

쓸모있는 CLI 팁 40가지

3시간 투자해서 개발시간 3초 단축하기

목차

- git tips (~10)
- vim tips (~10)
- POSIX shell tips (bash) (~3)
- fzf tips (~7)
- ETC (~3)

자기소개

- 성균관대학교 기계공학부 1학년
- 취미 위주 개발
- rest.nvim 메인테이너



깃허브 스타 부탁드립니다 :D

 [rest-nvim / rest.nvim](#) Public

A fast Neovim http client written in Lua

 GPL-3.0 license

 1.5k stars

 136 forks

 Branches

 Tags



Activity

 Star

 Notifications

 [boltlessengineer / NativeVim](#)

Public

Neovim config WITHOUT plugins

 MIT license

 189 stars

 3 forks

 Branches

 Tags



Activity

 Star

 Notifications



발표 목적 및 대상

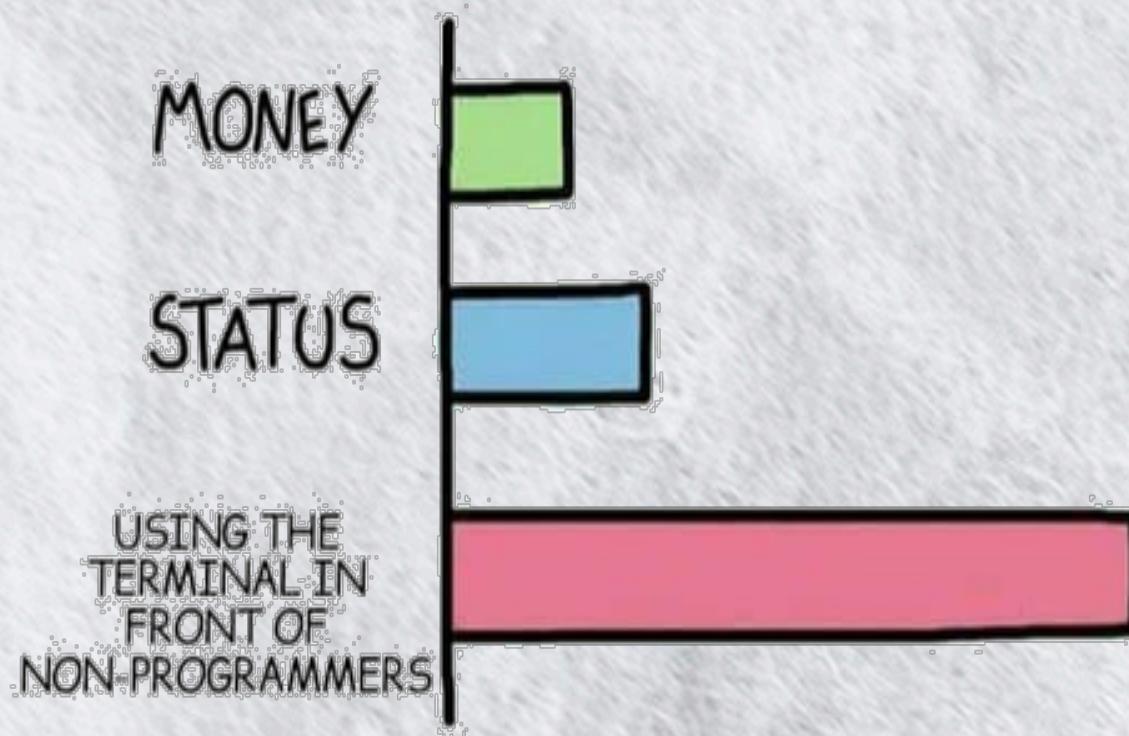
- “프로그램 동작 시간을 단축시켰다는건가?” X
- “와 이건 좀 미리 알았다면”
- “별건 아닌데 🐶 훤다”



