

2019-04-30 - golang-tools session - meeting notes

Previous session notes

[2019-03-26- golang-tools session - meeting notes](https://www.youtube.com/watch?v=xG-dNIK82rc)

Recording

<https://www.youtube.com/watch?v=xG-dNIK82rc>

Attendees

- Billie Cleek
- Jay Conrod
- Ian Cottrell
- Paul Jolly
- Muir Manders
- Michael Matloob
- Bryan Mills
- Ramya Rao
- Rebecca Stambler
- Marwan Sulaiman

Notes

Conference updates

- Rebecca: GopherCon Russia was cool. Video isn't posted yet, but will be soon. Gave an overview of gopls, why it was made, why it's helpful. Planning to give the talk again in Singapore and Denver.
- Michael: Will be giving a talk on go/analysis, walking through building an analyzer.

gopls update

- Rebecca: Nothing really specific, adding lots of small improvements, stabilizing. Have gotten lots of issues filed, trying to stay up to date. Currently tracking golang/go, vimgo, VSCode repos. Let me know if I should follow others.
- Rebecca: Ideally issues would be reported directly in the main repo, but it's inevitable some issues will be reported against editors.
- Rebecca: Main thing we need contributed is tests. Could use better test coverage.
- Ramya: General population will log issues in editor repos, they won't know about gopls. Editor repos will forward issues as needed.
- Ian: Planning to add "gopls bug" which should simplify bug filing. Will include information about specific issue. Probably won't include logs right away, since they can include sensitive information.

VSCoDe update

- Ramya: Latest release officially supports gopls. New users of language server will be prompted to install gopls. If you're using modules, you'll be prompted to use it. Feedback has been good. It's a little disruptive that missing imports are not added on file save without an extra setting.
- Ramya: Question about diagnostics: can we enable by default?
- Rebecca: Sometimes VSCoDe would not receive a clear diagnostic. Haven't seen it lately. Maybe let's give it another week and enable it again if no one reports the issue?
- Rebecca: On organizing imports on save, can we set by default?
- Ramya: The extension can set it by default for sure. We have to do it specifically for Go though.

vimgo update

- Billie: Released a version last week, 1.20. More work to be done with gopls, but we're stable enough. Planning to add better support for diagnostics and incremental changes for didChange. Also not quite sure how to do autoformatting. People have been having some problems with gopls, might not know how to clear module cache. `go clean -modcache` fixed it. Someone saw an issue where gopls ran for 30 minutes and never came back, no logs.
- Billie: Most helpful to have next would be seeing references, callees, guru features.

gopls: what comes next

- Ian: Features where we need scope beyond the current file. Everything we've done so far can start in the current file. To find references, you need to find referrers elsewhere. Doing this in large code bases is inefficient without extra machinery. May look at versions of those features that don't jump beyond the current package or module. This would cover most of the use cases people care about. Can consider larger scale infrastructure later.
- Billie: So guru won't be maintained for modules? Those features are still valuable
- Ian: Right, it's on life support. Making guru work in module mode is a large amount of effort, several weeks. Not worth the amount of effort compared to what we could do instead.
- Paul: Should address issues for newcomers: new packages, new files not on disk, things blow up.
- Rebecca: Main things that don't work yet are files that are only in overlays, main packages, xtest package. Everything else should work.
- Ian: If we guess it wrong, we end up with bad data in cache. Need to be very careful about it.
- Paul: How about watching changes in generated files?
- Ian: Not sure how many LSP clients support watching files. But it's becoming higher priority, and we'll do it in the near future.
- Muir: LSP has two jump-to-def: where it's declared and where its type is declared. Well-defined for named types, but not when cursor is on package name. Jump-to-def

goes to import line, jump-to-type has no defined behavior because packages aren't first-class types. Proposing jump-to-type taking you into one of the package sources.

- Rebecca: I was surprised when I saw the CL for this. Since there's no defined behavior, I'm not opposed, but I'm curious if other people would find that useful.
- Marwan: Could make the import path decl also be clickable.
- Ian: Have a pending CL for that, but it uses documentLink to take you to godoc.org.
- Ian: There's a third go-to: go-to-implementation.
- Paul: and go-to-documentation (documentLink?) is the fourth.
- Going to package source sounds good.

go/analysis

- Michael: We've settled on a mechanism to iterate through facts and putting the responsibility on analyzers. Have an experimental CL to submit today. Hoping Dominic and anyone else whose interested can take a look. Hopefully this will resolve the issue of whether we share facts between analyzers.
- Michael: Next step: we're going to start integrating staticcheck analyzers into our Google testing framework, since we also use go/analysis.
- Ian: Personal experimental fork of gopls uses staticcheck. It works really well in the editor. Won't get merged because x/tools can't depend on staticcheck. Need to think about ways for people to easily add analyzers though.

cmd/go

- Jay: working on get -u improvements. Planning to improve module diagnostic improvements during the freeze.
- Marwan: Will proxy.golang.org be on by default for 1.13, 1.14?
- Jay: we haven't reached a final decision yet.
- Bryan: Have been offline a bit over the last week: a close friend died, and I've been focused on working through that. Still thinking about authentication and improved replace. Working with Russ to decide what to do during the freeze.

Error propagation in gopls

- Rebecca: Trying to be better about surfacing diagnostics, when you have e.g., parse errors. Need to decide what should be done by editor, what should be done by gopls. Returning no format with no error means the editor can't make a decision.
- Paul: Sounds like the really thorny cases are around formatting.
- Rebecca: Emacs was showing errors when hover failed. Weird case.
- Paul: As more editors are integrating this, we'll start to see best practices.
- Rebecca: Wasn't shy about returning errors before because VSCode was throwing them away.

Global install command [#30515](#)

- Ian: There's a lot of subtle issues, too late to get it done before the freeze. Want to write a list of use cases and discuss solutions. Convinced that most use cases don't belong in cmd/go.

Adding go.mod files for gopls

- Ian: Lots of ideas for things we can add, particularly using code lenses. That means clients will need to talk to gopls as well for go.mod files.
- Paul: Would love to have a quick demo to understand these things.

Next time

- Paul: Tentatively May 28. Looking forward to get Katie on to talk about the new mirror (proxy.golang.org).