

2021-09-15 - golang-tools session: go/ast and go/types support for type parameters

Previous session notes

- https://docs.google.com/document/d/1fFwOXTOmPvR2E240z5WC1_VVs7h5vhcH4M81qT-2fW0/edit

Details

15:30 UTC

<https://meet.google.com/xuq-tcoc-dkp>

Hosted by rfindley@google.com

Agenda

- ~~Start recording the session!~~ Oops.
- Changes to go/ast
- Changes to go/types
- A transitional package
- Sample use cases
 - Writing an importer
 - Writing an analyzer
- Discussion
 - @rogppe suggested that we rename `Named.Orig` to `Origin`: `Origin` is a better expansion than `Original`. I agree, and spelling it out is better.
 - @mvdan suggested we switch from `SetTypeParams` APIs to generic constructors (`NewGenericNamed`, `NewGenericSignature`), and others agreed. The point being that we should expose a mutable API if it is not strictly necessary.
 - We've gone back and forth on this, and maybe we arrived at the wrong conclusion.
 - @rogppe suggested to use a pointer receiver for `ArgumentError` methods.
 - @rogppe asked what `Interface.IsConstraint` will mean if type restricting elements are permitted in any interface? Will it always return false? What if others rely on it meaning 'has type restricting elements'? Maybe if we choose to expose something like `Interface.UnderIs`, this method can be removed.
 - On that note, @jba asked about the name `UnderIs`. We should think of a better name before officially proposing this API.

Recording

To follow post meeting

Next meeting