

Implementing Reflection



Jeremy Morgan

SOFTWARE / DEVOPS ENGINEER

@JeremyCMorgan www.jeremymorgan.com



What is Reflection?



Types in Go



Basic Types

Aggregate Types

Reference Types

Interface Types



The Empty Interface

```
func DoSomething(t interface{}) {  
    switch reflect.TypeOf(t).Name() {  
        // change action based on type expected  
    }  
}
```

```
DoSomething(interfaceOne)
```

```
DoSomething(interfaceTwo)
```

Overview



Creating custom types

Accessing types at runtime

Creating types at runtime

Creating functions at runtime



Demo



Creating custom types



Demo



Accessing types at runtime



Demo



Creating types at runtime



Demo



Creating Functions



Summary



Created custom types

Accessing types at runtime

Creating types at runtime

Creating functions at runtime



What Have We Learned?



CLI Applications



Created some CLI applications

Read command line arguments

Utilized flags

Read keyboard inputs

Learned about bufio

FMT Package



Input scanning

Formatting output

Manipulating strings

Formatting other data types



Logging



Understanding error levels

Formatting log output

Creating useful log files

Utilizing the trace logger



Time



Wall clock vs. monotonic clock

Formatted time output

Calculated time spans

Calculated elapsed time for applications



Strings



How strings work in Go

Comparing strings

String splitting

Find and replace in strings

Trimming strings

Casing strings



Reflection



Created custom types

Accessing types at runtime

Creating types at runtime

Creating functions at runtime



Go out and write some code!



Thank You!



Jeremy Morgan

SOFTWARE / DEVOPS ENGINEER

@JeremyCMorgan www.jeremymorgan.com

