How to configure logging in Hibernate – SLF4j + Log4j

By mkyong | December 30, 2009 | Updated : August 30, 2012 | Viewed : 175,820 times

---

> **Try logback**
> Try logback logging framework, read this article for the "reasons to prefer logback over log4j. To integrate logback with Hibernate, refer this – How to configure logging in Hibernate – Logback

Hibernate uses Simple Logging Facade for Java (SLF4J) to redirect the logging output to your perfer logging frameworkis (log4j, JCL, JDK logging, lofback…). In this tutorial, we show you how to do logging in **Hibernate with SLF4j + Log4j** logging framework.

Technologies used in this article :

1. Hibernate 3.6.3.Final
2. slf4j-api-1.6.1
3. slf4j-log4j12-1.6.1
4. Eclipse 3.6
5. Maven 3.0.3

1. Get SLF4j + Log4j

To do logging in Hibernate, you need "**slf4j-api.jar**" and your preferred binding, like log4j "**slf4j-log4j12.jar**". Just declares the dependency in your `pom.xml` .

File : pom.xml

```
                                                                    Markup
<project ...>
```

```xml
    <repositories>
        <repository>
            <id>JBoss repository</id>
            <url>http://repository.jboss.org/nexus/content/groups/public/</url>
        </repository>
    </repositories>

    <dependencies>

        <dependency>
            <groupId>org.hibernate</groupId>
            <artifactId>hibernate-core</artifactId>
            <version>3.6.3.Final</version>
        </dependency>

        <!-- slf4j-log4j -->
        <dependency>
            <groupId>org.slf4j</groupId>
            <artifactId>slf4j-log4j12</artifactId>
            <version>1.6.1</version>
        </dependency>

    </dependencies>
</project>
```
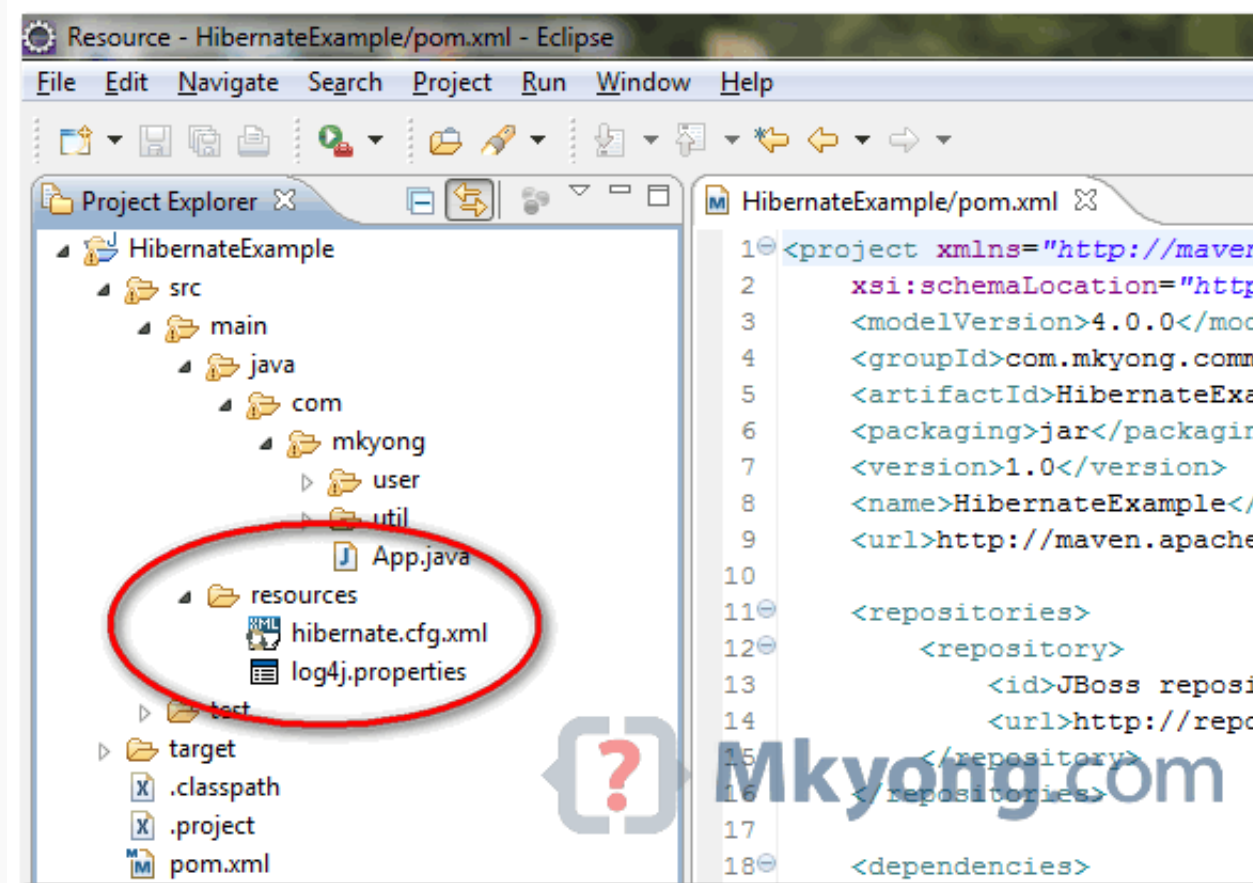
Where is slf4j-api.jar?
The **slf4j-api.jar** is defined as the dependency of "**hibernate-core**", so , you do not need to declare it again.