Perform
urgent work

Improve
startup time
for neovim
from 50 to 49ms

WHAT GIVES PEOPLE FEELINGS OF POWER

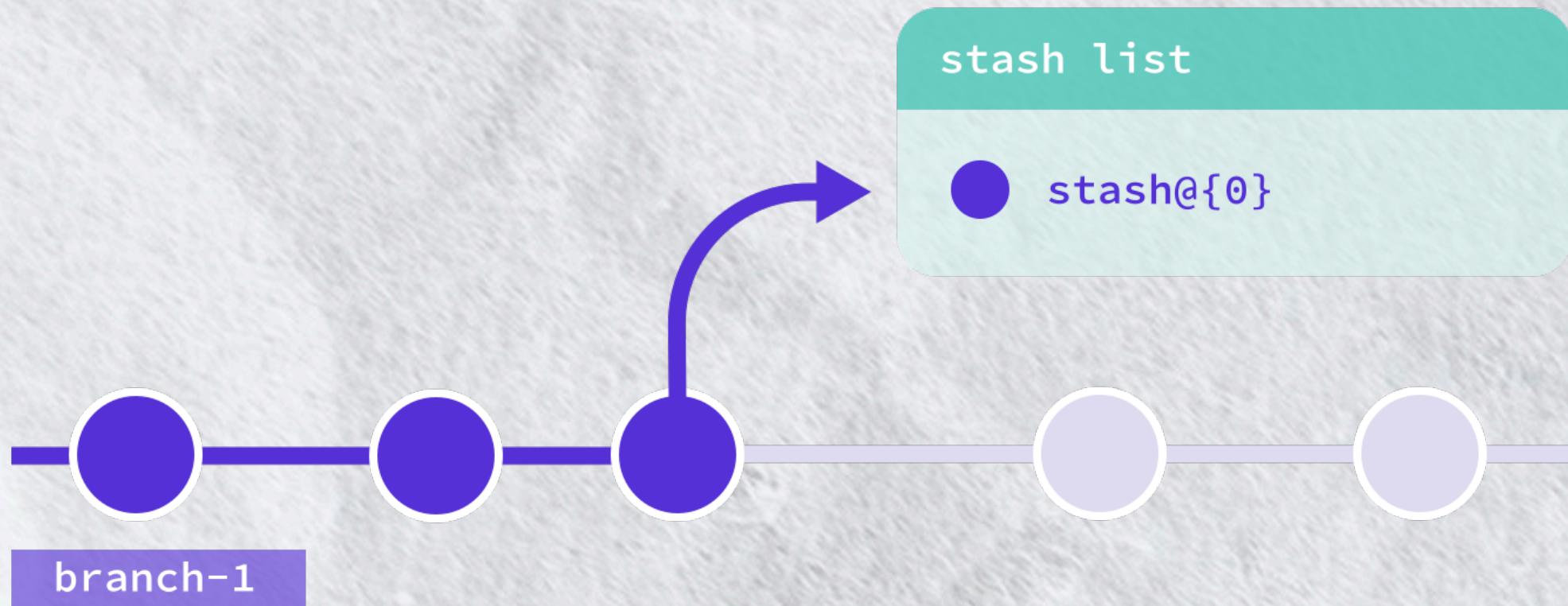


git

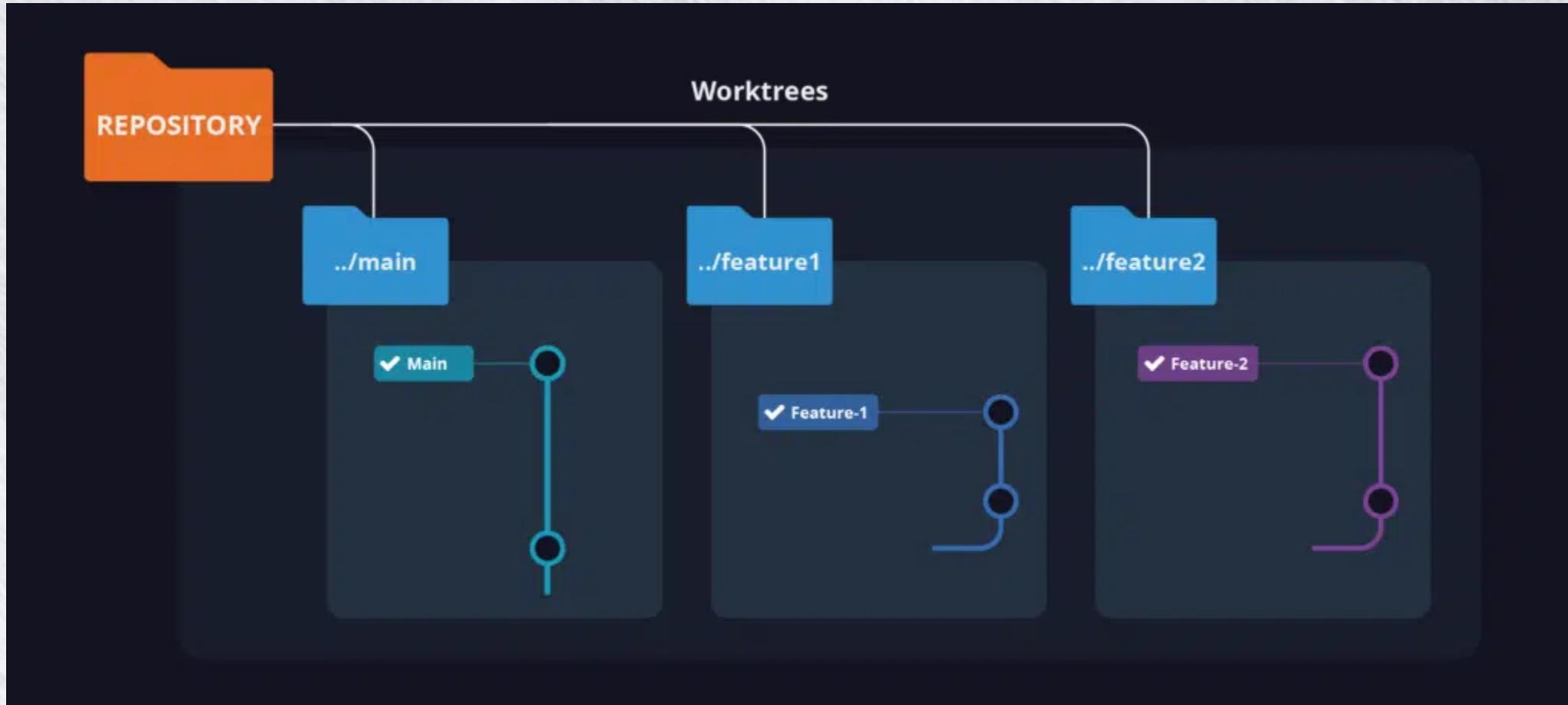


git

git stash



git worktree



git blame

\$ git blame

```
d22b6838 (NTBBloodbath 2024-01-26 11:35:12 -0400 54) local commands = {}
d22b6838 (NTBBloodbath 2024-01-26 11:35:12 -0400 55)
47beb74b (Seongmin Lee 2024-08-12 08:09:42 +0000 56) -- Lazy load
47beb74b (Seongmin Lee 2024-08-12 08:09:42 +0000 57) -- stylua: ignore start
47beb74b (Seongmin Lee 2024-08-12 08:09:42 +0000 58) local function dotenv() return require("rest-nvim.dotenv") end
47beb74b (Seongmin Lee 2024-08-12 08:09:42 +0000 59) local function request() return require("rest-nvim.request") end
47beb74b (Seongmin Lee 2024-08-12 08:09:42 +0000 60) local function logger() return require("rest-nvim.logger") end
47beb74b (Seongmin Lee 2024-08-12 08:09:42 +0000 61) local function parser() return require("rest-nvim.parser") end
47beb74b (Seongmin Lee 2024-08-12 08:09:42 +0000 62) local function ui() return require("rest-nvim.ui.result") end
47beb74b (Seongmin Lee 2024-08-12 08:09:42 +0000 63) local function config() return require("rest-nvim.config") end
47beb74b (Seongmin Lee 2024-08-12 08:09:42 +0000 64) -- stylua: ignore end
d22b6838 (NTBBloodbath 2024-01-26 11:35:12 -0400 65)
0fbef127 (Seongmin Lee 2024-08-12 12:57:18 +0000 66) —Open window based on command mods and return new window identifier
0fbef127 (Seongmin Lee 2024-08-12 12:57:18 +0000 67) —@param opts table
0fbef127 (Seongmin Lee 2024-08-12 12:57:18 +0000 68) —@return integer winnr
75c48564 (Seongmin Lee 2024-08-25 15:11:20 +0000 69) —@return boolean did_split
0fbef127 (Seongmin Lee 2024-08-12 12:57:18 +0000 70) local function split_open_cmd(opts)
8369aa1a (BoltlessEngineer 2024-08-20 17:43:32 +0900 71) local is_split = opts.smods.vertical or opts.smods.horizontal
8369aa1a (BoltlessEngineer 2024-08-20 17:43:32 +0900 72) local is_tab = opts.smods.tab ~= -1
8369aa1a (BoltlessEngineer 2024-08-20 17:43:32 +0900 73) if is_split or is_tab then
8369aa1a (BoltlessEngineer 2024-08-20 17:43:32 +0900 74)     vim.cmd(opts.mods .. " split")
8369aa1a (BoltlessEngineer 2024-08-20 17:43:32 +0900 75) end
75c48564 (Seongmin Lee 2024-08-25 15:11:20 +0000 76) return vim.api.nvim_get_current_win(), (is_split or is_tab)
75c48564 (Seongmin Lee 2024-08-25 15:11:20 +0000 77) end
75c48564 (Seongmin Lee 2024-08-25 15:11:20 +0000 78)
75c48564 (Seongmin Lee 2024-08-25 15:11:20 +0000 79) —@param opts table
75c48564 (Seongmin Lee 2024-08-25 15:11:20 +0000 80) local function open_result_ui(opts)
75c48564 (Seongmin Lee 2024-08-25 15:11:20 +0000 81) if ui().is_open() then
```

git blame

```
$ git blame -L<range> <file>
```

```
# git blame -L292,301 path/to/file
```

```
ubuntu@boltless-dev:~/repo/rest.nvim$ git blame -L292,300 lua/rest-nvim/utils.lua
1741ed8f (Seongmin Lee    2024-08-16 16:50:50 +0000 292) function utils.gq_lines(lines, filetype)
8c609d3b (Seongmin Lee    2024-08-25 09:23:50 +0000 293)     logger.debug("formatting with `gq`")
8369aa1a (BoltlessEngineer 2024-08-20 17:43:32 +0900 294)     local format_buf = vim.api.nvim_create_buf(false, true)
252d56a7 (Jiarong        2024-09-05 13:41:25 -0700 295)     local ok, errmsg = pcall(vim.api.nvim_set_option_value, "filetyp
b08b2c23 (Seongmin Lee    2024-09-04 14:59:48 +0000 296)     if not ok then
b08b2c23 (Seongmin Lee    2024-09-04 14:59:48 +0000 297)         local msg = ("Can't set filetype to '%s' (%s). Formatting is
b08b2c23 (Seongmin Lee    2024-09-04 14:59:48 +0000 298)         logger.warn(msg)
b08b2c23 (Seongmin Lee    2024-09-04 14:59:48 +0000 299)         vim.notify(msg, vim.log.levels.WARN, { title = "rest.nvim" }
b08b2c23 (Seongmin Lee    2024-09-04 14:59:48 +0000 300)         return lines, false
```

git log

```
$ git log -L<range>:<file>
```

```
commit 252d56a7334838e2406ebd58f780e3d9becfbf73
Author: Jiarong <alienover.l.d@gmail.com>
Date:   Thu Sep 5 13:41:25 2024 -0700

    fix: scope and buf number in setting options

diff --git a/lua/rest-nvim/utils.lua b/lua/rest-nvim/utils.lua
--- a/lua/rest-nvim/utils.lua
+++ b/lua/rest-nvim/utils.lua
@@ -292,9 +292,9 @@
     function utils.gq_lines(lines, filetype)
         logger.debug("formatting with `gq`")
         local format_buf = vim.api.nvim_create_buf(false, true)
-        local ok, errmsg = pcall(vim.api.nvim_set_option_value, "filetype", filetype, { scope = "local", buf = format_buf })
+        local ok, errmsg = pcall(vim.api.nvim_set_option_value, "filetype", filetype, { buf = format_buf })
         if not ok then
             local msg = ("Can't set filetype to '%s' (%s). Formatting is canceled"):format(filetype, errmsg)
             logger.warn(msg)
             vim.notify(msg, vim.log.levels.WARN, { title = "rest.nvim" })
             return lines, false
         end
     end

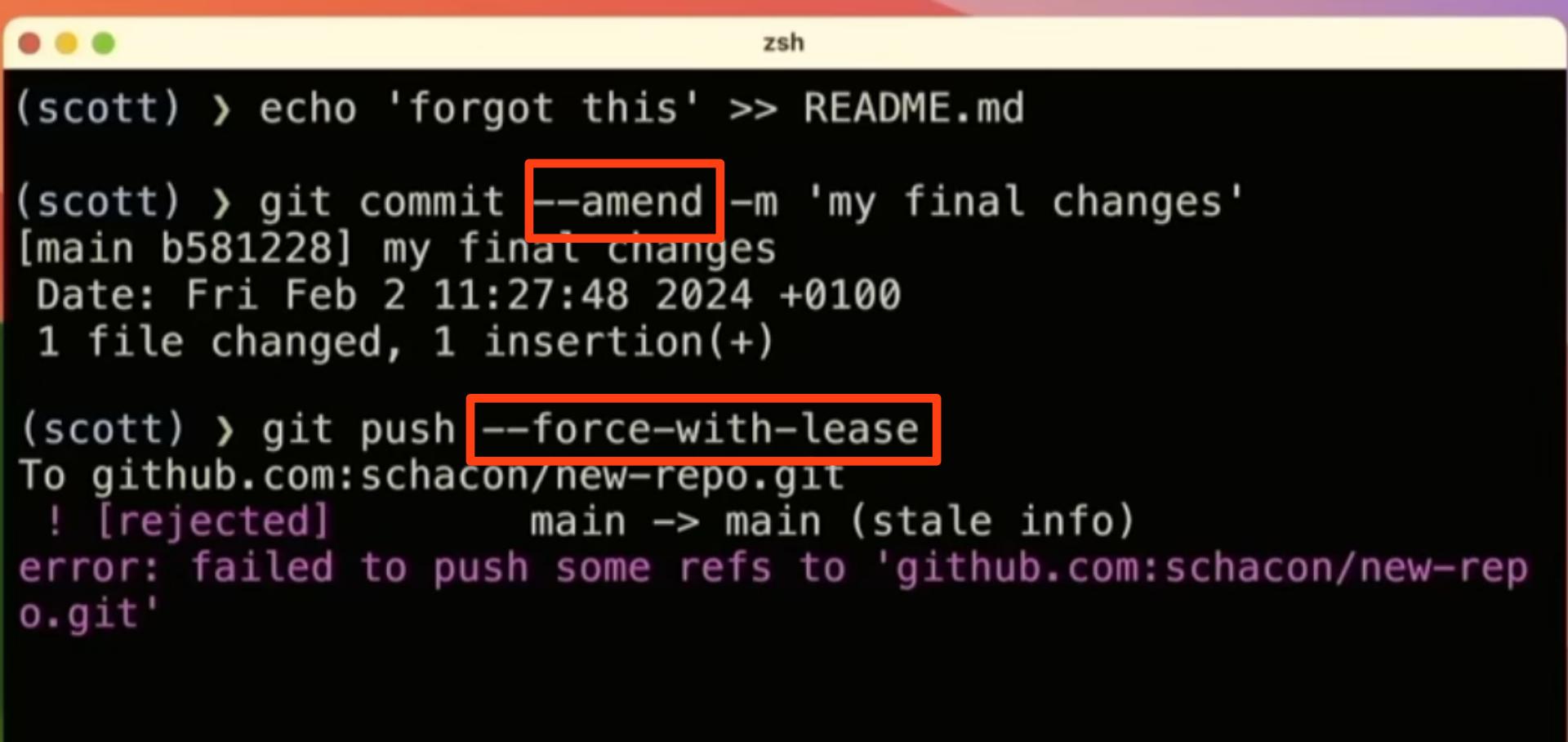
commit b08b2c23480a8c760cdf1863f9debedb0ed369ea
Author: Seongmin Lee <boltlessengineer@gmail.com>
Date:   Wed Sep 4 14:59:48 2024 +0000

    fix(gq): catch error while setting filetype

diff --git a/lua/rest-nvim/utils.lua b/lua/rest-nvim/utils.lua
:
```

