



Neovim

новая эпоха консольных редакторов

Панфилов Александр

Давайте знакомиться



- 7 лет в ИТ разработке, последние 4 года на позиции фронтенд разработчика
- Департамент ИТ блока «Корпоративно-инвестиционный бизнес», команды РФИКОД, УФН.Джо-Джо
- Любимые технологии: React, Next, Nest, CI/CD, MongoDB, MySQL

О чём поговорим

- Какие были редакторы ранее?
- Что такое neovim?
- В чём отличие neovim от vim?
- Минусы в готовых решениях.
- Переход со стандартного редактора кода на neovim?
- Будущее за neovim и vscode?
- Плюсы и минусы neovim.

Какие были редакторы ранее?





Что такое neovim?

The screenshot shows a Neovim window with a dark theme. On the left is a file tree for a Next.js project named 'admin-my-site'. The 'src' directory contains 'commons', 'components', 'entities', 'font', 'locales', and 'pages'. The 'pages' directory has subfolders 'Admin' and 'Page', with 'Page' containing '404Page' and 'Article'. '404Page' has files '_tests_.tsx', '404Page.tsx', 'index.ts', and 'style.ts'. 'Article' has files '_tests_.tsx', 'Article.tsx', 'api.ts', 'index.ts', and 'style.ts'. 'Articles' has files '_tests_.tsx' and 'Articles.tsx'. The main editor pane displays a TypeScript file 'Article.tsx' with code related to handling article responses. A tooltip is visible at the bottom of the editor, pointing to the word 'toast' with the message '● Cannot find name 'toast''. The status bar at the bottom shows the mode as 'NORMAL', the buffer as 'nextjs', the current file as 'src/.../Article/Article.tsx', and the cursor position as 'useEffect() callback'. Other status bar items include file counts (1), search count (23), and a timestamp (42:27).

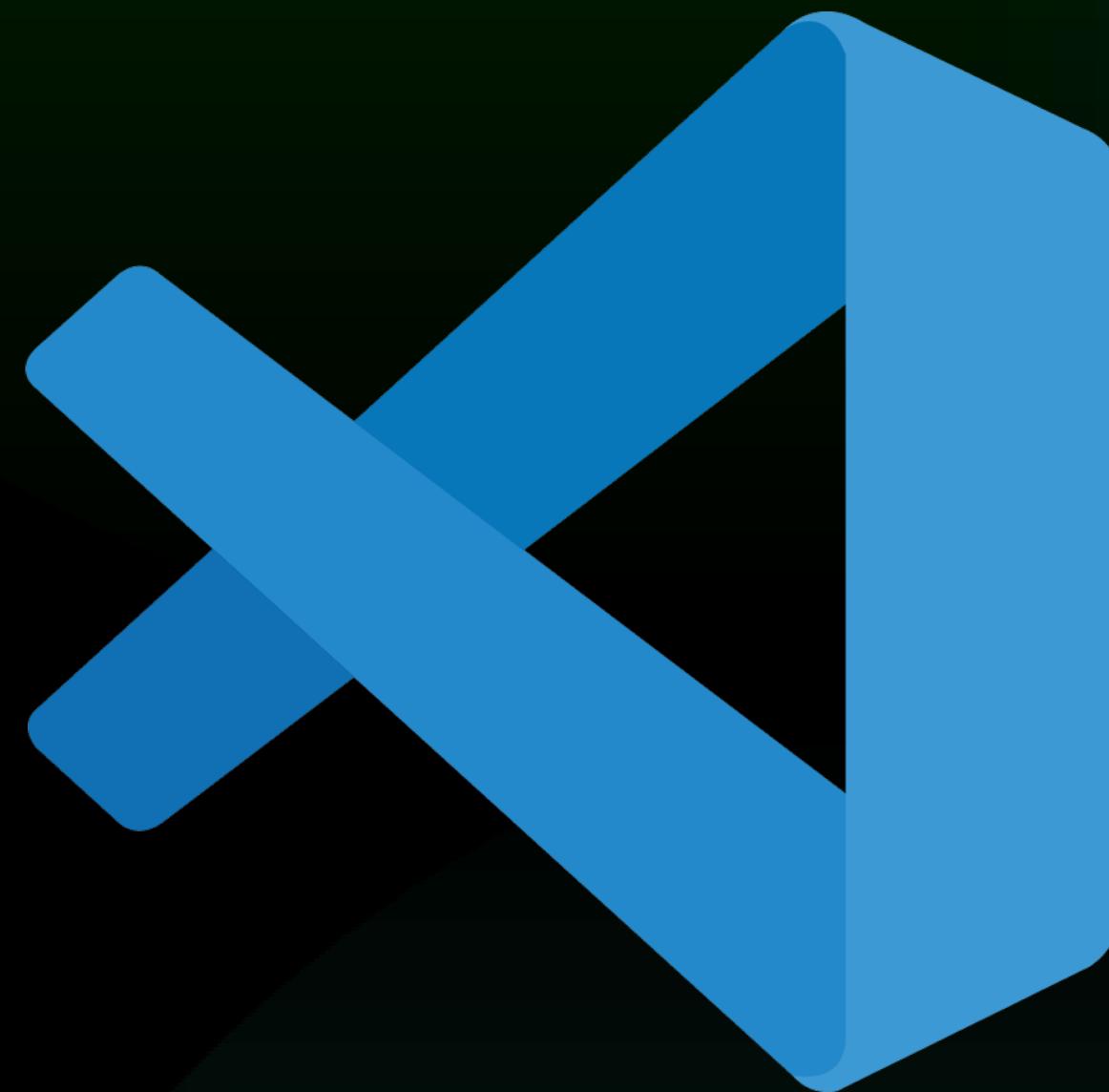
```
const { t, language } = useLanguage();
const { is360, is481 } = useIsMediaQuery();
// если в ответе есть responseBody (статья), то сразу ее отрисовываем
const [article, setArticle] = useState<ArticleI | null>(
  response?.responseBody ? response?.responseBody : null
);

const { imageUrl } = useImageName({
  imageName: article?.thumbnail,
});
const styles = useStyle();
const stylesPage = useStylesClasses({ theme: { is360, is481 } });

// сайд эффект остался для сценария, когда статья не загрузилась и в этом случае необходимо
// ошибка выводится в виде сообщения попада в правом верхнем углу
useEffect(() => {
  if (!response.status) {
    toast(t(response.message as string), {
      type: "error", // (property) ResponseI<void | ArticleI>.message?: string
      hideProgressBar: true,
      theme: "colored",
    });
  }
  if (response.redirectTo) {
    push(response.redirectTo);
  }
  return;
}

if (response.responseBody) {
```

Наиболее популярные редакторы кода



★ 164к



☆ 82k



☆ 36k

В чем отличие neovim от vim?