2. Log4j properties file

Create a "**log4j.properties**" file and put it into your project's classpath, see figure below :

File : log4.properties

Bash

```
# Direct log messages to a log file
log4j.appender.file=org.apache.log4j.RollingFileAppender
log4j.appender.file.File=C:\\mkyongapp.log
log4j.appender.file.MaxFileSize=1MB
log4j.appender.file.MaxBackupIndex=1
```

```
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L - %m%n

# Direct log messages to stdout
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target=System.out
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L - %m%n

# Root logger option
log4j.rootLogger=INFO, file, stdout

# Log everything. Good for troubleshooting
log4j.logger.org.hibernate=INFO

# Log all JDBC parameters
log4j.logger.org.hibernate.type=ALL
```
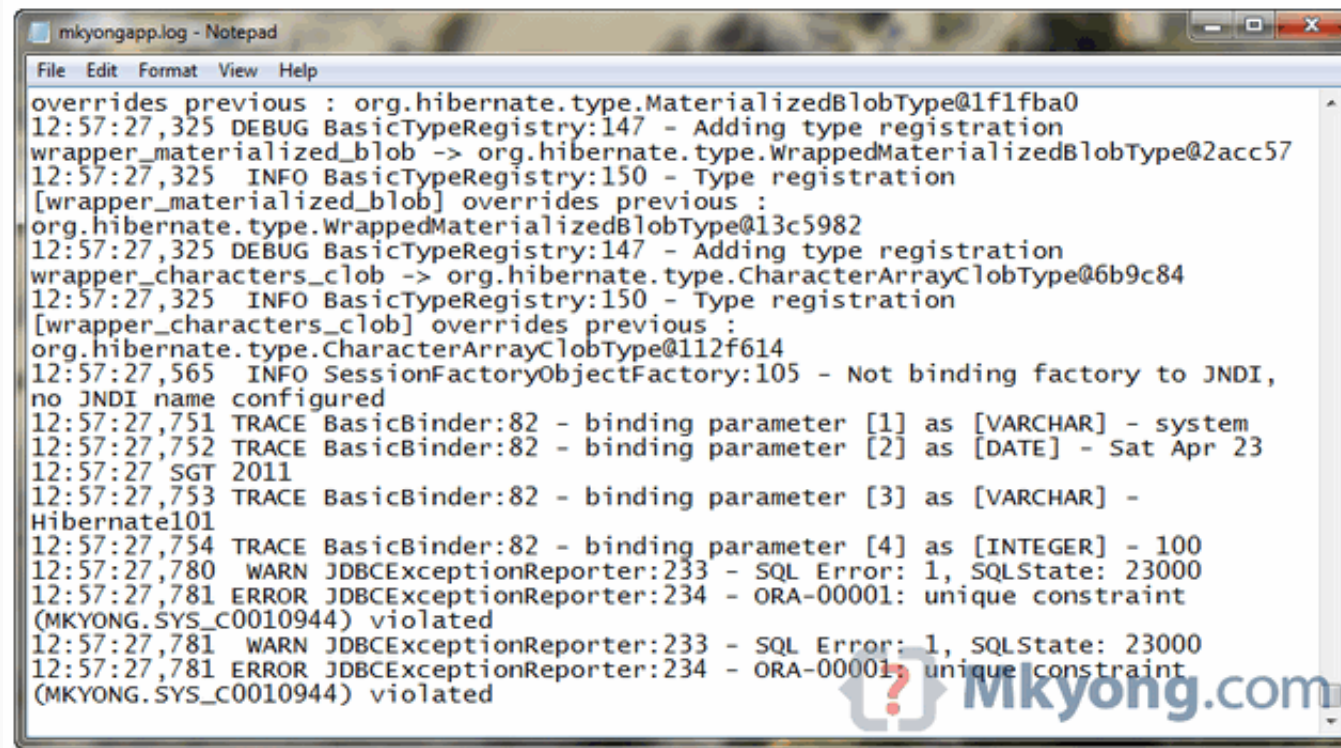
With this log4j configuration, it will redirect all the logging output to console and also a file at " `C:\\mkyongapp.log` ".

> **Note**
> Hibernate provides many settings to let developer to decide what to log. Always refer to this Hibernate Log Categories, choose some and implement it in your log file.

3. Output

Try run your Hibernate web application, all logging output will be logged in " `C:\\mkyongapp.log` " file. See figure below :

Download it – Log4j-Hibernate-Example.zip (7KB)

Reference

1. Simple Logging Facade for Java (SLF4J)
2. http://logging.apache.org/log4j/1.2/

Tags :   hibernate | log4j | logging