git push --force

git push --force-with-lease



A screenshot of a terminal window titled "zsh". The window has a red border and a white header bar with three colored dots (red, yellow, green) on the left. The main area of the terminal shows the following command history:

```
(scott) > echo 'forgot this' >> README.md
(scott) > git commit --amend -m 'my final changes'
[main b581228] my final changes
Date: Fri Feb 2 11:27:48 2024 +0100
1 file changed, 1 insertion(+)

(scott) > git push --force-with-lease
To github.com:schacon/new-repo.git
 ! [rejected]      main -> main (stale info)
error: failed to push some refs to 'github.com:schacon/new-repo.git'
```

The command `--amend` and the option `--force-with-lease` are highlighted with a red rectangular box.

git - signing commits

```
$ git config gpg.format ssh  
$ git config user.signingkey ~/.ssh/key.pub  
$ git commit -S
```

The screenshot shows a GitHub commit history for a repository. The commits are listed on the left, and on the right, there are green 'Verified' status indicators next to each commit hash. A red box highlights the first four 'Verified' status indicators.

Commit Message	Author	Date	Status	Hash
fix(ui): don't try to render image	boltlessengineer	committed last month	Verified	061f65d
fix: pre-request script sets variable locally	boltlessengineer	committed last month	Verified	c2e7ab4
test: organize test cases	boltlessengineer	committed last month	Verified	315c624
chore: bump tree-sitter-http version	boltlessengineer	committed last month	Verified	bb087be

git - gitattributes

.gitattributes

*.png diff=exif

git config diff.exif.textconv exiftool

build/** -diff

generated/** linguist-vendered

git - rerere (REuse REcorded RESolution)

```
$ git config --global rerere.enabled true
```

git - partial cloning

```
$ git clone --filter=blob:none
```

Vim





INS src/tree_sitter.rs [+] on ↵ main [] E 1, W 0 50% 00008 launch.lua E 0, W 0 All

```

13      |         |         |         |         |         |         |         |         |
12      |         |         |         |         |         |         |         |         |
11      |         |         |         |         |         |         |         |         |
10      |         |         |         |         |         |         |         |         |
9       }         }         |         |         |         |         |         |         |
8       Self::Scope(_) => unimplemented!("scope is not implemented");
7       }         |         |         |         |         |         |         |
6       }         |         |         |         |         |         |         |
5     }         |         |         |         |         |         |         |
4     impl Link { |         |         |         |         |         |         |
3   fn parse_from_node<'a>(node: Node<'_>, source: &'a [u8]) -> Option<Link> |         |         |
2   let destination = node.child_by_field_name("destination")?; |         |         |
1   destination.clone(); |         |         |         |         |         |         |
182  return Some(id()); |         |         |         |         |         |         |         |
1     range: range(); |         |         |         |         |         |         |         |
2     dest_range: range(); |         |         |         |         |         |         |         |
3     is_error(): bool; |         |         |         |         |         |         |         |
4     byte_range(): Range; |         |         |         |         |         |         |         |
5     children(): Vec<Node>; |         |         |         |         |         |         |         |
6     no_utf8_text(): Option<String>; |         |         |         |         |         |         |         |
7     utf16_text(): Option<String>; |         |         |         |         |         |         |         |
8     into() (as Into) |         |         |         |         |         |         |         |
9     clone() (as Clone); |         |         |         |         |         |         |         |
10    named_children(): Vec<Node>; |         |         |         |         |         |         |         |
11    current_workspace: Option<LinkRoot>; |         |         |         |         |         |         |         |
12    workspace: Option<LinkRoot>; |         |         |         |         |         |         |         |
13    |         |         |         |         |         |         |         |         |
14    |         |         |         |         |         |         |         |         |
15    |         |         |         |         |         |         |         |         |
16    |         |         |         |         |         |         |         |         |
17    path: Option<String>; |         |         |         |         |         |         |         |
18    destination: Option<Node>; |         |         |         |         |         |         |         |
19    |         |         |         |         |         |         |         |         |
20    |         |         |         |         |         |         |         |         |
21    |         |         |         |         |         |         |         |         |
22    |         |         |         |         |         |         |         |         |
23    |         |         |         |         |         |         |         |         |
24    |         |         |         |         |         |         |         |         |
25    |         |         |         |         |         |         |         |         |
26    |         |         |         |         |         |         |         |         |
27    |         |         |         |         |         |         |         |         |
28    |         |         |         |         |         |         |         |         |
29    ); |         |         |         |         |         |         |         |         |
30    }         |         |         |         |         |         |         |         |
31 // copied from helix-editor/helix-
32 // Adapter to convert rope chunks to bytes-
33 pub struct ChunksBytes<'a> {
34     chunks: ropey::iter::Chunks<'a>,
35 }

```

/home/ubuntu/projects/neorg-ls

rust_analyzer Spaces:4

```

7 -- vim.lsp.set_log_level(vim.lsp.log_levels.DEBUG)
6 -- local client_id = vim.lsp.start({
5 -- | name = "neorg-ls",
4 -- | cmd = { "./target/debug/neorg-language-server" },
3 -- | root_dir = vim.fs.dirname(vim.fs.find({ "index.norg" }, { upw
2 -- })),
1 -- vim.notify("server: " .. client_id)

8
1 require("lsp-debug-tools").start({
2   expected = { "norg" },
3   name = "neorg-ls",
4   cmd = { "./target/debug/neorg-language-server" },
5   root_dir = vim.fs.dirname(vim.fs.find({ "index.norg" }, { upw
6 })

fn(&self) -> Self

Returns a copy of the value.

# Examples

let hello = "Hello"; // &str implements Clone

assert_eq!("Hello", hello.clone());