VS



В чем отличие neovim от vim?

- Повышенная производительность и стабильность по сравнению с Vim;
- Поддержка асинхронных плагинов и скриптов;
- Улучшенная поддержка современных фреймворков пользовательского интерфейса и символов Unicode;
- Улучшенная интеграция с терминалом и поддержка пользовательского интерфейса.

Минусы в готовых решениях?

- Долго загружается информация в рабочем окружении;
- Мало горячих клавиш;
- Платный софт;
- Плохая поддержка конфигурационных файлов.

Режимы работы в neovim

```
<!doctype html>
<html lang="ru">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Анимация карточек для товаров</title>
    <link rel="stylesheet" href=".css/font.css" />
    <link rel="stylesheet" href=".css/base.css" />
    <link rel="stylesheet" href=".css/animateCardProducts.css" />
  </head>
  <body>
    <main class="animateCardCategory">
      <h1>Анимация карточек для Товаров</h1>

      <div class="animateCardCategoryWrapper">
        <div class="animateCardCategoryContainer">
          <h2>Пример анимации (интерактивный)</h2>
          @@include('chunk/animateCardProducts/card.html')
        </div>
    
```

Normal

Режимы работы в neovim

```
<!doctype html>
<html lang="ru">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Анимация карточек для товаров</title>
    <link rel="stylesheet" href=".css/font.css" />
    <link rel="stylesheet" href=".css/base.css" />
    <link rel="stylesheet" href=".css/animateCardProducts.css" />
  </head>
  <body>
    <main class="animateCardCategory">
      <h1>Анимация карточек для Товаров</h1>

      <div class="animateCardCategoryWrapper">
        <div class="animateCardCategoryContainer">
          <h2>Пример анимации (интерактивный)</h2>
          @@include('chunk/animateCardProducts/card.html')
        </div>
    
```

Insert

Режимы работы в neovim

```
<!doctype html>
<html lang="ru">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Анимация карточек для товаров</title>
    <link rel="stylesheet" href=".css/font.css" />
    <link rel="stylesheet" href=".css/base.css" />
    <link rel="stylesheet" href=".css/animateCardProducts.css" />
  </head>
  <body>
    <main class="animateCardCategory">
      <h1>Анимация карточек для Товаров</h1>

      <div class="animateCardCategoryWrapper">
        <div class="animateCardCategoryContainer">
          <h2>Пример анимации (интерактивный)</h2>
          @@include('./chunk/animateCardProducts/card.html')
        </div>
      </div>
    </main>
  </body>
</html>
```

Visual

```
9  keymap.set("n", "<C-a>", "gg<S-v>G")
8
7  -- jump list
6  keymap.set("n", "<C-m>", "<C-i>", opt)
5
4  -- new tab
3  keymap.set("n", "te", ":tabedit<Return>", opt)
2  keymap.set("n", "<tab>", ":tabnext<Return>", opt)
1  keymap.set("n", "<s-tab>", ":tabprev<Return>", opt)
30
1  -- split window
2  keymap.set("n", "ss", ":split<Return>", opt)
3  keymap.set("n", "sv", ":vsplit<Return>", opt)
4
5  -- move window
6  keymap.set("n", "sh", "<C-w>h")
7  keymap.set("n", "sk", "<C-w>k")
8  keymap.set("n", "sj", "<C-w>j")
9  keymap.set("n", "sl", "<C-w>l")
10
11 -- resize window
12 keymap.set("n", "<C-w><left>", "<C-w><"")
13 keymap.set("n", "<C-w><right>", "<C-w>>")
14 keymap.set("n", "<C-w><up>", "<C-w>+")
15 keymap.set("n", "<C-w><down>", "<C-w>-")
```

Neovim позволяет
переназначать или
добавлять
собственные
горячие клавиши

Пакетные менеджеры для neovim

-  Lazy.nvim Packer

```
~/.config/nvim/init.lua
```

```
require("config.lazy")
```

```
~/.config/nvim/lua/config/lazy.lua
```

```
-- Bootstrap lazy.nvim
local lazypath = vim.fn.stdpath("data") .. "/lazy/lazy.nvim"
if not (vim.uv or vim.loop).fs_stat(lazypath) then
    local lazyrepo = "https://github.com/folke/lazy.nvim.git"
    local out = vim.fn.system({ "git", "clone", "--filter=blob:none", "--branch=stable", lazyrepo, lazypa
if vim.v.shell_error ~= 0 then
    vim.api.nvim_echo({
        { "Failed to clone lazy.nvim:\n", "ErrorMsg" },
        { out, "WarningMsg" },
        { "\nPress any key to exit...", "NormalMsg" },
    }, true, {})
    vim.fn.getchar()
    os.exit(1)
end
vim.opt.rtp:prepend(lazypath)

-- Make sure to setup `mapleader` and `maplocalleader` before
-- loading lazy.nvim so that mappings are correct.
-- This is also a good place to setup other settings (vim.opt)
vim.g.mapleader = " "
vim.g.maplocalleader = "\\\""

-- Setup lazy.nvim
require("lazy").setup({
    spec = {
        -- import your plugins
        { import = "plugins" },
    },
    -- Configure any other settings here. See the documentation for more details.
    -- colorscheme that will be used when installing plugins.
    install = { colorscheme = { "habamax" } },
    -- automatically check for plugin updates
    checker = { enabled = true },
})
```

