 可以在 Neovim 下打开终端
- 在 insert 模式下按 C-w 可以向前删除一个 word
- :echo expand('%:p') 可以打印当前文件的绝对文件路径