

Introducing Go(Golang)



Answering “WHY”
you should use it

Evelyn Haslinger
Markus Zimmermann

eh@symflower.com
mz@symflower.com

Symflower - Automating QA



Product to autonomously generate and execute unit-tests



Support and training for all software testing needs



Implementation of modern development processes using the right tools

Write us at hello@symflower.com

Why Go?



Why you should get excited

Why Go? (1/4)

- Go is a minimal language by design ([Spec](#))
- Go included modern features into the language
 - (Hash)Maps, Functions are first-class
 - Concurrency (green-threads->goroutines)
 - Channels
 - ...
- Go is (usually) statically compiled(linked)
 - \$ ldd go-mutesting vs \$ ldd golint
 - No interpreter or “virtual machine”

Why Go? (2/4)

- Go is garbage-collected
- Statically typed but...
- Implicit interfaces
 - Checked at compile time
 - Dynamic when asked for
- Go already proved that it handles BIG projects
 - Docker, Kubernetes, CockroachDB, Prometheus

Why Go? (3/4)

- TOOLS! I cannot stress this enough so: TOOLS!!!
 - Everything you usually need is built-in
 - `$ go get`
 - `$ go install` (build system is included)
 - `$ go fmt` (Picked up by almost all other languages)
 - `$ go test` (includes static analysis)
 - `$ go ... -race (Race detector)`
 - `$ go tool pprof / $ go tool trace`

Why Go? (4/4)

- Packages
 - Official in the distribution: <https://golang.org/pkg/>
 - You can do almost everything with the STD
 - Official out of the distribution:
<https://golang.org/pkg/#subrepo> (golang.org/x)
- [Really good documentation](#)
- Runs everywhere ([cross-compile](#))

Go, take a look



<https://tour.golang.org>

Go is simple, Go is fast, Go is safe

Evelyn Haslinger
Markus Zimmermann

eh@symflower.com
mz@symflower.com



symflower
AUTOMATING QUALITY ASSURANCE

Evelyn Haslinger
Markus Zimmermann

eh@symflower.com
mz@symflower.com