Инициализация lazy

```
● neotree.lua  x | ● image.lua  ●
 9   return {
8     {
7       "vhyrro/luarocks.nvim",
6       priority = 1001, -- this plugin needs to run before anything else
5       opts = {
4         hererocks = true,
3         rocks = { "magick" },
2       },
1       config = true,
10      },
1      {
2        "3rd/image.nvim",
3        dependencies = { "luarocks.nvim" },
4        config = function()
5          require("image").setup({
6            backend = "kitty",
7            tmux_show_only_in_active_window = true,
8            max_height_window_percentage = 50,
9            hijack_file_patterns = { "*.png", "*.jpg", "*.jpeg", "*.gif", "*.webp", "*.svg" },
10           })
11         end,
12       },
13     }

```

Пример инициализации плагина на neovim

```
local ensure_packer = function()
    local fn = vim.fn
    local install_path = fn.stdpath('data')..'/site/pack/packer/start/packer.nvim'
    if fn.empty(fn.glob(install_path)) > 0 then
        fn.system({'git', 'clone', '--depth', '1', 'https://github.com/wbthomason/packer.nvim', install_path})
        vim.cmd [[packadd packer.nvim]]
    end
    return true
end

return require('packer').startup(function(use)
    use 'wbthomason/packer.nvim'
    -- My plugins here
    -- use 'foo1/bar1.nvim'
    -- use 'foo2/bar2.nvim'

    -- Automatically set up your configuration after cloning packer.nvim
    -- Put this at the end after all plugins
    if packer_bootstrap then
        require('packer').sync()
    end
end)
```