```

00007 todo.norg All

```

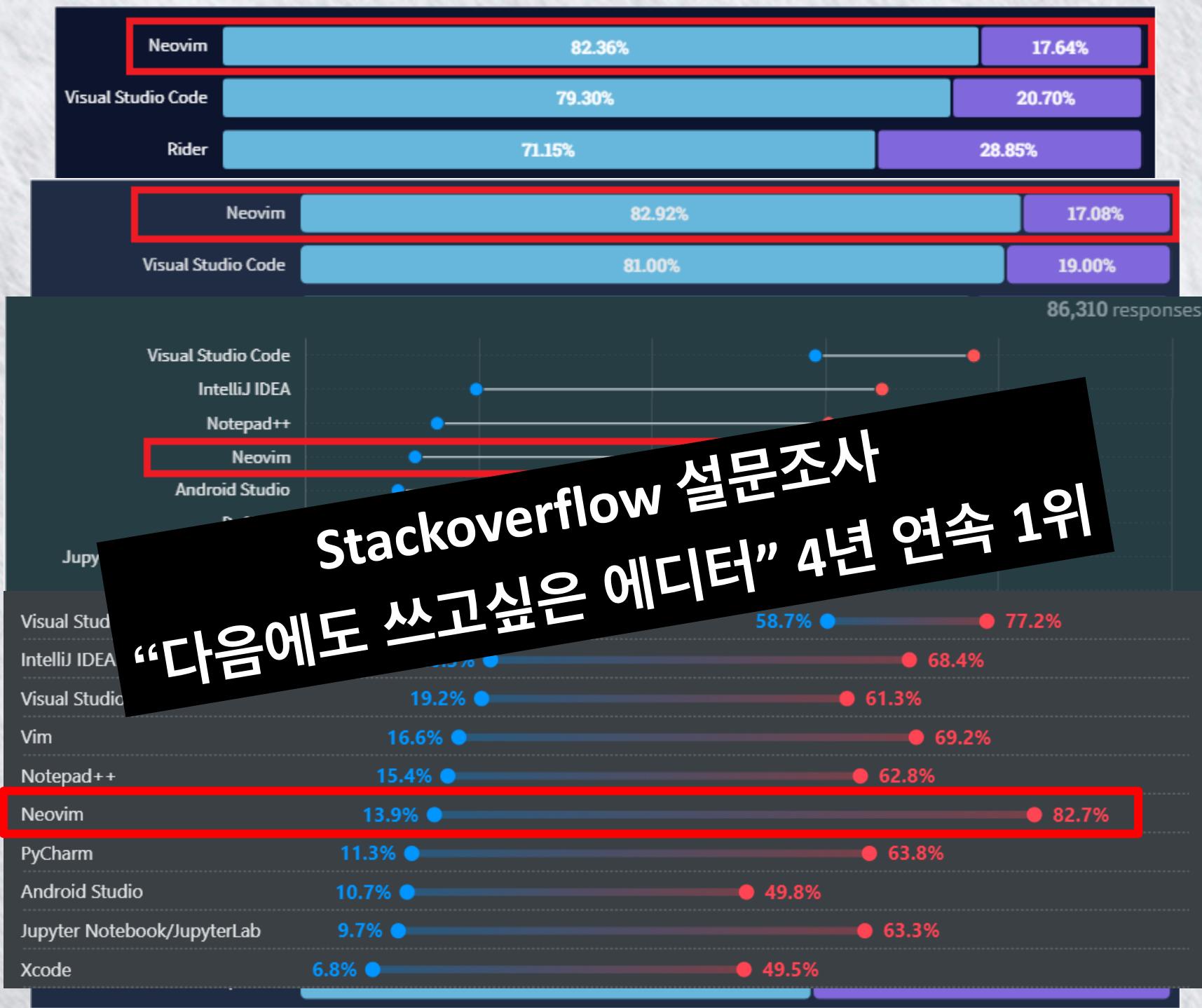
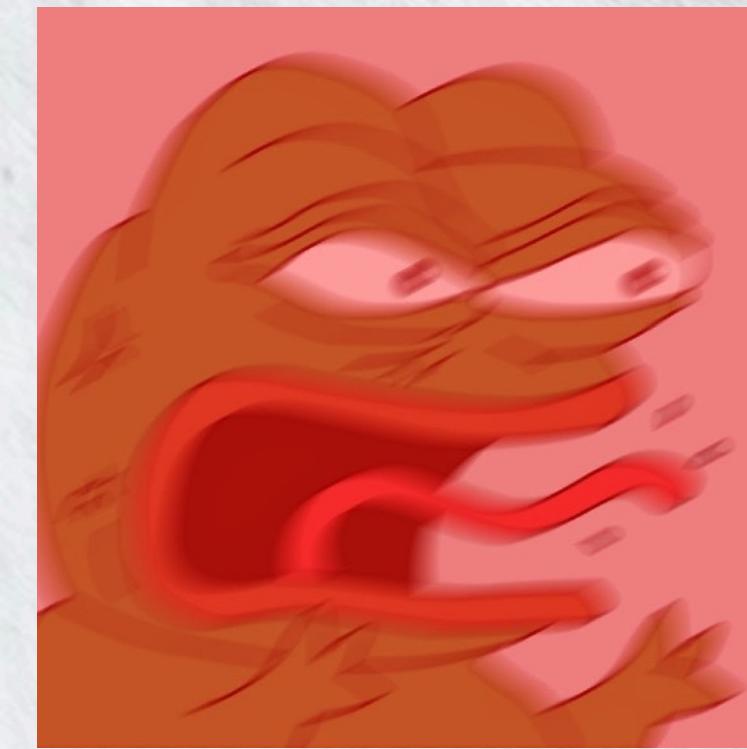
5 see: https://github.com/nvim-neorg/neorg/issues/513
4
3 - ( ) write script to check `println!` doesn't exist in document
|   (except inside tests)
2 Language Server should never include stdout functions
1 maybe using tree-sitter?

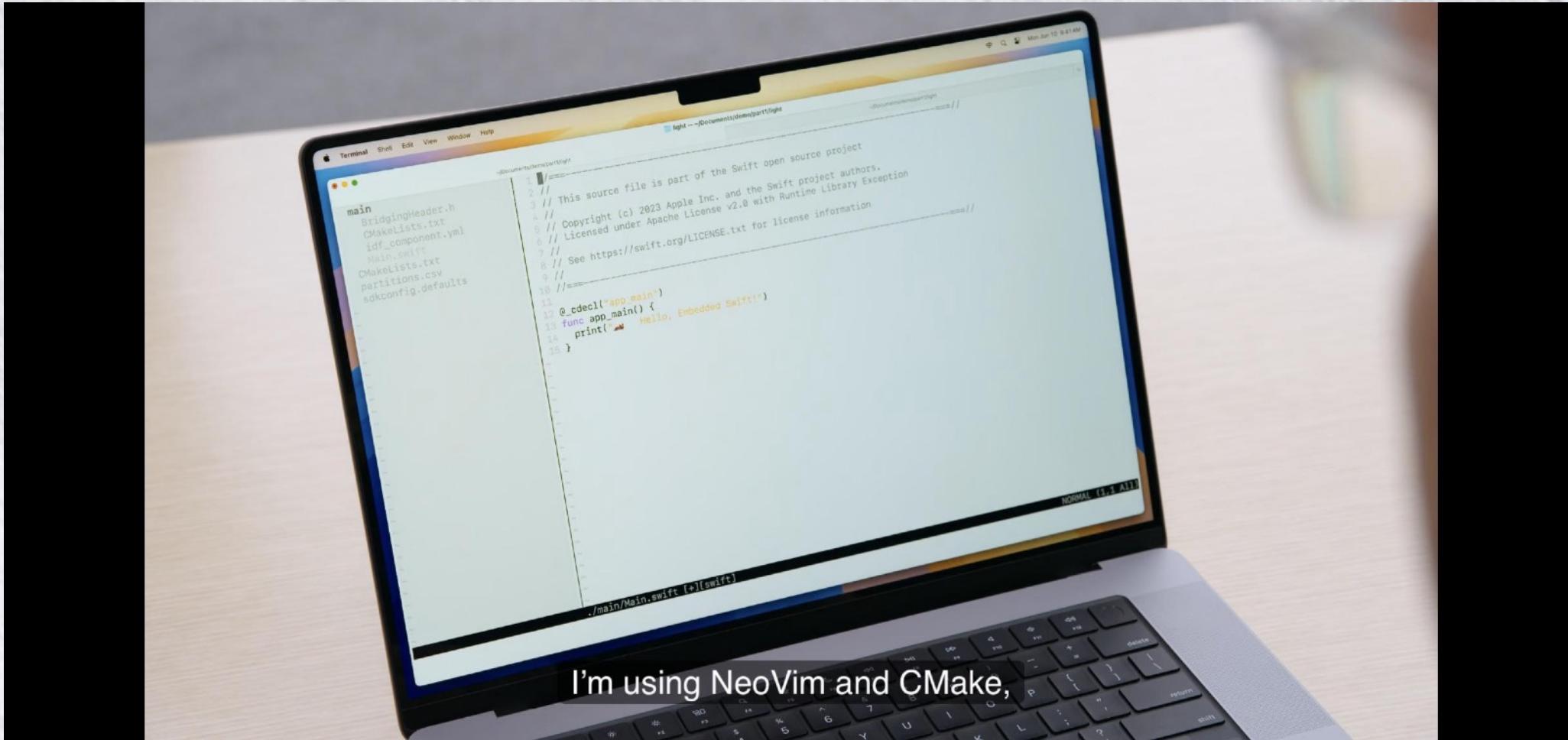
6
1 - ( ) include neorg-dirman as rust workspace child
2
3 * issue#1
4 ~ iterate over document symbols like `tree_to_symbols()`
5 ~ capture link inside each structural nodes using `capture_links()`
6 ~ save link including parent structural node's id
7 ===
8 - understand the attached modifier extensions and provide hover pr
|   eview for colored inline code.
|   e.g. `(x)` heading` (lang:norg)
9

```

너드들이나 쓰는 드보잡 에디터 아닌가요?







Apple WWDC 2024에서도 언급됨

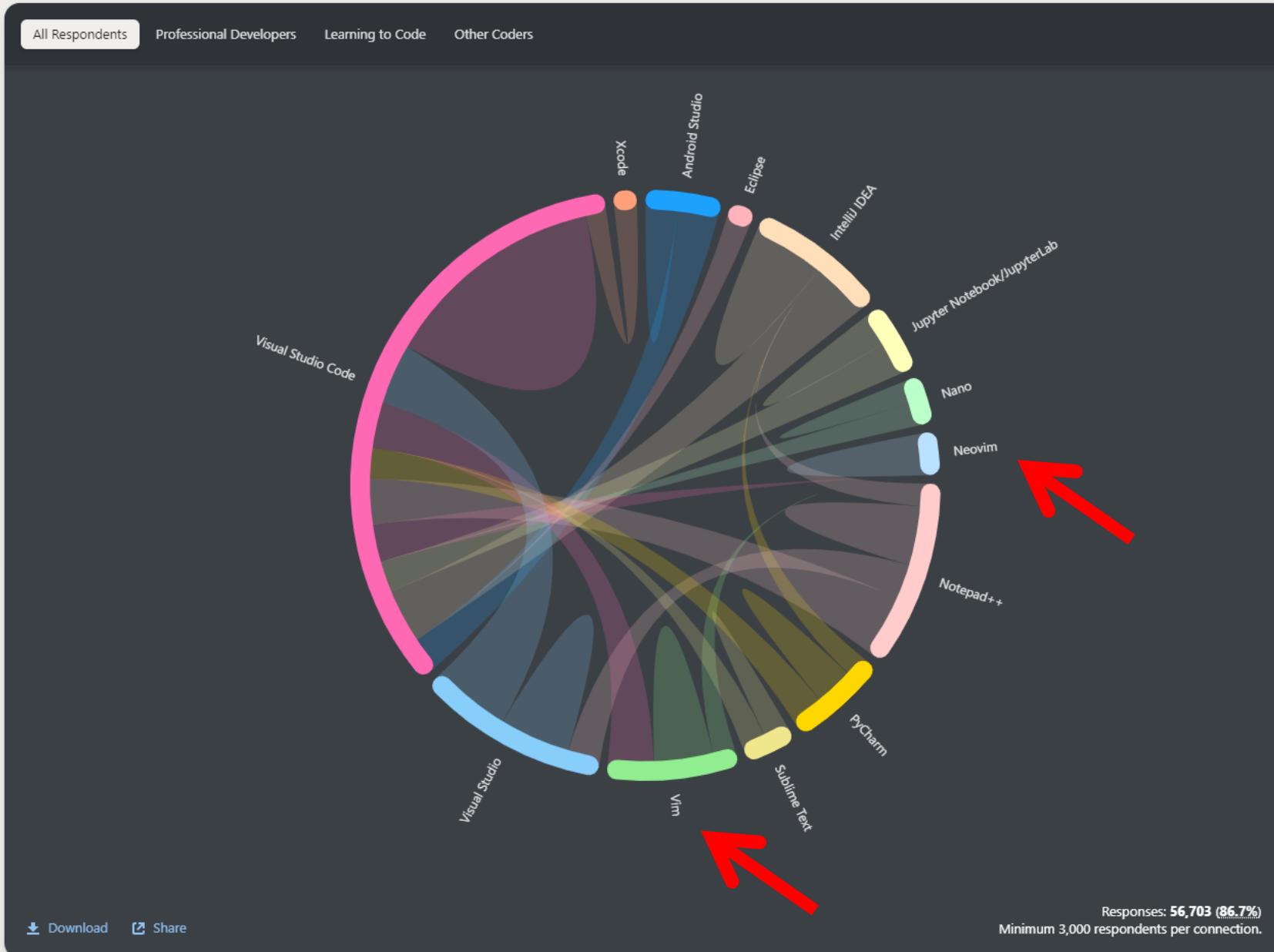
네, 너드용 들보잡 에디터 맞습니다.



