

# Go-critic Static analysis in Go

by Oleg Kovalov

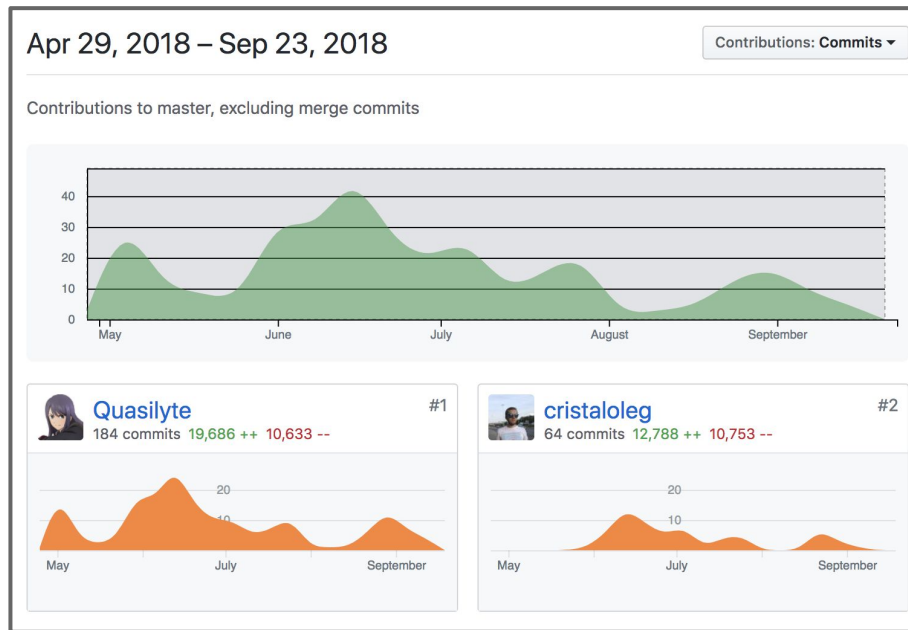


- Better code
- Human friendly
- Optimal code

Good examples:

- golint
- go vet
- gometalinter
- golangci-lint
- go-critic 😊

- 58 checks implemented + 54 in current todo
- Optimistic merging (10+ contributors)



- Diagnostics
  - Suspicious code
- Code style
  - Can be more readable
- Performance
  - Can be faster

Detects return statements those results evaluate to nil.

**Before:**

```
if err == nil {  
    return err  
}
```

**After:**

```
// typo in "==", change to "!="  
if err != nil {  
    return nil  
}
```

Detects literals that can be replaced with defined named const.

**Before:**

```
// pos has type of token.Pos.  
return pos != 0
```

**After:**

```
return pos != token.NoPos
```

Detects strings.Index calls that may cause unwanted allocs.

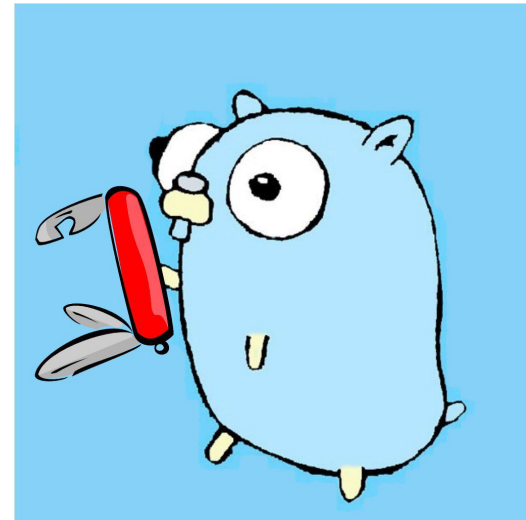
**Before:**

```
strings.Index(string(x), y)
```

**After:**

```
bytes.Index(x, []byte(y))
```

- astfmt - Print ast.Node with %s
- astp - Predicates for AST nodes
- astcopy - Deep copy for AST nodes
- astequal - Deep equal for AST nodes
- astinfo - Useful AST information
- strparse - Parse string to AST



Clickable link: [github.com/go-toolsmith](https://github.com/go-toolsmith)



- `strings.Index("<long string>", "i")`

vs

- `strings.IndexByte("<long string>", 'i')`

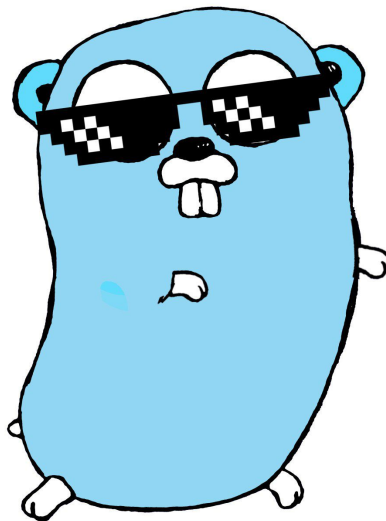
BenchmarkIndex-4	1000000000	19.3 ns/op
BenchmarkIndexByte-4	1000000000	16.4 ns/op

- go get go-critic
- make new <checker\_name>
- adding code examples (positive & negative)
- ~~copy-pasting~~ coding
- adding docs
- git push
- ....
- PROFIT

Code: <https://bit.ly/2NDbOtV>

Our trophies collection:

[go-critic.github.io/trophies.html](https://go-critic.github.io/trophies.html)



- [github.com/go-critic/go-critic](https://github.com/go-critic/go-critic)
- [github.com/go-toolsmith](https://github.com/go-toolsmith)
- [twitter.com/oleg\\_kovalov](https://twitter.com/oleg_kovalov)
- [github.com/cristaloleg](https://github.com/cristaloleg)



Thank you