Writing your own Generator in Go

$ whoami

- **Agni Bhattacharyya**
  - Loves programming, travelling, clicking photos.

- Currently working at **FunnelStory** (Software Engineer, IC lvl. 2)
  - Writing internal APIs in Go.
  - External API clients and integrations.
  - Creating custom AI agents do cool stuff.

- Past Experiences -
  - Worked on Python, Flask, Django.
  - Managed K8s clusters - upgrades, deployments, debugging, etc.
  - Infra changes via terraform, creating helm charts

- OSS
  - Contributor at Grafana

Code Generating

- Code generation is the process of programmatically producing source code that would otherwise be written manually

- Benefits -
  - Minimal bugs as no humans are involved.
  - Reduced boilerplate code
  - Saves developer time

Go generate

There are 3 main parts -

- Generators
- Go directives / Magic comments
- *go generate*

Examples

```
//go:generate command arg1 arg2
//go:generate go run generator/main.go
//go:generate mockgen -source=interface.go -destination=mock_interface.go
```


package project
//go:generate echo Hello, World
func Add(x, y int) int {
return x + y
}


$ go generate
Hello, World

DEMO

Conclusion

- Use generators when you have repetitive, predictable patterns - code or anything else like JSON, YAML, etc.
- Keep generators focused and single-purpose

- Next steps -
  - Explore different existing generators like stringer
  - Try to write small generators yourself.
  - Find repetitive patterns in your own codebase

Source code