Integrated development environment

74% of respondents used Visual Studio Code this past year, and of those that may want to try a different IDE next year are only interested in Vim or Neovim. Vim and Neovim users are mostly interested in staying put.

? Which **development environments** did you use regularly over the past year, and which do you want to work with over the next year? Please check all that apply.



Vim



Vim - word completion

IDE, AI 없이 그냥 자동완성이 필요할 때

:h i_CTRL-X_CTRL-P

Vi

```
14 function builder.build(req, ignore_stats)
13     local args = {}
12     --@param list table
11     --@param value any
10     local function insert(list, value)
9         if value then
8             |   table.insert(list, value)
7         end
6     end
5     insert(args, req.url)
4     insert(args, builder.extras(req))
3     insert(args, builder.method(req.method))
2     insert(args, builder.headers(req.headers))
1     insert(args, builder.cookies(req.cookies))
300    insert(args, b
1     if req.body then
2         if req.body.__TYPE == "external" then
3             |   insert(args, builder.file(req.body.data.path))
4         elseif req.body.__TYPE == "multipart_form_data" then
5             |   log.error("multipart-form-data body is not supported yet")
6         elseif vim.list_contains({ "json", "xml", "raw", "graphql" }, req.body.__TYPE) |then
7             |   insert(args, builder.raw_body(req.body.data))
8         else
9             |   log.error(("unkown body type: '%s'"):format(req.body.__TYPE))
10        end
11    end
12    if config.request.skip_ssl_verification then
```

lua/rest-nvim/client/curl/cli.lua[+] [unix]

[lua][ts:4][lua_ls] 300,19 81%

-- INSERT --

[rest_nvim] 1:bash- 2:nvim* 3:bash

-N

Vi

```
14 function builder.build(req, ignore_stats)
13 |   local args = {}
12 |   —param list table
11 |   —param value any
10 |   local function insert(list, value)
9 |     if value then
8 |       |     table.insert(list, value)
7 |     end
6 |   end
5 |   insert(args, req.url)
4 |   insert(args, builder.extras(req))
3 |   insert(args, builder.method(req.method))
2 |   insert(args, builder.headers(req.headers))
1 |   insert(args, builder.cookies(req.cookies))
300 |   insert(args, builder|
1 |     if req.body boltless
2 |       |     if req.b by
3 |         |       inse ternal" then
4 |           |         Builder file(req.body.data.path))
5 |           |           log. "multipart_form_data" then
6 |             |             body form-data body is not supported yet"
7 |             |             elseif v binary { ["json", "xml", "raw", "graphql" ], req.body.__TYPE) |then
8 |               |               inse raw_body(req.body.data))
9 |               |               based dy type: '%s'):format(req.body.__TYPE))
10 |             |             else boolean
11 |             |               log. build
12 |             |               builder
11 |           end
12 |         end
12 |         if config.request.skip_ssl_verification then
lua/rest-nvim/client/curl/cli.lua[+] [unix]
-- INSERT --
[rest_nvim] 1:bash- 2:nvim* 3:bash
```

Completed **Diagnosing**
lua_ls ✓

[lua][ts:4][lua_ls] 300,25 81%

-N

Vim - line completion

AI 자동완성을 일부 대체할 수 있는 내장 자동완성 기능

:h i_CTRL-X_CTRL-L

Vim - line completion

A screenshot of the Vim editor showing a Lua script. The cursor is at line 483, where the code is:

```
483 logger.debug("find request node child:",| child_type)|
```

The word "logger" is highlighted in blue, indicating it's being completed. A completion menu is visible below the line, listing several options starting with "logger":

- logger.debug
- logger.debug(body.data)
- logger.debug("save response body to:", path)
- logger.debug("response body saved done")
- logger.debug("find request node child:", child_type))
- name = filename:match(".*%.(.).http\$") or filename
- name = name .. "#" .. vim.b[source].._rest_nvim_count
- vim.b[source].._rest_nvim_count = vim.b[source].._rest_nvim_count + 1

The status bar at the bottom shows the file path "lua/rest-nvim/parser/init.lua[+]" and line number "483,61". The status bar also indicates "89%" completion.

:h i_CTRL-X_CTRL-L

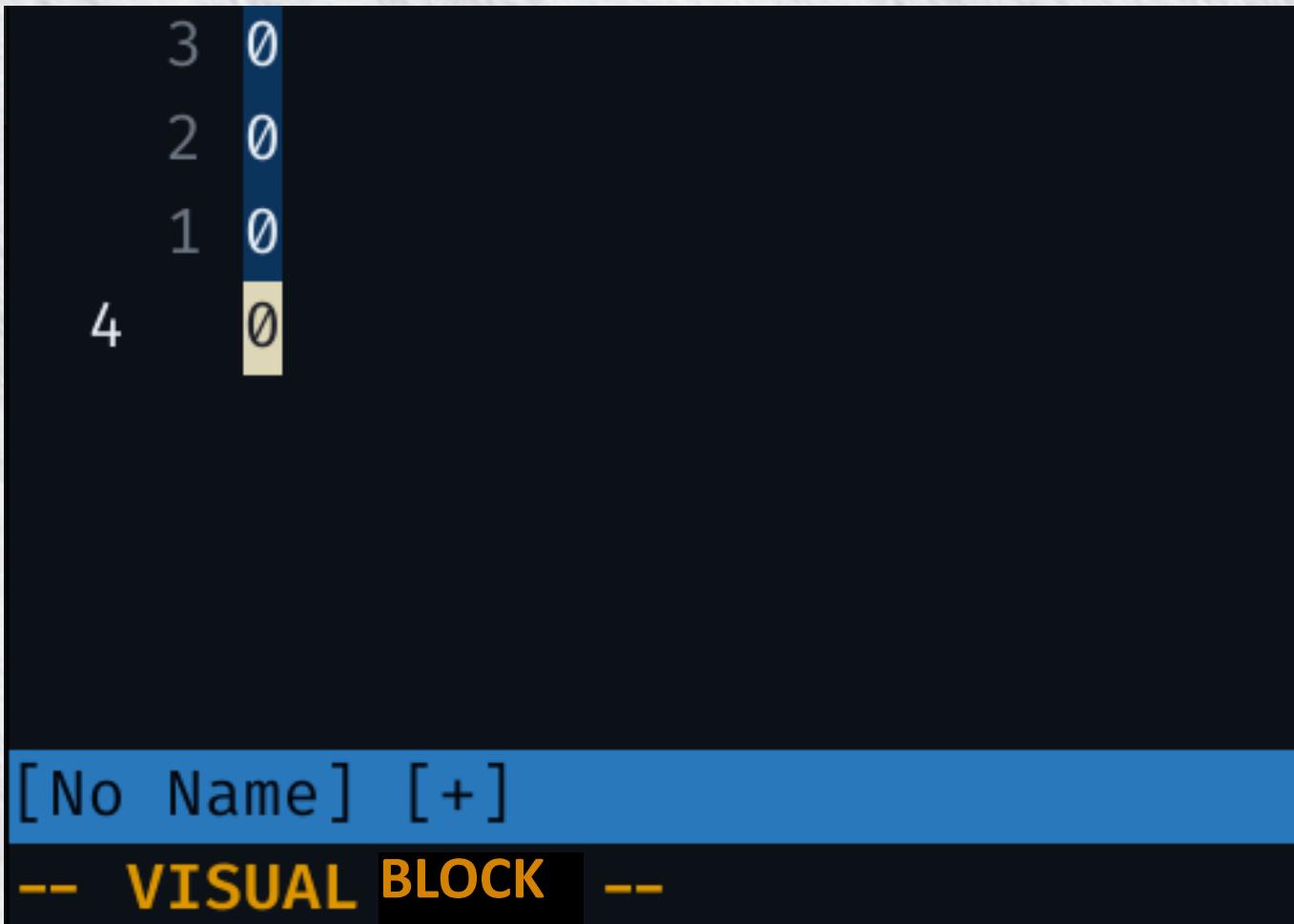
Vim - omni completion

```
12     handles - ts,
11     body = {
10         __TYPE = "json",
9          data = '{\n\t"blah": 1\n',
8        },
7      }, parser.parse(req_node, source)
6    end)
5  it("invalid json body", function()
4    local source = 'POST https://example.com\n\n{\n\t"blah": 1\n'
3    local _, tree = utils.ts_parse_source(source)
2    local req_node = assert(tree:root():child(0))
1    local spy_log_warn = spy.on(logger, "warn")
117   local blah = vim.fs.basename(file)|
```

