Golang Session

- Harsh Dusane

Topic : Logging Go Programs

Logging in Golang

- The standard library package log provides a basic infrastructure for log management in GO language that can be used for logging our GO programs. The main purpose of logging is to get a trace of what's happening in the program, where it's happening, and when it's happening. Logs can be providing code tracing, profiling, and analytics. Logging(eyes and ears of a programmer) is a way to find those bugs and learn more about how the program is functioning.

```
import (
        "log"
)
```

Sample Code

```go
// Program in GO language to demonstrates how to use base log package.
package main
import (
        "log"
)
func init(){
        log.SetPrefix("LOG: ")
        log.SetFlags(log.Ldate | log.Lmicroseconds | log.Llongfile)
        log.Println("init started")
}
func main() {
// Println writes to the standard logger.
        log.Println("main started")
// Fatalln is Println() followed by a call to os.Exit(1)
        log.Fatalln("fatal message")
// Panicln is Println() followed by a call to panic()
        log.Panicln("panic message")
}
```

Sample Code

```
// Program in GO language with real world example of logging.
package main
import (
"log"
"net/smtp"
)
func init(){
        log.SetPrefix("TRACE: ")
        log.SetFlags(log.Ldate | log.Lmicroseconds | log.Llongfile)
        log.Println("init started")
}
func main() {
// Connect to the remote SMTP server.
client, err := smtp.Dial("smtp.smail.com:25")
        if err != nil {
                log.Fatalln(err)
        }
client.Data()
}
```