LoggingExamples

Jim Northrop

Status

Project Health - [Build Status (Travis CI)]

Logback Tutorial

If you need a brief intro to Logback and how it works with SLF4J, then see this.

Notes About Logging

Here are some Java and Groovy Notes and Samples to use SLF4J logs. There is also an example of Spock test framework too.

When slf4j catches a log messages with a logger, that msg is given to an appender which decides what to do with the message. By default, the ConsoleAppender displays the message in the console. But it can be configured to write the msg to a file. It does this because of the configuration file.

> 💡 Jenkov has a GR8 tut about Java Logging here.

Java Configuration File

If no configuration class is specified, you can instead specify a configuration file (but no configuration class can be specified then!).

The **Java Logging API** has a default logging configuration file located at **"lib/logging.properties"**, inside the JRE directory. If you edit this file, you edit the default logging settings for the entire JRE, for every program executed. This is most often not what you want to do, though.

Logback Configurations

Look at the **config** folder. You will see two configuration files. The **logback.xml** file is used in this set of examples.

XML to write each log entry to both the console and an external logging file.

```
  <!-- Configure so that it outputs to both console and log file -->
  <root level="DEBUG">
    <appender-ref ref="FILE" />
    <appender-ref ref="STDOUT" />
  </root>
```

Log File Names

Naming Convention for Output Log Files

```
<fileNamePattern>logs/LoggingExamples.%d{yyyy-MM-dd}.log</fileNamePattern>
```

The prior pattern asks the loggers to write a file into the **log** folder (in this project root folder) with a name pattern. My tests on an Apple Macbook also write these msgs into the Apple system log folder.

My tests at the moment cause each log message to be written to a file named **LoggingExamples.2015-08-25.log**

Sample Log Entries

```
logback.xml 23:58:56.186 [main] INFO  MyExample:Instantiated MyExample
logback.xml 23:58:56.257 [main] DEBUG MyExample:Still alive... at interation 1
logback.xml 23:59:06.280 [main] WARN  MyExample:-- this is a warning
logback.xml 23:59:06.281 [main] DEBUG MyExample:Still alive... at interation 2
logback.xml 23:59:16.282 [main] WARN  MyExample:-- this is a warning
logback.xml 23:59:16.285 [main] DEBUG MyExample:Still alive... at interation 3
logback.xml 23:59:26.286 [main] WARN  MyExample:-- this is a warning
logback.xml 23:59:26.288 [main] DEBUG MyExample:Still alive... at interation 4
```

More Recent Findings

These were taken from MrHaki at this address.

See File: CombinedLogSample.groovy

Slf4j

Add dependencies for Slf4j API (and using Logback - failed as it needs config file: **File: src/main/resources/logback.groovy**)

And logback is config'd by a groovy script, see: configure logback or this xml config for logback.xml to an output file, so mkdir ./config at $projectDir like ./src then put **logback.xml** into it with something like this:

Example 1. logback.xml

<configuration>

```
<appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
  <encoder>
    <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>
  </encoder>
</appender>
```

```
<appender name="FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">
  <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
    <!-- Daily rollover -->
    <fileNamePattern>log/MyExample.%d{yyyy-MM-dd}.log</fileNamePattern>
```

```
  <!-- Keep 7 days' worth of history -->
  <maxHistory>7</maxHistory>
</rollingPolicy>
```

```
  <encoder>
    <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>
  </encoder>
</appender>
```

```
  <!-- Configure so that it outputs to both console and log file -->
  <root level="DEBUG">
    <appender-ref ref="FILE" />
    <appender-ref ref="STDOUT" />
  </root>
</configuration>
```

Besides an annotation for the Slf4j API other logging frameworks are supported with annotations:

Groovy Annotation

```
Logging framework        Annotation
java.util.logging        @Log - only .config .fine .finer .info methods
Log4j                    @Log4j - only works if .config file present otherwise no-op
Apache Commons Logging   @Commons
Slf4j API                @Slf4j - needs a backing log tool, see:
http://logback.qos.ch/manual/groovy.html
```

Project Construction

This project uses the **gradle** build tool. To make it easy to play with this project, the full gradle wrapper feature is included. This means you will not need to install gradle to run this set of log tests. You only need :

1. Java JVM 1.6 or more recent
2. active internet connection
3. a **Git** client to do a **git clone** **https://github.com/jnorthr/LoggingExamples.git** command

This will create a directory folder named **LoggingExamples** and then **cd** change directories into that folder.

