Contents

New API Logging SE9	. 2
Introduce	. 2
Use Logger Default Work	. 2
Use Extenal Logging Framwork [Log4j2]	. 3
Custom Framworks implement use SLF4J	. 4
Create Logging implement System	. 5
Add into moudule-infor.java	. 6
Ok Run Done	. 7
Reference	. 8

New API Logging SE9

Introduce

This API has been introduced in java to provide a common mechanism to handle all the platform logs and to expose a service interface that can be customized by libraries and applications. This way, the JDK platform logs can use the same logging framework as the application, and the project dependencies can be reduced.

Use Logger Default Work

Simple demo with print log into Consolog

Create Logger inplement the System.Logger

```
1 package com.sharefullcode.blogspot.loging;
 3⊖ import java.text.MessageFormat;
 4 import java.util.ResourceBundle;
 6 public class Consologer implements System.Logger{
 80
        @Override
Δ 9
        public String getName() {
10
           return "Consolog name";
11
12
13⊝
        @Override
        public boolean isLoggable(Level level) {
△14
15
           return true;
16
17
18⊖
        public void log(Level level, ResourceBundle bundle, String msg, Throwable thrown) {
△19
20
             System.out.printf("ConsoleLogger [%s]: %s - %s%n", level, msg, thrown);
21
22
23⊝
        @Override
        public void log(Level level, ResourceBundle bundle, String format, Object... params) {
24
25
            System.out.printf("ConsoleLogger [%s]: %s%n", level,
26
                     MessageFormat.format(format, params));
27
28
29 }
```

We need to implement a LoggerFinder that creates instances of our Consologer.

```
🖹 💈 🖇 🗀 🔲 Consologer.java
Package Explorer ⋈
                                                                1
                                                    package com.sharefullcode.blogspot.loging;
> 📂 commonfeaturesse9
v 🖒 CustomImplementLogger
                                                  3 import java.lang.System.Logger;

→ 
⊕ com.sharefullcode.blogspot

                                                     public class ConsologerFinder extends System.LoggerFinder{
      v 🖶 app
         > AppOne.java
                                                  7⊝
                                                         @Override

→ 

H loging

                                                △ 8
                                                         public Logger getLogger(String name, Module module) {
         > 🛭 Consologer.java
                                                             return new Consologer();
                                                 10
         > ConsologerFinder.java
                                                 11
    > I module-info.java
                                                 12 }
  > March JRE System Library [JavaSE-9]
                                                 13
```