Инициализация packer

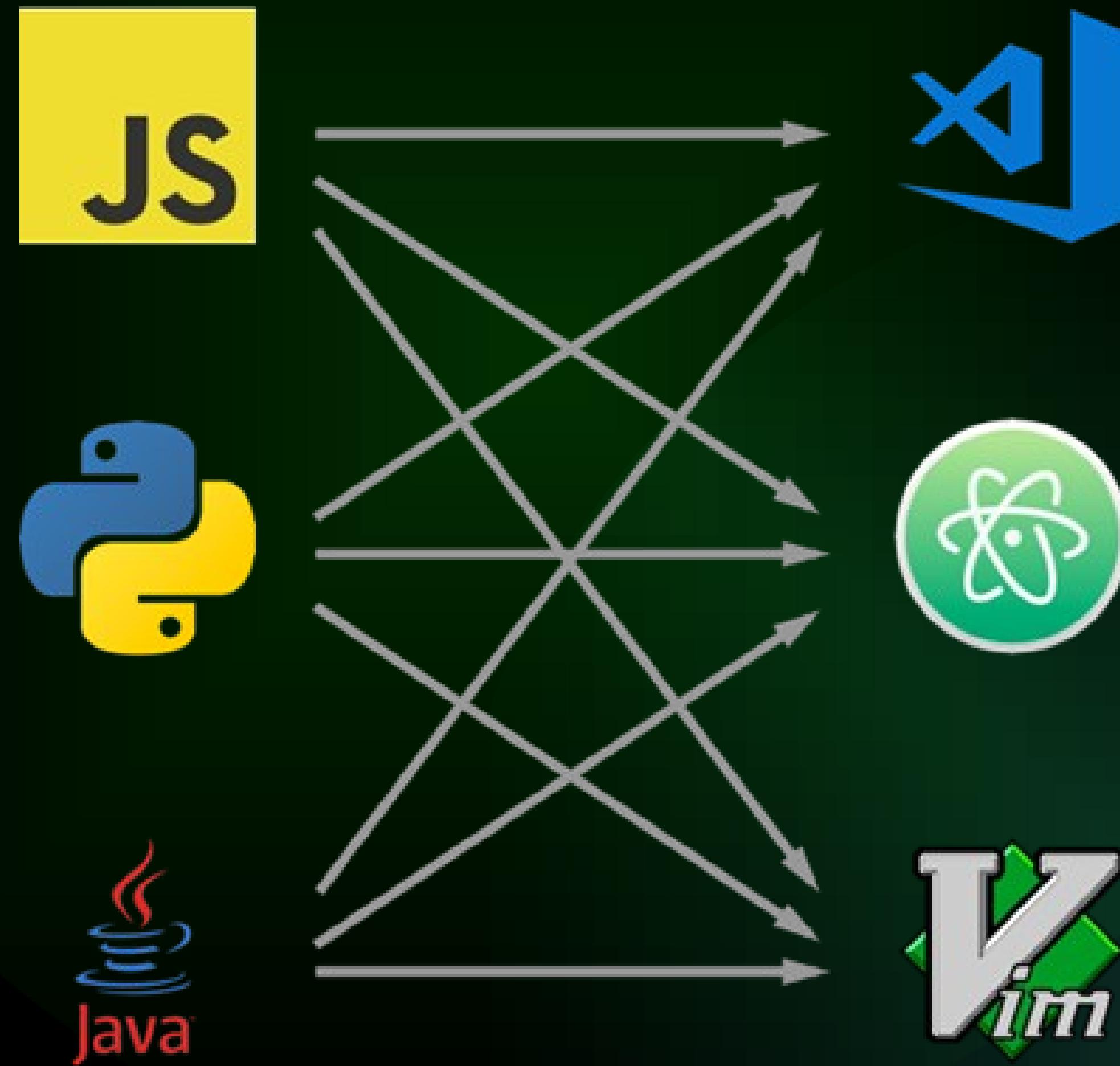
Командные оболочки



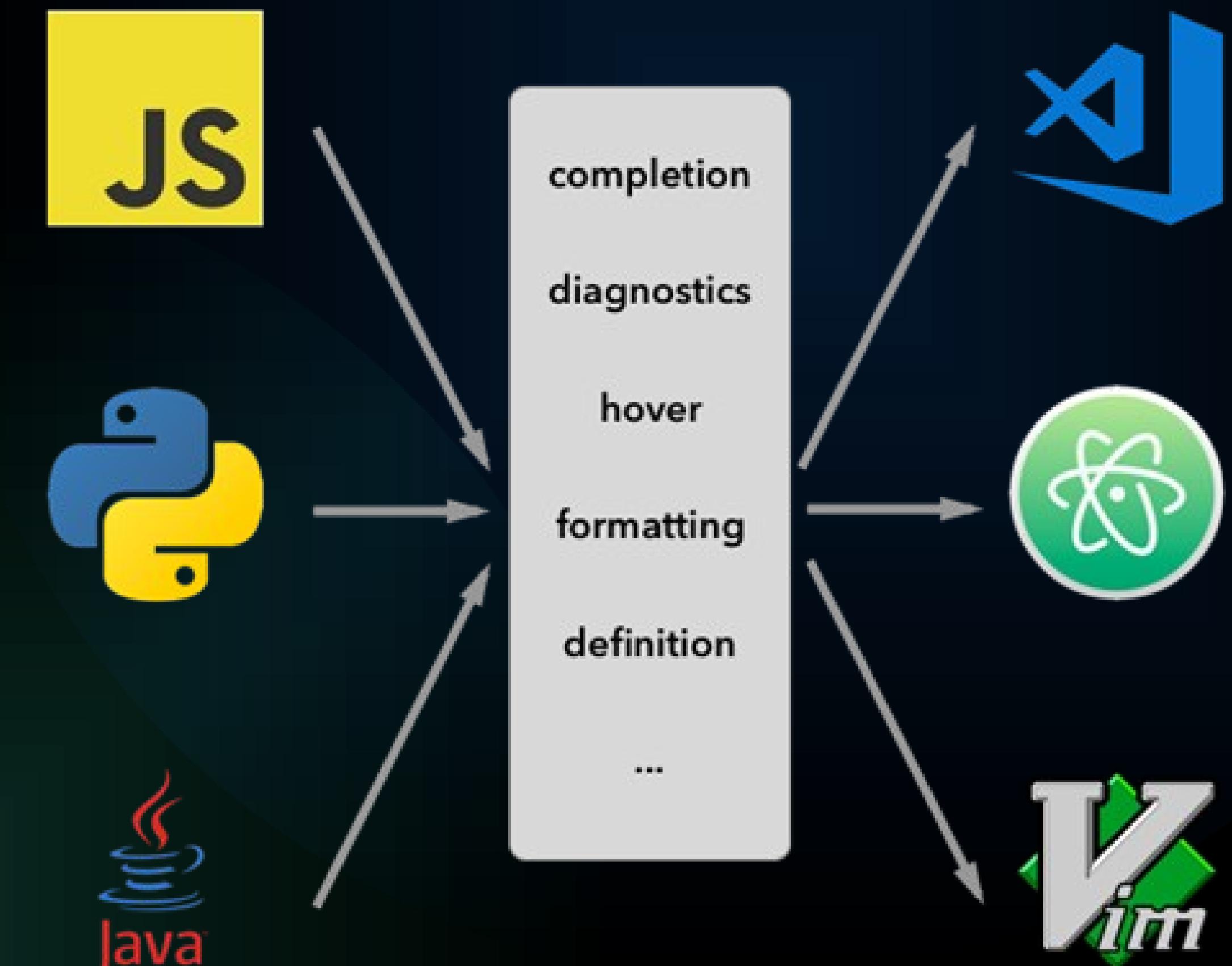
LSP

Lsp — language server protocol, языковой серверный протокол позволяет редактору напрямую общаться с языковым сервером, получать от него ошибки и подсвечивать их в коде и т.п.

NO LSP



LSP



Мультиплексоры



tmux



iTerm2 

```
> nvim
File: ~/Рабочий стол/gulp-html
0
> .git
> config
> data
> node_modules
> public
> src
>   css
>   font
>   html
>     chunk
>     icons
>     animateCardProducts.html
>     animateForCategory.html
>     index.html
>     styleguide.html
>   img
>     coal.png
>     forest-nature-path-woodland-wallpaper-1920x1080_ik03d
>     main1000_forest_3000x1875_ik03d
>     map.png
>     фон.png
>   js
>   scss
> task
  .gitignore
  gulpfile.js
  jsconfig.json
  package-lock.json
  package.json
  readme.md
```



Отображение картинки в консольном редакторе,
который открыт через kitty

Ключевые плагины для neovim

Neotree

Telescope

Mason

Treesitter

```

12 int tslua_add_language(lua_State *L)
11 {
10  if (lua_gettop(L) < 2 || !lua_isstring(L, 1) || !lua_isstring(L, 2)) {
9   return luaL_error(L, "string expected");
8 }
7
6 const char *path = lua_tostring(L, 1);
5 const char *lang_name = lua_tostring(L, 2);
4
3 if (pmap_has(cstr_t)(langs, lang_name)) {
2  return 0;
1 }

140 ■
1 #define BUFSIZE 128
2 char symbol_buf[BUFSIZE];
3 snprintf(symbol_buf, BUFSIZE, "tree_sitter_%s", lang_name);
4 #undef BUFSIZE
5
6 uv_lib_t lib;
7 if (uv_dlopen(path, &lib)) {
8  snprintf((char *)I0buff, IOSIZE, "Failed to load parser: uv_dlopen: %s",
9   uv_dlerror(&lib));
10 uv_dlclose(&lib);
11 lua_pushstring(L, (char *)I0buff);
12 return luaL_error(L);
13 }

14
15 TSLanguage *(*lang_parser)(void);
16 if (uv_dlsym(&lib, symbol_buf, (void **)&lang_parser)) {
17  snprintf((char *)I0buff, IOSIZE, "Failed to load parser: uv_dlsym: %s",
18   uv_dlerror(&lib));
19  uv_dlclose(&lib);
20  lua_pushstring(L, (char *)I0buff);
21  return luaL_error(L);
22 }

23
24 TSLanguage *lang = lang_parser();
25 if (lang == NULL) {
26  return luaL_error(L, "Failed to load parser: internal error");
27 }

28
29 pmap_put(cstr_t)(langs, xstrdup(lang_name), lang);
30
31 lua_pushboolean(L, true);
32 return 1;
33 }
34

