

---

# GO AND DISTRIBUTED SYSTEMS

---

A quick intro by Jeromy Johnson

---

# SOME USEFUL LIBRARIES

```
1 package main
2
3 import (
4     "encoding/json"
5     "net"
6 )
7
8 type MyMessage struct {
9     Sequence int
10    Message  string
11 }
12
13 func main() {
14     con, err := net.Dial("tcp", "localhost:9000")
15     if err != nil {
16         panic(err)
17     }
18     defer con.Close()
19
20     enc := json.NewEncoder(con)
21     err = enc.Encode(&MyMessage{
22         Sequence: 1,
23         Message:  "Hello World",
24     })
25     if err != nil {
26         panic(err)
27     }
28 }
```

- the standard lib is awesome
- net
  - socket programming
- encoding/json
  - struct marshaling
- net/rpc
  - remote function calls

# GOROUTINES ARE SWEET

```
1 package main
2
3 import (
4     "fmt"
5     "time"
6 )
7
8 func main() {
9     comm := make(chan int)
10    go func() {
11        defer close(comm)
12        for i := 0; i < 10; i++ {
13            comm <- i
14            time.Sleep(time.Second)
15        }
16    }()
17
18    for n := range comm {
19        fmt.Println(n)
20    }
21 }
```

- really simple green threads
- channels are pretty cool too