

# PROTOSBUF & CODE GENERATION

2016. BY MANFRED TOURON (@MOUL)

# OVERVIEW

- > GO-KIT IS AN AMAZING FRAMEWORK TO DEVELOP STRONG MICRO SERVICES
  - > BUT IT REQUIRES A LOT OF BOILERPLATE CODE
- > RETURN ON EXPERIENCE ON GO-KIT BOILERPLATE CODE GENERATION

# PROTOBUF?

- > LIMITED TO EXCHANGES (METHODS AND MODELS)
  - > EXTENDABLE WITH PLUGINS
    - > CONTRACT-BASED
      - > UNIVERSAL

# CODE GENERATION?

- > THE GOOD OLD `./GENERATE.SH` BASH SCRIPT
  - > `GO:GENERATE`
  - > `MAKE`
- > `PROTOBUF` + `PROTOC-GEN-GOTEMPLATE`

# GO-KIT

- > PROTOBUF-FIRST, RPC-FIRST SERVICE FRAMEWORK IN GOLANG
  - > ABSTRACT SERVICES, ENDPOINTS, TRANSPORTS
- > REQUIRES A LOT OF BOILERPLATE CODE IN MULTIPLE PACKAGES

# EXAMPLE: session.proto

```
syntax = "proto3";  
package session;  
  
service SessionService {  
    rpc Login(LoginRequest) returns (LoginResponse) {}  
}  
  
message LoginRequest {  
    string username = 1;  
    string password = 2;  
}  
  
message LoginResponse {  
    string token = 1;  
    string err_msg = 2;  
}
```

# EXAMPLE: session.go

```
package sessionsvc

import (
    "fmt"
    "golang.org/x/net/context"
    pb "github.com/moul/protoc-gen-gotemplate/examples/go-kit/services/session/gen/pb"
)

type Service struct{}

func New() pb.SessionServiceServer {
    return &Service{}
}

func (svc *Service) Login(ctx context.Context, in *pb.LoginRequest) (*pb.LoginResponse, error) {
    // custom code here
    return nil, fmt.Errorf("not implemented")
}
```

# EXAMPLE: {{.File.Package}}/gen/transport/http/http.go.tpl

```
// source: templates/{{.File.Package}}/gen/transport/http/http.go.tpl
package {{.File.Package}}_httptransport
import (
    gokit_endpoint "github.com/go-kit/kit/endpoint"
    httptransport "github.com/go-kit/kit/transport/http"
    endpoints "github.com/moul/protoc-gen-gotemplate/examples/go-kit/services/{{.File.Package}}/gen/endpoints"
    pb "github.com/moul/protoc-gen-gotemplate/examples/go-kit/services/{{.File.Package}}/gen/pb"
)

// result: services/user/gen/transport/http/http.go
package user_httptransport
import (
    gokit_endpoint "github.com/go-kit/kit/endpoint"
    httptransport "github.com/go-kit/kit/transport/http"
    endpoints "github.com/moul/protoc-gen-gotemplate/examples/go-kit/services/user/gen/endpoints"
    pb "github.com/moul/protoc-gen-gotemplate/examples/go-kit/services/user/gen/pb"
)
```



# EXAMPLE: `{{.File.Package}}/gen/transport/http/http.go.tpl`

```
// source: templates/{{.File.Package}}/gen/transport/http/http.go.tpl
{{range .Service.Method}}
func Make{{.Name}}Handler(ctx context.Context, svc pb.{{$file.Package | title}}ServiceServer, endpoint gokit_endpoint.Endpoint) *httptransport.Server {
    return httptransport.NewServer(
        ctx,
        endpoint,
        decode{{.Name}}Request,
        encode{{.Name}}Response,
        []httptransport.ServerOption{...},
    )
}
{{end}}
```

```
// result: services/user/gen/transport/http/http.go
func MakeGetUserHandler(ctx context.Context, svc pb.UserServiceServer, endpoint gokit_endpoint.Endpoint) *httptransport.Server {
    return httptransport.NewServer(
        ctx,
        endpoint,
        decodeGetUserRequest,
        encodeGetUserResponse,
        []httptransport.ServerOption{...},
    )
}
```