```

```

12 int tslua_add_language(lua_State *L)
11 {
10  if (lua_gettop(L) < 2 || !lua_isstring(L, 1) || !lua_isstring(L, 2)) {
9   return luaL_error(L, "string expected");
8 }
7
6 const char *path = lua_tostring(L, 1);
5 const char *lang_name = lua_tostring(L, 2);
4
3 if (pmap_has(cstr_t)(langs, lang_name)) {
2  return 0;
1 }

140 ■
1 #define BUFSIZE 128
2 char symbol_buf[BUFSIZE];
3 snprintf(symbol_buf, BUFSIZE, "tree_sitter_%s", lang_name);
4 #undef BUFSIZE
5
6 uv_lib_t lib;
7 if (uv_dlopen(path, &lib)) {
8  snprintf((char *)I0buff, IOSIZE, "Failed to load parser: uv_dlopen: %s",
9   uv_dlerror(&lib));
10 uv_dlclose(&lib);
11 lua_pushstring(L, (char *)I0buff);
12 return luaL_error(L);
13 }

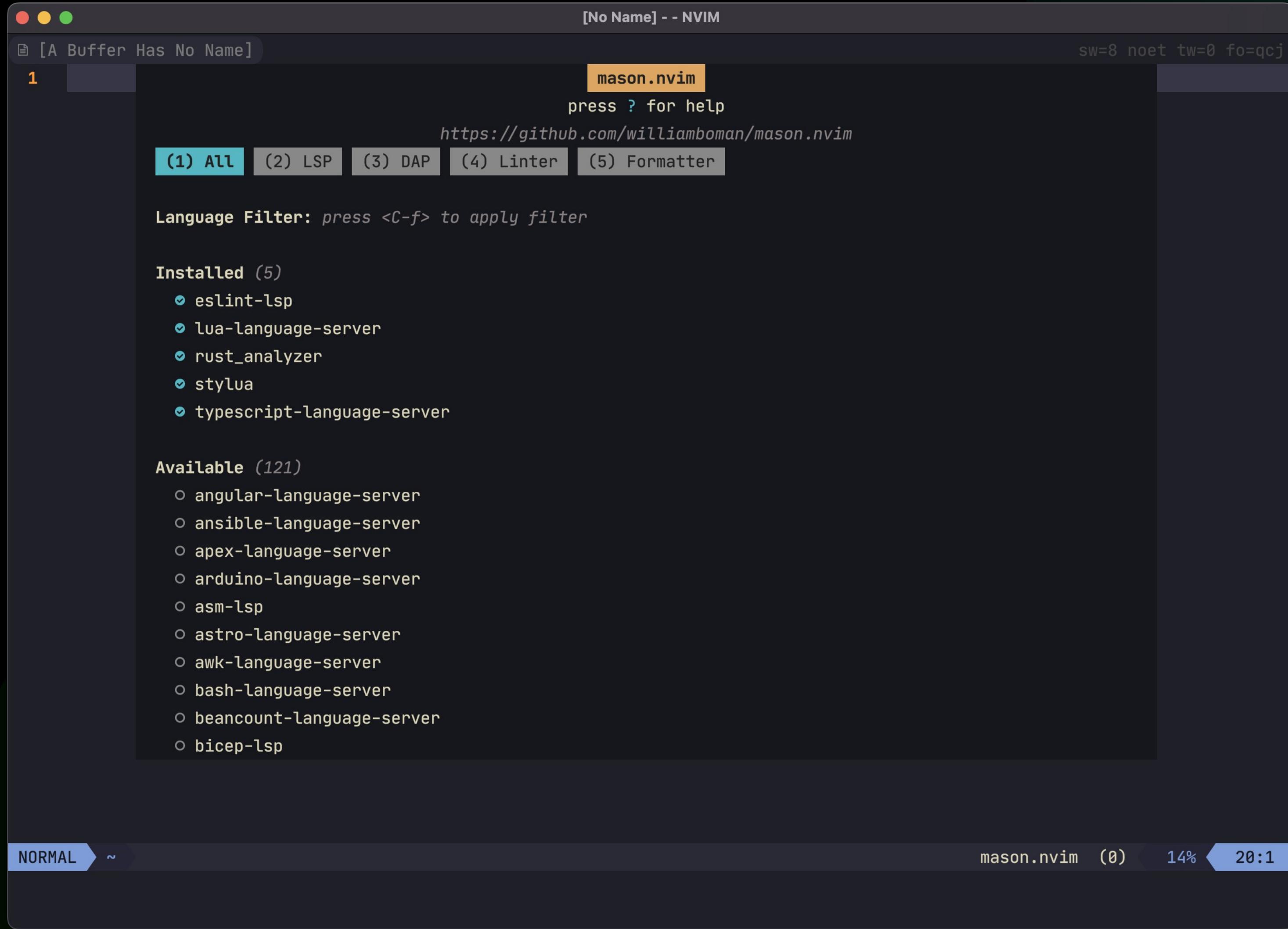
14
15 TSLanguage *(*lang_parser)(void);
16 if (uv_dlsym(&lib, symbol_buf, (void **)&lang_parser)) {
17  snprintf((char *)I0buff, IOSIZE, "Failed to load parser: uv_dlsym: %s",
18   uv_dlerror(&lib));
19  uv_dlcose(&lib);
20  lua_pushstring(L, (char *)I0buff);
21  return luaL_error(L);
22 }

23
24 TSLanguage *lang = lang_parser();
25 if (lang == NULL) {
26  return luaL_error(L, "Failed to load parser: internal error");
27 }

28
29 pmap_put(cstr_t)(langs, xstrdup(lang_name), lang);
30
31 lua_pushboolean(L, true);
32 return 1;
33 }
34

```

Treesitter



Mason

Order by

Press the corresponding key to execute the command.
Press <Esc> to cancel.

