

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 preferred logging framework (log4j, JCL, JDK logging, logback...). 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`

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

Markup

```
<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>

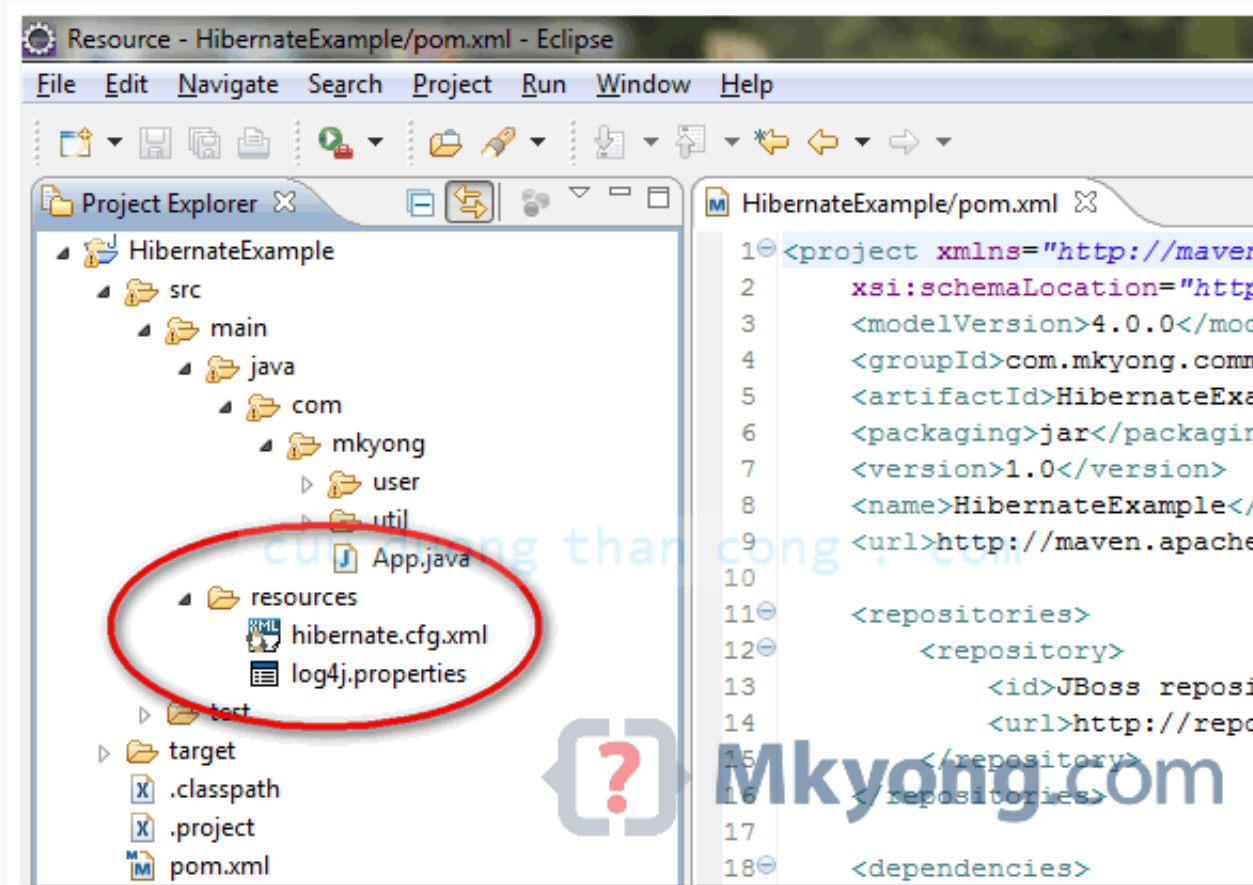
</dependencies>
</project>
```

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*

```
# 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
```

Bash

```
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
```

cuu duong than cong . com

With this log4j configuration, it will redirect all the logging output to console and also a file at “**C:\\mkyongapp.log**”.

