

Creating Log Messages



Maaike van Putten
Software Developer & Trainer

www.brightboost.nl



Overview/ Summary



- Overview of `java.util.logging`
- Getting a Logger
- Log levels
- Different log methods
- Best practices
- Solve the Carved Rock Fitness order problem using logging





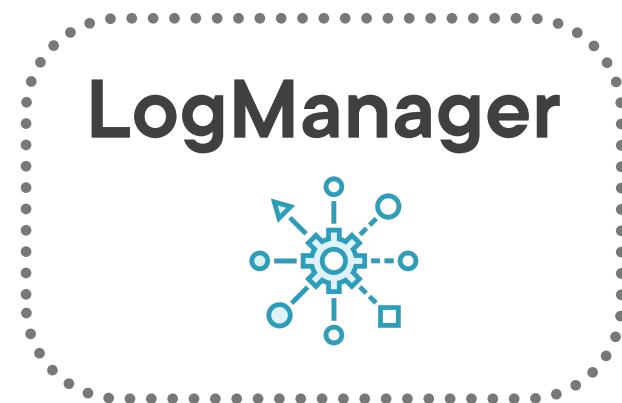
Different loggers available

Very popular library: log4j

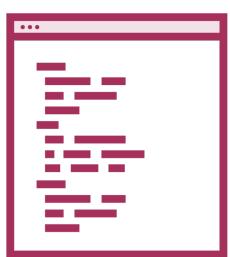
Since Java 1.4: java.util.logging



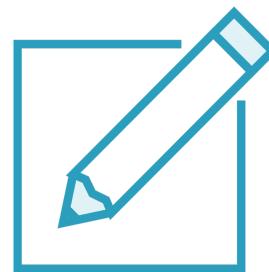
Overview Java Logging



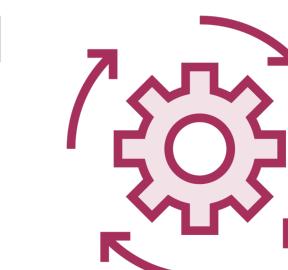
App



Logger



LogRecord



External log file



Filter



Filter



Formatter



Demo



- Get a logger
- Log!