		Size ▾	Type	Last Modified
>	tests/		- directory	2022-12-25 10:09 PM
>	lua/		- directory	2023-08-17 05:44 PM
<	neo-tree/		- directory	2023-08-26 11:27 AM
>	ui/		- directory	2023-08-26 11:00 AM
>	sources/		- directory	2023-08-17 05:44 PM
<	common/		- directory	2023-08-26 11:00 AM
>	filters/		- directory	2023-07-05 06:03 PM
•	commands.lua	25.48 KB	lua	2023-08-26 11:00 AM
•	components.lua	14.9 KB	lua	2023-08-26 11:00 AM
•	preview.lua	12.19 KB	lua	2023-08-06 11:21 AM
•	container.lua	11.26 KB	lua	2023-08-26 11:00 AM
•	file-items.lua	7.44 KB	lua	2023-08-26 11:00 AM
•	help.lua	3.44 KB	lua	2023-08-26 11:00 AM
•	node_expander.lua	2.84 KB	lua	2023-08-06 11:21 AM
•	file-nesting.lua	1.29 KB	lua	2022-12-25 10:09 PM
>	buffers/		- directory	2023-08-06 11:21 AM
>	document_symbols/		- directory	2023-07-05 06:03 PM
>	filesystem/		- directory	2023-08-17 05:44 PM
>	git_status/		- directory	2023-08-06 11:21 AM
L	manager.lua	19.81 KB	lua	2023-08-17 05:44 PM
>	setup/		- directory	2023-08-06 11:21 AM
>	git/		- directory	2023-08-06 11:21 AM
>	command/		- directory	2023-08-06 11:21 AM
>	events/		- directory	2023-07-05 06:03 PM
>	utils/		- directory	2023-08-26 11:00 AM
•	defaults.lua	27.67 KB	lua	2023-08-26 11:27 AM
•	log.lua	4.87 KB	lua	2022-12-25 10:09 PM
•	collections.lua	2.79 KB	lua	2023-03-18 10:27 PM
•	neo-tree.lua	2.42 KB	lua	2023-08-17 05:44 PM
>	doc/		- directory	2023-08-26 11:27 AM
>	plugin/		- directory	2023-08-06 11:21 AM
>	scripts/		- directory	2023-01-12 08:13 PM
•	README.md	31.19 KB	md	2023-08-26 11:27 AM
•	CONTRIBUTING.md	2.59 KB	md	2023-08-06 11:21 AM
•	release.sh	1.11 KB	sh	2022-12-25 10:09 PM
•	LICENSE	1.1 KB	file	2022-12-25 10:09 PM
•	Dockerfile	645 B	file	2022-12-25 10:09 PM
•	Makefile	365 B	file	2022-12-25 10:09 PM
	(9 hidden items)			

Neotree

Preview

```
# telescope.nvim

Gaze deeply into unknown regions using the power of the moon.

## What is Telescope?

Telescope is a highly extendable fuzzy finder over lists. Items are sh
own in a popup with a prompt to search over.
:
```