Your first console command will be either **gradlew** on Windows systems or **./gradlew** on Apples and Linux kit.

The **Gradle** build tool will reach across the internet the first time it's run. This is to download and install needed pieces of code to make the magic happen. So the first run will take a few minutes.

Gradle knows it's stuff by reading a configuration file named **build.gradle**. If you look at that file, you will see that i have pre-declared the default tasks to be executed. Look for the line reading: **defaultTasks 'clean', 'build', . . .** as these names are the names of tasks that **gradle** will run for you.

So run the **gradlew** wrapper and look at the results. Play with the source code files in the **src** folder. Did not have a lot of time to document what each script file does, but stick with it. Make a few changes to the groovy scripts. Add more **log.INFO** samples then try the wrapper again.

This sample project may teach you enough to get started doing some log messages in your own java or groovy projects. **Good luck !**

Sample Joblog

```
RedApple:jim /Volumes/REDUSB/LoggingExamples $ ./gradlew
```

```
Parallel execution is an incubating feature.
:clean
:compileJava UP-TO-DATE
:compileGroovy
:processResources UP-TO-DATE
:classes
:jar
:assemble
:compileTestJava UP-TO-DATE
:compileTestGroovy
:processTestResources UP-TO-DATE
:testClasses
:test
:check
:build
:runHW
Aug 26, 2015 12:43:39 AM org.codehaus.groovy.runtime.m12n.MetaInfExtensionModule
newModule
WARNING: Module [groovy-all] - Unable to load extension class
[org.codehaus.groovy.runtime.NioGroovyMethods]
HelloWorld.main()
00:43:41.653 [main] INFO  HelloWorld - this is an .info msg from Hello World
:runHW2
Aug 26, 2015 12:43:44 AM org.codehaus.groovy.runtime.m12n.MetaInfExtensionModule
newModule
WARNING: Module [groovy-all] - Unable to load extension class
[org.codehaus.groovy.runtime.NioGroovyMethods]
Aug 26, 2015 12:43:45 AM java_util_logging_Logger$info$4 call
INFO: Hello2 info logging !
Aug 26, 2015 12:43:45 AM java_util_logging_Logger$warning$5 call
WARNING: Hello2 warning logging !
Aug 26, 2015 12:43:45 AM java_util_logging_Logger$severe$6 call
SEVERE: Hello2 severe logging !
Hello world2! - uses standard java.util.logging.Logger
:runHello
Aug 26, 2015 12:43:47 AM org.codehaus.groovy.runtime.m12n.MetaInfExtensionModule
newModule
WARNING: Module [groovy-all] - Unable to load extension class
[org.codehaus.groovy.runtime.NioGroovyMethods]
HelloWorld.main()
00:43:49.078 [main] INFO  HelloWorld - this is an .info msg from Hello World
:runHello2
Aug 26, 2015 12:43:51 AM org.codehaus.groovy.runtime.m12n.MetaInfExtensionModule
newModule
WARNING: Module [groovy-all] - Unable to load extension class
[org.codehaus.groovy.runtime.NioGroovyMethods]
Aug 26, 2015 12:43:52 AM java_util_logging_Logger$info$4 call
INFO: Hello2 info logging !
Aug 26, 2015 12:43:52 AM java_util_logging_Logger$warning$5 call
WARNING: Hello2 warning logging !
Aug 26, 2015 12:43:52 AM java_util_logging_Logger$severe$6 call
```

```
SEVERE: Hello2 severe logging !
Hello world2! - uses standard java.util.logging.Logger
:runMrHaki
Aug 26, 2015 12:43:55 AM org.codehaus.groovy.runtime.m12n.MetaInfExtensionModule
