Using the Log Package

What is The Log Package?

Overview

Understanding error levels

Formatting log output

Creating useful log files

Utilizing the Trace Logger

Understanding error levels

What are Error Levels?

Some Basic Error Levels

- Information
- Warning
- Error
- Fatal

Error Levels

Information

Warning

Error

Fatal

Information Level

When to use

If you want to confirm something

If you're logging transactions

If you're keeping track of something

If you want to show runtime information

When not to use

A serious condition

Something misconfigured

Anything that compromises operation

Warning Level

When to use

You need to get the users attention

When something is misconfigured

When not to use

Operation is compromised

The program might fail

Error Level

When to use

Operation is compromised

Something very unexpected happened

When not to use

The program must shut down

Fatal Level

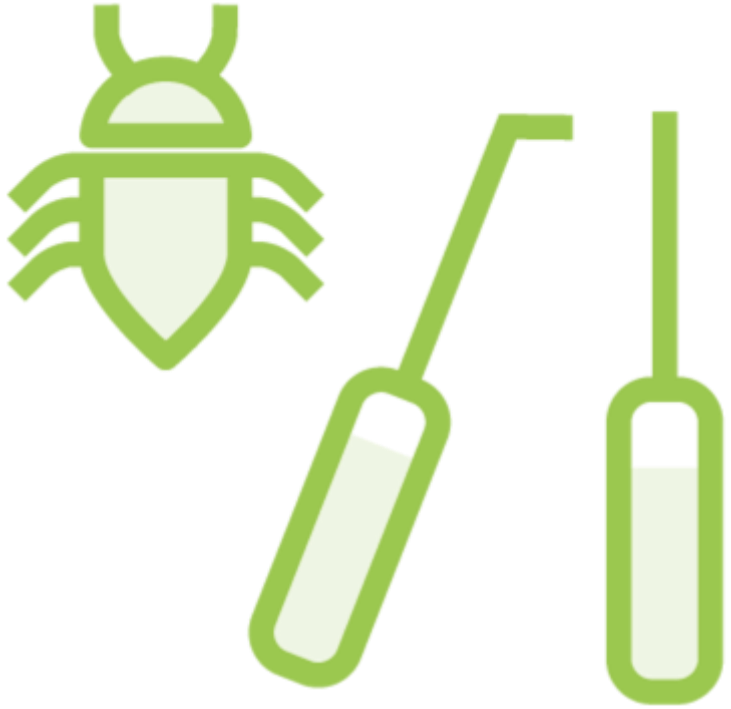
When to use

When the program must stop running

When not to use

When the program can still run

Debug

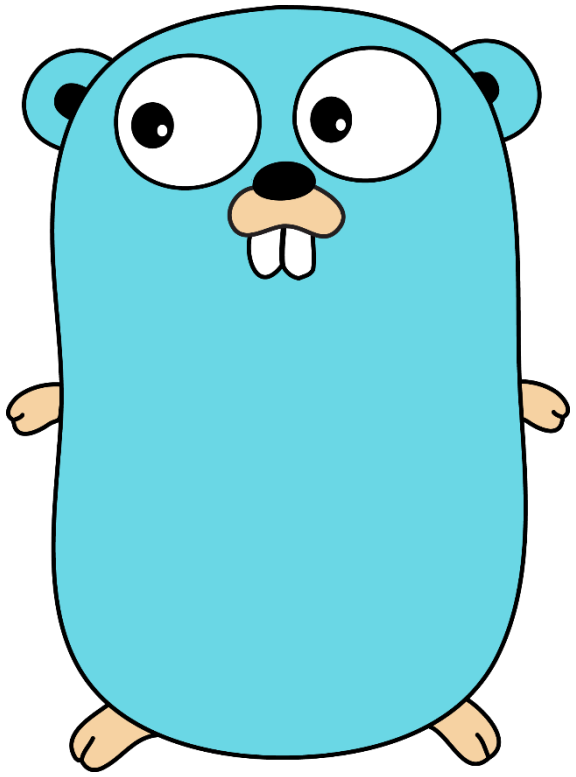


Demo

Formatting log output

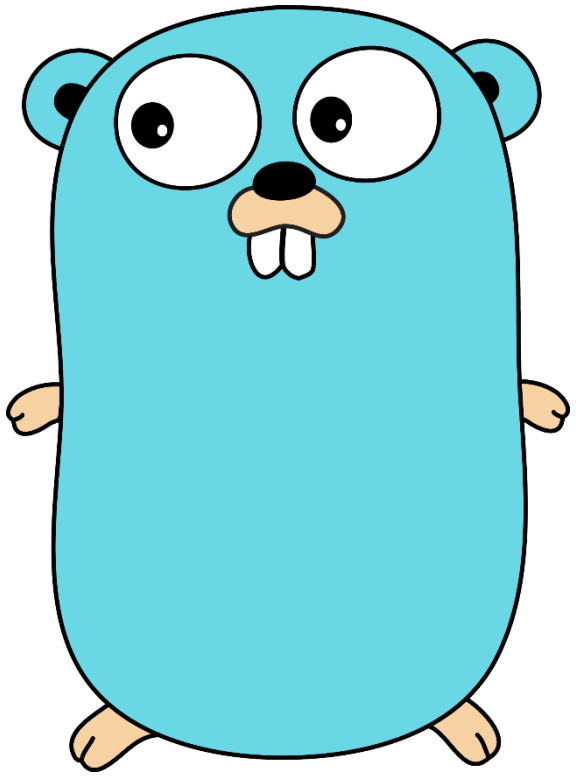
Creating Useful Log Files

Why Log things?



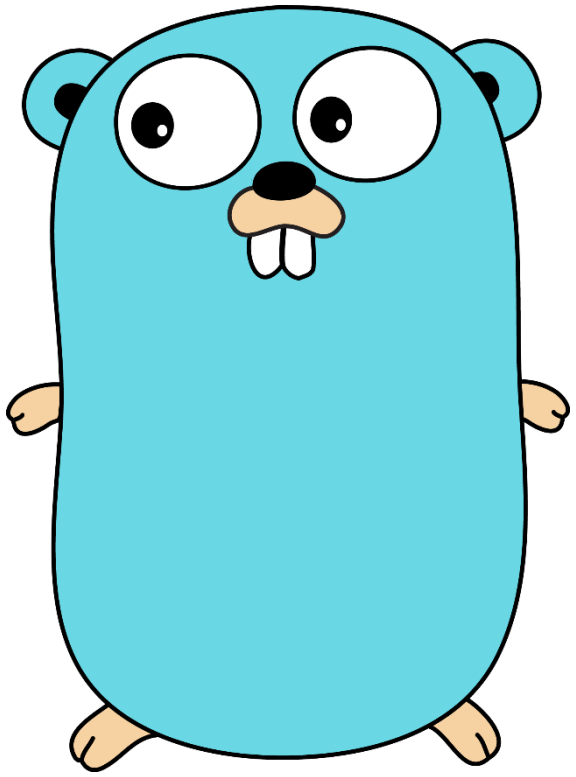
- Post-mortem analysis
- Spot bugs in your code
- Discover configuration issues
- Investigate performance problems

What Should We Log?



- Unusual events
- Error levels
- Configuration settings
- Telemetry information

What Not to Log



- Personal information
- Credential information
- Financial information
- IP addresses

Use error levels to your advantage, so you can be as verbose or concise as you like.

Demo

Utilizing the Trace Logger

Summary

- Understanding error levels
- Formatting log output
- Creating useful log files
- Utilizing the Trace Logger