Finally, we need to register out LoggerFinder as a Service so it can be discovered by JDK. If we don't provide an implementation, the SimpleConsoleLogger will be used by default.

```
1 module CustomImplementLogger {
    2
    3
                 provides java.lang.System.LoggerFinder with
               * com.sharefullcode.blogspot.loging.ConsologerFinder;
    4
    5
    6
              exports com.sharefullcode.blogspot.app;
    7 }
 📳 Problems 🍳 Javadoc 🖳 Declaration 📮 Console 🛭 🔫 Progress 🎏 Call Hierarchy
 <terminated> AppOne [Java Application] C:\Users\Laptop\Documents\Java\se9\jdk\bin\javaw.exe (Jan 4, 2021 9:07:31 PM - 9:07:32 PM)
 Jan 04, 2021 9:07:32 PM com.sharefullcode.blogspot.app.AppOne main
 INFO: Test
 Jan 04, 2021 9:07:32 PM com.sharefullcode.blogspot.app.AppOne main
 SEVERE: error test
 Jan 04, 2021 9:07:32 PM com.sharefullcode.blogspot.app.AppOne abc
 INFO: Test
 Jan 04, 2021 9:07:32 PM com.sharefullcode.blogspot.app.AppOne abc
 SEVERE: error test

☑ AppOne.java 
☒

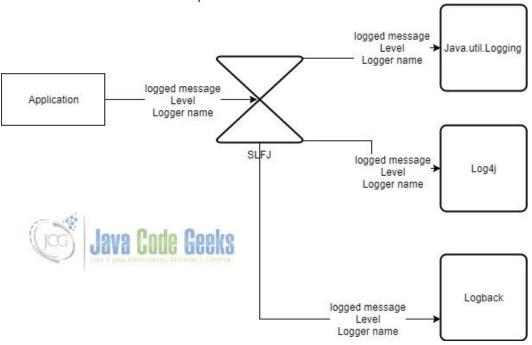
    ■ module-info.java 
    □

   package com.sharefullcode.blogspot.app;
                                                                                              module CustomImplementLogger {
                                                                                                 provides java.lang.System.LoggerFinder
with com.sharefullcode.blogspot.loging.ConsologerFinder;
   import java.lang.System.Logger.Level;
                                                                                                  exports com.sharefullcode.blogspot.app;
  5 public class AppOne {
        static System.Logger LOGGER = System.getLogger(AppOne.class.toString());
        public static void main(String[] args) {
   LOGGER.log(Level.INFO, "Test");
   LOGGER.log(Level.ERROR, "error test");
}
 11
12
13
14
15<sup>©</sup>
            abc();
        static void abc() {
            LOGGER.log(Level.INFO, "Test");
LOGGER.log(Level.ERROR, "error test");
 20
                                                                                                                         ■ X ¾ | B, 17 B | F F | → □ → □
🔐 Problems @ Javadoc 🚇 Declaration 📮 Console 🛭 🦐 Progress 🍰 Call Hierarchy
<terminated> AppOne [Java Application] C:\Users\Laptop\Documents\Java\se9\jdk\bin\javaw.exe (Jan 4, 2021 9:08:17 PM – 9:08:18 PM)
ConsoleLogger [INFO]: Test
ConsoleLogger [ERROR]: error test
ConsoleLogger [INFO]: Test
ConsoleLogger [ERROR]: error test
```

Use Extenal Logging Framwork [Log4j2]

One of the most useful uses of the Logging API in java 9 is to let applications route the JDK logs to the same logging framework the application is using.

Custom Framworks implement use SLF4J



https://examples.javacodegeeks.com/enterprise-java/slf4j/slf4j-tutorial-beginners/

Add File pom.xml

```
M *DemoApiLoggingWithLog4j2/pom.xml 

□

 1⊖ org/POM/4.0.0"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
 4
        <modelVersion>4.0.0</modelVersion>
       <groupId>sharefullcode
 6
       <artifactId>DemoApiLoggingWithLog4j2</artifactId>
       <version>0.0.1-SNAPSHOT</version>
 8
        <dependencies>
 9⊝
10
           <!-- https://mvnrepository.com/artifact/org.slf4j/slf4j-api -->
11
12⊝
           <dependency>
13
                <groupId>org.slf4j</groupId>
                <artifactId>slf4j-api</artifactId>
               <version>1.7.30
15
           </dependency>
16
17
           <!-- https://mvnrepository.com/artifact/org.slf4j/slf4j-log4j12 -->
18
19⊝
           <dependency>
               <groupId>org.slf4j</groupId>
20
21
               <artifactId>slf4j-log4j12</artifactId>
               <version>1.7.30</version>
22⊖
           </dependency>
23
24
25
           <!-- https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-core -->
26⊖
27
               <groupId>org.apache.logging.log4j</groupId>
28
               <artifactId>log4j-core</artifactId>
29
                <version>2.14.0</version>
           </dependency>
        </dependencies>
32
33 </project>
```

Create Logging implement System

```
■ DemoApiLoggingWithLog4j2/pom.xml
                                1 package com.sharefullcode.blogspot;
 3 import java.util.ResourceBundle;
 5 import org.slf4j.Logger;
 6 import org.slf4j.LoggerFactory;
 8 public class Slf4jCustomImplementation implements System.Logger{
№10
        private String name;
№11
        private Logger loggerSlf4j;
 12
 13⊝
        public Slf4jCustomImplementation(String name) {
14
           this.name = name;
15
           this.loggerSlf4j = LoggerFactory.getLogger(name);
16
        }
17
```