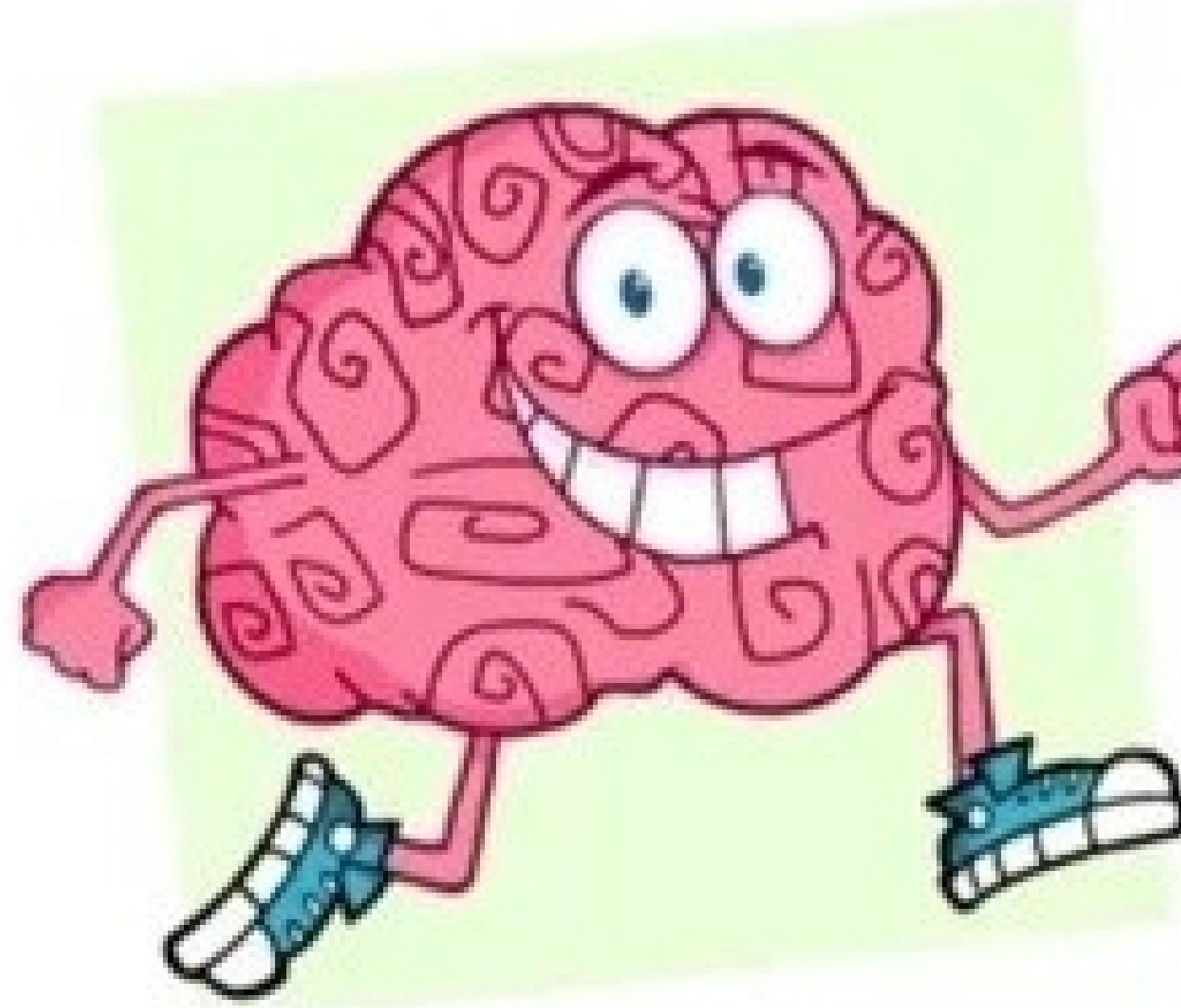
Find Files

82 / 82

- > README.md
- data/help/tags
- data/memes/planets/earth
- data/memes/planets/jupiter
- data/memes/planets/mars
- data/memes/planets/mercury
- data/memes/planets/moon
- data/memes/planets/neptune
- data/memes/planets/pluto
- data/memes/planets/saturn
- data/memes/planets/uranus
- data/memes/planets/venus
- data/options/options.lua
- full_line_completion.lua
- lua/telescope/WIP.lua

Insert [No Name] 0,1
Theme Dropdown 1,1 All

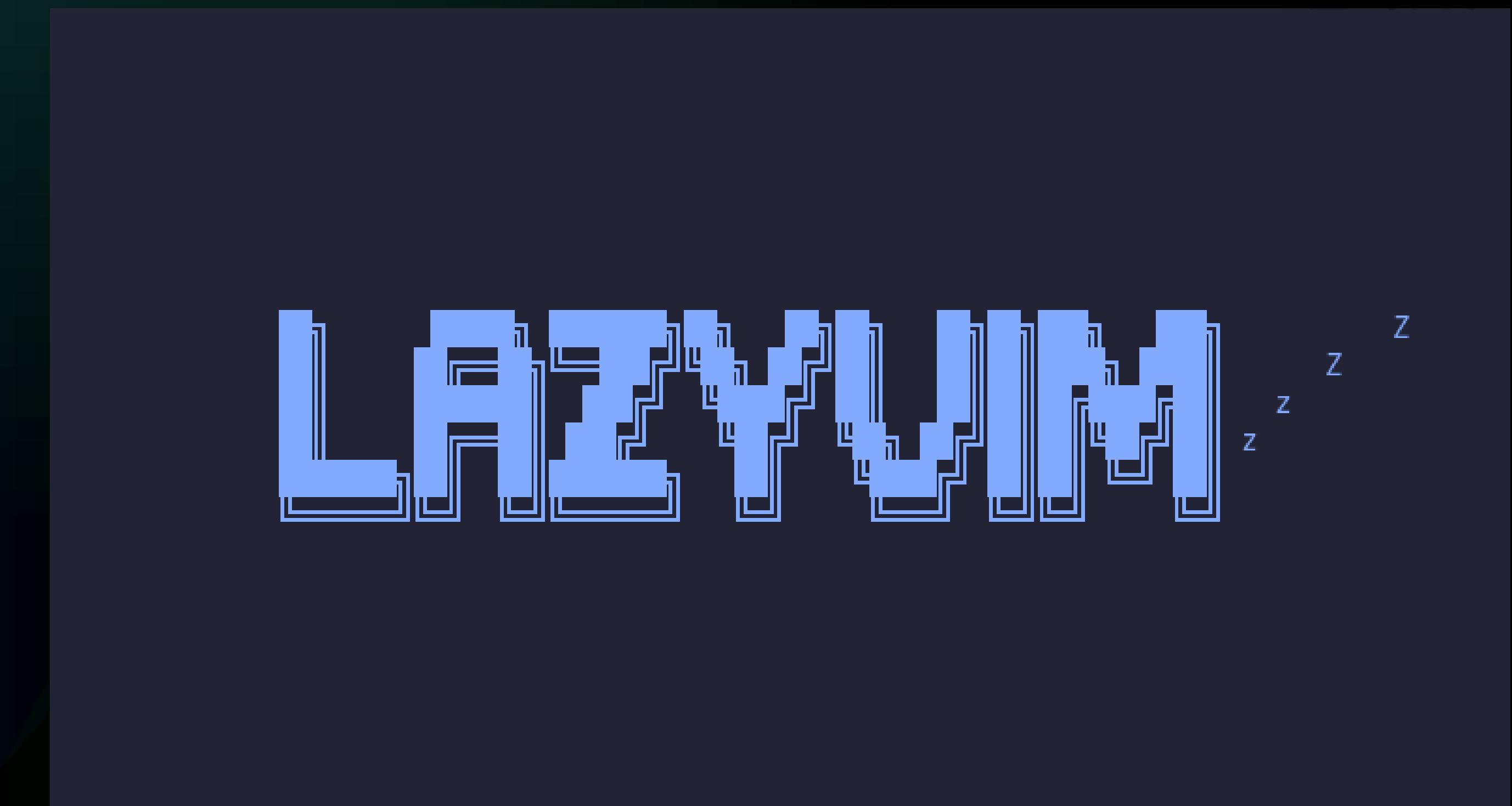
Telescope



Я: "НАДО ЗАПИСАТЬ, ПОКА НЕ ЗАБЫЛ"

МОЗГ: "НАДО ЗАБЫТЬ, ПОКА НЕ ЗАПИСАЛ"