newModule
WARNING: Module [groovy-all] - Unable to load extension class
[org.codehaus.groovy.runtime.NioGroovyMethods]
00:43:57.685 [main] DEBUG HelloWorldSlf4j - Execute HelloWorld.
00:43:57.708 [main] INFO  HelloWorldSlf4j - Simple sample to show log field is
injected.
00:43:57.722 [main] ERROR HelloWorldSlf4j - a severe msg
00:43:57.725 [main] WARN  HelloWorldSlf4j - a warning msg
log.getName()=HelloWorldSlf4j
00:43:57.828 [main] DEBUG MrHakiLogSlf4j.groovy - HelloWorldSlf4j.
:runMyExample
Aug 26, 2015 12:44:00 AM org.codehaus.groovy.runtime.m12n.MetaInfExtensionModule
newModule
WARNING: Module [groovy-all] - Unable to load extension class
[org.codehaus.groovy.runtime.NioGroovyMethods]
00:44:02.829 [main] INFO  MyExample - Instantiated MyExample
00:44:03.097 [main] DEBUG MyExample - Still alive... at interation 1
00:44:13.336 [main] WARN  MyExample - -- this is a warning
00:44:13.338 [main] DEBUG MyExample - Still alive... at interation 2
00:44:23.487 [main] WARN  MyExample - -- this is a warning
00:44:23.494 [main] DEBUG MyExample - Still alive... at interation 3
00:44:33.495 [main] WARN  MyExample - -- this is a warning
00:44:33.496 [main] DEBUG MyExample - Still alive... at interation 4
:runTest
Aug 26, 2015 12:44:46 AM org.codehaus.groovy.runtime.m12n.MetaInfExtensionModule
newModule
WARNING: Module [groovy-all] - Unable to load extension class
[org.codehaus.groovy.runtime.NioGroovyMethods]


Hello from CacheManagerTest
Aug 26, 2015 12:44:48 AM sun.reflect.NativeMethodAccessorImpl invoke0
INFO:
data does not have fred, so add
Aug 26, 2015 12:44:48 AM java_util_logging_Logger$info$0 call
INFO: data for fred added:true
flag from CacheManagerTest has(fred):true
ans from CacheManagerTest get(fred):CacheEntry(key:fred, name:Flintstone, expiry:0,
startTime:1440542687, payload:null)
ans from CacheManagerTest get(fredx):null
Aug 26, 2015 12:44:48 AM sun.reflect.NativeMethodAccessorImpl invoke0
INFO:
data does not have jim, so add
Aug 26, 2015 12:44:48 AM java_util_logging_Logger$info$0 call
INFO: data for jim added:true
added from CacheManagerTest put(jim):CacheEntry(key:jim, name:jimbo, expiry:66,
startTime:1440542688, payload:null)
```

```
ans from CacheManagerTest get(jim) has:CacheEntry(key:jim, name:jimbo, expiry:66,
startTime:1440542688, payload:null)


----------------
--> now add key+CacheEntry
Aug 26, 2015 12:44:48 AM sun.reflect.NativeMethodAccessorImpl invoke0
INFO:
data does not have eve, so add
Aug 26, 2015 12:44:48 AM java_util_logging_Logger$info$0 call
INFO: data for eve added:true
put CacheEntry from CacheManagerTest put(eve):CacheEntry(key:eve, name:, expiry:0,
startTime:1440542688, payload:horse feathers)
eve looks like this:CacheEntry(key:eve, name:, expiry:0, startTime:1440542688,
payload:horse feathers)


----------------
--> now add CacheEntry sam
Aug 26, 2015 12:44:48 AM sun.reflect.NativeMethodAccessorImpl invoke0
INFO:
data does not have sam, so add
Aug 26, 2015 12:44:48 AM java_util_logging_Logger$info$0 call
INFO: data for sam added:true
put CacheEntry from CacheManagerTest put(sam):CacheEntry(key:sam, name:, expiry:0,
startTime:1440542688, payload:play it again sam!)
sam looks like this:CacheEntry(key:sam, name:, expiry:0, startTime:1440542688,