The method remain is implement

```
■ DemoApiLoggingWithLog4j2/pom.xml
                                   ☑ SIf4jCustomImplementation.java ⋈
16
17
public String getName() {
19
             return name;
 20
 21
≥22⊝
        public boolean isLoggable(Level level) {
 23
            switch (level) {
 24
            case OFF:
 25
                 return false;
 26
            case TRACE:
 27
                return loggerSlf4j.isTraceEnabled();
 28
            case DEBUG:
 29
                return loggerSlf4j.isDebugEnabled();
 30
            case INFO:
 31
                 return loggerSlf4j.isInfoEnabled();
 32
            case WARNING:
 33
                 return loggerSlf4j.isWarnEnabled();
 34
            case ERROR:
 35
                 return loggerSlf4j.isErrorEnabled();
 36
            case ALL:
 37
            default:
 38
                 return true;
 39
40
        }
41
△42<sup>©</sup>
        public void log(Level level, ResourceBundle bundle, String msg, Throwable thrown) {
243
            // TODO Auto-generated method stub
44
45
        }
46
△47⊝
        public void log(Level level, ResourceBundle bundle, String format, Object... params) {
48
             // TODO Auto-generated method stub
49
50
        }
```

```
■ Package Explorer ⊠
                                                                                                                                                                                                 commonfeaturesse9
     public void log(Level level, ResourceBundle bundle, String format, Object... params) {
    if (!isloggable(level)) {
        return;
    }
}

→ 

⊕ com.sharefullcode.blogspot

                         > I Slf4jCustomImplementation.java
                                                                                                                                                                                                                                                   String msg = MessageFormat.format(format, params);
               src/main/resources

■ src/test/java

                                                                                                                                                                                                                                                    switch (level) {
              src/test/resources
        > ■ JRE System Library [J2SE-1.5]
                                                                                                                                                                                                                                                                   loggerSlf4j.trace(msg);
break;
              Maven Dependencies
        > 🥟 src
                                                                                                                                                                                                                                                    case DEBUG:
                                                                                                                                                                                                                                                                  loggerSlf4j.debug(msg);
break;

    barget
    barget
    barget
    barget
    contact    contact    barget
    contact    barget
    contact    contact    barget
    contact    contact    barget
    contact    contact    barget
    contact    co
                                                                                                                                                                                                                                                    case INFO:
> 👺 demoLog4j2
                                                                                                                                                                                                                                                   loggerSlf4j.info(msg);
break;
case WARNING:
                                                                                                                                                                                                                                                   loggerSlf4j.warn(msg);
break;
case ERROR:
                                                                                                                                                                                                                                                    loggerSlf4j.error(msg);
break;
case ALL:
                                                                                                                                                                                                                                                   default:
                                                                                                                                                                                                                                                                   loggerSlf4j.info(msg);
```

Create System.FinderLogger

```
commonfeaturesse9
                                             package com.sharefullcode.blogspot;
DemoApiLogging
                                             import java.lang.System.Logger;
✓ "> DemoApiLoggingWithLog4j2
 public class Slf4jLoggerFinder extends System.LoggerFinder{

→ 

⊕ com.sharefullcode.blogspot

     > I SIf4jCustomImplementation.java
                                                @Override
     > I Slf4jLoggerFinder.java
                                          8
                                                public Logger getLogger(String name, Module module) {
                                                   return new Slf4jCustomImplementation(name);
   10
     > 🗓 AppOne.java
                                          11
   src/main/resources
                                          12
   src/test/java
```