# EXAMPLE: `{{.File.Package}}/gen/transport/http/http.go.tpl`

```
// source: templates/{{.File.Package}}/gen/transport/http/http.go.tpl
func RegisterHandlers(ctx context.Context, svc pb.{{$file.Package | title}}ServiceServer, mux *http.ServeMux, endpoints endpoints.Endpoints) error {
    {{range .Service.Method}}
        log.Println("new HTTP endpoint: \"/{{.Name}}\" (service={{$file.Package | title}})")
        mux.Handle("/{{.Name}}", Make{{.Name}}Handler(ctx, svc, endpoints.{{.Name}}Endpoint))
    {{end}}
    return nil
}
```

```
// result: services/user/gen/transport/http/http.go
func RegisterHandlers(ctx context.Context, svc pb.UserServiceServer, mux *http.ServeMux, endpoints endpoints.Endpoints) error {

    log.Println("new HTTP endpoint: \"/CreateUser\" (service=User)")
    mux.Handle("/CreateUser", MakeCreateUserHandler(ctx, svc, endpoints.CreateUserEndpoint))

    log.Println("new HTTP endpoint: \"/GetUser\" (service=User)")
    mux.Handle("/GetUser", MakeGetUserHandler(ctx, svc, endpoints.GetUserEndpoint))

    return nil
}
```

```
protoc --gogo_out=plugins=grpc:. ./services/*/*.proto
```

```
protoc --gotemplate_out=template_dir=./templates:services ./
services/*/*.proto
```

```
→ user git:(slides) wc -l $(find gen -name "*.go")
```

```
63 gen/endpoints/endpoints.go
```

```
276 gen/pb/pb.go
```

```
75 gen/transport/grpc/grpc.go
```

```
68 gen/transport/http/http.go
```

```
482 total
```

```
→ user git:(slides) wc -l *.*
```

```
21 service.go
```

```
28 user.proto
```

```
49 total
```

3 SERVICES  
6 METHODS  
149 CUSTOM LINES  
1429 GENERATED LINES  
BUSINESS FOCUS

```
→ go-kit git:(slides) x wc -l services/**/*.  
19 services/session/service.go  
17 services/session/session.proto  
27 services/sprint/service.go  
37 services/sprint/sprint.proto  
21 services/user/service.go  
28 services/user/user.proto  
149 total  
→ go-kit git:(slides) x wc -l $(find services/*/gen -name "*.go")  
40 services/session/gen/endpoints/endpoints.go  
178 services/session/gen/pb/pb.go  
48 services/session/gen/transport/grpc/grpc.go  
43 services/session/gen/transport/http/http.go  
86 services/sprint/gen/endpoints/endpoints.go  
357 services/sprint/gen/pb/pb.go  
102 services/sprint/gen/transport/grpc/grpc.go  
93 services/sprint/gen/transport/http/http.go  
63 services/user/gen/endpoints/endpoints.go  
276 services/user/gen/pb/pb.go  
75 services/user/gen/transport/grpc/grpc.go  
68 services/user/gen/transport/http/http.go  
1429 total
```

# GENERATION USAGES

- > GO-KIT BOILERPLATE (SEE [EXAMPLES/GO-KIT](#))
  - > K8S CONFIGURATION
    - > DOCKERFILE
  - > DOCUMENTATION
  - > UNIT-TESTS
    - > FUN

# PROS

- > SMALL CUSTOM CODEBASE
- > TEMPLATES SHIPPED WITH CODE
- > HARDLY TYPED, NO REFLECTS
  - > GENERICITY
- > CONTRAT TERMS (PROTOBUF) RESPECTED
  - > NOT LIMITED TO A LANGUAGE



# CONS

- > THE AUTHOR NEEDS TO WRITE ITS OWN TEMPLATES
  - > SOMETIMES DIFFICULT TO GENERATE VALID CODE
- > NOT ENOUGH HELPERS AROUND THE CODE GENERATION YET

# IMPROVEMENT IDEAS

- > SUPPORT PROTOBUFS EXTENSIONS (I.E. ANNOTATIONS.PROBO)
  - > GENERATE ONE FILE FROM MULTIPLE SERVICES
  - > ADD MORE HELPERS AROUND THE CODE GENERATION

# CONCLUSION

- > USEFUL TO KEEP EVERYTHING STANDARD
- > THE AWESOMENESS OF GO-KIT WITHOUT THE HASSLE OF WRITING BOILERPLATE CODE
- > ALWAYS UP-TO-DATE WITH THE CONTRACTS

# QUESTIONS?

GITHUB.COM/MOUL/PROTOC-GEN-  
GOTEMPLATE

@MOUL