Caleb Miller

CST-235

20171217

Week 6 Short Assignment

[GitHub](https://github.com/spudgi/CST-235)

Using a simple logging formula that I have used in the past with java classes, I added a handler that directs the logging, a filter to limit what is logged, and a configuration file to mesh everything together.

Unfortunately, as always, I am getting a 500 HTTP Error, but I am sure my java is sound because I have used it in past classes.

Log4j.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE log SYSTEM "logger.dtd">

<log>

<record>

<date>2014-06-08T13:43:19</date>

<millis>1402215199326</millis>

<sequence>0</sequence>

<logger>com.javacodegeeks.corejava.util.logging.LoggerExample</logger>

<level>CONFIG</level>

<class>com.javacodegeeks.corejava.util.logging.HandlerExample</class>

<method>main</method>

<thread>1</thread>

<message>Configuration done.</message>

</record>

<record>

<date>2014-06-08T13:43:19</date>

<millis>1402215199376</millis>

<sequence>1</sequence>

<logger>com.javacodegeeks.corejava.util.logging.LoggerExample</logger>

<level>FINE</level>

<class>com.javacodegeeks.corejava.util.logging.HandlerExample</class>

<method>main</method>

<thread>1</thread>

<message>Finer logged</message>

</record>

<record>

<date>2014-06-08T13:43:19</date>

<millis>1402215199376</millis>

<sequence>2</sequence>

<logger>com.javacodegeeks.corejava.util.logging.LoggerExample</logger>

<level>FINER</level>

<class>com.javacodegeeks.corejava.util.logging.HandlerExample</class>

<method>main</method>

<thread>1</thread>

<message>Finest example on LOGGER handler completed.</message>

</record>

</log>

Main.java

**package** application;

**import** java.io.IOException;

**import** java.util.logging.Level;

**import** java.util.logging.Logger;

**public** **class** Main {

**private** **static** **final** Logger ***LOGGER*** = Logger.*getLogger*(Main.**class**.getName());

**public** **static** **void** main(String[] args) **throws** SecurityException, IOException {

***LOGGER***.info("Logger Name: "+***LOGGER***.getName());

***LOGGER***.warning("Can cause ArrayIndexOutOfBoundsException");

**int** []a = {1,2,3};

**int** index = 4;

***LOGGER***.config("index is set to "+index);

**try**{

System.***out***.println(a[index]);

}**catch**(ArrayIndexOutOfBoundsException ex){

***LOGGER***.log(Level.***SEVERE***, "Exception occur", ex);

}

}

}

Handler.java

**package** application;

**import** java.io.IOException;

**import** java.util.logging.ConsoleHandler;

**import** java.util.logging.FileHandler;

**import** java.util.logging.Level;

**import** java.util.logging.Logger;

**public** **class** Handler {

**private** **static** **final** Logger ***LOGGER*** = Logger.*getLogger*(Main.**class**.getName());

**public** **static** **void** main(String[] args) {

ConsoleHandler consoleHandler = **null**;

FileHandler fileHandler = **null**;

**try**{

consoleHandler = **new** ConsoleHandler();

fileHandler = **new** FileHandler("./javacodegeeks.log");

***LOGGER***.addHandler(consoleHandler);

***LOGGER***.addHandler(fileHandler);

consoleHandler.setLevel(Level.***ALL***);

fileHandler.setLevel(Level.***ALL***);

***LOGGER***.setLevel(Level.***ALL***);

***LOGGER***.config("Configuration done.");

***LOGGER***.removeHandler(consoleHandler);

***LOGGER***.log(Level.***FINE***, "Finer logged");

}**catch**(IOException exception){

***LOGGER***.log(Level.***SEVERE***, "Error occur in FileHandler.", exception);

}

***LOGGER***.finer("Finest example on LOGGER handler completed.");

}

}

Filter.java

**package** application;

**import** java.util.logging.LogRecord;

**import** java.util.logging.Logger;

**public** **class** Filter **implements** java.util.logging.Filter {

**private** **static** **final** Logger ***LOGGER*** = Logger.*getLogger*(Main.**class**.getName());

**public** **static** **void** main(String[] args) {

***LOGGER***.setFilter(**new** Filter());

***LOGGER***.severe("This is SEVERE message");

***LOGGER***.warning("This is important warning message");

}

@Override

**public** **boolean** isLoggable(LogRecord record) {

**if**(record == **null**)

**return** **false**;

String message = record.getMessage()==**null**?"":record.getMessage();

**if**(message.contains("important"))

**return** **true**;

**return** **false**;

}

}

Configuration.java

**package** application;

**import** java.io.FileInputStream;

**import** java.io.IOException;

**import** java.util.logging.Level;

**import** java.util.logging.LogManager;

**import** java.util.logging.Logger;

**public** **class** Configuration {

**private** **static** **final** LogManager ***logManager*** = LogManager.*getLogManager*();

**private** **static** **final** Logger ***LOGGER*** = Logger.*getLogger*("confLogger");

**static** {

**try** {

***logManager***.readConfiguration(**new** FileInputStream("./javacodegeeks.properties"));

} **catch** (IOException exception) {

***LOGGER***.log(Level.***SEVERE***, "Error in loading configuration",exception);

}

}

**public** **static** **void** main(String[] args) {

***LOGGER***.fine("Fine message logged");

}

}



