

# Grab Golang Bootcamp

Week 3 - Session 2



# gRPC CLI

- Can be used to introspect server protos and send/receive test RPCs

```
cang.ta@ITVN000860-MAC ~GP (master••)$ grpc_cli ls localhost:50051 -l
filename: grpc_reflection_v1alpha/reflection.proto
package: grpc.reflection.v1alpha;
service ServerReflection {
  rpc ServerReflectionInfo(stream grpc.reflection.v1alpha.ServerReflectionRequest) returns (stream grpc.reflection.v1alpha.ServerReflectionResponse) {}
}

filename: hello.proto
package: helloworld;
service Greeter {
  rpc SayHello(helloworld.HelloRequest) returns (helloworld.HelloReply) {}
}
```

# gRPC CLI - Install

- Prerequisites
  - Install OS packages
    - <https://github.com/grpc/grpc/blob/master/BUILDING.md>
- Install gRPC CLI
  - <https://github.com/grpc/grpc-go/blob/master/Documentation/server-reflection-tutorial.md>

## gRPC CLI - How to use

- `grpc_cli help`
- `grpc_cli ls localhost:50051 -l`
- `grpc_cli call localhost:50051  
helloworld.Greeter.SayHello  
"name: 'world' "`

# Context

- Be used to cancel, or halt an operation mid-way, especially with goroutines.
- Add additional-info along, e.g. user's email.
- Links
  - <https://golang.org/pkg/context/>
  - <https://blog.golang.org/context>
  - [http://p.agnihotry.com/post/understanding the context package in golang/](http://p.agnihotry.com/post/understanding_the_context_package_in_golang/)
  - <https://www.sohamkamani.com/blog/golang/2018-06-17-golang-using-context-cancellation/>

# Exercise

- Create a gRPC server and client for user management.
  - Error handling on client side
  - Create user
  - List user
  - Get user detail
  - Update user detail
  - Delete user
  - (Optional) HTTP RESTful APIs (use gin or [grpc-gateway](#))

# Q & A

## THANK YOU!