Add into moudule-infor.java

provides ... with ... Register service in the module-info

```
☐ SIf4jLoggerFinder.java 
☐ DemoApiLoggingWithLog4j2 ☐

1 module DemoApiLoggingWithLog4j2 ☐

2 exports com.sharefullcode.blogspot.app;
exports com.sharefullcode.blogspot;

4

5 provides java.lang.System.LoggerFinder with com.sharefullcode.blogspot.Slf4jLoggerFinder;

7

8 requires org.slf4j;

9 }
```

```
E 😫 🖁 🗖 🗖 📝 AppOne.java 🛭
□ Package Explorer ⋈
                                                              package com.sharefullcode.blogspot.app;
  commonfeaturesse!
 import java.lang.System.Logger.Level;
DemoApiLoggingWithLog4j2
  public class AppOne

→ 

⊕ com.sharefullcode.blogspot

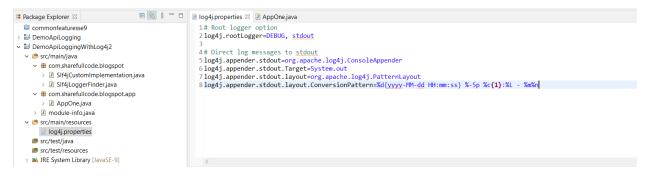
        > I SIf4jCustomImplementation.java
                                                                  private final static System.Logger LOGGER = System.getLogger(AppOne.class.getSimpleName());
        SIf4il oggerFinder.iava
                                                                  public static void main(String[] args) {
    LOGGER.log(Level.INFO, "inFOR LOG");
      AppOne.java
     > 🛽 module-info.java
     src/main/resources
     src/test/java
     src/test/resources
  > M JRE System Library [JavaSE-9]
  > Maven Dependencies
  > 🗁 src

    Problems @ Javadoc   □ Declaration  □ Console  □  □ Progress  □ Call Hierarchy

    target
                                                         <terminated> AppOne (2) [Java Application] C\Users\Laptop\Documents\Java\se9\jdk\bin\javaw.exe (Jan 7, 2021 3:56:35 PM – 3:56:35 PM)
     Imx.mog M
                                                        log4j:WARN No appenders could be found for logger (AppOne).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
```

Not yet create file log4j config

Src/resources/Log4j.properties



Ok Run Done

```
🖹 🚼 🖁 📅 🗖 🖟 AppOne.java 🛭 🗎 log4j.properties 🔑 Slf4jLoggerFinder.java
□ Package Explorer ≅
  commonfeaturesse9
                                                                       package com.sharefullcode.blogspot.app;
  import java.lang.System.Logger.Level;

    BemoApiLoggingWithLog4j2

   public class AppOne {

√ 

⊕ com.sharefullcode.blogspot

          >  Slf4jCustomImplementation.java
                                                                             private final static System.Logger LOGGER = System.getLogger(AppOne.class.getSimpleName());
          > 🛭 SIf4jLoggerFinder.java
                                                                             public static void main(String[] args) {

w ## com.sharefullcode.blogspot.app

                                                                                  LOGGER.log(Level.LU, "ALL LOG");
LOGGER.log(Level.DEBUG, "DEBUG LOG");
LOGGER.log(Level.DEBUG, "DEBUG LOG");
LOGGER.log(Level.DFC, "THFO LOG");
LOGGER.log(Level.DFF, "OFF LOG");
LOGGER.log(Level.DFF, "OFF LOG");
LOGGER.log(Level.TRACE, "TRACE LOG");
         > 1 module-info.java
   log4j.properties
      src/test/java
                                                                                   LOGGER.log(Level.WARNING, "WARNING LOG");
      src/test/resources
   > M JRE System Library [JavaSE-9]
                                                                   18 }
   > Maven Dependencies
   > 🗁 src
      target
      Imx.moq 🔝
 > 👺 demoLog4j2
                                                                  🖺 Problems @ Javadoc 🖳 Declaration 📮 Console 🖾 🤜 Progress 🎏 Call Hierarchy
                                                                 <terminated> AppOne (2) [Java Application] C:\Users\Laptop\Documents\Java\se9\jdk\bin\javaw.exe (Jan 7, 2021 4:02:30 PM - 4:02:31 PM)
                                                                 2021-01-07 16:02:31 INFO AppOne:94 - ALL LOG
2021-01-07 16:02:31 DEBUG AppOne:81 - DEBUG LOG
                                                                 2021-01-07 16:02:31 ERROR AppOne:90 - ERROR LOG
2021-01-07 16:02:31 INFO AppOne:84 - INFO LOG
2021-01-07 16:02:31 WARN AppOne:87 - WARNING LOG
```

Reference

This section is non commercial mainly sharing and advance knowlage for java. This tutorials has referenced document from the list below if you has complain for license, i will remove all from internet. Thank you all everything.

https://www.baeldung.com/java-9-logging-api

https://examples.javacodegeeks.com/enterprise-java/slf4j/slf4j-tutorial-beginners/