payload:play it again sam!)
has(sam)=true
Aug 26, 2015 12:44:48 AM sun.reflect.NativeMethodAccessorImpl invoke0
INFO:
data has sam to remove
Aug 26, 2015 12:44:48 AM java_util_logging_Logger$info$0 call
INFO: data del sam removed ?true
del(sam)=true
Aug 26, 2015 12:44:48 AM sun.reflect.NativeMethodAccessorImpl invoke0
INFO:
data has jim to update from map
Aug 26, 2015 12:44:49 AM sun.reflect.NativeMethodAccessorImpl invoke0
INFO: ... k=key and v=<jim>
Aug 26, 2015 12:44:49 AM sun.reflect.NativeMethodAccessorImpl invoke0
INFO: ... k=name and v=<jnorthr>
Aug 26, 2015 12:44:49 AM sun.reflect.NativeMethodAccessorImpl invoke0
INFO: ... k=expiry and v=
Aug 26, 2015 12:44:49 AM java_util_logging_Logger$info$0 call
INFO: data jim updated:CacheEntry(key:jim, name:jnorthr, expiry:26,
startTime:1440542688, payload:null)
result from CacheManagerTest fix(jim) using a map:CacheEntry(key:jim, name:jnorthr,
expiry:26, startTime:1440542688, payload:null) and now contains:CacheEntry(key:jim,
name:jnorthr, expiry:26, startTime:1440542688, payload:null)
Aug 26, 2015 12:44:49 AM sun.reflect.NativeMethodAccessorImpl invoke0
```

```
INFO:
data has jim to update from map
Aug 26, 2015 12:44:49 AM sun.reflect.NativeMethodAccessorImpl invoke0
INFO: ... k=key and v=<jim>
Aug 26, 2015 12:44:49 AM sun.reflect.NativeMethodAccessorImpl invoke0
INFO: ... k=payload and v=<update jim payload>
Aug 26, 2015 12:44:49 AM sun.reflect.NativeMethodAccessorImpl invoke0
INFO: ... k=expiry and v=③
Aug 26, 2015 12:44:49 AM java_util_logging_Logger$info$0 call
INFO: data jim updated:CacheEntry(key:jim, name:jnorthr, expiry:3,
startTime:1440542688, payload:update jim payload)
result from CacheManagerTest cmt.fix(jim)'s payload using a map:CacheEntry(key:jim,
name:jnorthr, expiry:3, startTime:1440542688, payload:update jim payload)
 and now contains:CacheEntry(key:jim, name:jnorthr, expiry:3, startTime:1440542688,
payload:update jim payload)
is cache ok: ok(jim)=true
is cache ok after 6 sec.s: ok(jim)=false



----------------
Show all CacheEntry :
Aug 26, 2015 12:44:55 AM java_util_logging_Logger$info$0 call
INFO: [0] = fred=CacheEntry(key:fred, name:Flintstone, expiry:0, startTime:1440542687,
payload:null);
Aug 26, 2015 12:44:55 AM java_util_logging_Logger$info$0 call
INFO: [1] = jim=CacheEntry(key:jim, name:jnorthr, expiry:3, startTime:1440542688,
payload:update jim payload);
Aug 26, 2015 12:44:55 AM java_util_logging_Logger$info$0 call
INFO: [2] = eve=CacheEntry(key:eve, name:, expiry:0, startTime:1440542688,
payload:horse feathers);

flag from CacheManagerTest del(fred):true
flag from CacheManagerTest del(jim):true
Aug 26, 2015 12:44:55 AM sun.reflect.NativeMethodAccessorImpl invoke0
INFO:
data has fred to remove
Aug 26, 2015 12:44:55 AM java_util_logging_Logger$info$0 call
INFO: data del fred removed ?true
Aug 26, 2015 12:44:55 AM sun.reflect.NativeMethodAccessorImpl invoke0
INFO:
data has jim to remove
Aug 26, 2015 12:44:55 AM java_util_logging_Logger$info$0 call
INFO: data del jim removed ?true
ans from CacheManagerTest get(jim):null

CacheManagerTest now holds:[eve:CacheEntry(key:eve, name:, expiry:0,
startTime:1440542688, payload:horse feathers)]

---------
Another put but this time as a CacheEntry