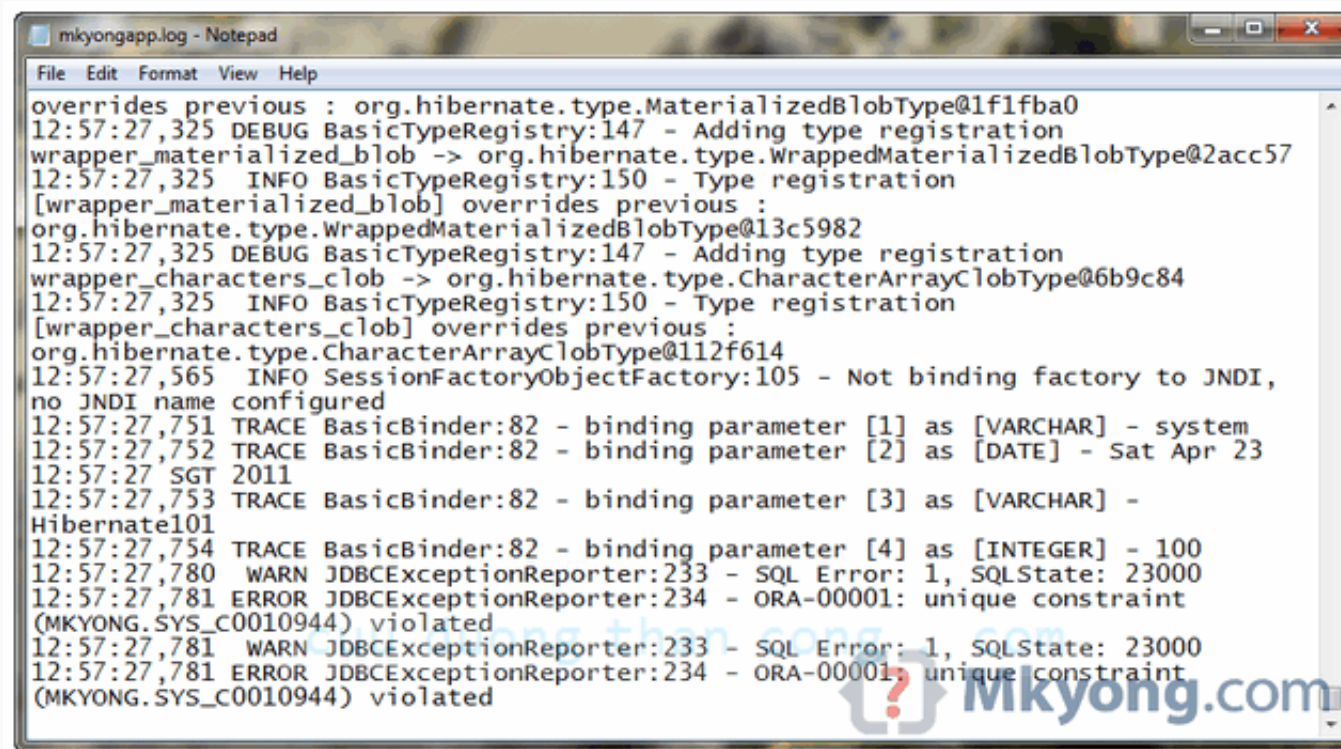
Note

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.

cuu duong than cong . com

3. Output

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



```
mkyongapp.log - Notepad
File Edit Format View Help
overrides previous : org.hibernate.type.MaterializedBlobType@1f1fba0
12:57:27,325 DEBUG BasicTypeRegistry:147 - Adding type registration
wrapper_materialized_blob -> org.hibernate.type.WrappedMaterializedBlobType@2acc57
12:57:27,325 INFO BasicTypeRegistry:150 - Type registration
[wrapper_materialized_blob] overrides previous :
org.hibernate.type.WrappedMaterializedBlobType@13c5982
12:57:27,325 DEBUG BasicTypeRegistry:147 - Adding type registration
wrapper_characters_clob -> org.hibernate.type.CharacterArrayClobType@6b9c84
12:57:27,325 INFO BasicTypeRegistry:150 - Type registration
[wrapper_characters_clob] overrides previous :
org.hibernate.type.CharacterArrayClobType@112f614
12:57:27,565 INFO SessionFactoryObjectFactory:105 - Not binding factory to JNDI,
no JNDI name configured
12:57:27,751 TRACE BasicBinder:82 - binding parameter [1] as [VARCHAR] - system
12:57:27,752 TRACE BasicBinder:82 - binding parameter [2] as [DATE] - Sat Apr 23
12:57:27 SGT 2011
12:57:27,753 TRACE BasicBinder:82 - binding parameter [3] as [VARCHAR] -
Hibernate101
12:57:27,754 TRACE BasicBinder:82 - binding parameter [4] as [INTEGER] - 100
12:57:27,780 WARN JDBCExceptionReporter:233 - SQL Error: 1, SQLState: 23000
12:57:27,781 ERROR JDBCExceptionReporter:234 - ORA-00001: unique constraint
(MKYONG.SYS_C0010944) violated
12:57:27,781 WARN JDBCExceptionReporter:233 - SQL Error: 1, SQLState: 23000
12:57:27,781 ERROR JDBCExceptionReporter:234 - ORA-00001: unique constraint
(MKYONG.SYS_C0010944) violated
```

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

cuu duong than cong . com

Reference

1. [Simple Logging Facade for Java \(SLF4J\)](#)
2. <http://logging.apache.org/log4j/1.2/>

Tags : [hibernate](#) [log4j](#) [logging](#)

cuu duong than cong . com

cuu duong than cong . com