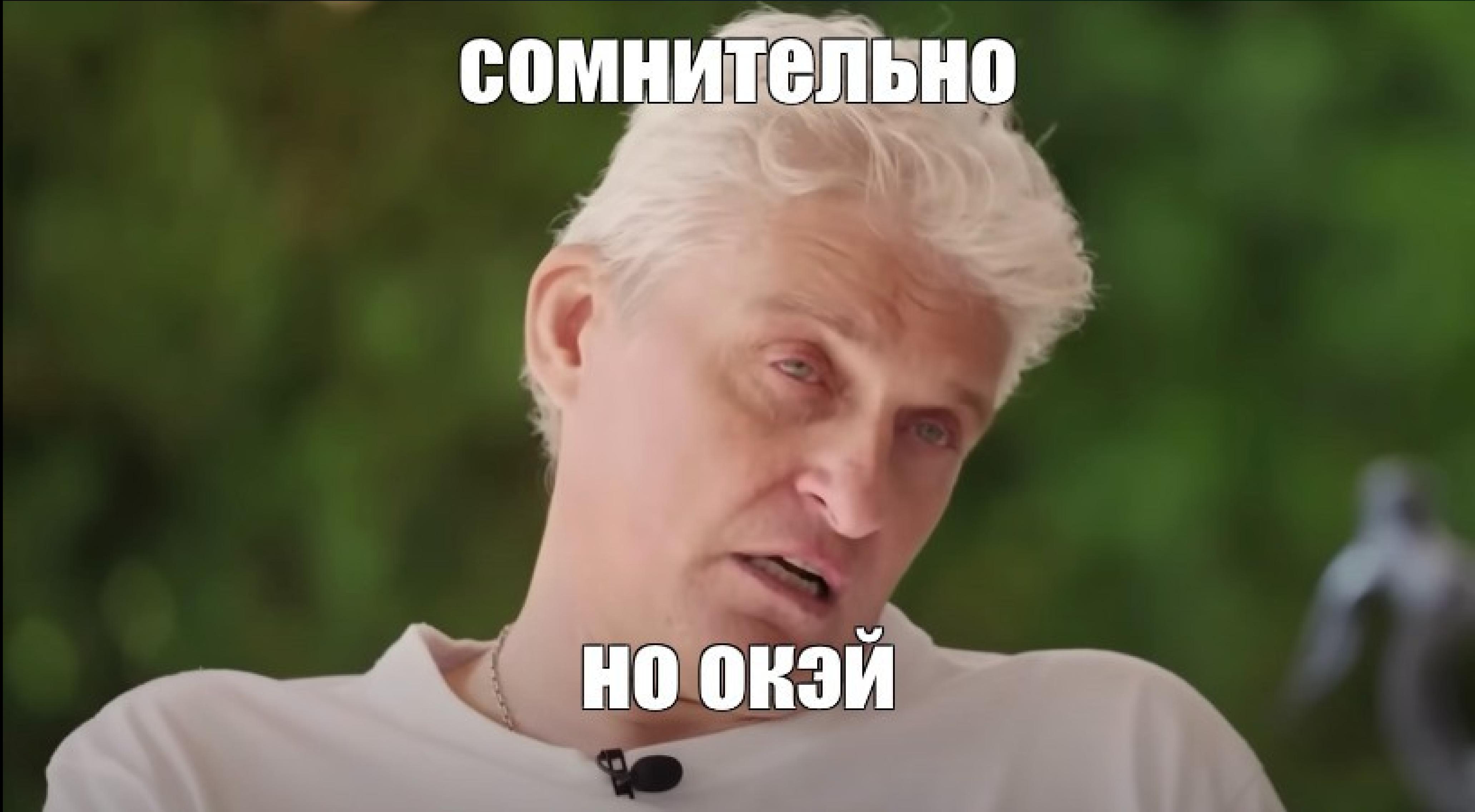
Готовые сборки neovim



Переход со стандартного редактора кода на neovim

- Подробно изучить, что из себя представляет neovim;
- Найти материал для изучения;
- Самостоятельно практиковаться;
- Настроить под себя одно из готовых решений.

Будущее за neovim и vscode?



Будущее за neovim и vscode?

- Бесплатная лицензия
- Меньшее требование к железу;
- Neovim будет все больше теснить vscode в среднесрочной перспективе.

The screenshot shows the nvim editor interface. On the left is a file tree for a Next.js project named 'admin-my-site'. The 'src' directory contains 'commons', 'components', 'entities', 'font', 'locales', and 'pages'. The 'pages' directory has subfolders 'Admin' and 'Page'. The 'Page' folder contains '404Page' and 'Article'. The 'Article' folder contains 'Article.tsx', 'api.ts', 'index.ts', and 'style.ts'. The 'Article.tsx' file is currently open in the main editor area. The code in 'Article.tsx' uses hooks like `useLanguage`, `useIsMediaQuery`, and `useState` to handle article data. It includes a `useEffect` hook that checks if `response.status` is not defined. If it's not defined, it uses the `toast` function to show an error message. The `toast` function is highlighted with a red underline, indicating a TypeScript error: 'Cannot find name 'toast''. The status bar at the bottom shows the file path 'src/.../Article/Article.tsx', the mode 'NORMAL', and other status indicators.

```
const { t, language } = useLanguage();
const { is360, is481 } = useIsMediaQuery();
// если в ответе есть responseBody (статья), то сразу ее отрисовываем
const [article, setArticle] = useState<ArticleI | null>(
  response?.responseBody ? response?.responseBody : null
);

const { imageUrl } = useImageName({
  imageName: article?.thumbnail,
});
const styles = useStyle();
const stylesPage = useStylesClasses({ theme: { is360, is481 } });

// сайд эффект остался для сценария, когда статья не загрузилась и в этом случае необходимо
// ошибка выводится в виде сообщения попала в правом верхнем углу
useEffect(() => {
  if (!response.status) {
    toast(t(response.message as string), {
      type: "error",
      hideProgressBar: true,
      theme: "colored",
    });
  }

  if (response.redirectTo) {
    push(response.redirectTo);
  }
  return;
}

if (response.responseBody) {
```

Пример работы настроенного редактора кода
на neovim

Плюсы Neovim

- Простота настройки и легкость переноса на другие рабочие машины;
- Быстрее и стабильней работа;
- Открытие больших файлов данных.

Минусы Neovim

- Тяжелый переход;
- Агрессивная политика обновления плагинов;
- Тяжело применить в большой корпоративной компании из-за отсутствия файла установщика

Спасибо!



Панфилов Александр