1 parser.parse(req_no basename(file) Function
2 __@diagnostic disa dir(path, opts) Function
3 assert.spy(spy_log_ dirname(file) Function '{\n\t"blah": 1 '')
4 end) Function
5 it("parse xml", functio joinpath(...) Function
6 local source = [[PO normalize(path, opts) Function
7 parents(start) Function
8 <?xml version="1.0" encoding="u root(source, marker) Function
9 <Request>
10 <Login>login</Login>
11 <Password>password</Password>
12 </Request>
13]]

:h i_CTRL-X_CTRL-O

Vim - number increment/decrement



A screenshot of a Vim window. The visual block mode is active, indicated by the text "VISUAL BLOCK" in yellow at the bottom. The cursor is positioned over the number "4". The numbers 3, 2, 1, and 4 are displayed in white on a dark background, with the number under the cursor being slightly larger. The status bar at the bottom shows "[No Name] [+]" and "-- VISUAL BLOCK --".

```
:h CTRL-A | :h 'nrformats'
```

Vim - number increment/decrement

1 1
1 2
2 3
3 4

[No Name] [+]
4 lines changed

:h v_CTRL-A | :h 'nrformats'

Vim - number increment/decrement

A screenshot of a Vim session. The main window displays a list of four lines:

1	5
2	10
3	15
4	20

The number '5' is highlighted with a yellow background. The status bar at the bottom shows '[No Name] [+]' and '4 lines changed'.

```
:h v_g_CTRL-A | :h 'nrformats'
```

Vim - builtin formatting

```
1 setlocal formatprg=jq
2

/tmp/init.vim [unix]                                     [vim][ts:4] 2,0-1      All
1   "page": 1, "per_page": 6,
1 "total": 12, "total_pages":
2 2 }

/tmp/test-asdf.json[+] [unix]                           [json][ts:4][jsonls] 1,1      All
```

:h gq | :h 'formatprg'

Vim - builtin formatting

```
1 setlocal formatprg=jq
2

/tmp/init.vim [unix]                                     [vim][ts:4] 2,0-1      All
1
1 | "page": 1,
2 | "per_page": 6,
3 | "total": 12,
4 | "total_pages": 2
5 }

/tmp/test-asdf.json[+] [unix]                           [json][ts:4][jsonls] 1,1      All
```

:h gq | :h 'formatprg'

Vim – command-line-window

```
8     }, parser.parse(req_node, source))
7 end)
6 it("invalid json body", function()
5   local source = 'POST https://example.com\n\n{\n\n\t"blah": 1\n'
4   local _, tree = utils.ts_parse_source(source)
3   local req_node = assert(tree:root():child(0))
2   local spy_log_warn = spy.on(logger, "warn")
1   parser.parse(req_node, source)
118  --@diagnostic disable-next-line: undefined-field
1   assert.spy(spy_log_warn).called_with("invalid json: '{\n\n\t\"blah\": 1'")
2 end)
3 it("parse xml", function()
4   local source = [[POST https://example.com
5
6 <?xml version="1.0" encoding="utf-8"?>
7 <Request>
8   <Login>login</Login>
9   <Password>password</Password>
spec/parser/http_parser_spec.lua [unix]
: 5 set bg=light | colorscheme sweetie
: 4 set bg=light | colorscheme github_light_default
: 3 set laststatus=2
: 2 lua require("utils.highlights").set("StatusLineNC", { reverse = true, inherit = "StatusLine" })
: 1 set laststatus=3
: 2448 !git blame -w -C -C -C -L112,121 %
```

[No Name] [unix]
-- INSERT --

:h c_CTRL-F | :h qw

Vim - tag stack

code definition stack에서
빠르게 이전으로 돌아가게 도와주는 기능

```
9 — NOTE: Currently, all options except for `install_args` can be pass
8 —@brief ]]
7
6 local rocks = {}
5
4 —@package
3 —@deprecated
2 function rocks.packadd(rock_name, opts)
1 |   vim.deprecate("rocks.packadd", "Neovim's built-in 'packadd'", "3.0")
59     require("rocks.runtime").packadd(rock_name, opts)
1 end
2
3 return rocks
lua/rocks/init.lua
```

Vim - tag stack

code definition stack에서
빠르게 이전으로 돌아가게 도와주는 기능

```
6 —@class rocks.PackaddOpts
5 —@field bang? boolean
4
3 —@param rock_name rock_name
2 —@param opts? rocks.PackaddOpts
1 —@return boolean found
30 function runtime.packadd(rock_name, opts)
1   —@cast rock_name rock_name
2   opts = vim.tbl_deep_extend("force", {
3     |   bang = false,
4     }, opts or {})
5   local ok, err = pcall(vim.cmd.packadd, { rock_name, bang = opts.ba
6   if not ok and err and err:find("Directory not found in 'packpath'"  
lua/rocks/runtime.lua
```

Vim - tag stack

code definition stack에서
빠르게 이전으로 돌아가게 도와주는 기능

```
6 |     opts = vim.tbl_deep_extend("force", {
5 |         bang = false,
4 |     }, opts or {})
3 |     local ok, err = pcall(vim.cmd.packadd, { rock_name, bang = opts.b
2 |     if not ok and err and err:find("Directory not found in 'packpath'
1 |         vim.schedule(function()
38|             vim.notify(err, vim.log.levels.ERROR)
1 |         end)
2 |     elseif not ok then
3 |         return false
4 |     end
5 |     return true
6 end
lua/rocks/runtime.lua
```

Vim - tag stack

code definition stack에서
빠르게 이전으로 돌아가게 도와주는 기능

```
6 — a custom provider (such as the system notification provider). By
5 — writes to |:messages|.
4 — @param msg string Content of the notification to show to the user.
3 — @param level integer|nil One of the values from |vim.log.levels|.
2 — @param opts table|nil Optional parameters. Unused by default.
1 — @diagnostic disable-next-line: unused-local
618 function vim.notify(msg, level, opts) -- luacheck: no unused args
1   if level == vim.log.levels.ERROR then
2     vim.api.nvim_err写入(msg)
3   elseif level == vim.log.levels.WARN then
4     vim.api.nvim_echo({ { msg, 'WarningMsg' } }, true, {})
5   else
6     vim.api.nvim_echo({ { msg } }, true, {})
/usr/local/share/nvim/runtime/lua/vim/_editor.lua[RO]
```

Vim - tag stack

code definition stack에서
빠르게 이전으로 돌아가게 도와주는 기능

```
6 — a custom provider (such as the system notification provider). By
5 — writes to |:messages|.
4 — @param msg string Content of the notification to show to the user.
3 — @param level integer|nil One of the values from |vim.log.levels|.
2 — @param opts table|nil Optional parameters. Unused by default.
1 — @diagnostic disable-next-line: unused-local
618 function vim.notify(msg, level, opts) -- luacheck: no unused args
    1 | if level == vim.log.levels.ERROR then

:ttags
# TO tag      FROM line  in file/text
1 1 packadd          59  lua/rocks/init.lua
2 1 notify           38  ~repo/rocks.nvim/lua/rocks/runtime.lua
>
Press ENTER or type command to continue
```

Vim - tag stack

code definition stack에서
빠르게 이전으로 돌아가게 도와주는 기능

```
6 |     opts = vim.tbl_deep_extend("force", {
5 |         bang = false,
4 |     }, opts or {})
3 |     local ok, err = pcall(vim.cmd.packadd, { rock_name, bang = opts.ban
2 |     if not ok and err and err:find("Directory not found in 'packpath'") then
1 |         vim.schedule(function()
38 |             vim.notify(err, vim.log.levels.ERROR)
1 |         end)
2 |     elseif not ok then
3 |         return false
4 |     end
5 |     return true
6 end
lua/rocks/runtime.lua
```

Vim - tag stack

code definition stack에서
빠르게 이전으로 돌아가게 도와주는 기능

```
9 — NOTE: Currently, all options except for `install_args` can be passed
8 —@brief ]]
7
6 local rocks = {}
5
4 —@package
3 —@deprecated
2 function rocks.packadd(rock_name, opts)
1 |   vim.deprecate("rocks.packadd", "Neovim's built-in 'packadd'", "3.0")
59     require("rocks.runtime").packadd(rock_name, opts)
1 end
2
3 return rocks
lua/rocks/init.lua
```

Vim - tag stack

code definition stack에서
빠르게 이전으로 돌아가게 도와주는 기능

```
9 — NOTE: Currently, all options except for `install_args` can be passed
8 —@brief ]]
7
6 local rocks = {}
5
4 —@package
3 —@deprecated
2 function rocks.packadd(rock_name, opts)
1 |     vim.deprecate("rocks.packadd", "Neovim's built-in 'packadd'", "3.0")
:tags
# TO tag          FROM line  in file/text
> 1  1 packadd      59  require("rocks.runtime").packadd(rock_name, op
2  1 notify        38  ~/repo/rocks.nvim/lua/rocks/runtime.lua
Press ENTER or type command to continue
```

Vim - user-manual

The screenshot shows the Vim user manual interface. At the top, it says "usr_toc.txt Nvim" and "VIM USER MANUAL - by Bram Moolenaar". Below that is a navigation bar with "Table Of Contents" and "user-manual". A purple horizontal line separates the header from the content area. The content area starts with a section titled "Overview". Under "Overview", there is a "Getting Started" section with links to "About the manuals" and "The first steps in Vim". At the bottom of the screen, there is a status bar with the path "/usr/local/share/nvim/runtime/doc/usr_toc.txt[Help][-][RO] [unix]" and help information "[help][ts:8] 5,28-56 Top :h user-manual".

