## 需要jar 和打开mybatis日志配置

slf4j-api-1.7.25.jar、logback-classic-1.2.3.jar、logback-core-1.2.3.jar、

logback-ext-spring-0.1.4.jar

|  |
| --- |
| <logback.version>1.2.3</logback.version>  <dependency>  <groupId>org.slf4j</groupId>  <artifactId>slf4j-api</artifactId>  <version>1.7.25</version>  </dependency>  <dependency>  <groupId>ch.qos.logback</groupId>  <artifactId>logback-classic</artifactId>  <version>${logback.version}</version>  </dependency>  <dependency>  <groupId>ch.qos.logback</groupId>  <artifactId>logback-core</artifactId>  <version>${logback.version}</version>  </dependency>  <dependency>  <groupId>org.logback-extensions</groupId>  <artifactId>logback-ext-spring</artifactId>  <version>0.1.4</version>  </dependency> |

mybatis-config.xml配置文件添加配置，打开日志主要是*logImpl*这个参数

因为mybatis原则上是不支持logback，但是logback是SLF4J的完美实现者（有点像java的接口和实现类感觉）

|  |
| --- |
| <configuration>  <settings>  <!-- 全局映射器启用缓存 -->  <setting name=*"cacheEnabled"* value=*"true"*/>  <!-- 查询时，关闭关联对象即时加载以提高性能 -->  <setting name=*"lazyLoadingEnabled"* value=*"true"*/>  <!-- 设置关联对象加载的形态，此处为按需加载字段(加载字段由SQL指 定)，不会加载关联表的所有字段，以提高性能 -->  <setting name=*"aggressiveLazyLoading"* value=*"false"*/>  <!-- 日志实现方式。SLF4J、LOG4J2、LOG4J、STDOUT\_LOGGING -->  <setting name=*"logImpl"* value=*"****SLF4J****"*/>  </settings>  </configuration> |

## 配置logback.xml

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <configuration>  <appender name=*"STDOUT"* class=*"ch.qos.logback.core.ConsoleAppender"*>  <encoder charset=*"UTF-8"*>  <pattern>  %d{yyyy-MM-dd HH:mm:ss.SSS} [%thread] %-5level %logger{50} - %msg%n  </pattern>  </encoder>  </appender>  <appender name=*"info"* class=*"ch.qos.logback.core.rolling.RollingFileAppender"*>  <file>${catalina.home}/logs/ssm-init-info.log</file>  <rollingPolicy class=*"ch.qos.logback.core.rolling.TimeBasedRollingPolicy"*>  <fileNamePattern>${catalina.home}/logs/ssm-init-info.%d{yyyy-MM-dd}.log</fileNamePattern>  <maxHistory>30</maxHistory>  <maxFileSize>20MB</maxFileSize>  </rollingPolicy>  <filter class=*"ch.qos.logback.classic.filter.LevelFilter"*>  <level>INFO</level>  <onMatch>ACCEPT</onMatch>  <onMismatch>DENY</onMismatch>  </filter>  <encoder>  <pattern>%d{HH:mm:ss.SSS} %-5level %logger{35} - %msg%n</pattern>  </encoder>  </appender>  <appender name=*"debug"* class=*"ch.qos.logback.core.rolling.RollingFileAppender"*>  <file>${catalina.home}/logs/ssm-init-debug.log</file>  <rollingPolicy class=*"ch.qos.logback.core.rolling.TimeBasedRollingPolicy"*>  <fileNamePattern>${catalina.home}/logs/ssm-init-debug.%d{yyyy-MM-dd}.log</fileNamePattern>  <maxHistory>30</maxHistory>  <maxFileSize>60MB</maxFileSize>  </rollingPolicy>  <!-- 这个得注释掉，因为加上了会重复打印，原因看下面就知道了 -->  <!-- <filter class="ch.qos.logback.classic.filter.LevelFilter">  <level>DEBUG</level>  <onMatch>ACCEPT</onMatch>  <onMismatch>DENY</onMismatch>  </filter> -->  <encoder>  <pattern>%d{HH:mm:ss.SSS} %-5level %logger{35} - %msg%n</pattern>  </encoder>  </appender>  <appender name=*"error"* class=*"ch.qos.logback.core.rolling.RollingFileAppender"*>  <file>${catalina.home}/logs/ssm-init-error.log</file>  <rollingPolicy class=*"ch.qos.logback.core.rolling.TimeBasedRollingPolicy"*>  <fileNamePattern>${catalina.home}/logs/ssm-init-error.%d{yyyy-MM-dd}.log</fileNamePattern>  <maxHistory>30</maxHistory>  <maxFileSize>20MB</maxFileSize>  </rollingPolicy>  <filter class=*"ch.qos.logback.classic.filter.LevelFilter"*>  <level>ERROR</level>  <onMatch>ACCEPT</onMatch>  <onMismatch>DENY</onMismatch>  </filter>  <encoder>  <pattern>%d{HH:mm:ss.SSS} %-5level %logger{35} - %msg%n</pattern>  </encoder>  </appender>  <!-- 自己项目代码里的日志级别调为debug级别 -->  <logger name=*"com.shuframework"* level=*"DEBUG"*>  <!-- <appender-ref ref="debug"></appender-ref> -->  <appender-ref ref=*"info"*></appender-ref>  <appender-ref ref=*"error"*></appender-ref>  </logger>  <!-- 就是这个监控了mybatis日志输出，配合上面的“dao”  <logger name="com.shuframework.\*.mapper" level="DEBUG" /> -->  <!-- 以下这些也没用 -->  <!-- <logger name="com.apache.ibatis" level="DEBUG" />  <logger name="java.sql.Connection" level="DEBUG" />  <logger name="java.sql.Statement" level="DEBUG" />  <logger name="java.sql.PreparedStatement" level="DEBUG" />  <logger name="org.mybatis" level="DEBUG" additivity="false">  <appender-ref ref="debug" />  </logger>  <logger name="org.apache.ibatis" level="DEBUG" additivity="false">  <appender-ref ref="debug" />  </logger>  <logger name="java.sql" level="DEBUG" additivity="false">  <appender-ref ref="debug" />  </logger> -->  <!-- 这个是日志总控制开关，这里面我把“debug“加上去 ，所以上面那个注释的就不要 -->  <root level=*"DEBUG"*>  <appender-ref ref=*"debug"*></appender-ref>  <appender-ref ref=*"STDOUT"* />  <appender-ref ref=*"com.shuframework"* />  </root>  </configuration> |

## 配置web.xml

|  |
| --- |
| <context-param>  <param-name>logbackConfigLocation</param-name>  <param-value>classpath:logback.xml</param-value>  </context-param>  <listener>  <listener-class>ch.qos.logback.ext.spring.web.LogbackConfigListener</listener-class>  </listener> |

注意：

1. 这个监听器需要在ContextLoaderListener 之前配置

附件（资料）

Logback日志配置详解 http://blog.csdn.net/l2009430167/article/details/49335603