```

```
flag from CacheManagerTest del(max):false
--- so now ce:CacheEntry(key:max, name:MaxWell, expiry:21, startTime:1440542695,
payload:null)
Aug 26, 2015 12:44:55 AM sun.reflect.NativeMethodAccessorImpl invoke0
INFO:
data does not have max, so add
Aug 26, 2015 12:44:55 AM java_util_logging_Logger$info$0 call
INFO: data for max added:true
added from CacheManagerTest put(max):CacheEntry(key:max, name:MaxWell, expiry:21,
startTime:1440542695, payload:null)
did max add ? ans from CacheManagerTest get(max):CacheEntry(key:max, name:MaxWell,
expiry:21, startTime:1440542695, payload:null)
Aug 26, 2015 12:44:55 AM sun.reflect.GeneratedMethodAccessor2 invoke
INFO:
data has max to update
Aug 26, 2015 12:44:55 AM java_util_logging_Logger$info$0 call
INFO: data max updated:CacheEntry(key:max, name:, expiry:0, startTime:1440542695,
payload:MaxWell's silver hammer came down on her head. Bang, bang - Maxwell's hammer
made sure she was dead.)

added from CacheManagerTest fix(max):CacheEntry(key:max, name:, expiry:0,
startTime:1440542695, payload:MaxWell's silver hammer came down on her head. Bang,
bang - Maxwell's hammer made sure she was dead.)
did max update ? ans from CacheManagerTest get(max):CacheEntry(key:max, name:,
expiry:0, startTime:1440542695, payload:MaxWell's silver hammer came down on her head.
Bang, bang - Maxwell's hammer made sure she was dead.)

----------
CacheManagerTest now holds:[eve:CacheEntry(key:eve, name:, expiry:0,
startTime:1440542688, payload:horse feathers), max:CacheEntry(key:max, name:,
expiry:0, startTime:1440542695, payload:MaxWell's silver hammer came down on her head.
Bang, bang - Maxwell's hammer made sure she was dead.)]

Show all CacheEntry :
Aug 26, 2015 12:44:55 AM java_util_logging_Logger$info$0 call
INFO: [0] = eve=CacheEntry(key:eve, name:, expiry:0, startTime:1440542688,
payload:horse feathers);
Aug 26, 2015 12:44:55 AM java_util_logging_Logger$info$0 call
INFO: [1] = max=CacheEntry(key:max, name:, expiry:0, startTime:1440542695,
payload:MaxWell's silver hammer came down on her head. Bang, bang - Maxwell's hammer
made sure she was dead.);
goodbye from CacheManagerTest
:runTestH2
Aug 26, 2015 12:44:57 AM org.codehaus.groovy.runtime.m12n.MetaInfExtensionModule
newModule
WARNING: Module [groovy-all] - Unable to load extension class
[org.codehaus.groovy.runtime.NioGroovyMethods]




---------------------------------------------
```

```
Hello from CacheManagerTestH2
... loading driver
... create cache table as H2mem data store
Aug 26, 2015 12:45:00 AM java_util_logging_Logger$info$0 call
INFO: ... create cache table as H2mem data store
CacheEntry put(String fred, CacheEntry entry)
has(String fred) ?
found 0 rows
flag=false
Aug 26, 2015 12:45:01 AM sun.reflect.NativeMethodAccessorImpl invoke0
INFO:
data does not have fred, so add
db.executeUpdate(inserted :fred
has(String fred) ?
Aug 26, 2015 12:45:01 AM java_util_logging_Logger$info$0 call
INFO: data for fred added:true
found 1 rows
has(String fred) ?
found 1 rows
CacheEntry get(String fred)=true
found 6 rows
result was :class groovy.sql.GroovyRowResult
------------------------

--->ID=1
--->KEY=fred
--->NAME=Flintstone
--->EXPIRY=0
--->STARTTIME=1440542700
--->PAYLOAD=null