:h user-manual

Vim - :sus

```
ubuntu@boltless-dev:~/repo/rest.nvim$ nvim  
[1]+  Stopped                  nvim  
ubuntu@boltless-dev:~/repo/rest.nvim$ █
```



:h :suspend

POSIX shell



POSIX shell - cursor movement

CTRL-A : 맨 앞으로 이동

CTRL-E : 맨 뒤로 이동

CTRL-X CTRL-X : 커서를 최근 위치로 이동

POSIX shell - cut / paste

CTRL-W : 커서 앞 글자 삭제

CTRL-U : 커서 위치부터 맨 앞까지 삭제

CTRL-Y : 방금 삭제한 것 붙여넣기

POSIX shell - etc

CTRL-X CTRL-E : 현재 명령어를 \$EDITOR로 편집
CTRL-R : history에서 명령어 검색

POSIX shell (example)

```
boltless in 🌐 boltless-macmini in rest.nvim on ✨ main via 🌙 v5.4.7
❯ curl -Sf https://sh.rustup.rs | sh
```

POSIX shell (example)

```
boltless in 🌐 boltless-macmini in rest.nvim on ↗ main via 🌙 v5.4.7
❯ curl -Sf https://sh.rustup.rs | sh
```

<C-a>

POSIX shell (example)

```
boltless in 🌐 boltless-macmini in rest.nvim on ✘ main via 🌙 v5.4.7
❯ sudo curl -Sf https://sh.rustup.rs | sh
```

<C-a>sudo_

POSIX shell (example)

```
boltless in 🌐 boltless-macmini in rest.nvim on ✨ main via 🌙 v5.4.7
❯ sudo curl -Sf https://sh.rustup.rs | sh
```

<C-a>sudo_<C-e>

POSIX shell (example)

```
boltless in ℐ boltless-macmini in rest.nvim on ↵ main via 🌙 v5.4.7
› sudo curl -Sf https://sh.rustup.rs | █
```

<C-a>sudo_<C-e><C-w>

POSIX shell (example)

```
boltless in 🌐 boltless-macmini in rest.nvim on ✏ main via 🌙 v5.4.7
❯ sudo curl -Sf https://sh.rustup.rs
```

<C-a>sudo_<C-e><C-w>x2

POSIX shell (example)

```
boltless in ℐ boltless-macmini in rest.nvim on ↵ main via 🌙 v5.4.7
❯ sudo curl -Sf █
```

<C-a>sudo_<C-e><C-w>x3

POSIX shell (example)

```
boltless in ℍ boltless-macmini in rest.nvim on ↵ main via 🌙 v5.4.7
› sudo curl |ssf █
```

<C-a>sudo_<C-e><C-w>x3<BS>x5-sSf_

POSIX shell (example)

```
boltless in 🌐 boltless-macmini in rest.nvim on ✘ main via 🌙 v5.4.7
❯ sudo curl -sSf https://sh.rustup.rs | sh
```

a>sudo_<C-e><C-w>x3<BS>x5-sSf_<C-y>

fzf



command-line fuzzy finder

fzf (예시0: 기본 동작)

fzf

```
# fzf : 파일 선택 (stdin 없을 시의 기본 동작)
```

fzf (예시1: interactive vim)

```
vim $(fzf --multi)
```

fzf : 파일 선택 (stdin 없을 시의 기본 동작)

--multi : 1개 이상을 선택 가능하게 하는 옵션

fzf (예시2: interactive cd)

```
cd $(find . -type d | fzf)
```

```
# find . -type d : 현재경로(.)에서 모든 폴더(d) 출력
```

fzf (예시3: git checkout)

```
git branch --all | fzf | xargs git checkout
```

git branch --all : 모든 git 브랜치 출력

xargs git checkout : stdin을 git의 마지막 인자로 실행

fzf (예시4: kill process)

```
kill -9 $(ps -ef | fzf | awk '{print $2}')
```

ps -ef : 모든 프로세스 출력

awk '{print \$2}' : stdin의 2번째 문자만 필터링

xargs kill -9 : stdin을 kill의 마지막 인자로 실행

fzf (예시5: docker attach)

```
docker attach $(  
    docker ps --format "{{.ID}} {{.Names}}" | \/  
    fzf | \  
    awk '{print $1}'  
)
```

docker ps : 모든 docker 프로세스 출력

awk '{print \$1}' : stdin의 1번째 문자만 필터링

docker attach : process ID에 연결

gh (github cli)

GitHub CLI is here

Take GitHub to the command line

```
$ gh pr status
```

Relevant pull requests in cli/cli

Current branch

There is no pull request associated with [branch]

Created by you

You have no open pull requests

Requesting a code review from you

#1640 Add auth command and flag usage to

✓ Checks passing

```
$ gh --version
```

gh version 1.0.0

<https://github.com/cli/cli/releases/tag/v1.0>

gh (github cli)

```
$ gh issue list
```

ID	TITLE	LABELS	UPDATED
#455	Feature: use variables in bod ...	enhancement	about 2 days ago
#448	Feature: GraphQL CMP Support	enhancement	about 3 days ago
#441	Feature(hooks): automatically ...	enhancement	about 10 days ago
#440	Feature: confirm request comm ...	enhancement	about 9 days ago
#424	Feature: Change file type of ...	enhancement	about 9 days ago
#414	v3 migration issue		about 18 days ago
#413	Feature: provide raw result pane	enhancement	about 20 days ago
#398	notice: status of the project ...		about 1 month ago
#382	JSON formatting for applicati ...		about 9 days ago
#361	Swagger support	format-compat	about 13 days ago
#337	Not installed luarocks with p ...	installation	about 19 days ago
#309	Unable to execute request due ...	v2	about 5 months ago
#273	Integration with OAuth	enhancement	about 6 months ago
#270	Add Floating window for results	enhancement	about 21 days ago
#254	Pass the settings --cert, --k ...		about 3 months ago
#244	Can we have support for runni ...	enhancement	about 21 days ago
#235	Redirected response is not fo ...	bug	about 21 days ago
#233	Feature Request: Add API meth ...	enhancement	about 21 days ago
#195	export to hurl format	format-compat	about 21 days ago
#168	tests should check for the ht ...		about 1 year ago
#144	[Feature Request] Import Post ...	format-compat	about 1 year ago
#107	Amazon Auth	enhancement	about 24 days ago
#98	more test example for http re ...		about 7 months ago
#73	Trying to setup options in ht ...	enhancement	about 20 days ago
#66	Feature request: Adding snippets	enhancement, help wanted, go ...	about 2 years ago

```
$ gh pr list
```

ID	TITLE	BRANCH	CREATED AT
#452	feat(gql): download schema	gorillamoe:feat/download-gql- ...	about 2 days ago
#443	Feature: redirected requests	feat/redirected-requests	about 9 days ago
#432	fix(ui): always use `rest_nvi ...	fix/use-rest_nvim_result-ft	about 12 days ago
#415	feat: curl commands from `sh...	feat/ft-curl-support	about 20 days ago
#225	test: use neorocks	teto:setup-neorocks	about 1 year ago
#115	Interpret "Authorization: Bas ...	srihari93:basic-auth	about 2 years ago
#100	[feat] Response body stored (...	blackaddress:response_body_stored	about 2 years ago

gh 가 없을 경우의 PR review workflow

1) check fork repo URL & branch from github



2) fetch it to local

```
$ git fetch https://github.com/someone/forked-repo.git \
  feature:feature-branch
```

3) checkout to that branch

```
$ git checkout feature:feature-branch
```

gh 를 활용한 PR review workflow

\$ gh pr list

```
ubuntu@boltless-dev:~/repo/rest.nvim$ gh pr list

Showing 7 of 7 open pull requests in rest-nvim/rest.nvim

ID      TITLE                                BRANCH
#452    feat(gql): download schema          gorillamoe:feat/down
#443    Feature: redirected requests        feat/redirected-requ
#432    fix(ui): always use `rest_nvi...     fix/use-rest_nvim_re
```

\$ gh pr checkout 452

```
ubuntu@boltless-dev:~/repo/rest.nvim$ gh pr checkout 452
From https://github.com/rest-nvim/rest.nvim
 * [new ref]      refs/pull/452/head → feat/download-gql-schema
Switched to branch 'feat/download-gql-schema'
ubuntu@boltless-dev:~/repo/rest.nvim$
```



fzf (예시6: gh pr checkout)

```
gh pr list --state open | \
fzf | \
awk '{print $1}' | \
xargs gh pr checkout
```

gh pr list --state open : 모든 pr 출력

gh pr checkout : stdin을 docker의 마지막 인자로 실행

jq (command line json processor)