Different log levels

Log levels represent severity of situation

Important to use the right one



SEVERE

WARNING

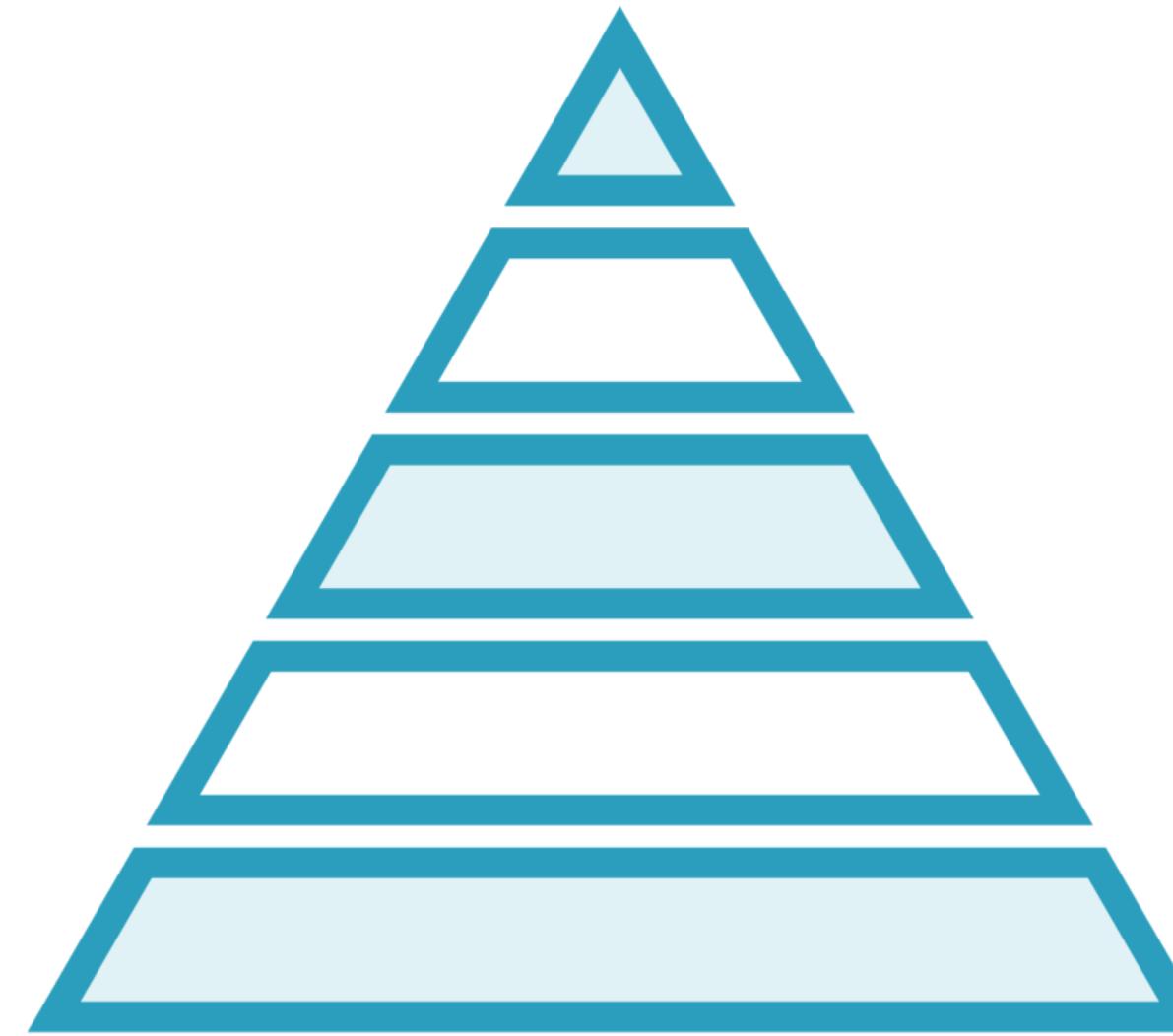
INFO

CONFIG

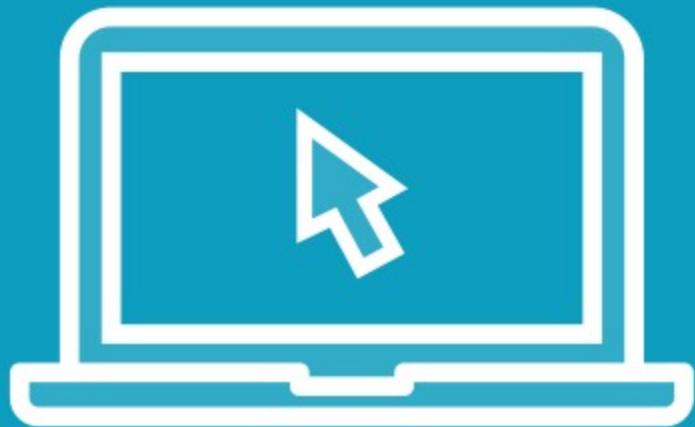
FINE

FINER

FINEST



Demo



- Set the level
- Check the log





Log Handlers

Sends the message to the log place

Handlers format the message using a formatter





ConsoleHandler

FileHandler

StreamHandler

SocketHandler

MemoryHandler



Logging Best Practice

Be precise

No sensitive data

Right log level

**Machine and human
readable**

Don't log too much

Don't log too little



Different Log Methods

```
LOGGER.log(Level.INFO, "This is an example message");
```

```
LOGGER.log(Level.INFO, LogExample.class.getName(), "sourceMethodName", "This is an example  
message");
```

```
LOGGER.logrb(Level.INFO, ResourceBundle.getBundle("en_US"), "This is an example message");
```



Demo



- Use different log methods
- Check the log



Demo



- Let's add logging to our backend
- And order again
- Check the log messages



Up Next:
Configuring and Formatting Logging