-----------------------------

CacheEntry put(String Mary, CacheEntry entry)
has(String Mary) ?
found 0 rows
Aug 26, 2015 12:45:02 AM sun.reflect.NativeMethodAccessorImpl invoke0
INFO:
data does not have Mary, so add
flag=false
Aug 26, 2015 12:45:02 AM java_util_logging_Logger$info$0 call
INFO: data for Mary added:true
db.executeUpdate(inserted :Mary
has(String Mary) ?
found 1 rows
has(String Mary) ?
found 1 rows
CacheEntry get(String Mary)=true
found 6 rows
result was :class groovy.sql.GroovyRowResult
------------------------
```

```
--->ID=2
--->KEY=Mary
--->NAME=Flintstone.
--->EXPIRY=123
--->STARTTIME=345
--->PAYLOAD=clob2: '<html><h1>Mary woz ere</h1></html>'
-----------------------------

a1 from CacheManagerTestH2 constructor get(Mary):CacheEntry(key:Mary,
name:Flintstone., expiry:123, startTime:345, payload:clob2: '<html><h1>Mary woz
ere</h1></html>')
===================================



===================================
has(String fred) ?
found 1 rows
flag from CacheManagerTestH2 has(fred):true
===================================
has(String fred) ?
found 1 rows
CacheEntry get(String fred)=true
found 6 rows
result was :class groovy.sql.GroovyRowResult
------------------------

--->ID=1
--->KEY=fred
--->NAME=Flintstone
--->EXPIRY=0
--->STARTTIME=1440542700
--->PAYLOAD=null
-----------------------------

ans from CacheManagerTestH2 get(fred):CacheEntry(key:fred, name:Flintstone, expiry:0,
startTime:1440542700, payload:null)
===================================
has(String fredx) ?
found 0 rows
CacheEntry get(String fredx)=false
ans from CacheManagerTestH2 get(fredx):null
===================================
CacheEntry put(String jim, Map map)
has(String jim) ?
found 0 rows
Aug 26, 2015 12:45:02 AM sun.reflect.NativeMethodAccessorImpl invoke0
INFO:
data does not have jim, so add
CacheEntry put(CacheEntry entry) key:jim
CacheEntry put(String jim, CacheEntry entry)
has(String jim) ?
```

```
found 0 rows
flag=false
Aug 26, 2015 12:45:02 AM sun.reflect.NativeMethodAccessorImpl invoke0
INFO:
data does not have jim, so add
db.executeUpdate(inserted :jim
has(String jim) ?
Aug 26, 2015 12:45:02 AM java_util_logging_Logger$info$0 call
INFO: data for jim added:true
found 1 rows
has(String jim) ?
found 1 rows
CacheEntry get(String jim)=true
found 6 rows
result was :class groovy.sql.GroovyRowResult
------------------------

--->ID=3
--->KEY=jim
--->NAME=jimbo
--->EXPIRY=66
--->STARTTIME=1440542702
--->PAYLOAD=clob5: '<html></html>'
-----------------------------

has(String jim) ?
found 1 rows
added from CacheManagerTestH2 put(jim):CacheEntry(key:jim, name:jimbo, expiry:66,
startTime:1440542702, payload:clob5: '<html></html>')
has(String jim) ?
Aug 26, 2015 12:45:02 AM java_util_logging_Logger$info$0 call
INFO: data for jim added:true
found 1 rows
CacheEntry get(String jim)=true
found 6 rows
result was :class groovy.sql.GroovyRowResult
------------------------

--->ID=3
--->KEY=jim
--->NAME=jimbo
--->EXPIRY=66
--->STARTTIME=1440542702
--->PAYLOAD=clob8: '<html></html>'
-----------------------------

<deleted a bunch of lines here>

goodbye from CacheManagerTestH2

BUILD SUCCESSFUL
```

```
Total time: 2 mins 38.789 secs
RedApple:jim /Volumes/REDUSB/LoggingExamples $
```