./jq

jq

```
$ curl -sSL https://reqres.in/api/users
```

```
ubuntu@boltless-dev:~/repo/rest.nvim$ curl -s https://reqres.in/api/users
{"page":1,"per_page":6,"total":12,"total_pages":2,"data":[{"id":1,"email":"george.bluth@reqres.in","first_name":"George","last_name":"Bluth","avatar":"https://reqres.in/img/faces/1-image.jpg"}, {"id":2,"email":"janet.weaver@reqres.in","first_name":"Janet","last_name":"Weaver","avatar":"https://reqres.in/img/faces/2-image.jpg"}, {"id":3,"email":"emma.wong@reqres.in","first_name":"Emma","last_name":"Wong","avatar":"https://reqres.in/img/faces/3-image.jpg"}, {"id":4,"email":"eve.holt@reqres.in","first_name":"Eve","last_name":"Holt","avatar":"https://reqres.in/img/faces/4-image.jpg"}, {"id":5,"email":"charles.morris@reqres.in","first_name":"Charles","last_name":"Morris","avatar":"https://reqres.in/img/faces/5-image.jpg"}, {"id":6,"email":"tracey.ramos@reqres.in","first_name":"Tracey","last_name":"Ramos","avatar":"https://reqres.in/img/faces/6-image.jpg"}],"support":{"url":"https://reqres.in/#support-heading","text":"To keep ReqRes free, contributions towards server costs are appreciated!"}}ubuntu@boltless-dev:~/repo/rest.nvim$
```

jq

```
$ curl -sSL https://reqres.in/api/users | jq
```

```
ubuntu@boltless-dev:~/repo/rest.nvim$ curl -s https://reqres.in/api/users | jq [26/378]
{
  "page": 1,
  "per_page": 6,
  "total": 12,
  "total_pages": 2,
  "data": [
    {
      "id": 1,
      "email": "george.bluth@reqres.in",
      "first_name": "George",
      "last_name": "Bluth",
      "avatar": "https://reqres.in/img/faces/1-image.jpg"
    },
    {
      "id": 2,
      "email": "janet.weaver@reqres.in",
      "first_name": "Janet",
      "last_name": "Weaver",
      "avatar": "https://reqres.in/img/faces/2-image.jpg"
    },
    {
      "id": 3,
      "email": "emma.watson@reqres.in",
      "first_name": "Emma",
      "last_name": "Watson",
      "avatar": "https://reqres.in/img/faces/3-image.jpg"
    },
    {
      "id": 4,
      "email": "michelle.williams@reqres.in",
      "first_name": "Michelle",
      "last_name": "Williams",
      "avatar": "https://reqres.in/img/faces/4-image.jpg"
    },
    {
      "id": 5,
      "email": "david.thewizard@reqres.in",
      "first_name": "David",
      "last_name": "TheWizard",
      "avatar": "https://reqres.in/img/faces/5-image.jpg"
    },
    {
      "id": 6,
      "email": "karen.kwan@reqres.in",
      "first_name": "Karen",
      "last_name": "Kwan",
      "avatar": "https://reqres.in/img/faces/6-image.jpg"
    }
  ]
}
```

jq

```
$ curl -sSL https://reqres.in/api/users | jq .data[0]
```

```
ubuntu@boltless-dev:~/repo/rest.nvim$ curl -s https://reqres.in/api/users | jq .data[0]
{
  "id": 1,
  "email": "george.bluth@reqres.in",
  "first_name": "George",
  "last_name": "Bluth",
  "avatar": "https://reqres.in/img/faces/1-image.jpg"
}
ubuntu@boltless-dev:~/repo/rest.nvim$
```

jq

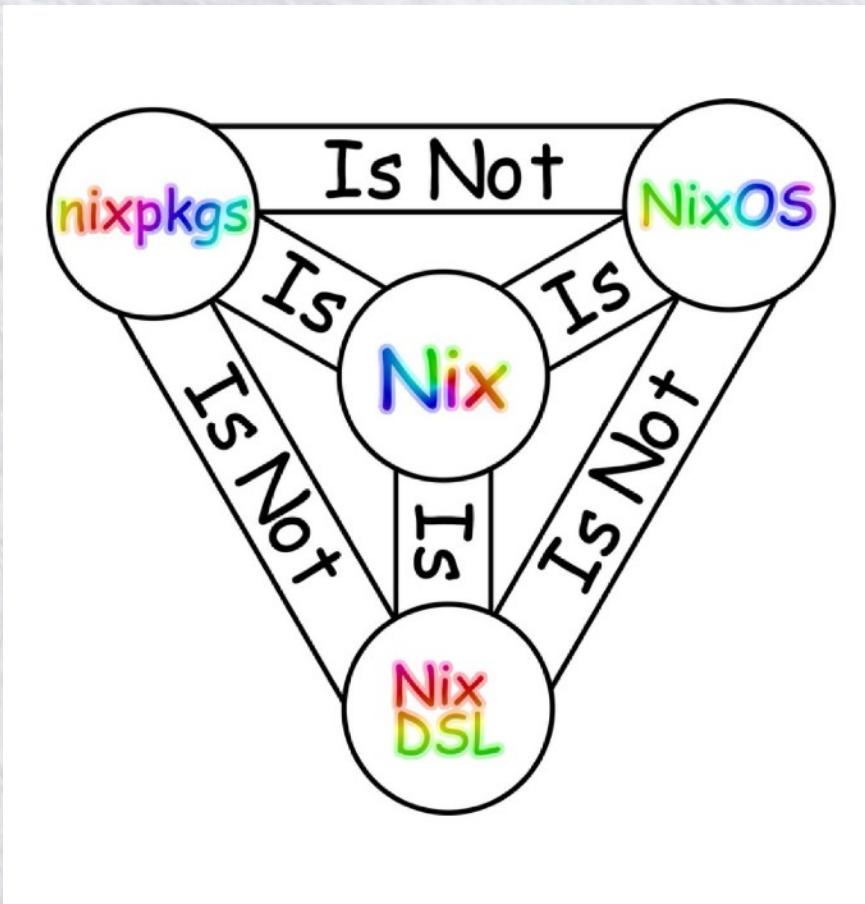
```
$ curl -sSL https://reqres.in/api/users | jq .data[].email
```

```
ubuntu@boltless-dev:~/repo/rest.nvim$ curl -s https://reqres.in/api/users | jq .data[].email
"george.bluth@reqres.in"
"janet.weaver@reqres.in"
"emma.wong@reqres.in"
eve.holt@reqres.in"
"charles.morris@reqres.in"
"tracey.ramos@reqres.in"
ubuntu@boltless-dev:~/repo/rest.nvim$
```

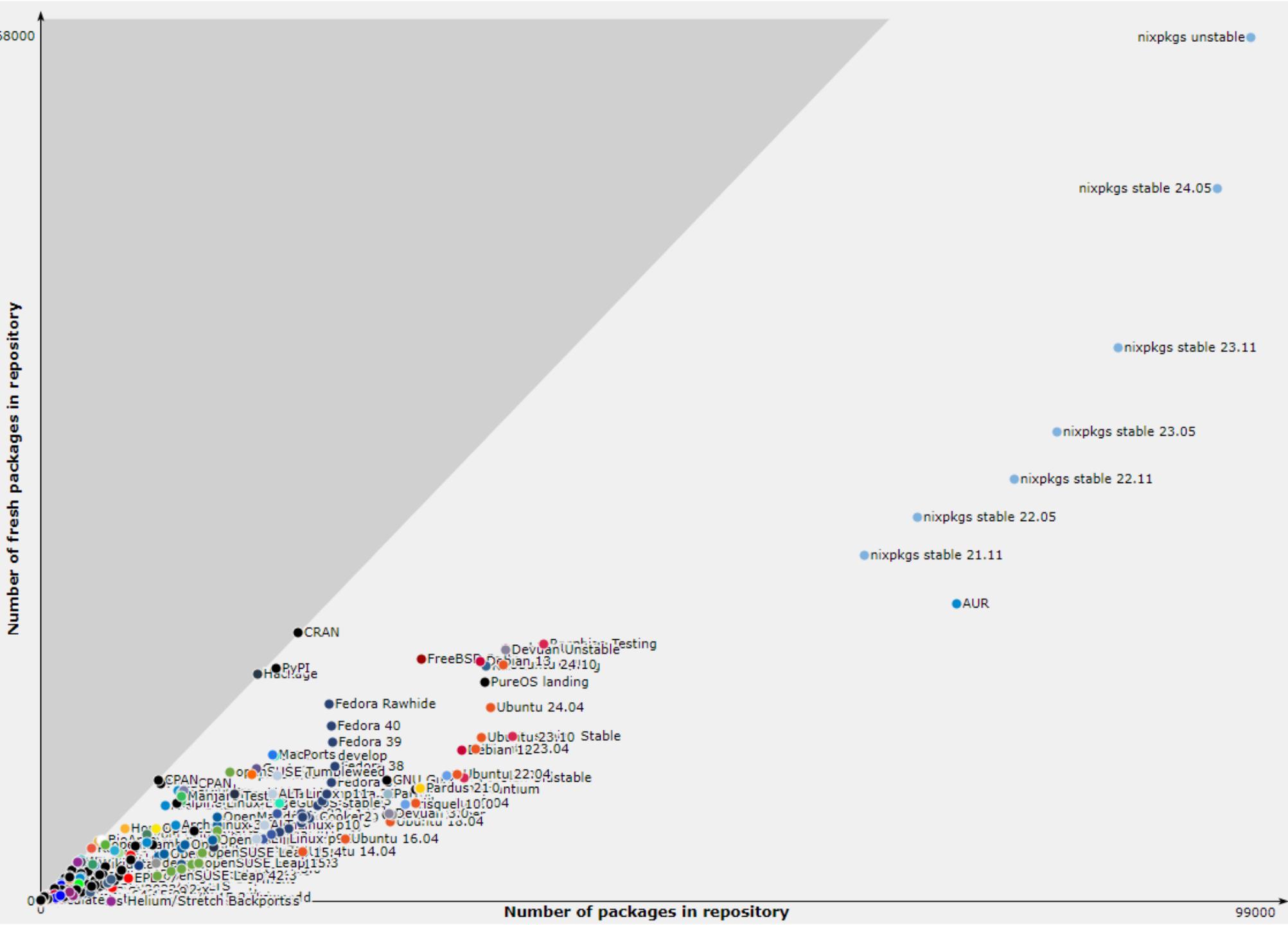
nix



nix



nix



pdf:

<https://share.boltless.dev/hanmadang.pdf>

감사합니다.